



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ
ΣΥΣΤΗΜΑΤΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΑΣΥΡΜΑΤΟΥ ΕΛΕΓΚΤΗ
ΓΙΑ ΒΙΟΜΗΧΑΝΙΚΕΣ ΕΦΑΡΜΟΓΕΣ

Μιλτιάδης Νάκος

Επιβλέπων: Δρ. Κωνσταντίνος Αγγέλης
ΔΕΠ, Καθηγητής

Άρτα, Ιούλιος 2021

**DESIGN AND IMPLEMENTATION
OF AN ELECTRONIC WIRELESS CONTROLLER SYSTEM
FOR INDUSTRIAL APPLICATIONS**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Αρτα

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Κωνσταντίνος Αγγέλης,
ΔΕΠ, Καθηγητής
2. Μέλος επιτροπής
Ευριπίδης Γλαβάς,
ΔΕΠ, Καθηγητής
3. Μέλος επιτροπής
Αλέξανδρος Τζάλλας,
ΔΕΠ, Επίκουρος Καθηγητής

© Νάκος Μιλτιάδης 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εκ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Νάκος Μιλτιάδης

ΕΥΧΑΡΙΣΤΙΕΣ

Με το πέρας της παρούσας Διπλωματικής εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου Δρ. Κωνσταντίνο Αγγέλη για την πολύτιμη βοήθεια που μου παρείχε, όχι μόνο κατά την εκπόνηση της εργασίας αλλά και καθ' όλη την διάρκεια της διετούς πορείας των μεταπτυχιακών σπουδών μου.

Επίσης, στο σημείο αυτό που βρίσκομαι σήμερα και κοιτώντας το παρελθόν εκεί που όλα ξεκίνησαν με την έναρξη των προπτυχιακών σπουδών μου μερικά χρόνια νωρίτερα αλλά και μετέπειτα ως νέος εργαζόμενος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου τόσο για την πίστη που εναπόθεσε σε εμένα όλα αυτά τα χρόνια όσο και για την στήριξη που μου παρείχε, καθώς και στους συγγενείς και φίλους μου –στους δικούς μου ανθρώπους– για την υπομονή και κατανόησή τους όλο αυτό το διάστημα.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος ηλεκτρονικού ασύρματου ελεγκτή για βιομηχανικές εφαρμογές, σύστημα ελέγχου δηλαδή που στηρίζεται σε ηλεκτρονικές διατάξεις σχεδιασμένες με τρόπο τέτοιο ώστε να επικοινωνούν μεταξύ τους ασυρματικά, παρέχοντας δυνατότητα εκτέλεσης και ελέγχου εφαρμογών από απόσταση.

Χάρης στην ένταξη των ηλεκτρονικών κυκλωμάτων – τόσο αναλογικών όσο και ψηφιακών – σε όλους σχεδόν τους τομείς των τεχνολογικών εφαρμογών, η τεχνολογία γνώρισε ραγδαία εξέλιξη τα τελευταία χρόνια, και ακριβώς όπως η επιστήμη της ηλεκτρονικής είναι πλέον απαραίτητη σε όλες τις μορφές των σύγχρονων συσκευών και μηχανημάτων, το ίδιο ισχύει και για την χρήση της ασύρματης τεχνολογίας για επικοινωνία τόσο μεταξύ των ίδιων των ανθρώπων όσο και μεταξύ ανθρώπου και μηχανής.

Τα συστήματα ελέγχου που στηρίζονται σε ηλεκτρονική και ασύρματη τεχνολογία θεωρούνται πλέον όχι απλά σαν μια βιώσιμη αλλά σαν μια απαραίτητη λύση για την σημερινή εποχή, όπου η απαίτηση για γρήγορους ρυθμούς και ευκολότερη έκβαση του καθημερινού τρόπου ζωής και διαβίωσης καθίσταται ως πάγια ανάγκη.

Ο ελεγκτής που πραγματεύεται η παρούσα διπλωματική εργασία μπορεί να χρησιμοποιηθεί σε πληθώρα εφαρμογών, με εγκατάσταση του εν λόγω συστήματος σε εξωτερικό ή εσωτερικό περιβάλλον για τον έλεγχο ενός χώρου και χειρισμό του με ασύρματο τρόπο. Ο σχεδιασμός και η υλοποίηση επομένως ενός τέτοιου ηλεκτρονικού ασύρματου συστήματος ελέγχου αποτελεί μια χρήσιμη τεχνολογική ύπαρξη για εφαρμογές της καθημερινής ζωής των ανθρώπων, καθώς επίσης και τεχνολογική βάση για περαιτέρω εξέλιξη και χρήση του σε μελλοντικές, νέες εφαρμογές, πέραν αυτών για τις οποίες ήδη προορίζεται.

Λέξεις-κλειδιά: Μικροελεγκτές, βιομηχανικά συστήματα ελέγχου, ασύρματη επικοινωνία, περιφερειακές μονάδες, ηλεκτρονικά εξαρτήματα.

ABSTRACT

The purpose of this master thesis is the design and implementation of an electronic wireless controller system for industrial applications, ie a control system based on electronic devices designed to communicate with each other wirelessly, enabling the execution and control of applications remotely.

Thanks to the integration of electronic circuits –both analog and digital– in almost all areas of technological applications, technology has evolved rapidly in recent years, and just as the science of electronics is now essential in all forms of modern devices and machinery, the same goes for the use of wireless technology for communication both between people themselves and between man and machine.

Control systems based on electronic and wireless technology are now considered not just as a viable but as a necessary solution for our times, where the need for fast pace and easier outcome of daily life and living becomes a constant need.

The controller concerning this master thesis can be used in a variety of applications, by installing this system in an external or internal environment to control a space and operate it wirelessly. The design and implementation of such an electronic wireless control system is therefore a useful technological existence for applications regarding people's daily lives, as well as a technological basis for its further development and use in future new applications, beyond those for which it is already intended.

Keywords: Microcontrollers, industrial control systems, wireless communication, peripheral units, electronic components.

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|--|------------|
| ΕΥΧΑΡΙΣΤΙΕΣ | I |
| ΠΕΡΙΛΗΨΗ | II |
| ABSTRACT | III |
| ΠΕΡΙΕΧΟΜΕΝΑ | IV |
| ΚΕΦΑΛΑΙΟ 1: Εισαγωγικές έννοιες | 1 |
| 1.1 Ασύρματες επικοινωνίες | 1 |
| 1.2 Συστήματα Ελέγχου | 2 |
| 1.2.1 Συστήματα Ασύρματου Ελέγχου για Βιομηχανικές Εφαρμογές | 3 |
| 1.3 Σκοπός της Διπλωματικής Εργασίας | 4 |
| ΚΕΦΑΛΑΙΟ 2: Ηλεκτρονικά εξαρτήματα – Υλικά κατασκευής | 5 |
| 2.1 Η Γενική Ιδέα | 5 |
| 2.2 Συσκευή Εισόδου | 6 |
| 2.2.1 Ενσύρματος Αναλογικός Μοχλός..... | 7 |
| 2.2.1.1 Μονάδα Αναλογικού Μοχλού 2 Αξόνων | 12 |
| 2.3 Προγραμματιζόμενα Ολοκληρωμένα Κυκλώματα | 16 |
| 2.3.1 Ο Μικροελεγκτής Arduino | 17 |
| 2.3.1.1 Arduino UNO R3 | 19 |
| 2.3.1.2 Arduino NANO EVERY | 25 |
| 2.3.1.3 Μικροελεγκτές AVR ATmega | 28 |
| 2.4 Μονάδες Ασύρματης Επικοινωνίας | 32 |
| 2.4.1 Πομποδέκτες Ραδιοσυχνότητας NRF24L01 | 32 |
| 2.4.1.1 Πομποδέκτες NRF24L01 με ενσωματωμένη κεραία | 43 |
| 2.5 Συσκευές Εξόδου | 44 |
| 2.5.1 Σερβοκινητήρες Συνεχούς Ρεύματος | 44 |
| 2.5.1.1 Ψηφιακοί Σερβοκινητήρες Συνεχούς Ρεύματος | 54 |
| 2.6 Επιλογές Τροφοδοσίας | 56 |
| 2.6.1 Τροφοδοσία Arduino UNO R3 | 56 |
| 2.6.1.1 Σύνδεση βύσματος Power Jack με μπαταρία | 56 |
| 2.6.1.2 Σύνδεση θύρας USB με H/Y | 61 |
| 2.6.2 Τροφοδοσία Arduino NANO EVERY | 64 |

| | | |
|--------------------|---|------------|
| 2.6.2.1 | Σύνδεση Ακροδεκτών με μπαταρία | 64 |
| 2.6.2.2 | Σύνδεση θύρας mini USB με H/Y | 65 |
| 2.6.2.3 | Κεφαλίδες Ακροδεκτών | 66 |
| 2.6.3 | Τροφοδοσία Πομποδεκτών NRF24L01 | 69 |
| 2.6.4 | Τροφοδοσία Σερβοκινητήρων | 70 |
| 2.6.4.1 | Σύνδεση Ακροδεκτών με μπαταρία | 70 |
| 2.7 | Ηλεκτρικά Καλώδια Δοκιμών | 72 |
| 2.8 | Πλακέτες διασύνδεσης χωρίς κολλήσεις | 74 |
| 2.9 | Δευτερεύουσες Συσκευές – Πρόσθετες Περιφερειακές Μονάδες | 76 |
| 2.9.1 | Αισθητήρας Υπερήχων | 77 |
| 2.9.2 | Πιεζοηλεκτρικός Βομβητής / Buzzer | 81 |
| 2.9.3 | Δίοδοι Εκπομπής Φωτός / LEDs | 83 |
| 2.9.4 | Μονάδα Αισθητήρα Επιταχυνσιόμετρου-Γυροσκοπίου 3 Αξόνων | 84 |
| 2.10 | Συγκεντρωτικός Πίνακας Συσκευών & Εξαρτημάτων | 89 |
| 2.11 | Συμπεράσματα | 98 |
| ΚΕΦΑΛΑΙΟ 3: | Σχεδιασμός – Προγραμματισμός – Υλοποίηση | 99 |
| 3.1 | Υποσύστημα Πομπού (Κύκλωμα TS) | 100 |
| 3.1.1 | Σύνδεση Χειριστηρίου & Arduino NANO EVERY | 100 |
| 3.1.1.1 | Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας | 106 |
| 3.1.1.2 | Προσθήκη 2 Σερβοκινητήρων για Έλεγχο Απόκρισης I/O ως Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή (Σ.Η.Ε.Ε.) για Βιομηχανικές Εφαρμογές | 131 |
| 3.1.1.2.1 | Έλεγχος Απόκρισης του Σ.Η.Ε.Ε. με ένταξη περαιτέρω περιφερειακών συσκευών I/O | 166 |
| 3.1.2 | Σύνδεση πομποδέκτη NRF24L01 & Arduino NANO EVERY | 177 |
| 3.1.2.1 | Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας | 180 |
| 3.2 | Υποσύστημα Δέκτη (Κύκλωμα RS) | 182 |
| 3.2.1 | Σύνδεση πομποδέκτη NRF24L01 & Arduino UNO R3 | 182 |
| 3.2.1.1 | Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας | 184 |
| 3.2.2 | Σύνδεση 2 Σερβοκινητήρων & Arduino UNO R3 | 188 |
| 3.3 | Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) για Βιομηχανικές Εφαρμογές | 192 |
| 3.3.1 | Ανάπτυξη Κώδικα για το Κύκλωμα TS | 196 |

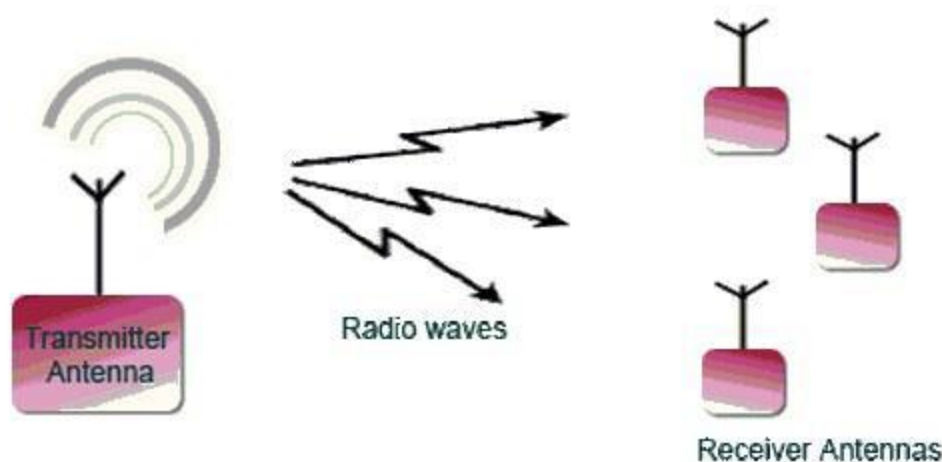
| | | |
|------------|---|------------|
| 3.3.2 | Ανάπτυξη Κώδικα για το Κύκλωμα RS | 197 |
| 3.3.3 | Έλεγχος Απόκρισης I/O για λειτουργία του συστήματος ως Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) | 198 |
| 3.4 | Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) με Πρόσθετες Περιφερειακές Μονάδες | 205 |
| 3.4.1 | 2 ^{ος} Αναλογικός Μοχλός Δύο Αξόνων & 2 Ψηφιακοί Σερβοκινητήρες Περιστροφής 360 μοιρών | 206 |
| 3.4.2 | Δοκιμή Διασύνδεσης Arduino UNO R3 & Αισθητήρα Υπερήχων HC-SR04 | 217 |
| 3.4.3 | Δοκιμή Διασύνδεσης Arduino UNO R3 & Buzzer | 220 |
| 3.4.3.1 | Σύστημα Συναγερμού – Ειδοποίησης & Ελέγχου της Κίνησης | 222 |
| 3.4.4 | Δοκιμή Διασύνδεσης των Arduino UNO & NANO με την Μονάδα Αισθητήρα Επιταχυνσιόμετρου-Γυροσκοπίου | 227 |
| 3.4.5 | Έλεγχος Απόκρισης του Σ.Η.Α.Ε. με Πρόσθετες Περιφερειακές Μονάδες | 245 |
| 3.5 | Ενδεικτική Ολοκληρωμένη Κατασκευή ενός Σ.Η.Α.Ε. για Βιομηχανικές Εφαρμογές | 271 |
| 3.5.1 | Διαδικασία Υλοποίησης της Ενδεικτικής Ολοκληρωμένης Κατασκευής του Σ.Η.Α.Ε. | 300 |
| 3.6 | Συμπεράσματα | 315 |
| | ΒΙΒΛΙΟΓΡΑΦΙΑ | 316 |

Κεφάλαιο 1:

Εισαγωγικές έννοιες

1.1 Ασύρματες επικοινωνίες

Με τον όρο ασύρματη επικοινωνία, γενικά, αναφερόμαστε στην μεταφορά σημάτων χωρίς την χρήση καλωδίων, με τον μηχανισμό λειτουργίας των ασύρματων επικοινωνιών να βασίζεται σε δύο κύρια μέρη, τον πομπό και τον δέκτη, αλλά και μία κεραία μέσω της οποίας συνδέονται μεταξύ τους αντί σύνδεσης μέσω σύρματος/καλωδίου.



Εικόνα 1.1 Ασύρματη επικοινωνία – Η κεραία του πομπού (transmitter antenna) αποστέλλει σήματα στις κεραίες των δεκτών (receiver antennas)

Η ασύρματη τεχνολογία έχει βοηθήσει σε μεγάλο βαθμό στην απλοποίηση του τρόπου της δικτύωσης, επιτρέποντας σε εφαρμογές που πραγματοποιούνταν με ενσύρματο τρόπο να εκτελούνται δίχως την ανάγκη τοποθέτησης πρόσθετων ή δύσκολα τοποθετήσιμων καλωδιώσεων. Επίσης, από πλευράς κόστους εγκατάστασης η ασύρματη μετάδοση μέσω κεραιών εκπομπής και λήψης έχει χαμηλότερο κόστος σε αντίθεση με τις ενσύρματες επικοινωνίες.

Για την πραγματοποίηση ασύρματης επικοινωνίας, η ασύρματη τεχνολογία χρησιμοποιεί διάφορους τύπους κεραιών, επηρεάζεται άμεσα από το περιβάλλον διάδοσης και χαρακτηρίζεται από διάφορα φαινόμενα που παρουσιάζονται σε αυτό, όπως είναι για παράδειγμα η ανάκλαση, η διάθλαση, η σκέδαση και η απορρόφηση λόγω ανομοιογένειας της επιφάνειας της γης αλλά και της ίδιας της ατμόσφαιρας που την περιβάλλει και η οποία αποτελείται από ραδιοσήματα που προκαλούν παρεμβολές.

1.2 Συστήματα Ελέγχου

Γενικά ως **σύστημα ελέγχου** ή **σύστημα ελεγκτή** (*control system* ή *controller system*) χαρακτηρίζεται μια ομάδα συσκευών και εξαρτημάτων που είναι σχεδιασμένη να λειτουργεί ενοποιημένα για να δίνει εντολές, να διαχειρίζεται, να κατευθύνει ή να ρυθμίζει την συμπεριφορά άλλων συσκευών ή συστημάτων χρησιμοποιώντας βρόχους ελέγχου.



Εικόνα 1.2 Σχηματικό διάγραμμα ενός απλού συστήματος ελέγχου – Σύστημα απόκρισης 1 εισόδου με 1 έξοδο

Ως σύστημα ελέγχου μπορεί να θεωρηθεί μια οικιακή συσκευή καθημερινής χρήσης όπως ένας ελεγκτής οικιακής θέρμανσης χρησιμοποιώντας έναν θερμοστάτη που ελέγχει έναν οικιακό λέβητα έως και μεγάλα βιομηχανικά συστήματα ελέγχου που χρησιμοποιούνται για τον έλεγχο διαδικασιών ή μηχανών.

Σε αντίθεση με την γενική έννοια των συστημάτων ελέγχου, ο όρος **βιομηχανικό σύστημα ελέγχου** (*Industrial Control System*) χαρακτηρίζει τους διάφορους τύπους συστημάτων ελέγχου και συναφών οργάνων που χρησιμοποιούνται για τον έλεγχο της βιομηχανικής διαδικασίας και των βιομηχανικών εφαρμογών.

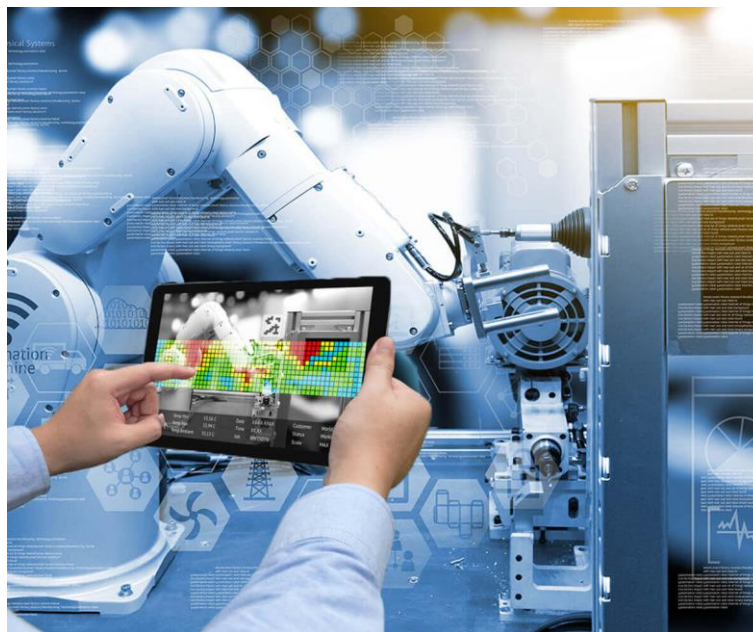
1.2.1 Συστήματα Ασύρματου Ελέγχου για Βιομηχανικές Εφαρμογές

Η ζήτηση εγκατεστημένων ασύρματων συστημάτων ελέγχου για βιομηχανικές εφαρμογές συνεχίζει να αυξάνεται κατά τα τελευταία χρόνια, λόγω των χαρακτηριστικών που είναι εγγενείς στην ασύρματη επικοινωνία, όπως ευελιξία, απλή εγκατάσταση και χαμηλό κόστος υλικών και εργασίας.

Ωστόσο, υπάρχουν πολλοί τομείς της τεχνολογίας που εξακολουθούν να εκφράζουν δισταγμό ως προς την επένδυση σε ασύρματες λύσεις λόγω παραμέτρων που σχετίζονται με την αξιοπιστία και την ασφάλεια των υφιστάμενων ασύρματων προτύπων αλλά και τη συμβατότητα με τις καθιερωμένες υποδομές δικτύου ελέγχου.

Ο σκεπτικισμός σχετικά με τα ασύρματα συστήματα ελέγχου εξακολουθεί να υφίσταται και παρότι η δυνατότητα κλιμάκωσης και η ευελιξία σπάνια αμφισβητούνται, τα θέματα ασφάλειας, διαλειτουργικότητας και αξιοπιστίας από μόνα τους προκαλούν αμφισβήτηση ως προς την αντικατάσταση των ήδη εγκατεστημένων ενσύρματων συστημάτων με αντίστοιχα ασύρματα.

Εν τω μεταξύ, οι ασύρματες τεχνολογίες και τα πρωτόκολλα επικοινωνίας έχουν πλέον ωριμάσει αρκετά, παρέχοντας βελτιωμένη ασφάλεια, προστασία από παρεμβολές και αξιοπιστία ως προς την λειτουργία και την απόδοση.



Εικόνα 1.3 Ασύρματη οθόνη αφής για παρακολούθηση και χειρισμό βιομηχανικών διεργασιών

1.3 Σκοπός της Διπλωματικής Εργασίας

Η παρούσα Διπλωματική εργασία εστιάζει στον σχεδιασμό, τον προγραμματισμό και την υλοποίηση ενός συστήματος απόκρισης εισόδου/εξόδου βάσει του οποίου παρέχεται η δυνατότητα εκτέλεσης και ελέγχου εφαρμογών Βιομηχανικής φύσεως. Επειδή οι Βιομηχανικές Εφαρμογές περιλαμβάνουν πολυάριθμους τομείς και λειτουργικές διαδικασίες, είναι λογικό και κατανοητό να μην μπορούν να καλυφθούν όλες –τουλάχιστον από πλευράς υλοποίησης– σαν ενιαίο αντικείμενο εκπόνησης αποκλειστικά και μόνο μιας διπλωματικής εργασίας.

Βάσει των ανωτέρω, γίνεται ανάπτυξη ενός συστήματος ελέγχου που αποσκοπεί μεν σε συγκεκριμένη λειτουργία, μα με τρόπο τέτοιο που με αλλαγές ή τροποποιήσεις στον προγραμματιστικό του κώδικα και τα περιφερειακά του μέρη να δύναται να λειτουργεί σε περισσότερους του ενός τομείς που άπτονται της κατηγορίας των Βιομηχανικών Εφαρμογών. Συγκεκριμένα υλοποιείται ένα ασύρματο σύστημα απόκρισης εισόδου/εξόδου μέσω του οποίου ένας χρήστης δύναται από χειριστήριο να ελέγχει ασύρματα και από ικανοποιητική απόσταση την λειτουργία δύο σερβοκινητήρων αλλά και να δέχεται ειδοποίηση στο χειριστήριο για την αλλαγή μιας κατάστασης στον γειτονικό του χώρο. Η υλοποίηση πραγματοποιείται με τρόπο τέτοιο ώστε με περαιτέρω τοποθέτηση ή τροποποίηση στην έξοδο του συστήματος απόκρισης –ή σε οποιοδήποτε άλλο σημείο επί του συστήματος– διαφορετικών μηχανισμών το εν λόγω σύστημα να δύναται να αξιοποιηθεί σε πληθώρα εφαρμογών.

Με την εύρεση των απαραίτητων υλικών και εξαρτημάτων για την κατασκευή να διαδραματίζεται στο 2^ο κεφάλαιο και την υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή να λαμβάνει μέρος στο κεφάλαιο 3 με μεγάλη έμφαση στην ευελιξία του ως προς τις εφαρμογές στις οποίες δύναται να χρησιμοποιείται βάσει του εκάστοτε σχεδιασμού και προγραμματισμού του, στο 3^ο και τελευταίο κεφάλαιο παρατίθεται επίσης μία ενδεικτική ολοκληρωμένη κατασκευή ενός Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή ως σύστημα απόκρισης I/O για προοριζόμενο για Βιομηχανικές Εφαρμογές.

Με περαιτέρω προσαρμογές, αλλαγές ή και επεκτάσεις στο λογισμικό και υλικό του περιεχόμενο, το εν λόγω κατασκευασμένο Σ.Η.Α.Ε. καθίσταται ικανό να καλύπτει και άλλες βιομηχανικές εφαρμογές πέραν αυτών για τις οποίες ήδη προορίζεται βάσει των περιφερειακών μονάδων που το απαρτίζουν, καθώς ο σχεδιασμός του ως σύστημα απόκρισης I/O είναι τέτοιος που με συνεχή προσθήκη περιφερειακών συσκευών και εξαρτημάτων σε αυτό δύναται να αυξάνει και το εύρος των εφαρμογών που μπορεί να καλύψει.

Κεφάλαιο 2: **Ηλεκτρονικά εξαρτήματα – Υλικά κατασκευής**

2.1 Η Γενική Ιδέα

Ως γενική ιδέα, προτείνεται η ενσύρματη επικοινωνία μιας συσκευής εισόδου (*χειριστήριο, πληκτρολόγιο*) με ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα (*μικροελεγκτής*) και περιφερειακές συσκευές ώστε να μπορούν να λειτουργήσουν ως ένα ενοποιημένο υποσύστημα που δέχεται εντολές από έναν χρήστη, τις επεξεργάζεται και τις μεταφέρει με την μορφή δεδομένων ασύρματα σε ένα δεύτερο υποσύστημα, παρόμοιου σχεδιασμού με το πρώτο, που λαμβάνει τα δεδομένα και τα οδηγεί σε επιθυμητή έξοδο, δημιουργώντας έτσι ένα σύστημα ηλεκτρονικού ασύρματου ελεγκτή.

Για την εύρυθμη λειτουργία των δύο υποσυστημάτων ως ένα ενιαίο ηλεκτρονικό σύστημα ασύρματου ελέγχου ικανό να λειτουργεί σε οποιοδήποτε επιθυμητό περιβάλλον, αλλά και την αξιοποίηση των υποσυστημάτων αυτών τόσο ως κινητές όσο και ως ακίνητες μονάδες για περαιτέρω αυτονομία και ευελιξία χειρισμού και τοποθέτησης, απαιτείται η μεταξύ τους επικοινωνία να πραγματοποιείται αποκλειστικά με ασύρματο τρόπο αλλά η τροφοδοσία τους να μπορεί να παρέχεται επιλεκτικά τόσο από φορητές όσο και από σταθερές εξωτερικές πηγές τροφοδοσίας.

2.2 Συσκευή Εισόδου

Στην επιστήμη της Πληροφορικής, των Υπολογιστικών Συστημάτων και των Τηλεπικοινωνιών, ως συσκευή εισόδου θεωρείται οποιαδήποτε περιφερειακή μονάδα λειτουργεί ως μέρος ενός συστήματος ή συνόλου μηχανημάτων και μέσω της οποίας εισάγονται πληροφορίες –με την μορφή δεδομένων– από το εξωτερικό περιβάλλον στο σύστημα αυτό.

Τα είδη συσκευών εισόδου μπορεί να διατίθενται σε διάφορα σχήματα και μεγέθη και οι άνθρωποι χρησιμοποιούν τέτοιες συσκευές σε ευρύ φάσμα της καθημερινής τους ζωής, είτε αυτές φέρουν την μορφή πηδαλίου (*τιμόνι*) και συστημάτων πέδησης-επιτάχυνσης (*φρένο-γκάζι*) για τον έλεγχο της κίνησης ενός οχήματος, τηλεχειριστηρίων για ασύρματο χειρισμό τηλεοράσεων και κλιματιστικών, κομβίων και οθονών αφής για την διαχείριση τραπεζικών λογαριασμών και καταθέσεων σε κάποιο ΑΤΜ, ενσύρματων ή μη πληκτρολογίων και ποντικιών για χειρισμό επιτραπέζιων και φορητών ηλεκτρονικών υπολογιστών, μοχλών για έλεγχο θυρών και ανοιγμάτων γενικότερα, χειριστηρίων για βιντεοπαιχνίδια κ.α.

Όσον αφορά τον τρόπο παροχής εντολών για έλεγχο μιας εφαρμογής μέσω του ηλεκτρονικού ασύρματου συστήματος ελέγχου που υλοποιείται στην παρούσα διπλωματική εργασία, για εντολές εισόδου παρεχόμενες από τον χρήστη, θα μπορούσε να γίνει χρήση είτε μιας ενσύρματης συσκευής εισόδου για την απλότητα του σχεδιασμού είτε μιας ασύρματης για περαιτέρω ενίσχυση του σήματος προς το σημείο ελέγχου, αλλά προσπαθώντας να διατηρήσουμε χαμηλό τον προϋπολογισμό του κόστους κατασκευής –καθότι κατά κόρον οι διαδικασίες ασύρματης επικοινωνίας κοστίζουν συγκριτικά με τις συνδεσμολογίες μέσω καλωδίων– γίνεται επιλογή μιας ενσύρματης συσκευής εισόδου και συγκεκριμένα μιας συσκευής ελέγχου εισόδου με χειροκίνητα ρυθμιζόμενη αντίσταση.

2.2.1 Ενσύρματος Αναλογικός Μοχλός

Μία από τις απλούστερες σύγχρονες παραλλαγές των συσκευών ελέγχου εισόδου αποτελείται από έναν αρκετά μικρό μοχλό που βρίσκεται ενσωματωμένος και περιστρέφεται επάνω σε μια βάση και είναι ευρέως γνωστή ως **αναλογικός μοχλός** (*analog joystick*). Με κάθε περιστροφή του μοχλού, ο μηχανισμός της βάσης λαμβάνει τις κινήσεις αυτές ως είσοδο στο όλο σύστημα και έτσι μεταφέρει εντολές στην έξοδο (*ακροδέκτες*).



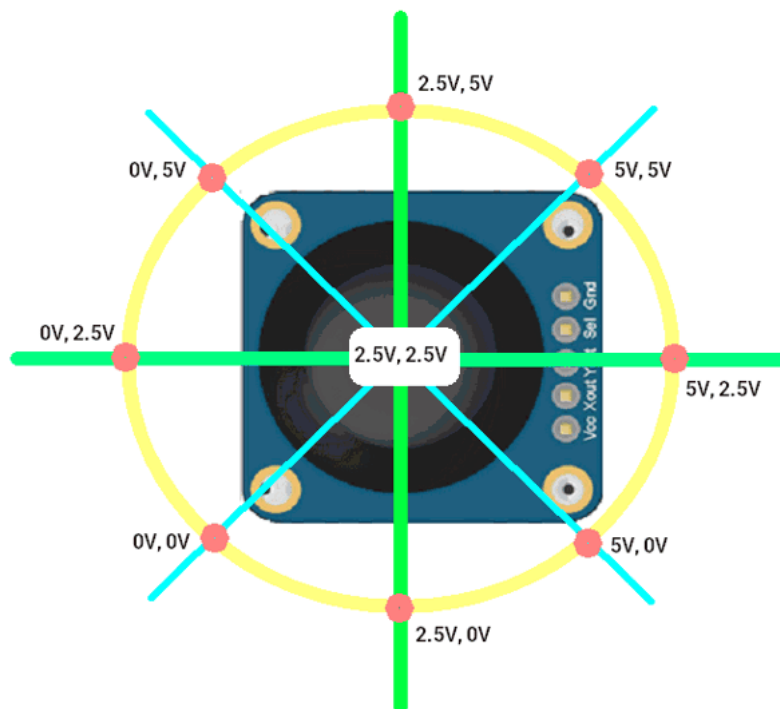
Εικόνα 2.1 Διάφορα μοντέλα αναλογικών χειριστηρίων με μοχλό ολίσθησης

Λόγω της απλότητας του σχεδιασμού τους, οι αναλογικοί μοχλοί χρησιμοποιούνται σε πολλές βιομηχανικές εφαρμογές.

Ο αναλογικός μοχλός μετατρέπει την μηχανική είσοδο λόγω κίνησης σε ηλεκτρικό σήμα μέσω ενός βιομηχανικού ποτενσιόμετρου, μιας χειροκίνητα ρυθμιζόμενης (μεταβλητής) αντίστασης τριών ακροδεκτών με ολισθαίνουσα ή αλλιώς περιστρεφόμενη επαφή που σχηματίζει έναν ρυθμιζόμενο διαιρέτη τάσης (*adjustable voltage divider*). Μετρά ηλεκτρεγερτική δύναμη εξισορροπώντας την έναντι της διαφοράς δυναμικού που παράγεται με την διέλευση ρεύματος μέσω μεταβλητής αντίστασης.

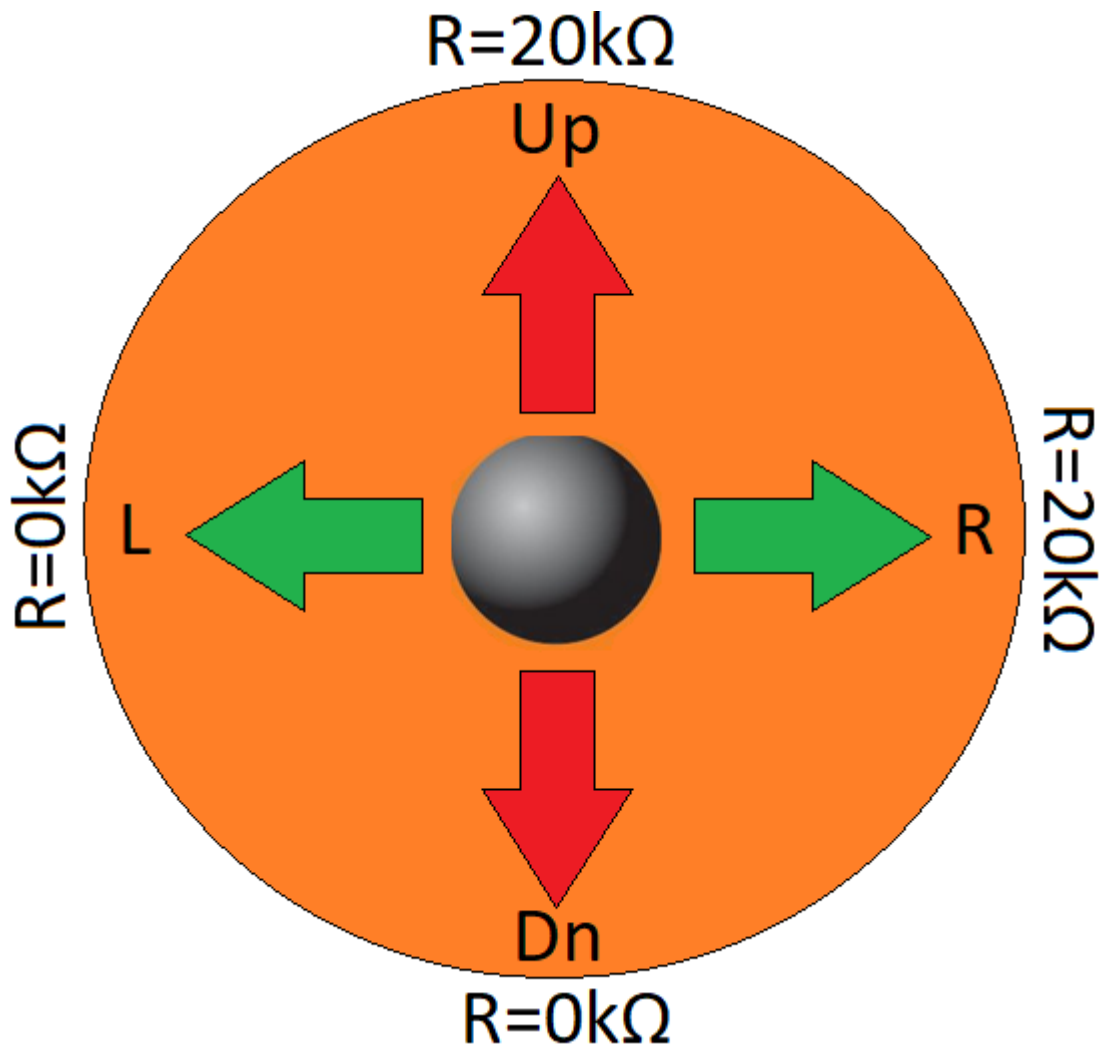
Η συνηθέστερη έκδοση των αναλογικών μοχλών ορίζει ότι ο μοχλός που βρίσκεται ενσωματωμένος επάνω στην βάση πρέπει να κινείται σε άξονα δύο διαστάσεων. Διαθέτοντας δύο ποτενσιόμετρα για να λαμβάνει την μηχανική είσοδο από έναν χρήστη, το σύστημα παράγει δύο τάσεις που σχετίζονται με την κίνηση στους δύο άξονες – μια που αντιστοιχεί σε μετατόπιση ως προς τον άξονα X και μια που αντιστοιχεί σε μετατόπιση ως προς τον άξονα Y – ενώ τα ποτενσιόμετρα συνδέονται μεταξύ τάσης τροφοδοσίας (+ VCC) και γείωσης (GND) για την δημιουργία του προαναφερθέντος διαιρέτη τάσης.

Στην εικόνα που ακολουθεί οι παραγόμενες τάσεις εξαρτώνται από τη θέση του μοχλού ως προς τους άξονες X και Y.



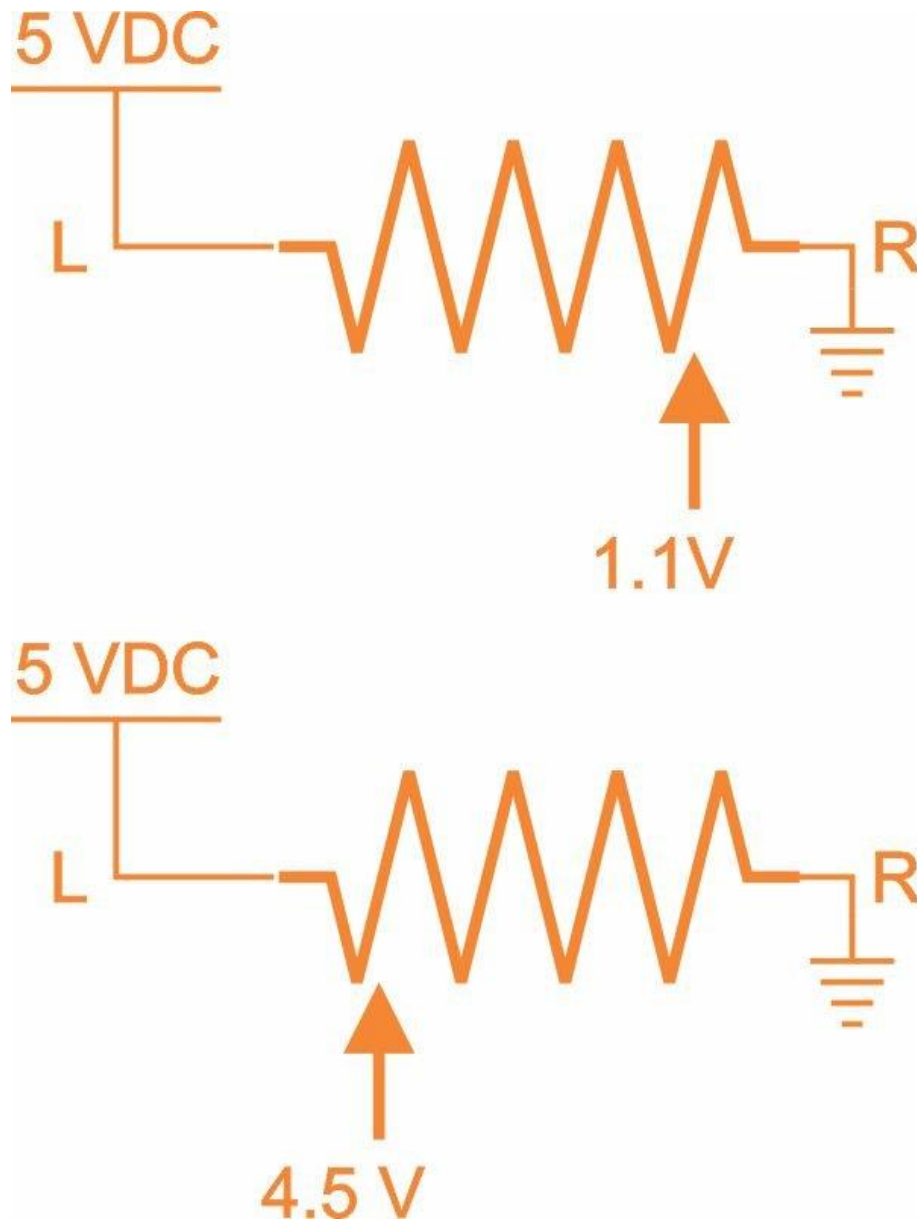
Εικόνα 2.2 Τα ποτενσιόμετρα του συστήματος του αναλογικού μοχλού «διαβάζουν» την κίνησή του και αντιστοιχίζουν αυτήν την μηχανική μετατόπιση σε μεταβλητή αντίσταση

Βάσει των χαρακτηριστικών σχεδιασμού ενός αναλογικού μοχλού, αν για παράδειγμα διαθέτει μια μεταβλητή αντίσταση των $20\text{k}\Omega$, η κεντρική θέση (ακινησία) του μοχλού θα αντιστοιχεί σε αντίσταση τιμής $R=10\text{k}\Omega$, με το αριστερό άκρο να αντιστοιχεί σε αντίσταση $R=0\text{k}\Omega$ και το δεξί άκρο σε $R=20\text{k}\Omega$ για τον άξονα των X , με αντίστοιχες τιμές να ανταποκρίνονται στην άνω και κάτω μετατόπιση που αναφέρεται στον άξονα των Y .



Εικόνα 2.3 Η μετατόπιση του αναλογικού μοχλού μεταβάλλει την αντίσταση των δύο ποτενσιομέτρων και κατά συνέπεια την τάση.

Σε έναν αναλογικό μοχλό, αν για παράδειγμα του αποδίδεται τάση τροφοδοσίας της τάξεως των 5V, σχηματίζοντας μια σταθερή διαφορά δυναμικού μεταξύ των εξωτερικών ακίδων της μεταβλητής αντίστασης, η διαφορά δυναμικού που μετριέται μεταξύ ακίδας και γείωσης θα είναι ανάλογη της μετατόπισης που κυμαίνεται μεταξύ 0-5V, όπως φαίνεται στην εικόνα 2.4 που ακολουθεί.



Εικόνα 2.4 Με μηχανική (χειροκίνητη) μετατόπιση του μοχλού στον άξονα X (Left/Right) παρατηρείται σχηματισμός ανάλογης διαφοράς δυναμικού, με ακριβώς αντίστοιχο φαινόμενο να προκύπτει με μετατόπιση ως προς τον Y άξονα (Up/Down). Τα ποτενσιόμετρα λειτουργούν ως διαιρέτες τάσης με τιμή ανάλογη της μετατόπισης του μοχλού.

Ο ενσύρματος αναλογικός μοχλός ρυθμιζόμενης αντίστασης προορίζεται για χρήση ως συσκευή εισόδου σε ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα, έναν μικροελεγκτή για την ακρίβεια.

Οι μικροελεγκτές δεν έχουν την δυνατότητα να μετρούν άμεσα μια αντίσταση, μπορούν όμως να ερμηνεύουν άμεσα την διαφορά δυναμικού που αναπτύσσεται επάνω στις γραμμές εισόδου του συστήματος των μοχλών, οπότε ως συσκευή που χρησιμοποιεί μια ρυθμιζόμενη αντίσταση για την δημιουργία μεταβλητής διαφοράς δυναμικού, ο αναλογικός μοχλός μπορεί κάλλιστα να λαμβάνει τον ρόλο τέτοιου είδους μεταβλητής αντίστασης.

Οι ενσύρματοι αναλογικοί μοχλοί είναι αρκετά φθηνά, εύκολοι στην χρήση και διατίθενται στην αγορά σε αφθονία, δεν παύουν όμως να διαθέτουν και μειονεκτήματα.

Εκτός του ενσύρματος τρόπου λειτουργίας τους που παρέχει περιορισμένη απόσταση χειρισμού, καταστάσεις όπως συγκέντρωση σκόνης λόγω πολυετούς ακινησίας, σχηματισμός οξειδωσης λόγω επαφής με το περιβάλλον, εμφάνιση φυσικής φθοράς λόγω εκτεταμένης χρήσης ή άλλοι παράγοντες όπως η ανάπτυξη υγρασίας, προκαλούν ανεπιθύμητες διακυμάνσεις στις αναγνώσεις των επαφών, καθιστώντας πολλές φορές λιγότερο αξιόπιστη την χρήση του αναλογικού μοχλού ως συσκευή ελέγχου εισόδου.

Παρόλα αυτά, απαιτείται αρκετά μεγάλη καταπόνηση και εκτεταμένη χρήση αυτών για εμφάνιση τέτοιων ανεπιθύμητων καταστάσεων και άρα για την εν λόγω κατασκευή ένας ενσύρματος αναλογικός μοχλός τέτοιου τύπου καλύπτει τα μέγιστα στην εξυπηρέτηση του σκοπού για τον οποίο προορίζεται.

Για τον σχεδιασμό και την υλοποίηση του συστήματος ηλεκτρονικού ασύρματος ελεγκτή συγκεκριμένα, γίνεται επιλογή μιας έκδοσης ενσύρματος αναλογικού μοχλού που διαθέτει δύο ποτενσιόμετρα αντίστασης 10kΩ έκαστο.

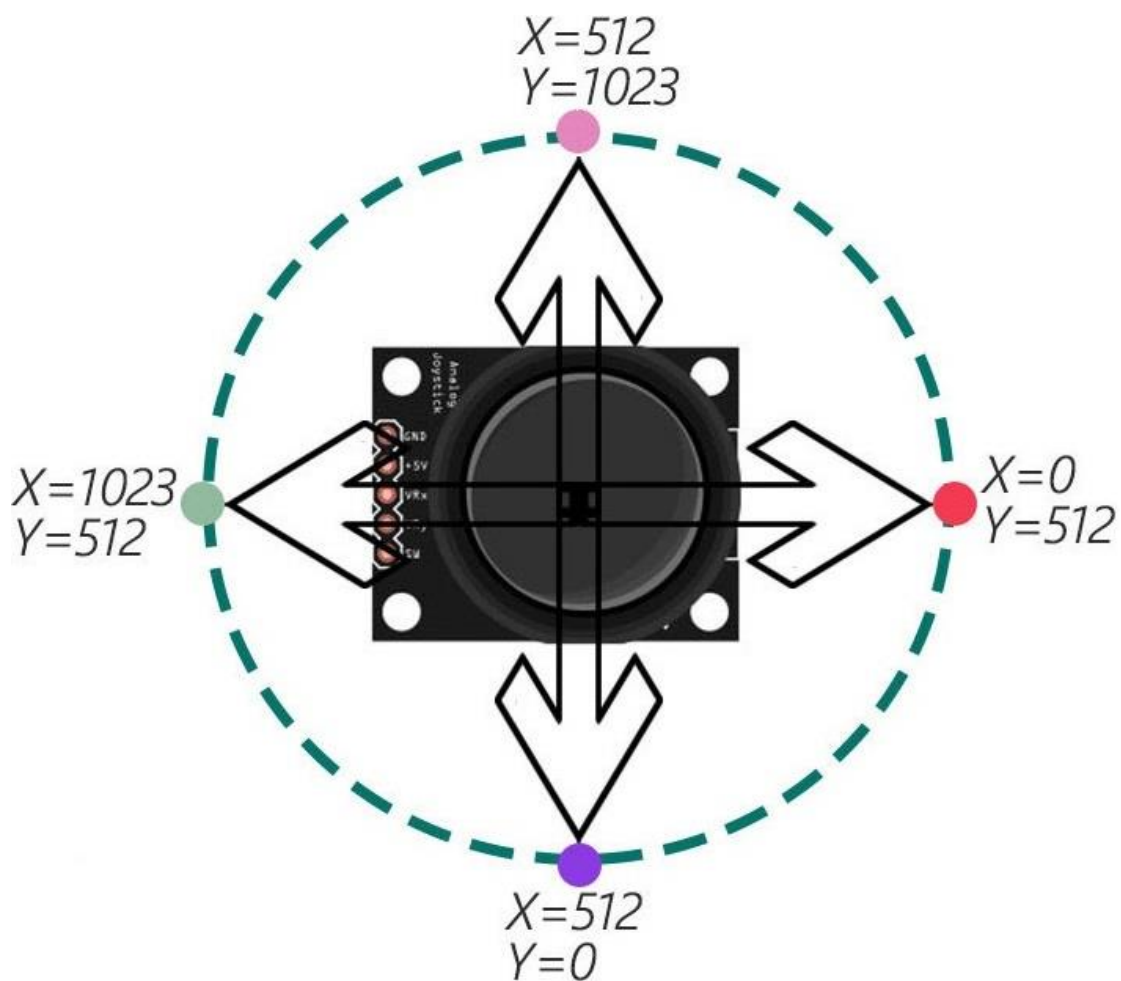
2.2.1.1 Μονάδα Αναλογικού Μοχλού 2 Αξόνων

Αποτελώντας ένα ποιοτικό και χαμηλού κόστους ενσύρματο αναλογικό μοχλό, η **Μονάδα Αναλογικού Μοχλού 2 Αξόνων** (*Dual-axis Joystick Module*) με δύο ποτενσιόμετρα των 10kΩ έκαστο χρησιμοποιείται συχνά σε διάφορες ηλεκτρονικές διεπαφές ελέγχου και στην ρομποτική, λόγω κατασκευής της είναι αρκετά ανθεκτική και απαλή στην αφή για τον χρήστη και άρα αποτελεί μια ιδανική επιλογή για αξιοποίησή της ως συσκευή εισόδου δεδομένων από έναν χρήστη στο σύστημα.



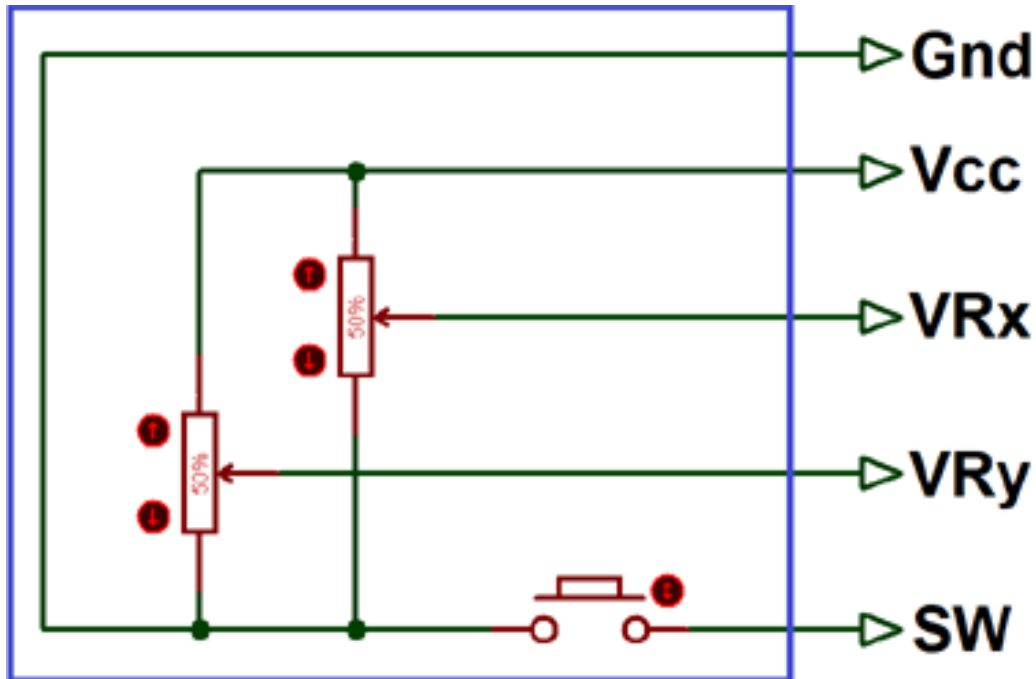
Εικόνα 2.5 Ενσύρματος αναλογικός μοχλός 2 αξόνων, δύο ποτενσιόμετρων των 10kΩ

Ο εν λόγω μοχλός διαθέτει δύο ποτενσιόμετρα, παρέχοντας αναλογικά σήματα X και Y τα οποία λαμβάνονται από τις αναλογικές εισόδους των μικροελεγκτών με τιμές εύρους 0-1023 σε μια τυπική ανάγνωση μιας μονάδας μικροελεγκτή (για τυπικές εισόδους ανάγνωσης μετατροπέα αναλογικού σήματος σε ψηφιακό ADC 10-bit των Arduino), με μια μεσαία τιμή γύρω στο $(1024/2 =) 512$ να διαβάζεται και από τις δύο εξόδους X και Y του χειριστηρίου όταν ο μοχλός βρίσκεται σε ηρεμία (μεσαίο σημείο). Καθώς ο μοχλός ολισθαίνει, μία ή και οι δύο έξοδοι θα καταγράφουν υψηλότερες ή χαμηλότερες τιμές ανάλογα με τον τρόπο ολίσθησης.



Εικόνα 2.6 Ο αναλογικός μοχλός 2 αξόνων δύναται να εκτελεί περιστροφική κίνηση 360 μοιρών, ολισθαίνοντας στις 2 διαστάσεις των αξόνων X & Y

Ο ίδιος ο μοχλός δύναται επίσης να λειτουργεί ως ηλεκτρικό κομβίο ή διακόπτης μπουτόν (*switch / push button*) που όμως αποτελεί ψηφιακή είσοδο σε αντίθεση με τα αναλογικά σήματα των δύο ποτενσιόμετρων. Με ελαφριά πίεση προς την βάση, ο μοχλός μετατρέπεται σε μπουτόν καθώς με το πάτημα η αρχικά ανοικτή επαφή του συνδέεται εσωτερικά σε αγωγό γείωσης (*GND*).

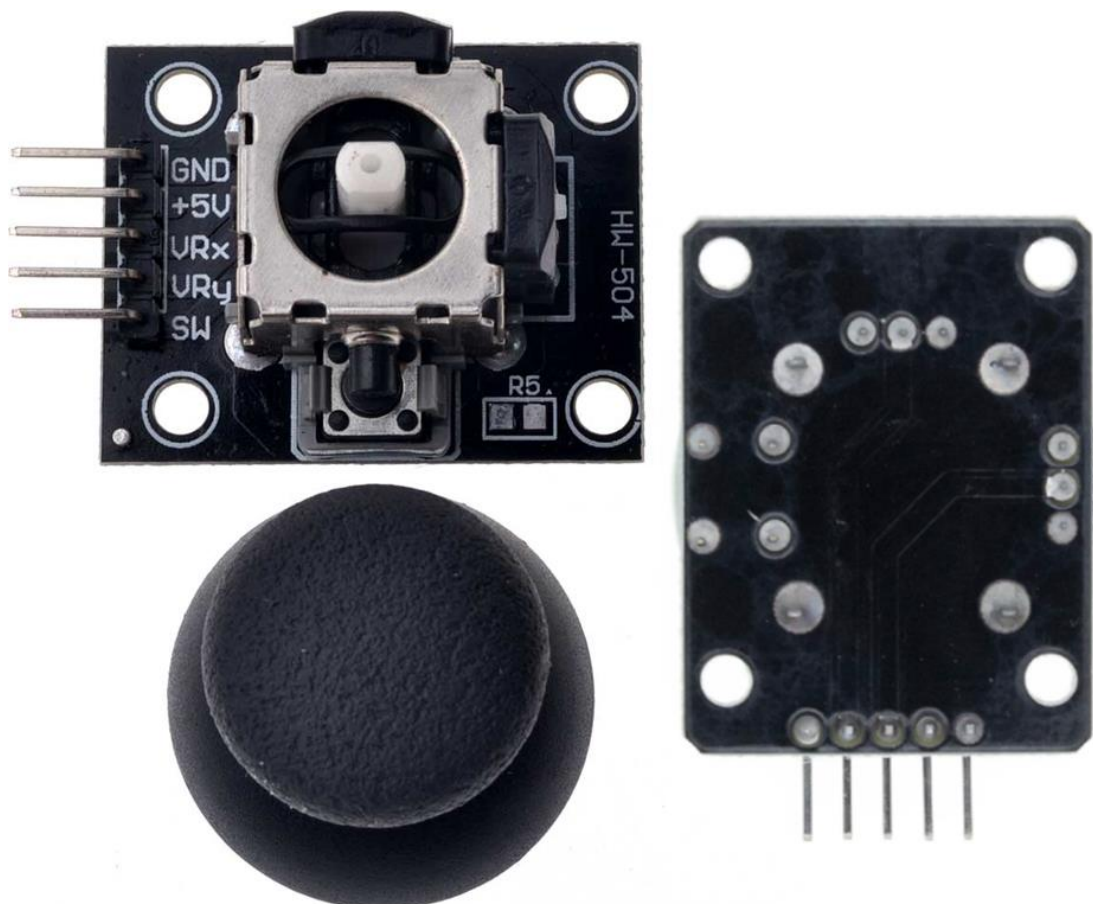


Εικόνα 2.7 Εσωτερική διασύνδεση του συστήματος ενός αναλογικού μοχλού 2 αξόνων, διαθέτοντας δύο ποτενσιόμετρα (*VRX, VRY*) και ένα μπουτόν (*SW*)

Ο αναλογικός μοχλός δύο αξόνων διαθέτει επίσης ένα σύστημα ελατηρίου (*spring system*) για αυτόματη επαναφορά του μοχλού στην αρχική του θέση ηρεμίας (*μεσαίο σημείο*) όταν αυτός δεν χρησιμοποιείται από τον χρήστη και άρα δεν μετατοπίζεται.

Τεχνικές Λεπτομέρειες Αναλογικού μοχλού

- Τιμή αντίστασης εκάστου ποτενσιόμετρου: 10kΩ
- Τάση λειτουργίας: 5V
- Συμβατότητα ακροδεκτών με τους μικροελεγκτές Arduino:
 1. GND: γείωση
 2. +5V: Τάση 5V DC
 3. VRx (*analog*): Τάση ανάλογη της μετατόπισης του μοχλού στον X άξονα
 4. VRy (*analog*): Τάση ανάλογη της μετατόπισης του μοχλού στον Y άξονα
 5. SW (*digital*): διακόπτης μπουτόν
- Διαστάσεις προϊόντος: 4 x 2,6 x 3,2 cm



Εικόνα 2.8 Άνω και κάτω όψη του αναλογικού μοχλού 2 αξόνων, με ορατό το εσωτερικό του άξονα του μοχλού

2.3 Προγραμματιζόμενα Ολοκληρωμένα Κυκλώματα

Με την τεχνολογία να εξελίσσεται ραγδαία και να επεκτείνεται σε όλους τους τομείς της σύγχρονης εποχής, τέτοια κυκλώματα που απαρτίζουν πλέον τις περισσότερες ηλεκτρονικές και ηλεκτρικές συσκευές είναι οι μικροελεγκτές.

Οι μικροελεγκτές (*microcontrollers* ή *MCUs*) είναι προγραμματιζόμενα ολοκληρωμένα κυκλώματα – ουσιαστικά ένας τύπος κεντρικής μονάδας επεξεργασίας (*CPU*) – που μπορούν να λειτουργούν με ελάχιστα εξωτερικά εξαρτήματα λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτουν.

Όπως ακριβώς ένας μικροϋπολογιστής, οι μικροελεγκτές διαθέτουν επεξεργαστή, μνήμη, διάφορα περιφερειακά κυκλώματα και θύρες εισόδου/εξόδου για επικοινωνία με άλλες συσκευές, με την βασική ιδιαιτερότητα ότι όλα τα υλικά τους μέρη διατίθενται σε ένα μόνο κύκλωμα ή αλλιώς μικροτσιπ (*microchip*), εξ ου και ο χαρακτηρισμός τους ως ολοκληρωμένα κυκλώματα (*integrated circuits*).

Εξαιτίας του αρκετά μικρού μεγέθους τους, της υψηλής επεξεργαστικής/υπολογιστικής τους ισχύος, της ευελιξίας ως προς την επικοινωνία και συνεργασία τους με άλλες συσκευές και εξαρτήματα, του χαμηλού κόστους σχεδιασμού και κατασκευής τους, αλλά και της αυτονομίας και αξιοπιστίας που προκύπτουν εκ των ανωτέρω, καθιστούν την χρήση μικροελεγκτών σε διάφορους τομείς της τεχνολογίας – είτε πρόκειται για χρήση στην καθημερινή ζωή των ανθρώπων είτε για εξειδικευμένο σκοπό, σε Βιομηχανικές ή μη εφαρμογές – μείζονος σημασίας.

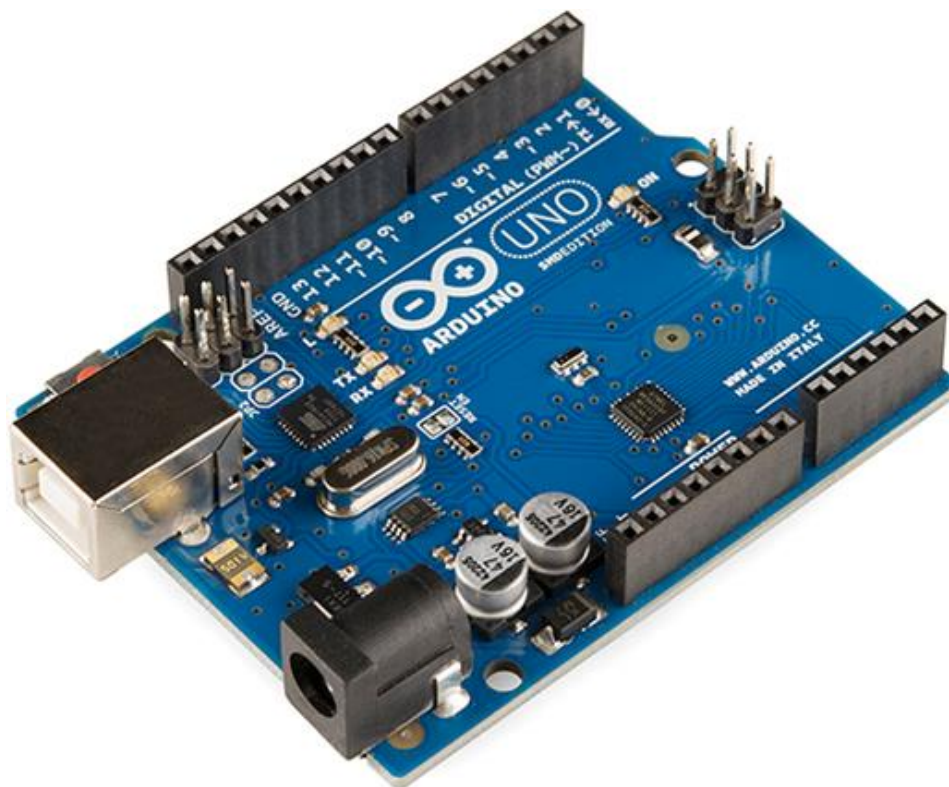
Για τον σχεδιασμό και μετέπειτα υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή γίνεται επιλογή δύο συγκεκριμένων μικροελεγκτών από την οικογένεια μικροελεγκτών που ονομάζεται *Arduino*.

2.3.1 Ο Μικροελεγκτής Arduino

Ο μικροελεγκτής Arduino είναι μια ηλεκτρονική πλακέτα ανοικτού κώδικα που αναπτύχθηκε από την εταιρεία Arduino.cc και αποτελείται από έναν μικροελεγκτή και μονάδες εισόδου/εξόδου.

Η πλακέτα Arduino δέχεται προγραμματιστικό κώδικα γλώσσας C και C++, συνοδευόμενο από μια σειρά βιβλιοθηκών υλοποιημένων επίσης στις ίδιες γλώσσες προγραμματισμού.

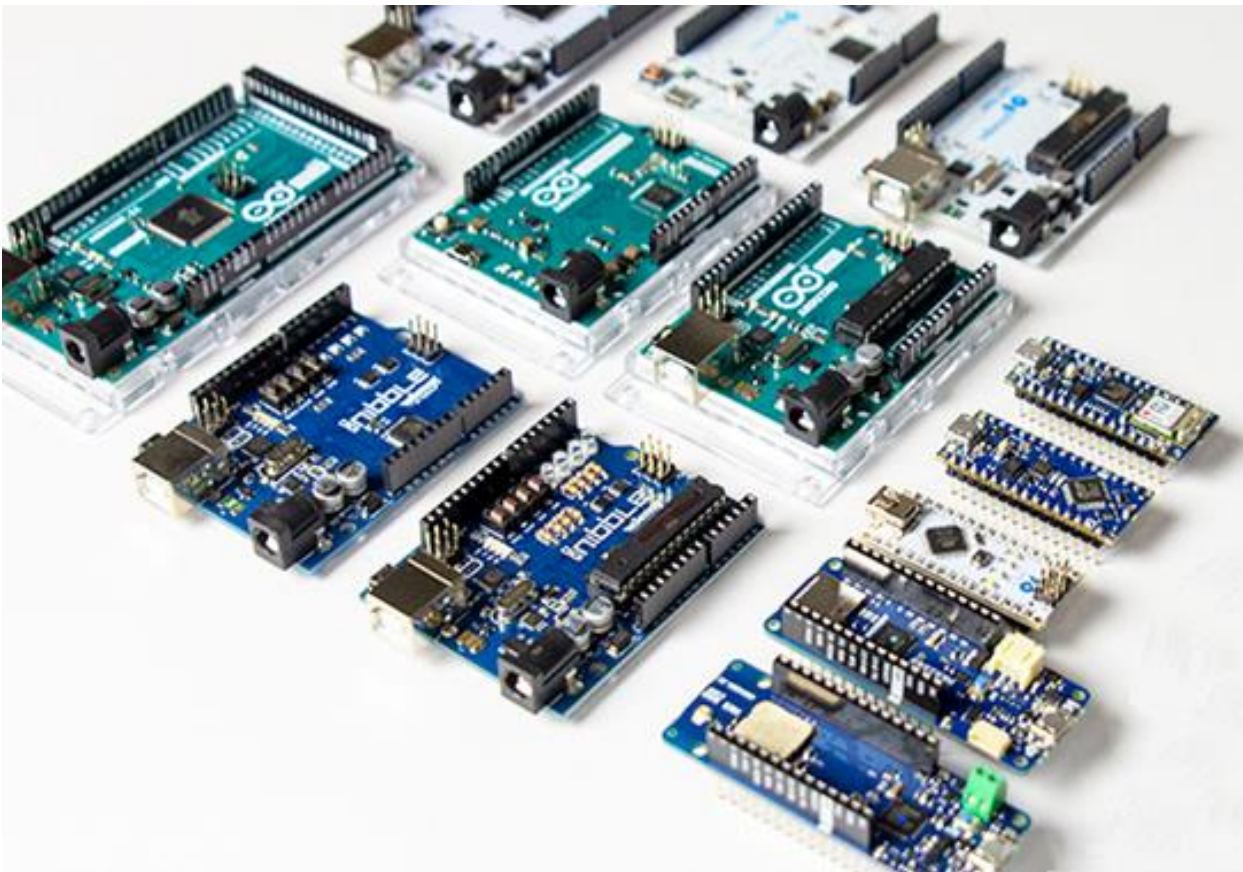
Οι περισσότερες πλακέτες Arduino αποτελούνται από έναν μικροελεγκτή Atmel AVR των 8-bit (*ATmega328 & ATmega168* για νεότερες εκδόσεις και *ATmega8* για παλαιότερες) και συμπληρωματικά εξαρτήματα (μνήμη *FLASH*, ακροδέκτες, θύρα *USB*, προγραμματισμό *bootloader* κλπ) για την διευκόλυνση του χρήστη στον προγραμματισμό και την απλή και εύκολη ενσωμάτωση του Arduino σε άλλα ηλεκτρονικά κυκλώματα και εξαρτήματα.



Εικόνα 2.9 Ηλεκτρονική πλακέτα Arduino UNO R3 (έκδοση SMD/Surface-Mount-Device), μια από τις συνηθέστερες εκδόσεις ηλεκτρονικών πλακετών της οικογένειας μικροελεγκτών Arduino.

Το Arduino υλοποιείται σε διάφορες εκδόσεις, με την πλειονότητα των διαφοροποιήσεων κατασκευής να δίνει έμφαση κυρίως στο μέγεθος της πλακέτας, στην κατανάλωση ισχύος από αυτήν, στην χωρητικότητα της μνήμης του μικροελεγκτή και την επεξεργαστική του ισχύ και φυσικά στο κόστος προμήθειας.

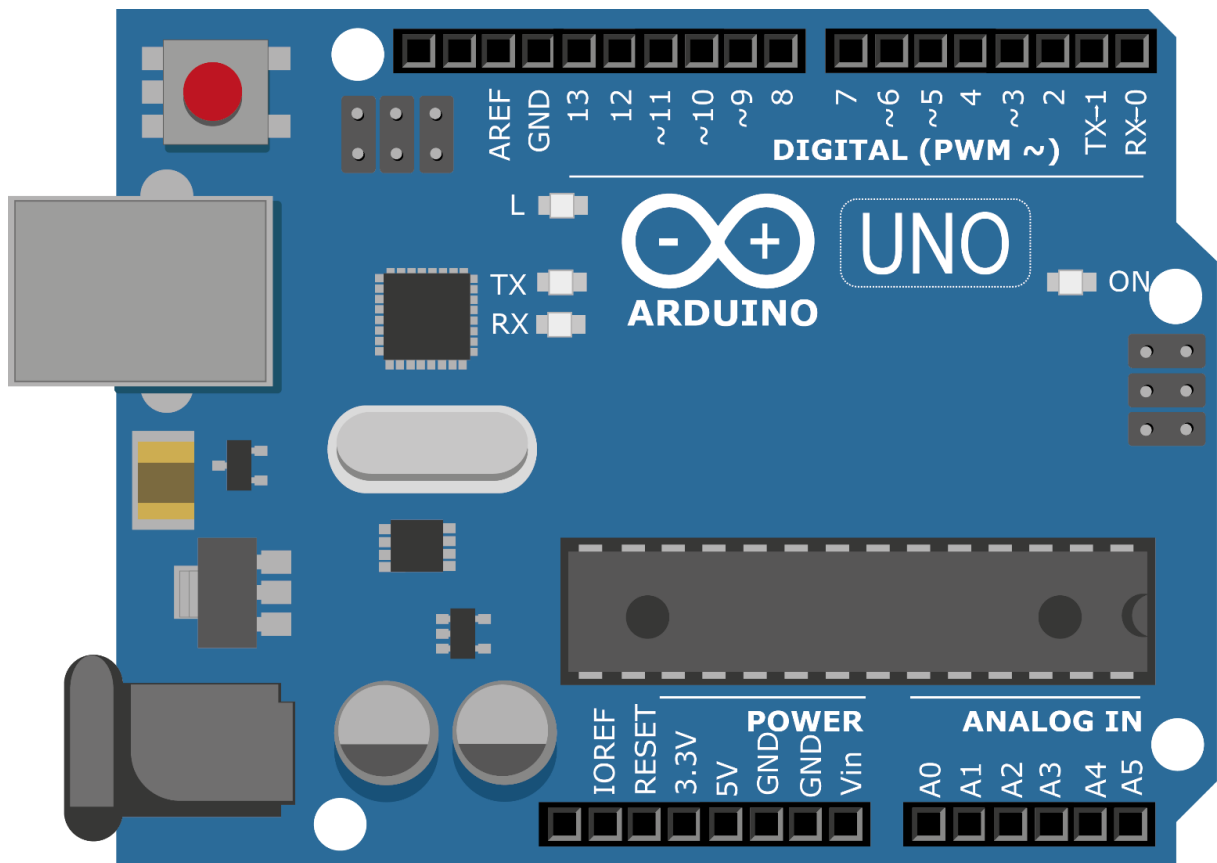
Καθώς στην ενότητα 2.1 του παρόντος κεφαλαίου έγινε αναφορά στον σχεδιασμό δύο υποσυστημάτων με λειτουργία πομπού και δέκτη αντίστοιχα για την υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή, για τους ανωτέρω λόγους γίνεται επιλογή δύο διαφορετικών εκδόσεων Arduino στο εκάστοτε υποσύστημα, αντί δύο ίδιων μοντέλων. Συγκεκριμένα, η έκδοση Arduino NANO EVERY επιλέγεται για τον σχεδιασμό του υποσυστήματος πομπού ενώ η έκδοση Arduino UNO R3 επιλέγεται για τον σχεδιασμό του υποσυστήματος δέκτη.



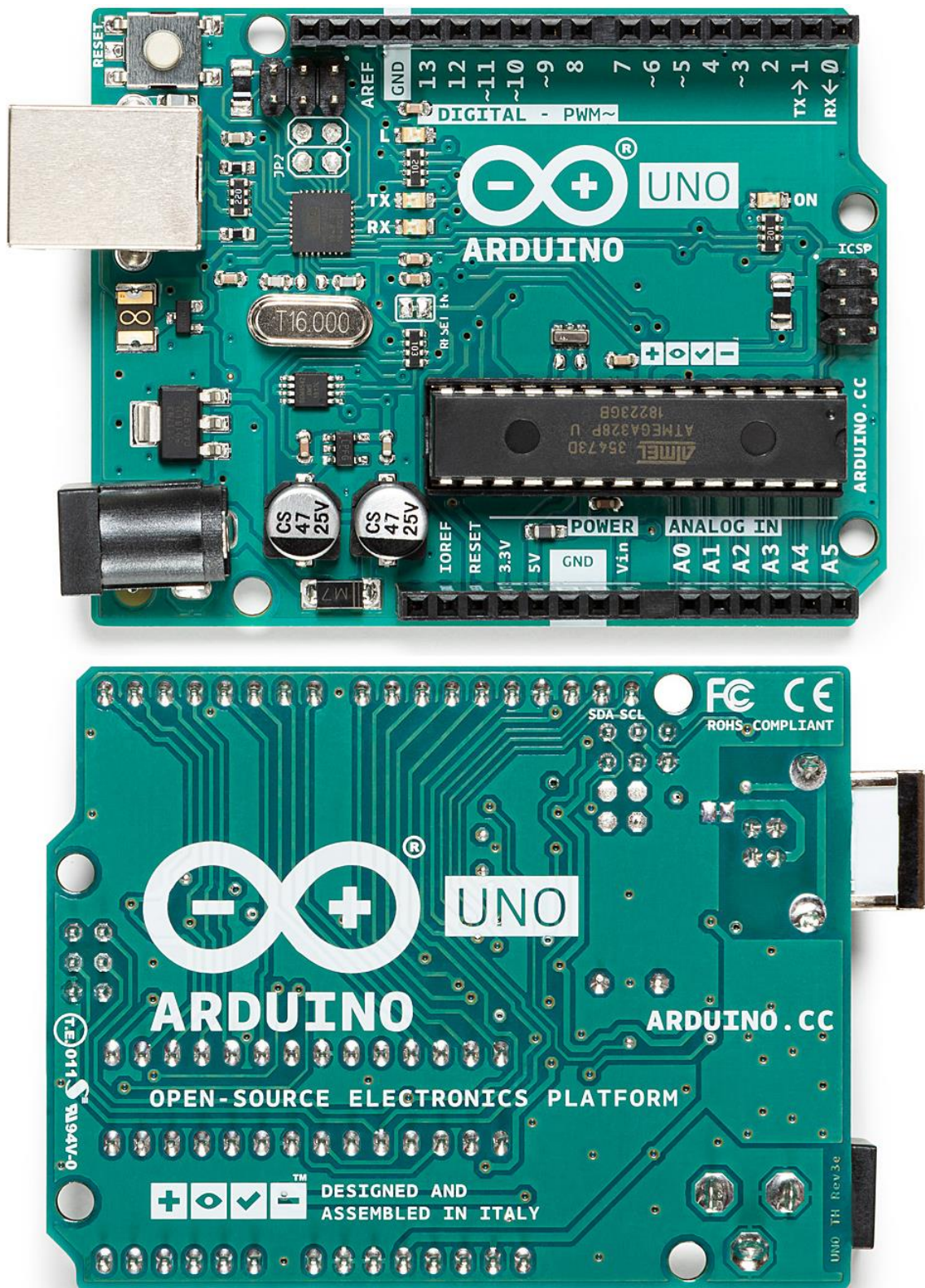
Εικόνα 2.10 Διάφορα μοντέλα μικροελεγκτών της οικογένειας Arduino

2.3.1.1 Arduino UNO R3

Για τον σχεδιασμό του υποσυστήματος δέκτη επιλέγεται η χρήση της ευρέως διαδεδομένης ηλεκτρονικής πλακέτας Arduino UNO και συγκεκριμένα της έκδοσης **Arduino UNO R3 – DIP** (αφαιρούμενος *Dual In-line Package* μικροελεγκτής *ATmega328 AVR*), λόγω του απλού και ευέλικτου τρόπου χρήσης και λειτουργίας που παρέχει ο σχεδιασμός του, ο οποίος ανταποκρίνεται ικανοποιητικά τόσο σε έμπειρους προγραμματιστές όσο και σε σχετικά άπειρους στον τομέα χρήστες.



Εικόνα 2.11 Απεικόνιση της ηλεκτρονικής πλακέτας μικροελεγκτή *Arduino UNO R3* (έκδοση *DIP*), σχεδιασμένη σε περιβάλλον προσομοίωσης



Εικόνα 2.12 Άνω και κάτω όψη μιας ηλεκτρονικής πλακέτας Arduino UNO R3 – DIP

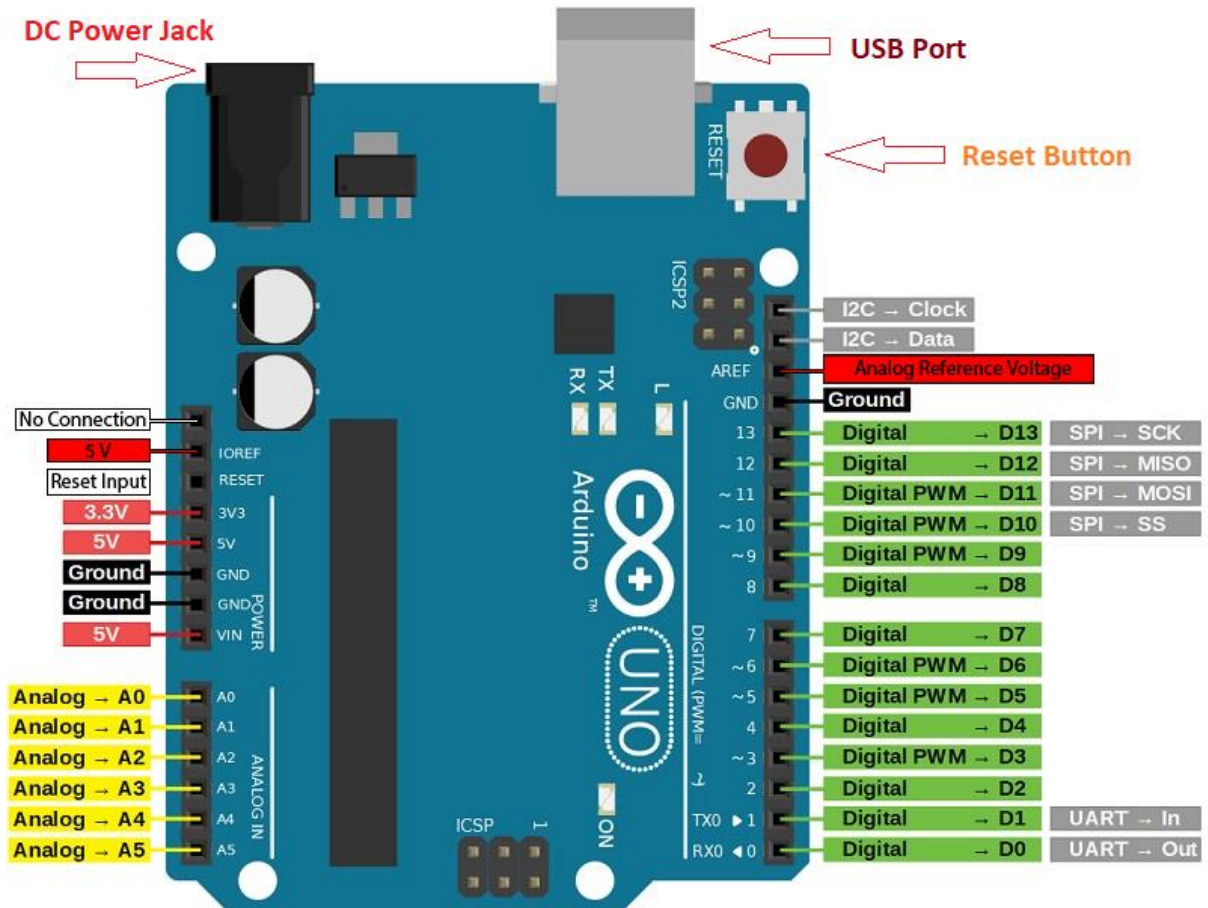
Τεχνικές προδιαγραφές του Arduino UNO R3

- Μικροελεγκτής ATmega328P
- Τάση λειτουργίας (*γραμμικός ρυθμιστής*) της πλακέτας 5V
- Τάση εισόδου τροφοδοσίας της πλακέτας στα 6-20V με προτεινόμενο εύρος εύρυθμης λειτουργίας και ασφάλειας της πλακέτας στα 7-12V
- 14 ψηφιακοί ακροδέκτες (*pins*) I/O, εκ των οποίων 6 μπορούν να παρέχουν έξοδο PWM
- 1 ασύγχρονο πομποδέκτη UART
- 1 μηχανισμό ICSP για σειριακό προγραμματισμό κυκλώματος δίχως να απαιτείται μεταβολή των στοιχείων της πλακέτας
- 1 σειριακή περιφερειακή διεπαφή SPPI
- 6 αναλογικοί ακροδέκτες (*pins*) εισόδου
- 20mA (DC) ανά pin I/O
- 50mA (DC) για το pin 3.3V
- Μνήμη Flash 32KB, εκ των οποίων 0,5KB διατίθενται για booting
- LED ενσωματωμένο και συνδεδεμένο με το pin13
- SRAM 2KB
- EEPROM 1KB
- Ταχύτητα ρολογιού (*κρυσταλλικός ταλαντωτής*) 16MHz
- Τροφοδοσία:
 - μέσω θύρας USB σε H/Y
 - μέσω βύσματος power jack με καλώδιο AC/DC adapter στο δίκτυο της ΔΕΗ
 - μέσω βύσματος power jack με καλώδιο 2,1mm center-positive plug σε μπαταρία
 - μέσω του pin VIN της πλακέτας με κατάλληλη σύνδεση σε άλλη ρυθμιζόμενη τροφοδοσία
- Διαστάσεις:
 - Μήκος 68,6mm
 - Πλάτος 53,4mm
 - Βάρος 25g

Ακροδέκτες παροχής ενέργειας (*POWER pins*)

- **LED:** Ενσωματωμένο LED εξαρτώμενο από τον ψηφιακό ακροδέκτη 13 – το LED ανάβει όταν pin13 έχει υψηλή τιμή, αντιθέτως παραμένει σβηστό.
- **Ανώνυμος ακροδέκτης:** Δεν χρησιμοποιείται, υπάρχει μόνο για να διασφαλίσει συμβατότητα με μελλοντικά προϊόντα
- **IOREF:** Ο ακροδέκτης αναφοράς IOREF παρέχει την τάση με την οποία λειτουργεί ο μικροελεγκτής. Με χρήση κατάλληλων περιφερειακών (*Arduino shields*) η τάση αυτή μπορεί να αξιοποιηθεί για επιλογή αντίστοιχης πηγής τροφοδοσίας ή δυνατότητα λειτουργίας στις εξόδους με τάση 5V ή 3,3V.
- **RESET:** Συνήθως χρησιμοποιείται για την προσθήκη κομβίου επαναφοράς (reset button) για εξειδικευμένες λειτουργίες σε συνδυασμό με προσθήκη περιφερειακών μονάδων (*shields*)
- **3V3:** Μέσω αυτού του pin παρέχεται – από τον ενσωματωμένο ρυθμιστή της πλακέτας – τροφοδοσία 3,3V σε περιφερειακές μονάδες (*μέγιστο ρεύμα εξόδου 50mA*). Παρότι δύναται να παρέχεται τροφοδοσία στο Arduino μέσω του pin 3V3 με κατάλληλη σύνδεση (*pins 3V3–GND*) σε μονάδα τροφοδοσίας, δεν προτείνεται καθώς παρακάμπτεται ο ρυθμιστής και άρα σε περιπτώσεις παροχής τάσης διαφορετικής των 3,3V υπάρχει κίνδυνος πρόκλησης ζημιάς στο Arduino
- **5V:** Το εν λόγω pin παρέχει μια ρυθμιζόμενη τάση 5V εξαιτίας του γραμμικού ρυθμιστή (*linear regulator*) τάσης της πλακέτας. Παρότι δύναται να παρέχεται τροφοδοσία στο Arduino μέσω του pin αυτού με κάποια τροφοδοτική μονάδα και κατάλληλη συνδεσμολογία (*pins 5V–GND*), δεν προτείνεται καθώς παρακάμπτεται ο ρυθμιστής και άρα σε περιπτώσεις παροχής τάσης διαφορετικής των 5V υπάρχει κίνδυνος πρόκλησης ζημιάς στην πλακέτα.
- **GND (x3):** Ακροδέκτες γείωσης
- **VIN:** Ο ακροδέκτης αυτός έχει δύο χρήσεις λόγω της εσωτερικής παράλληλης συνδεσμολογίας του με το βύσμα power jack:
 - Όταν η πλακέτα Arduino τροφοδοτείται με τάση μέσω του βύσματος power jack με καλώδιο, ο ακροδέκτης αυτός δύναται να μεταφέρει την τάση αυτή και σε κάποια άλλη ηλεκτρονική μονάδα, ως μορφή τροφοδοσίας της, με κατάλληλο τρόπο συνδεσμολογίας
 - Όταν δεν γίνεται χρήση του power jack ή USB, η πλακέτα Arduino δύναται να τροφοδοτείται μέσω του ακροδέκτη VIN από άλλης μορφής ρυθμιζόμενη πηγή

ισχύος (power bank, μπαταρία, τροφοδοτική πηγή εργαστηρίου και γενικά οποιασδήποτε μορφής ηλεκτρονική μονάδα τροφοδοσίας) με κατάλληλη συνδεσμολογία (pins VIN–GND)



Εικόνα 2.13 Ακροδέκτες ενός Arduino UNO R3

Ακροδέκτες εξειδικευμένης λειτουργίας

- Ψηφιακοί (*DIGITAL pins*)

- **RX0 (0) και TX0 (1) – ασύγχρονος πομποδέκτης UART:** Τα pins 0 και 1 χρησιμοποιούνται για λήψη και μετάδοση σειριακών δεδομένων TTL, αντίστοιχα, και συνδέονται στους αντίστοιχους ακροδέκτες του σειριακού τσιπ ATmega8U2 USB-to-TTL
- **2 και 3 – εξωτερικοί διακόπτες:** Δύνανται να διαμορφώνονται με τρόπο τέτοιο που να προκαλούν διακοπές στα κυκλώματα.
- **3, 5, 6, 9, 10 και 11 – διαμόρφωση πλάτους παλμού PWM:** Παρέχουν έξοδο PWM 8-bit με τη λειτουργία analogWrite().
- **10 (SS), 11 (MOSI), 12 (MISO) και 13 (SCK) – σειριακές περιφερειακές διεπαφές SPI:** Υποστηρίζουν επικοινωνία SPI χρησιμοποιώντας την βιβλιοθήκη SPI.
- **AREF (αναλογική αναφορά):** Αναλογικός/ψηφιακός μετατροπέας αναφοράς τάσης (ADC) για τις αναλογικές εισόδους.

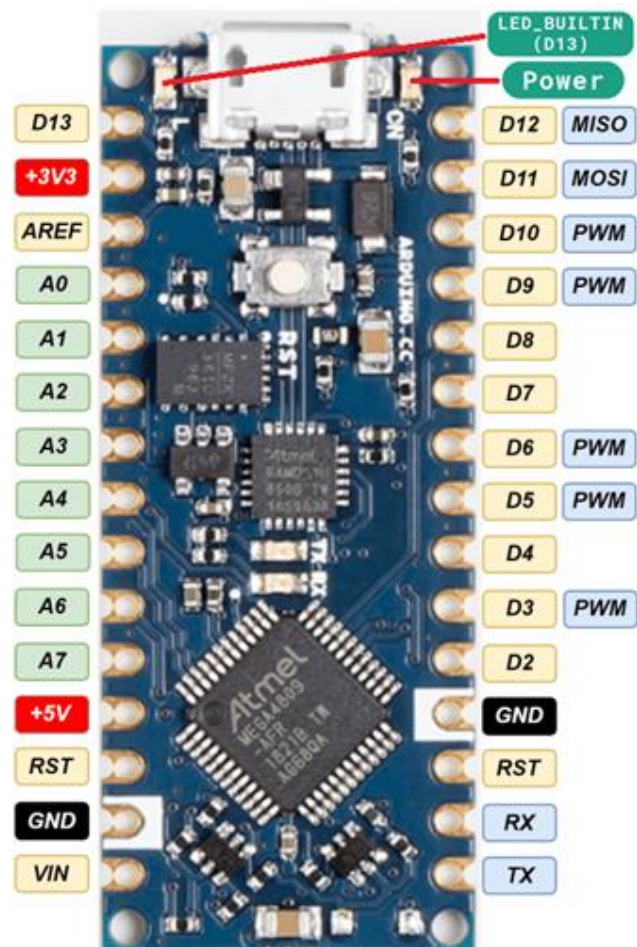
- Αναλογικοί (*ANALOG pins*)

- **A4 (pin SDA) και A5 (pin SCL) – διεπαφή δύο καλωδίων TWI/I²C:** Υποστηρίζουν επικοινωνία TWI χρησιμοποιώντας τη βιβλιοθήκη Wire.

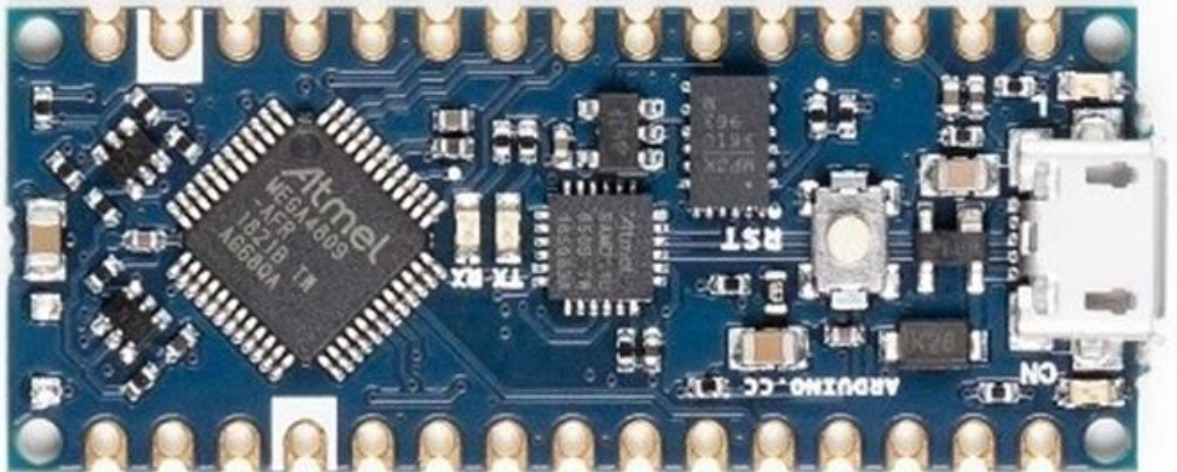
2.3.1.2 Arduino NANO EVERY

Για το υποσύστημα πομπού γίνεται επιλογή της πλακέτας μικροελεγκτή **Arduino NANO EVERY**, καθώς το αρκετά μικρό της μέγεθος και η ιδιαίτερη κατασκευή επιτρέπουν την εύκολη ενσωμάτωσή της στο εν λόγω υποσύστημα. Τα χαρακτηριστικά της πλακέτας NANO EVERY παρουσιάζονται κατωτέρω:

- **Επεξεργαστής (CPU):** Atmel Mega4809 AVR
- **Ταχύτητα επεξεργαστή (Clock Speed):** 20MHz
- **Μνήμη προγράμματος (Flash):** 48KB
- **SRAM:** 6KB
- **EEPROM:** 256 Byte
- **GPIO:** 14 Ψηφιακοί (5 PWM) / 8 Αναλογικοί (ADC 10bit)
- **Τάση εισόδου (VIN):** 7-21V
- **Τάση λειτουργίας:** 5V
- **Ρεύμα / pin:** 20mA DC (50mA για το pin 3.3V)
- **Διαστάσεις:** 45mm × 18mm
- **Βάρος:** 5g

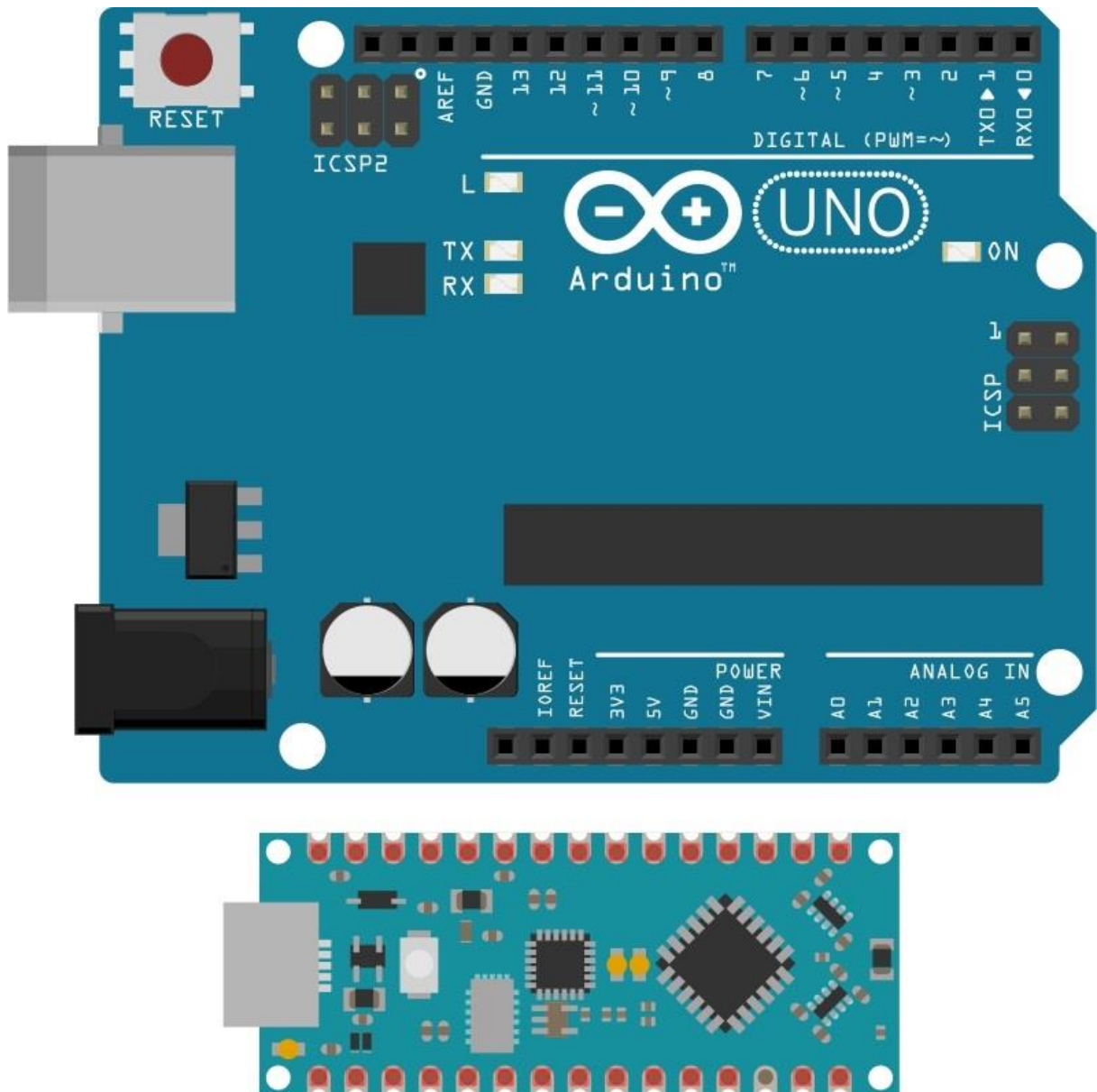


Το Arduino NANO EVERY δύναται να τροφοδοτείται είτε μέσω της θύρας τύπου Mini-B USB που προορίζεται για τον προγραμματισμό του μικροελεγκτή At-Mega4809 είτε με κατάλληλη σύνδεση των ακροδεκτών 5V-GND με εξωτερική πηγή τροφοδοσίας (π.χ. μπαταρία).



Εικόνα 2.14 Άνω και κάτω όψη της ηλεκτρονικής πλακέτας Arduino NANO EVERY

Συγκρίνοντας τα χαρακτηριστικά τους, οι βασικές διαφορές μεταξύ των μοντέλων UNO και NANO είναι ότι η ηλεκτρονική πλακέτα NANO EVERY διαθέτει 2 επιπλέον αναλογικούς ακροδέκτες, ίδιους ψηφιακούς ακροδέκτες αλλά με 1 PWM λιγότερο, 1 θύρα τύπου Mini-B USB αντί για USB και δεν περιλαμβάνει βύσμα power jack.



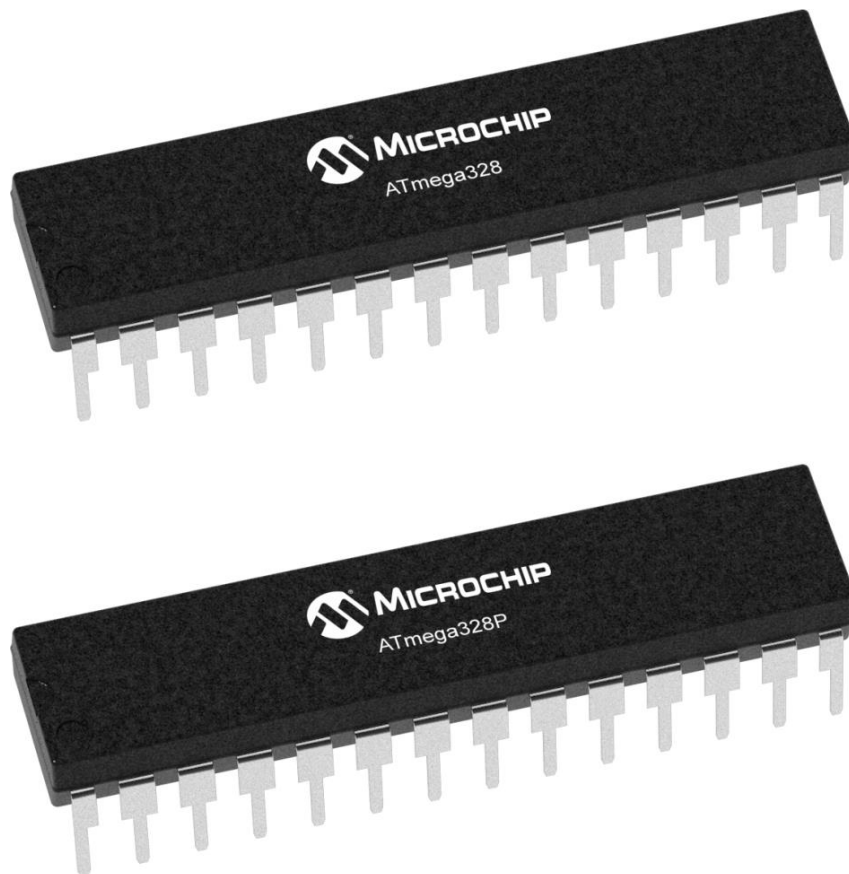
Εικόνα 2.15 Ηλεκτρονικές πλακέτες ενός Arduino UNO R3 (άνω) και ενός Arduino NANO EVERY (κάτω) σε περιβάλλον προσομοίωσης

2.3.1.3 Μικροελεγκτές AVR ATmega

Παρότι ο όρος «μικροελεγκτές *Arduino*» έχει άρρηκτα ταυτιστεί με τις ηλεκτρονικές πλακέτες της οικογένειας *Arduino* κατά τα τελευταία χρόνια, στην ουσία ένας μικροελεγκτής *Arduino* είναι μόνο η μονάδα εκείνη που είναι υπεύθυνη για τον προγραμματισμό και άρα όλες τις διεργασίες που δύναται να εκτελεί η ηλεκτρονική πλακέτα και τυχόν περιφερειακές μονάδες που συνδέονται σε αυτήν.

Η πλακέτα *Arduino UNO R3* διαθέτει μικροελεγκτή ATmega328P, ενώ η πλακέτα *Arduino NANO EVERY* διαθέτει μικροελεγκτή ATmega4809 παρά το γεγονός ότι οι προηγούμενες εκδόσεις *Arduino NANO* διαθέτουν μικροελεγκτή ATmega328.

Οι μικροελεγκτές Atmega328 και Atmega328P είναι ίδιοι από αρχιτεκτονική άποψη.



Εικόνα 2.16 Μικροελεγκτές ATmega328 (*Arduino NANO V3*) και ATmega328P (*Arduino UNO R3*)

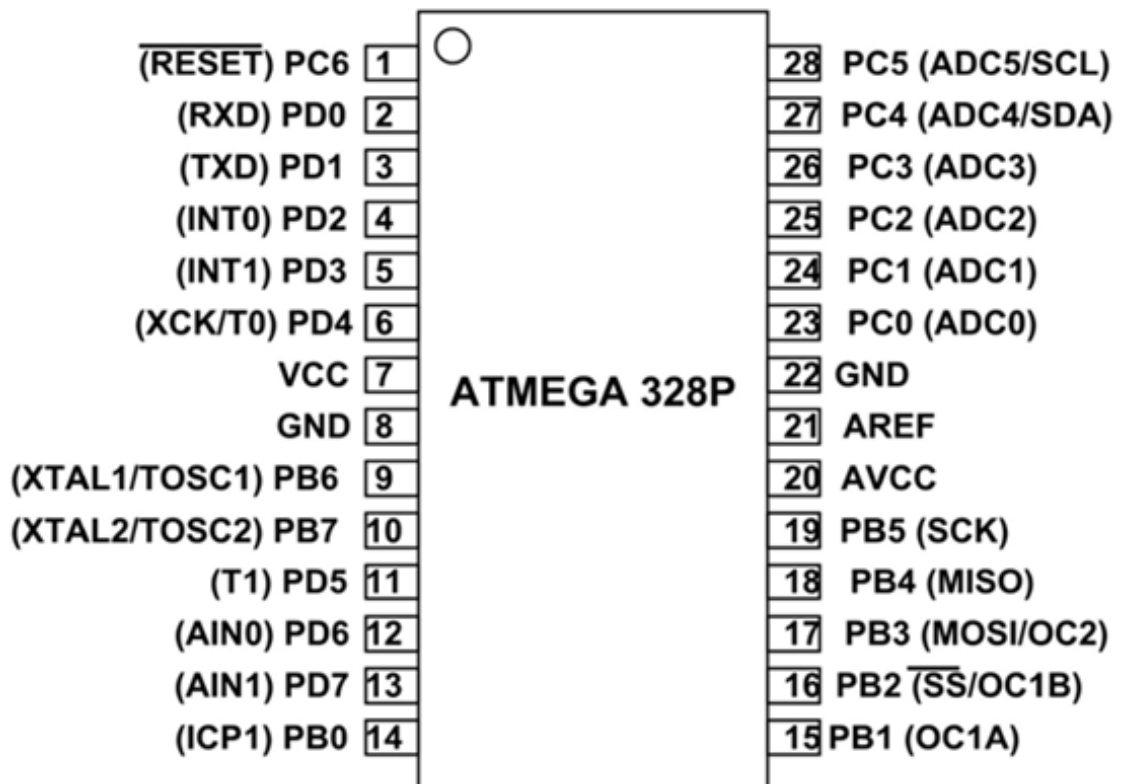
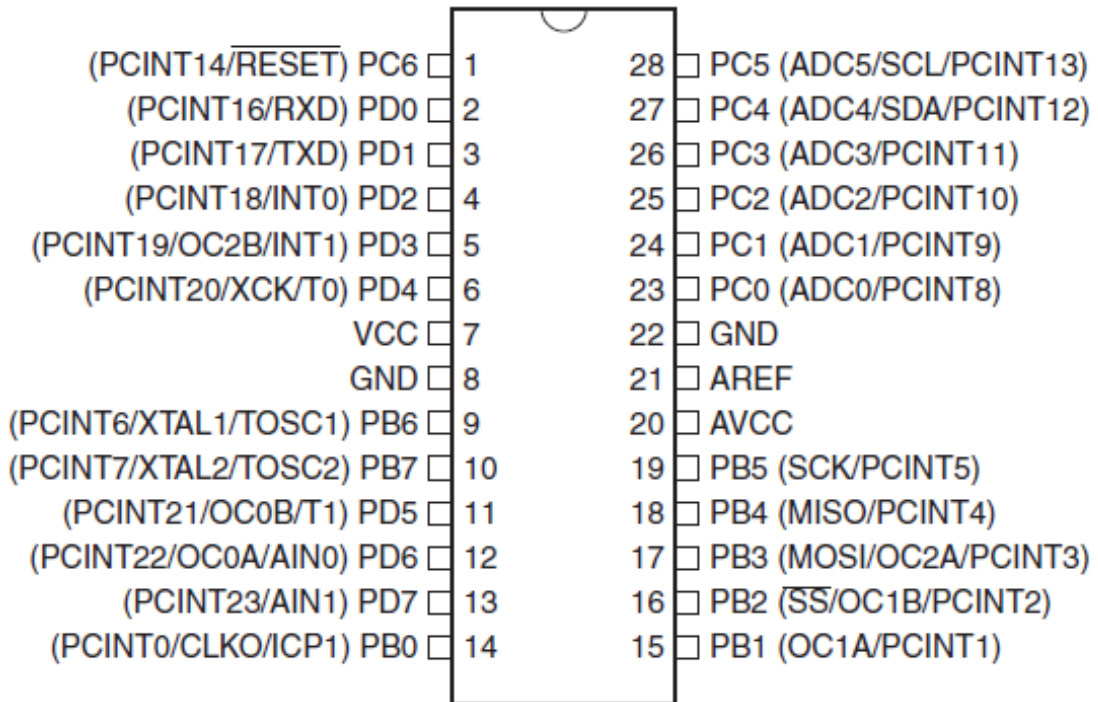
Από πλευράς κατασκευής, η EEPROM είναι 1KB για τον Atmega328 και 1024B για τον Atmega328P, ενώ όλες οι υπόλοιπες παράμετροι είναι οι ίδιες (*8-bit επεξεργαστής RISC αρχιτεκτονικής 28 ακροδεκτών, 2KB SRAM, 23 γραμμές γενικής χρήσης I/O, 32 καταχωρητές γενικής χρήσης, 3 εύκαμπτοι χρονοδιακόπτες/μετρητές, εσωτερικοί και εξωτερικοί διακόπτες, σειριακή θύρα SPI, 10-bit μετατροπέας A/D 6 καναλιών και τάση λειτουργίας 1,8-5,5V*).

Άλλες διαφορές που υπάρχουν μεταξύ τους είναι ότι το Atmega328P καταναλώνει λιγότερη ισχύ –της τάξεως των μA – σε σχέση με το Atmega328.

Επιπλέον, οι υπογραφές των δύο τσιπ είναι διαφορετικές, πράγμα που σημαίνει ότι όταν χρησιμοποιείται μια ηλεκτρονική πλακέτα Arduino θα πρέπει οι παράμετροι στο Arduino IDE να ρυθμιστούν στο σωστό αντίστοιχο τσιπ ώστε να γίνεται η σωστή αναγνώριση, διαφορετικά το λογισμικό εμφανίζει μήνυμα δηλώνοντας ότι χρησιμοποιήθηκε λάθος τσιπ για την τρέχουσα εφαρμογή.

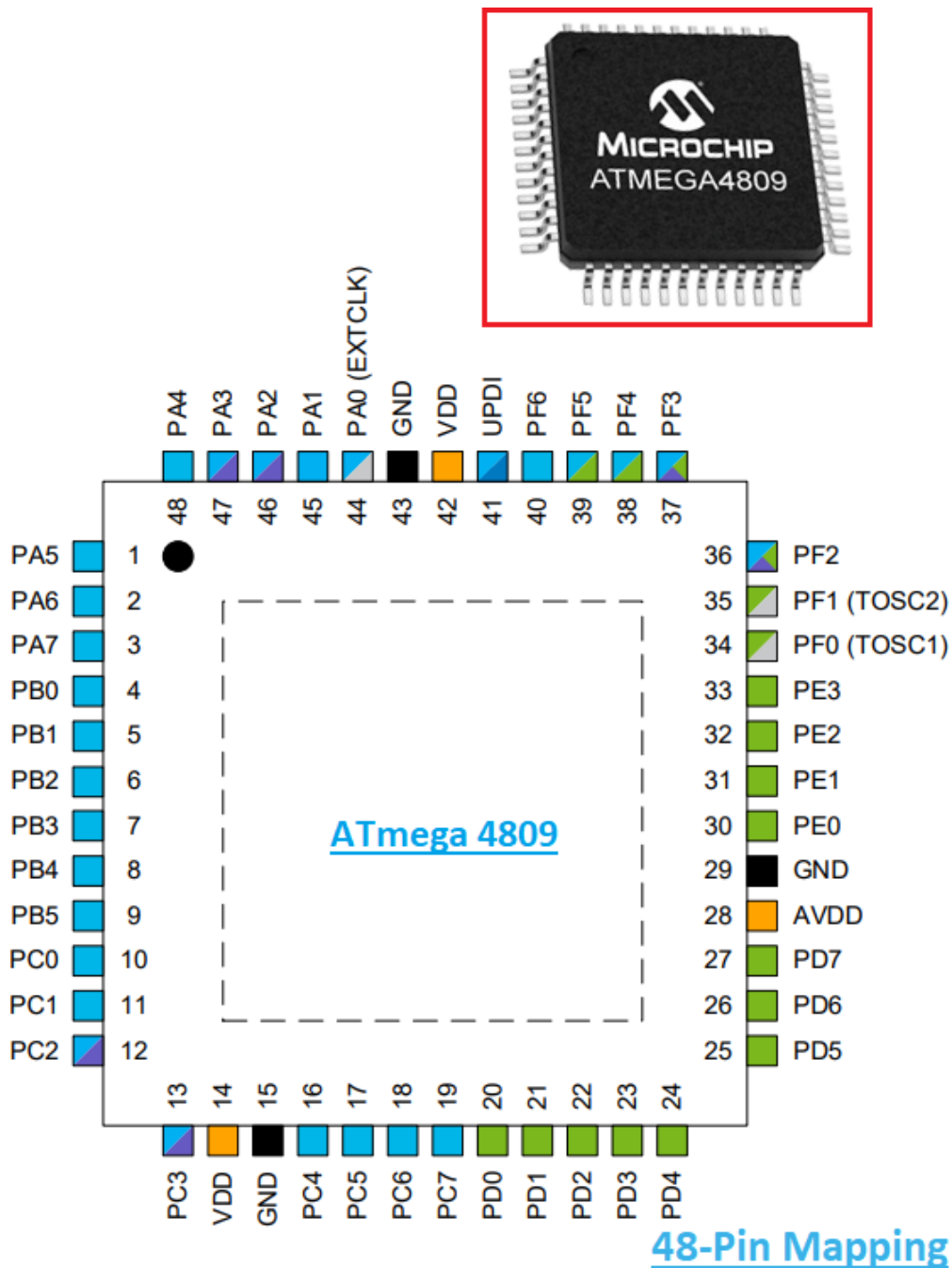
Εκτός των ανωτέρω, το Atmega328 διατίθεται με χαμηλότερο κόστος στην αγορά, ενώ το Atmega328P καταναλώνει λιγότερη ισχύ –της τάξεως των μA – μία δυνατότητα που καθίσταται σωτήρια σε ορισμένες εφαρμογές όπου η εργασία με χαμηλή ισχύ είναι απαραίτητη.

Atmega328



Εικόνα 2.17 Χαρτογράφηση ακροδεκτών των ATmega328 (Arduino NANO V3) και ATmega328P (Arduino UNO R3)

Ο μικροελεγκτής ATmega4809 που χρησιμοποιεί η πλακέτα Arduino NANO EVERY αποτελείται από ένα τσιπ 48 ακροδεκτών που διαθέτει έναν 8-bit επεξεργαστή που λειτουργεί με ταχύτητα έως και 20MHz, μνήμης Flash 48KB, SRAM 6KB, EEPROM 256 bytes και λειτουργίες χαμηλής κατανάλωσης ισχύος.



Εικόνα 2.18 Μικροελεγκτής ATmega4809 του Arduino NANO EVERY (άνω) και χαρτογράφηση των ακροδεκτών του (κάτω)

2.4 Μονάδες Ασύρματης Επικοινωνίας

2.4.1 Πομποδέκτες Ραδιοσυχνότητων NRF24L01

Μια ηλεκτρονική συσκευή ικανή να παρέχει ασύρματη επικοινωνία ικανοποιητικής εμβέλειας μεταξύ των υποσυστημάτων πομπού & δέκτη είναι η ασύρματη μονάδα πομποδέκτη NRF24L01 (*NRF24L01 wireless transceiver module*).



Εικόνα 2.19 Μοντέλο πομποδέκτη ραδιοσυχνότητων NRF24L01 με ενσωματωμένη κεραία

Ο πομποδέκτης NRF24L01 αποτελεί μια αρκετά φθηνή λύση για ασύρματη επικοινωνία καθώς μπορεί να αποκτηθεί με πολύ χαμηλό κόστος, διαθέτει κεραία εκπομπής στα 2,4GHz και αρκετά μικρό μέγεθος που καθιστά την ενσωμάτωσή του σε ηλεκτρονική πλακέτα αρκετά απλή από άποψη χώρου κατασκευής και ευκολίας τοποθέτησης.

Ο πομποδέκτης NRF24L01 προσφέρεται σε διάφορα μοντέλα που μπορεί να διαθέτουν ενσωματωμένη κεραία, πρόσθετο ενισχυτή σήματος ή και υποδοχή SMA για σύνδεση με εξωτερική κεραία ώστε να παρέχεται ακόμη μεγαλύτερη εμβέλεια.

Όντας ένας ημι-αμφίδρομος (*half-duplex*) πομποδέκτης, όταν συνδέεται σε έναν μικροελεγκτή Arduino και ρυθμίζεται κατάλληλα μπορεί να μεταφέρει πληροφορίες με αμφίδρομο τρόπο, λειτουργώντας στην ραδιοφωνική ζώνη 2,4GHz ISM η οποία αποτελεί τμήμα του ραδιοφάσματος ISM (*Industrial-Scientific-Medical*) που προορίζεται διεθνώς για Βιομηχανικούς, Επιστημονικούς και Ιατρικούς σκοπούς.

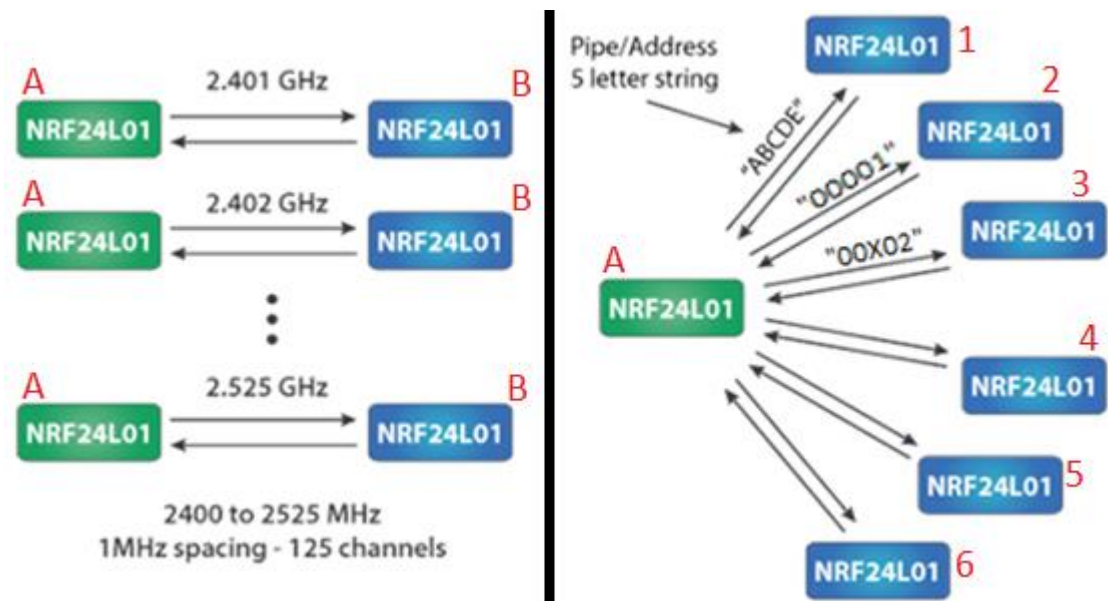
Βασικά χαρακτηριστικά του NRF24L01

- 125 κανάλια ραδιοσυχνότητας (2,4 έως 2,525 GHz)
- Λειτουργία ζώνης ISM 2,4GHz χωρίς άδεια παγκοσμίως
- Διαμόρφωση GFSK
- Κοινή διεπαφή για αποστολή και λήψη (*TX και RX*)
- Ενσωματωμένο πρωτόκολλο βασικής ζώνης Enhanced ShockBurst για ασύρματες επικοινωνίες χαμηλής ισχύος
- Επικοινωνία SPI για αλληλεπίδραση με μικροελεγκτή

Ο πομποδέκτης αποστέλλει και λαμβάνει πληροφορίες με την μορφή δεδομένων σε μια συγκεκριμένη συχνότητα που ονομάζεται κανάλι και φέρει οποιαδήποτε τιμή περιλαμβάνεται στην ζώνη 2,4GHz ISM και άρα μεταξύ των τιμών 2,4GHz - 2,525GHz (2400MHz - 2525MHz).

Θέτοντας δύο ή περισσότερους πομποδέκτες σε ένα συγκεκριμένο κανάλι και ρυθμίζοντας κατάλληλα τις διευθύνσεις τους, μπορούν να μεταφέρονται μέσω αυτών δεδομένα και έτσι να επιτυγχάνεται ασύρματη επικοινωνία.

Με χρήση ενός πομποδέκτη NRF24L01 μπορούν να αξιοποιούνται έως και (2525MHz – 2400MHz =) 125 διαφορετικά κανάλια (*συχνότητες*) και άρα να σχηματίζεται ένα δίκτυο από 125 μόντεμ που λειτουργούν ανεξάρτητα σε έναν χώρο, με το κάθε κανάλι να διαχειρίζεται έως και 6 διευθύνσεις (*pipes/addresses*) ταυτόχρονα, ήτοι έως και 6 διαφορετικούς πομποδέκτες.



Εικόνα 2.20 Ορισμός 2 πομποδεκτών A και B σε συγκεκριμένο κανάλι κάθε φορά (αριστερή εικόνα) και ορισμός 7 πομποδεκτών σε συγκεκριμένο κανάλι με ρύθμιση σε 6 διαφορετικών διευθύνσεων για ταυτόχρονη επικοινωνία μεταξύ τους (δεξιά εικόνα)

Αυτό το ιδιαίτερο χαρακτηριστικό μπορεί να παρέχει σε ένα σύστημα ελέγχου ευρεία ποικιλία διαφορετικών βιομηχανικών εφαρμογών σε πολλά σημεία ενός χώρου και σε πραγματικό χρόνο όταν του παρέχονται συγκεκριμένες εντολές εισόδου από έναν χρήστη, όπως παρακολούθηση της κατάστασης του καιρού, έλεγχος κινητών και ακίνητων κατασκευών, διαβίβαση πληροφοριών και εικοσιτετράωρη παρακολούθηση μιας περιοχής για διάφορους λόγους.

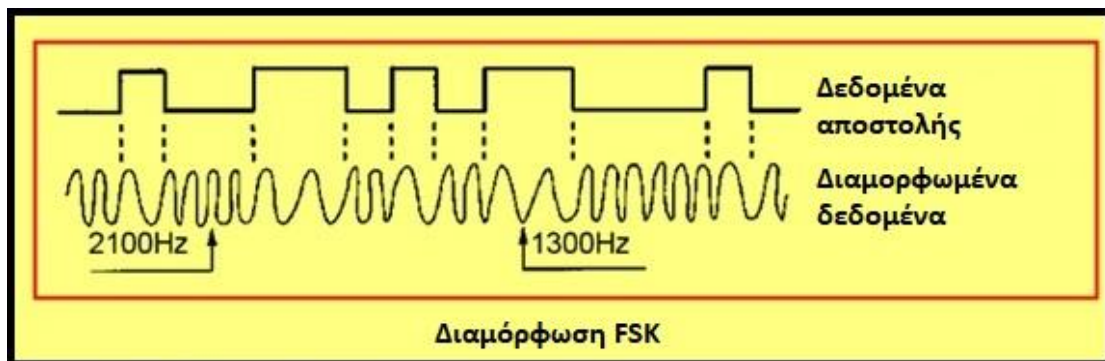
Προσθέτοντας στα ανωτέρω την δυνατότητα περισσότερων του ενός χρηστών και από διαφορετικά σημεία εισόδου να παρέχουν εντολές σε διαφορετικά και απομακρυσμένα μεταξύ τους –αλλά και από τους χρήστες– επιθυμητά σημεία εφαρμογής του ελέγχου, γίνεται ξεκάθαρα αντιληπτό ότι η απόκτηση ενός σχεδόν απόλυτου ελέγχου εφαρμογών σε μια ή και περισσότερες επιθυμητές περιοχές μπορεί να γίνει εύκολα πραγματικότητα με κατάλληλο σχεδιασμό, προγραμματισμό και υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελέγχου με αξιοποίηση της μονάδας NRF24L01.

Λειτουργώντας στις ζώνες συχνοτήτων 2,4GHz ISM, ο πομποδέκτης NRF24L01 δύνανται να χρησιμοποιείται δίχως άδεια ή καταβολή χρηματικού ποσού για εφαρμογές που περιλαμβάνουν βιομηχανικό, επιστημονικό ή ιατρικό εξοπλισμό.

Η μονάδα πομποδέκτη NRF24L01 χρησιμοποιεί διαμόρφωση GFSK, μια παραλλαγή της γενικής κατηγορίας διαμόρφωσης μετατόπισης συχνότητας γνωστή ως FSK, μέσω της οποίας ψηφιακά δεδομένα παρουσιάζονται ως αλλαγές στη συχνότητα ενός φέροντος σήματος.

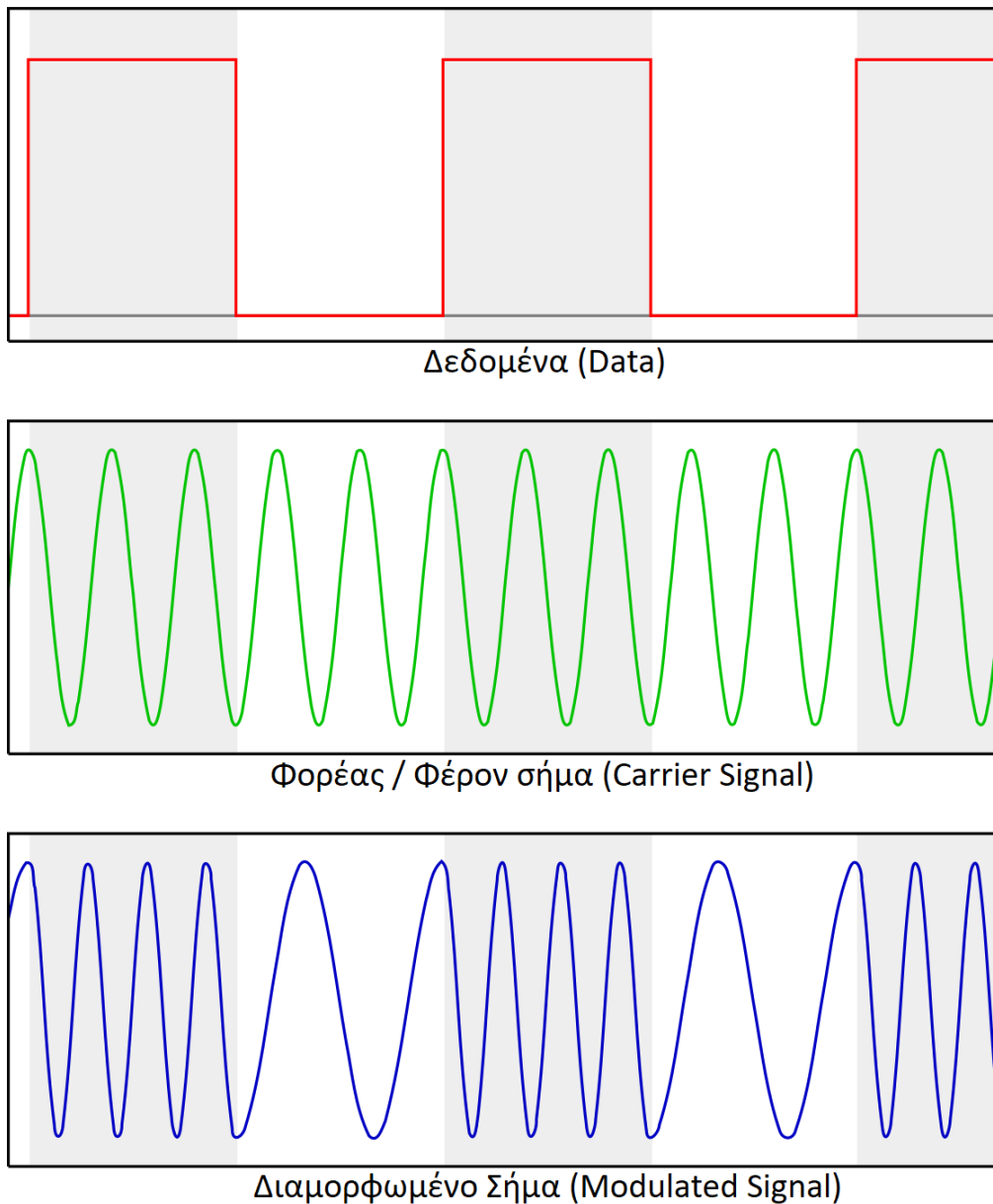
Γενικά, ως **διαμόρφωση FSK** (*Frequency-Shift Keying modulation*) ορίζεται η διαδικασία μεταβολής της συχνότητας του φέροντος σήματος των πομποδεκτών σύμφωνα με τα δεδομένα (σε *bits*) που μεταφέρονται κατά την ασύρματη επικοινωνία. Αυτή η διαδικασία μεταβολής της συχνότητας είναι επίσης γνωστή και ως ζωνοπερατή μετάδοση δεδομένων.

Η διαμόρφωση FSK εκχωρεί διαφορετικές συχνότητες στον φορέα ανάλογα με το bit που μεταδίδεται. Επομένως, όταν μεταδίδεται “0” bit, ο φορέας αναλαμβάνει μια συχνότητα που αντιστοιχεί σε “0” bit κατά τη διάρκεια μετάδοσης ενός bit. Όταν μεταδίδεται ένα bit “1”, η συχνότητα φορέα μεταβάλλεται σε τιμή που αντιστοιχεί στο bit “1” και παρομοίως παραμένει σε αυτήν τη συχνότητα κατά τη διάρκεια μετάδοσης ενός bit, όπως φαίνεται στο ακόλουθο σχήμα:



Εικόνα 2.21 Διαμόρφωση FSK

Εναλλακτικά, για παράδειγμα, μπορούν να χρησιμοποιηθούν τέσσερις διαφορετικές συχνότητες μετάδοσης (00, 01, 11 και 10 bits), καθεμία από τις οποίες αντιστοιχεί σε 2 bit, με την λειτουργία αυτή να ονομάζεται **4FSK**. Η μέθοδος αυτή αυξάνει το ρυθμό των μεταδιδόμενων bits, αλλά αυξάνει επίσης την ζώνη συχνοτήτων που χρησιμοποιείται για τη μετάδοση.



Εικόνα 2.22 Παράδειγμα διαμόρφωσης FSK – Διαμόρφωση της συχνότητας του φέροντος σήματος (carrier signal) στο ρυθμό των παλμών των δεδομένων (data)

Η πληροφορία που εκφράζεται με “τάση”/“όχι-τάση” ή αλλιώς “1”/“0” αντίστοιχα, μέσω συσκευής μίξεως μεταβάλλει την συχνότητα του φέροντος σήματος που παράγεται από έναν ταλαντωτή σταθερής συχνότητας στον ρυθμό των μεταβολών της πληροφορίας (μεταβολές “1” ή “0”), με αποτέλεσμα να έχουμε αύξηση της συχνότητας του φέροντος σήματος με “1” και αντίστοιχη μείωση με “0”, όπως παρουσιάζεται αναλυτικά στην εικόνα 2.22. Το διαμορφωμένο αυτό σήμα εκπέμπεται από τον πομπό προς τον δέκτη για μεταφορά της πληροφορίας.

Στην **διαμόρφωση GFSK** (*Gaussian FSK modulation*) ή αλλιώς *Γκαουσιανή FSK* τα δεδομένα κωδικοποιούνται ως παραλλαγές της συχνότητας σε έναν φορέα με τρόπο παρόμοιο της διαμόρφωσης FSK. Επομένως, ο διαμορφωτής που χρησιμοποιείται εδώ μπορεί να είναι ο ίδιος που χρησιμοποιείται για τη διαμόρφωση FSK. Ωστόσο, οι παλμοί περνούν από ένα φίλτρο Gauss πριν εισέλθουν στον διαμορφωτή παλμών για να μειώσουν το φασματικό πλάτος του ίδιου.

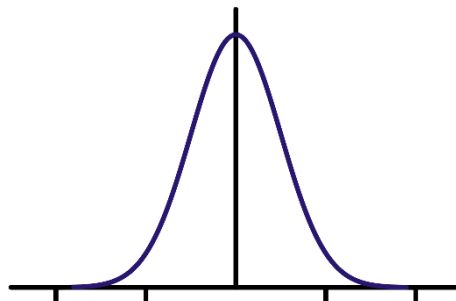
Το φίλτρο Gaussian είναι ένα είδος μορφοποιητή παλμού που χρησιμοποιείται για την εξομάλυνση της μετάβασης μεταξύ των τιμών των παλμών και την αποφυγή θορύβου, χρησιμοποιείται δηλαδή για την μείωση του διαστήματος μετατόπισης της συχνότητας.



Εικόνα 2.23 Φίλτρο Gauss της διαμόρφωσης GFSK

Η διαμόρφωση GFSK, αντί να ρυθμίζει άμεσα τη συχνότητα με τα σύμβολα ψηφιακών δεδομένων και αλλάζοντας «στιγμαία» τη συχνότητα στην αρχή κάθε περιόδου συμβόλων, φιλτράρει τους παλμούς δεδομένων με ένα φίλτρο Gauss για να κάνει τις μεταβάσεις πιο ομαλές.

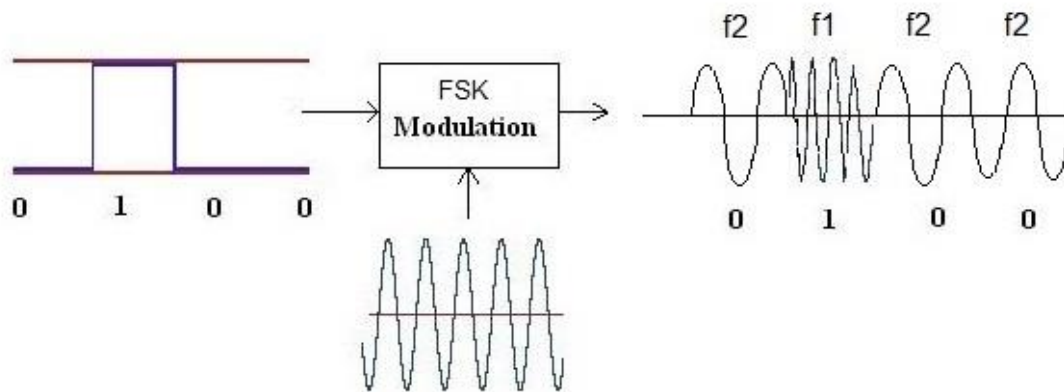
Με χρήση του φίλτρου Gauss επιτυγχάνεται μείωση της ισχύος της πλευρικής ζώνης, μείωση των παρεμβολών σε γειτονικά κανάλια αλλά έχει ως μειονέκτημα την αύξηση παρεμβολών μεταξύ συμβόλων.



Εικόνα 2.24 Σχήμα της παλμικής απόκρισης ενός τυπικού φίλτρου Gauss

Ένας διαμορφωτής GFSK διαφέρει από έναν απλό διαμορφωτή FSK καθώς πριν ακόμη η κυματομορφή (παλμοί) βασικής ζώνης εισέλθει στον διαμορφωτή FSK για την διαδικασία της διαμόρφωσης, περνάει πρώτα από ένα φίλτρο Gauss. Η ενέργεια αυτή καθιστά τις μεταβάσεις (παλμούς) πιο ομαλές και επομένως περιορίζει το διαμορφωμένο πλάτος φάσματος (φασματικό πλάτος) και ονομάζεται παλμική διαμόρφωση ή διαμόρφωση παλμού.

Η τεχνική διαμόρφωσης FSK χρησιμοποιείται για τη διαμόρφωση δυαδικής συμβολοσειράς των 1 και 0 κατάλληλη για μετάδοση χρησιμοποιώντας συχνότητα φορέα RF από το ένα άκρο στο άλλο.



Εικόνα 2.25 Διαμόρφωση FSK

Παράδειγμα απλού σχήματος FSK είναι το δυαδικό FSK (*Binary FSK*).

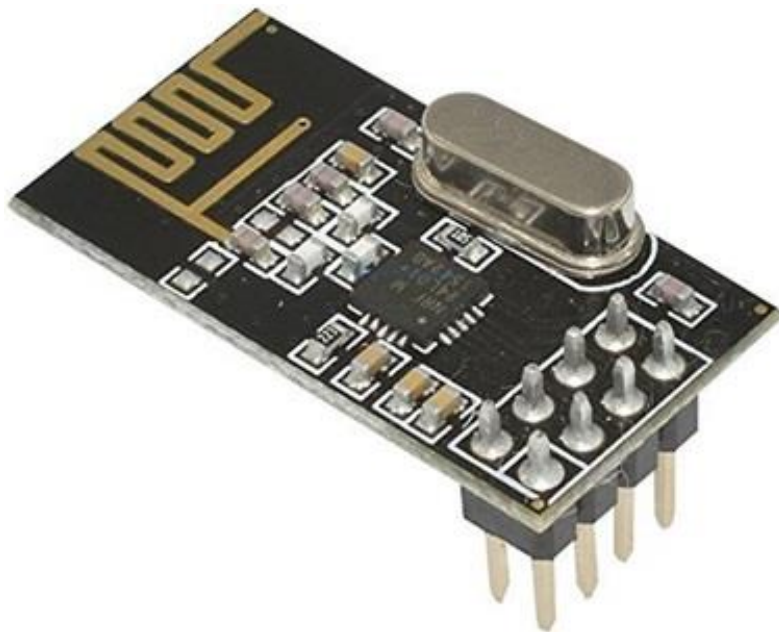
Στο BFSK, οι δυαδικές πληροφορίες των 1 και 0 αντιπροσωπεύονται από αλλαγή συχνότητας (π.χ. το δυαδικό 1 αντιπροσωπεύεται από την συχνότητα $f1$ και το δυαδικό 0 αντιπροσωπεύεται από την συχνότητα $f2$).

Στο GFSK, το φασματικό πλάτος είναι περιορισμένο λόγω του φίλτρου.

Το ενσωματωμένο πρωτόκολλο βασικής ζώνης Enhanced ShockBurst βασίζεται σε επικοινωνία με πακέτα, καθιστώντας παράλληλα ομαλή την μεταφορά δεδομένων μεταξύ του πομποδέκτη και του μικροελεγκτή, με την διασύνδεση των δύο μονάδων και μετέπειτα προγραμματισμό να επιτυγχάνεται μέσω σειριακής περιφερειακής επικοινωνίας (*SPI*).

Τεχνικές προδιαγραφές του NRF24L01

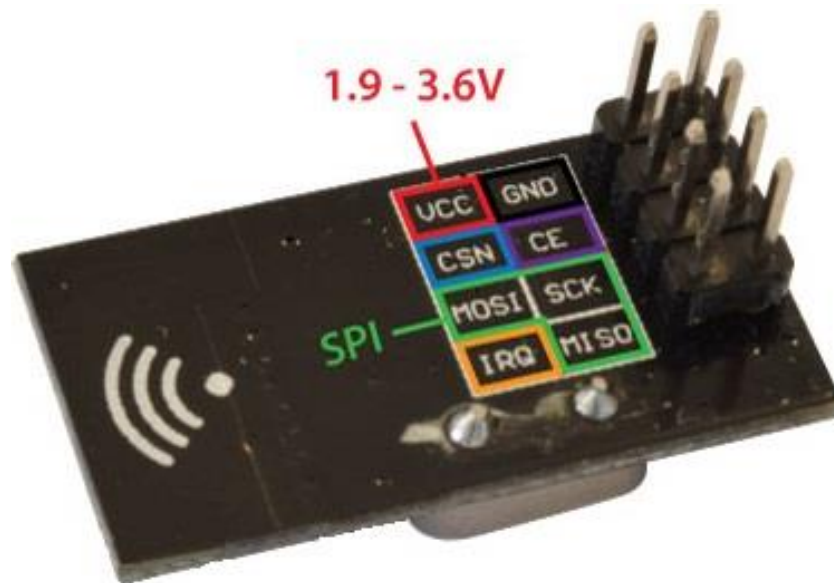
- Ταχύτητα επικοινωνίας καναλιού (*Baud rate*): 250Kbps - 2Mbps
- Εύρος επικοινωνίας: έως 100m
- Μέγιστη τρέχουσα κατανάλωση: ~12mA
- Τάση λειτουργίας: 1,9V - 3,6V (1μΑ για *standby mode*)
- Τροφοδοσία μέσω ακροδεκτών (*pins*): 5V
- Εναλλαγή 125 καναλιών: Δυνατότητα αλλαγής συχνότητας εκπομπής μεταξύ 2,4GHz - 2,525GHz για επικοινωνία με διαφορετικά σημεία σε μία περιοχή



Εικόνα 2.26 Άνω όψη της πομποδέκτη NRF24L01 με ενσωματωμένη κεραία

Η κατανάλωση ισχύος της μονάδας πομποδέκτη NRF είναι περίπου 12mA κατά τη μετάδοση, η οποία είναι μικρότερη και από αυτήν μιας διόδου LED. Η τάση λειτουργίας της μονάδας NRF κυμαίνεται από 1,9 – 3,6V, αλλά το θετικό χαρακτηριστικό της είναι ότι οι υπόλοιποι ακροδέκτες είναι σχεδιασμένοι να τροφοδοτούνται με τάση έως και 5V, έτσι η συνδεσμολογία ενός NRF24L01 με ένα Arduino καθίσταται εύκολη και η λειτουργία τους απρόσκοπτη, δίχως την ανάγκη προσθήκης περαιτέρω διεπαφών (πχ *logic level converter*).

Ακροδέκτες του NRF24L01



Εικόνα 2.27 Ακροδέκτες (pins) του πομποδέκτη NRF24L01

Όπως παρουσιάζεται και στην ανωτέρω εικόνα, οι ακροδέκτες VCC και GND χρησιμοποιούνται για την τροφοδοσία της μονάδας NRF24L01.

Οι ακροδέκτες MOSI, SCK, MISO προορίζονται για επικοινωνία SPI και άρα μπορούν να συνδεθούν με τους αντίστοιχους ακροδέκτες των περισσότερων πλακετών Arduino.

Οι ακροδέκτες CSN και CE μπορούν να συνδεθούν σε οποιονδήποτε ψηφιακό ακροδέκτη των Arduino και χρησιμοποιούνται για να τίθεται ο πομποδέκτης RF σε καταστάσεις αναμονής ή ενεργής λειτουργίας (*standby/active modes*), καθώς και για την εναλλαγή μεταξύ λειτουργίας μετάδοσης ή εντολής (*transmit/command modes*).

Ο ακροδέκτης IRQ είναι ένας ακροδέκτης διακοπής που χρησιμοποιείται για προγραμματισμένες διακοπές της μονάδας NRF.

| Module NRF24L01 | MOSI | MISO | SCK | CSN |
|--------------------|------|------|-----|-----|
| Arduino UNO R3 | 11 | 12 | 13 | 10 |
| Arduino NANO EVERY | 11 | 12 | 13 | 10 |

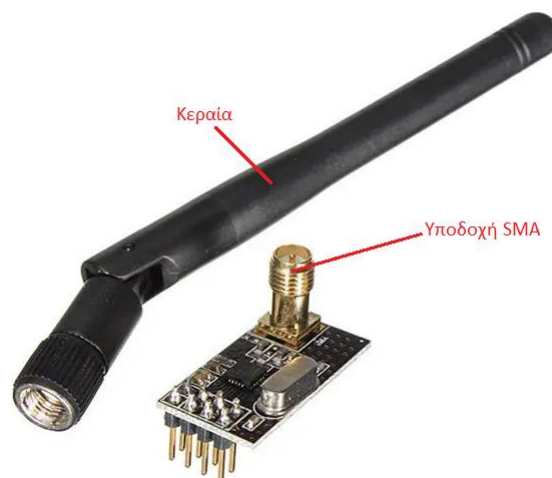
Εικόνα 2.28 Πίνακας σύνδεσης ακροδεκτών (pins) μεταξύ του NRF24L01 και των δύο Arduino

Ο πομποδέκτης NRF24L01 διατίθεται στην αγορά σε τρεις βασικές παραλλαγές με την συνηθέστερη στην αγορά να περιλαμβάνει ενσωματωμένη κεραία, όπως αυτή που παρουσιάζεται στην εικόνα 2.29. Ο ιδιαίτερος αυτός σχεδιασμός καθιστά μεν τον πομποδέκτη πιο συμπαγή και εύκολο στην χρήση και τοποθέτηση, αλλά περιορίζει το εύρος μετάδοσης σε μια αρκετά μικρή κλίμακα αν συσχετιστεί με άλλα μοντέλα, καλύπτοντας μια απόσταση περίπου 100 μέτρων.



Εικόνα 2.29 Μοντέλο πομποδέκτη NRF24L01 με ενσωματωμένη κεραία

Η δεύτερη παραλλαγή του NRF24L01 (εικ. 2.30), αντί της ενσωματωμένης κεραίας διαθέτει υποδοχή SMA που υποστηρίζει σύνδεση με εξωτερική κεραία για καλύτερη εμβέλεια μετάδοσης. Με σύνδεση διαφορετικών κεραιών, το εύρος μετάδοσης διαφοροποιείται σημαντικά ενώ παράλληλα καλύπτονται οι ανάγκες της εκάστοτε εφαρμογής για την οποία χρησιμοποιείται.



Εικόνα 2.30 Μοντέλο πομποδέκτη NRF24L01 με υποδοχή SMA για προσθήκη κεραίας

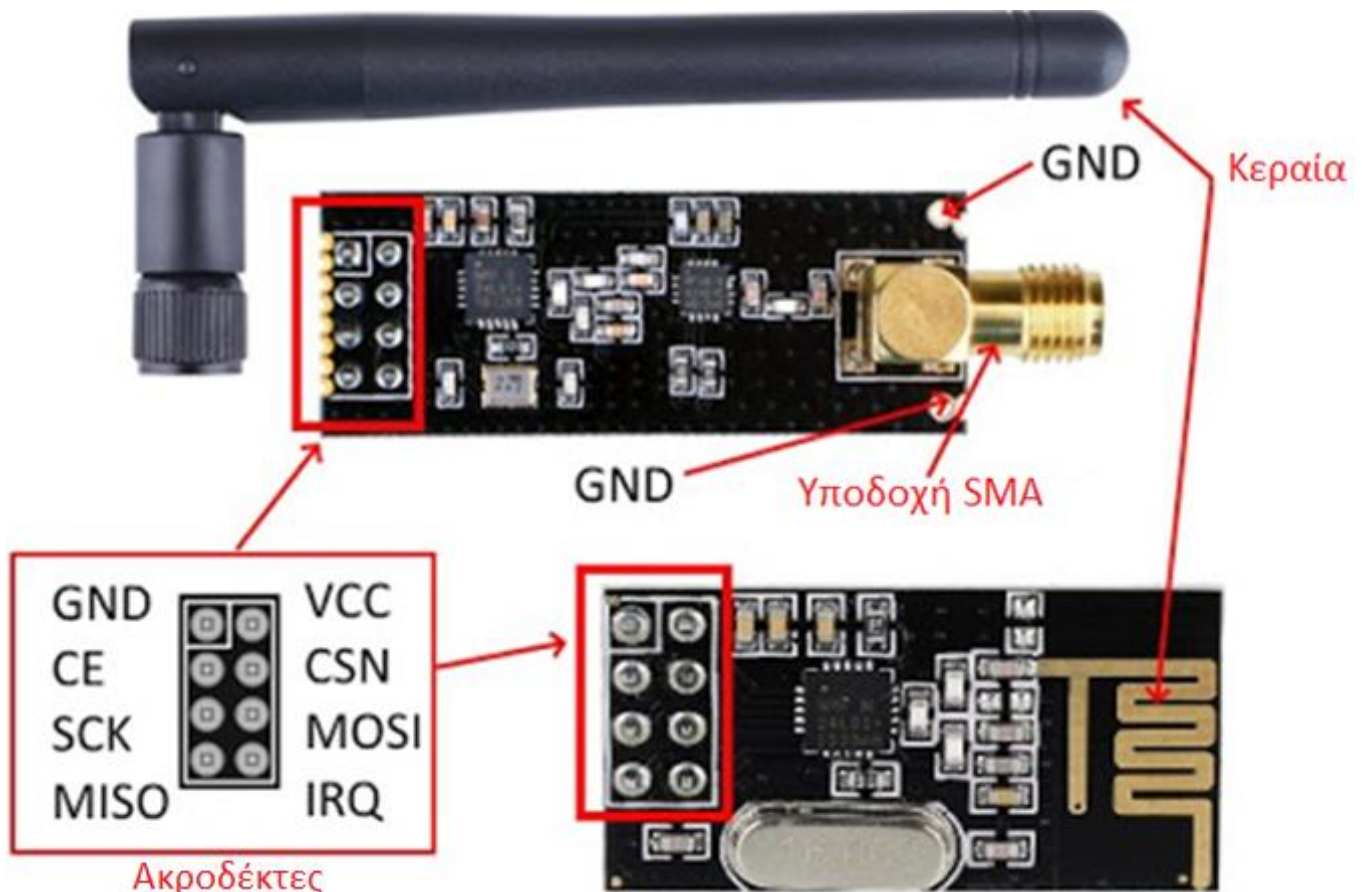
Η τρίτη παραλλαγή του μοντέλου NRF24L01 διαθέτει υποδοχή SMA για σύνδεση κεραίας και ένα τσιπ RFX2401C (PA + LNA) που περιλαμβάνει ενισχυτή ισχύος (*power amplifier*) και ενισχυτή χαμηλού θορύβου (*low-noise amplifier*), έχοντας ως επιπλέον πλεονέκτημα την ενίσχυση του σήματος και άρα επιτρέποντας ακόμη μεγαλύτερη εμβέλεια μετάδοσης που εκτιμάται έως και 1000m σε ανοιχτό χώρο.



Εικόνα 2.31 Μοντέλο πομποδέκτη NRF24L01 με υποδοχή SMA για εξωτερική κεραία και RFX2401C chip (PA + LNA)

2.4.1.1 Πομποδέκτες NRF24L01 με ενσωματωμένη κεραία

Ένα θετικό χαρακτηριστικό σε όλα τα μοντέλα NRF24L01 είναι ότι διαθέτουν την ίδια σειρά ακροδεκτών και ως αποτέλεσμα ο τρόπος συνδεσμολογίας τους σε μια διάταξη καθίσταται απλοϊκός και εύκολος.



Εικόνα 2.32 Όλες οι παραλλαγές του πομποδέκτη NRF24L01 διαθέτουν τους ίδιους ακροδέκτες (pins)

Για την επίτευξη της ασύρματης επικοινωνίας μεταξύ των υποσυστημάτων πομπού και δέκτη, επιλέγεται κατά κόρων η χρήση του μοντέλου πομποδέκτη NRF24L01 με ενσωματωμένη κεραία, αλλά στις τελευταίες υλοποιήσεις της Διπλωματικής εργασίας γίνεται επίσης χρήση του μοντέλου πομποδέκτη NRF24L01 υποδοχής SMA για εξωτερική κεραία και τσιπ RFX2401C (PA + LNA) για περαιτέρω πειραματικούς ελέγχους.

2.5 Συσκευές Εξόδου

2.5.1 Σερβοκινητήρες Συνεχούς Ρεύματος

Ο σερβοκινητήρας (*servomotor*) είναι μια διάταξη που με μεταβολή στην (ηλεκτρική) είσοδό του επιτρέπει τον ακριβή έλεγχο της κίνησης ενός εξαρτήματος για διάφορες εφαρμογές όπως η περιστροφή ενός γραναζιού, το άνοιγμα μιας βαλβίδας ή το κλείσιμο ενός διακόπτη, σε αντίθεση με τον ηλεκτροκινητήρα ή μοτέρ (*electric motor*) ο οποίος διατηρεί συνεχή περιστροφική κίνηση για όσο του παρέχεται ισχύς από την (ηλεκτρική) είσοδο.

Ανάλογα με τον σχεδιασμό τους, οι σερβοκινητήρες μπορεί να είναι περιστροφικοί ή γραμμικοί και άρα ελέγχουν την γωνιακή ή γραμμική θέση, την ταχύτητα και την επιτάχυνση ενός συγκεκριμένου εξαρτήματος ή συστήματος.



Εικόνα 2.33 Περιστροφικοί και γραμμικοί σερβοκινητήρες διαφόρων μεγεθών

Σε αντίθεση με τους ηλεκτρικούς και τους βηματικούς (*stepper*) κινητήρες που εκτελούν περιστροφική κίνηση 360 μοιρών, οι σερβοκινητήρες περιορίζονται σε ένα εύρος περιστροφής 90, 180, 270 ή 360 μοιρών αναλόγως το μοντέλο και χρησιμοποιούνται ευρέως σε βιομηχανικές εφαρμογές και συστήματα που περιλαμβάνουν μικροελεγκτές.



Εικόνα 2.34 Εφαρμογές περιστροφικών σερβοκινητήρων σε αυτοκινούμενα οχήματα, βιντεοκάμερες και ρομποτικούς βραχίονες

Οι σερβοκινητήρες χωρίζονται σε δύο κατηγορίες:

- Σερβοκινητήρες εναλλασσόμενου ρεύματος (*AC servo motors*) με είσοδο ψηφιακού ελέγχου που κοστίζουν εκατοντάδες ή χιλιάδες ευρώ και χρησιμοποιούνται σε εφαρμογές που απαιτούν γρήγορες αποκρίσεις, όπως σε ανελκυστήρες κτιρίων ή στα συστήματα ελέγχου των μεταφορικών μέσων (*πλοία, υποβρύχια, αεροπλάνα*)
- Σερβοκινητήρες συνεχούς ρεύματος (*DC servo motors*) με είσοδο ψηφιακού ή αναλογικού σήματος που είναι αρκετά φθηνοί και διατίθενται σε διάφορα μεγέθη και με αφθονία για χρήση σε πληθώρα βιομηχανικών εφαρμογών (*σύστημα διαφόρων ειδών ελέγχου μιας περιοχής, έξυπνο σπίτι*).

Η απλούστερη και συνηθέστερη μορφή ενός DC σερβοκινητήρα αποτελείται από έναν μικρό ηλεκτρικό κινητήρα που οδηγεί ένα σύστημα γρναζιών και ένα ποτενσιόμετρο συνδεδεμένο με έναν άξονα. Επί του άξονα τοποθετείται ειδικά σχεδιασμένος μοχλός που αξιοποιείται ως μηχανική έξοδος για διάφορες εφαρμογές, δύναται να εκτελεί περιστροφική ή γραμμική μετατόπιση ανάλογα το σχεδιασμό του μοντέλου και συναντάται σε διάφορα σχήματα και μεγέθη.

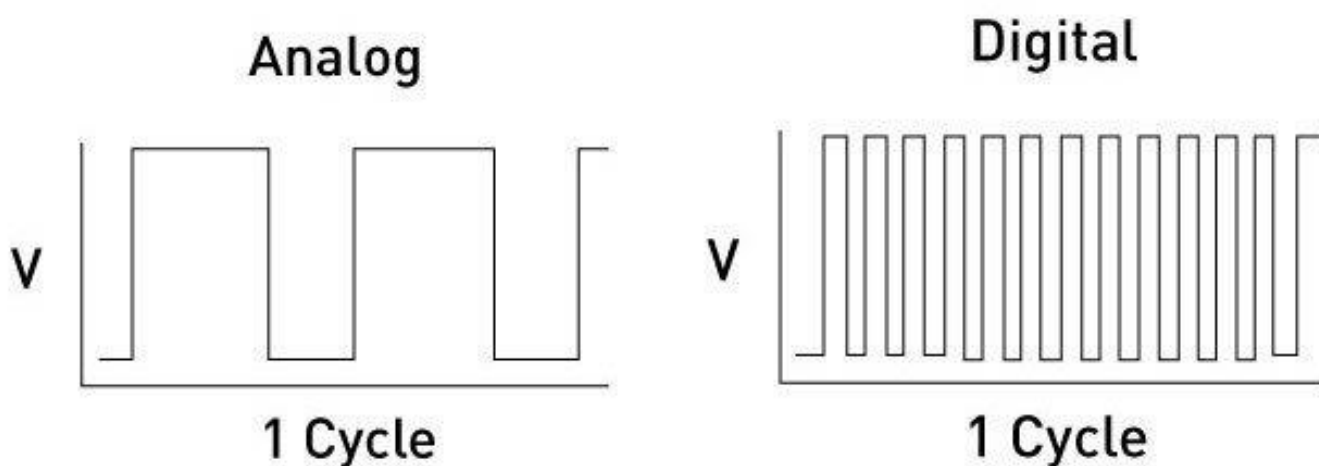


Εικόνα 2.35 Τυπικός περιστροφικός σερβοκινητήρας με διαφορετικούς μοχλούς επί του άξονά του

Όσον αφορά την λειτουργία ενός DC σερβοκινητήρα με ένα Arduino, τόσο οι αναλογικοί όσο και οι ψηφιακοί σερβοκινητήρες λειτουργούν το ίδιο με την μόνη διαφορά στον τρόπο με τον οποίο σηματοδοτούν και επεξεργάζονται πληροφορίες.

Οι αναλογικοί σερβοκινητήρες λειτουργούν βάσει σημάτων τάσης που προέρχονται από διαμόρφωση πλάτους παλμού (*PWM*). Όταν ένας αναλογικός σερβοκινητήρας βρίσκεται σε ηρεμία, η μέθοδος PWM είναι απενεργοποιημένη μέχρι ο σερβοκινητήρας να δεχθεί κάποια εντολή από τον χρήστη ώστε να αρχίσει να περιστρέφεται δημιουργώντας ροπή. Αυτός όμως ο τρόπος λειτουργίας δημιουργεί καθυστερήσεις στην παραγωγή της επιθυμητής ροπής, γεγονός που καθιστά τους αναλογικούς σερβοκινητήρες μη επιλέξιμους για εξειδικευμένες εφαρμογές.

Οι ψηφιακοί σερβοκινητήρες διαθέτουν έναν μικρό μικροεπεξεργαστή που λαμβάνει και κατευθύνει πληροφορίες (εντολές) σε παλμούς τάσης υψηλής συχνότητας, επιτυγχάνοντας αποστολή παλμών σε ποσότητα έξι φορές περισσότερο από αυτήν που εκτελεί ένα αναλογικό σήμα. Το γεγονός αυτό καθιστά τους ψηφιακούς σερβοκινητήρες γρηγορότερους και την ροπή που παράγουν συνεχή και με γρηγορότερη απόκριση ως προς τις εκάστοτε εντολές εισόδου από τον χρήστη.

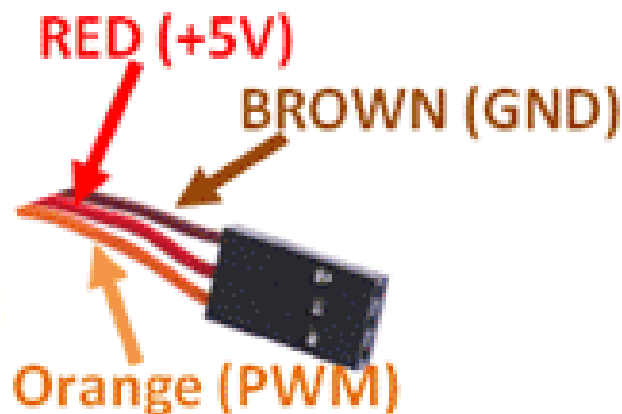


Εικόνα 2.36 Αποστολή πληροφοριών με την μορφή παλμών τάσης από έναν αναλογικό και έναν ψηφιακό σερβοκινητήρα, αντίστοιχα

Όπως παρουσιάζεται και στις εικόνες της συγκεκριμένης υποενότητας, οι σερβοκινητήρες συνήθως διαθέτουν τρία ενσωματωμένα καλώδια που καταλήγουν σε συνδετήρα με τρεις ακροδέκτες (*3-pin connector*).

Τα τρία καλώδια τα οποία καταλήγουν σε ακροδέκτες μπορεί να διαφέρουν σε χρώμα ανά τύπο μοντέλου, οι ακροδέκτες όμως καθ' αυτοί διατηρούν τα χαρακτηριστικά τους ως εξής:

- **CONTROL (Signal Line)**: είναι η είσοδος του σήματος ελέγχου PWM και το καλώδιο έχει συνήθως κίτρινο ή πορτοκαλί χρώμα
- **POWER (VIN)**: είναι η θετική τάση που τροφοδοτεί τον σερβοκινητήρα και το καλώδιο φέρει συνήθως κόκκινο ή κόκκινο-καφέ χρώμα
- **GND**: αναφέρεται στην γείωση και το καλώδιο είναι συνήθως μαύρου ή καφέ χρώματος



Εικόνα 2.37 Ακροδέκτες με συνήθη χρώματα ενός τυπικού σερβοκινητήρα.



Εικόνα 2.38 Διάφορα μοντέλα αναλογικών & ψηφιακών περιστροφικών σεβροκινητήρων και είδη μοχλών τοποθέτησης επί του άξονά τους

Προδιαγραφές ενός σερβοκινητήρα

Δύο από τις προδιαγραφές μείζονος σημασίας κατά την επιλογή ενός σερβοκινητήρα είναι η ταχύτητα και η ροπή που μπορεί να αναπτύξει.

Η ταχύτητα (*speed*) ορίζεται ως ο χρόνος που απαιτείται για την περιστροφική κίνηση του άξονα και μετριέται σε δευτερόλεπτα ανά μοίρες (*sec / °*).

Παίρνοντας ως παράδειγμα το μοντέλο αναλογικού σερβοκινητήρα **Analog Servo Motor SG-90 περιστροφής 180°** της κατωτέρω εικόνας, ο οποίος χρησιμοποιείται συχνά σε δοκιμές εφαρμογών που σχετίζονται με ηλεκτρονικά κυκλώματα και συστήματα αυτοματισμού, η ταχύτητα περιστροφής (*operating speed*) βάσει χαρακτηριστικών του είναι 0.12sec/60° που υποδεικνύει ότι ο άξονας περιστρέφεται κατά 60 μοίρες σε 0.12 δευτερόλεπτα όταν τροφοδοτείται με τάση 4,8V δίχως φορτίο.



Specifications:

Weight: 9g
Dimension: 23x12.2x29mm
Stall torque: 1.8kg/cm(4.8v)
Gear type: POM gear set
Operating speed: 0.12 sec/60degree(4.8v)
Operating voltage: 4.8v
Temperature range: 0°C_ 55°C
Dead band width: 1us
Power Supply: Through External Adapter
servo wire length: 25 cm
Servo Plug: JR (Fits JR and Futaba)

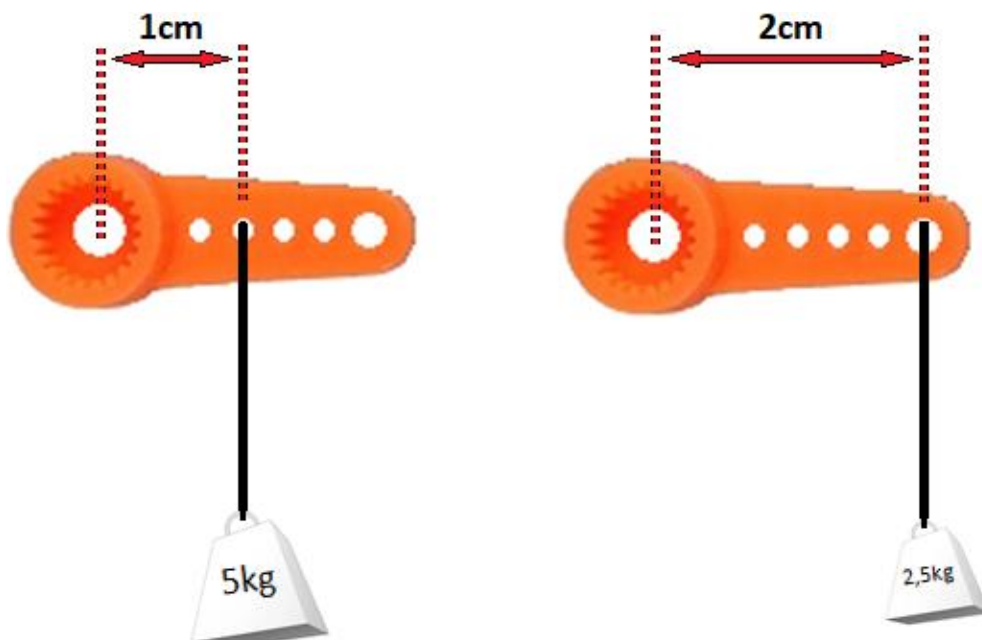
Εικόνα 2.39 Χαρακτηριστικά του αναλογικού σερβοκινητήρα μοντέλου SG90, βάρους 9g

Η ροπή (*torque*) ορίζεται ως το μέγεθος της δύναμης που μπορεί να ασκήσει ένας σερβοκινητήρας σε έναν μοχλό – ή αλλιώς πόσο βάρος μπορεί να διαχειριστεί ένας μοχλός – κατά την εκτέλεση περιστροφικής κίνησης του άξονα.

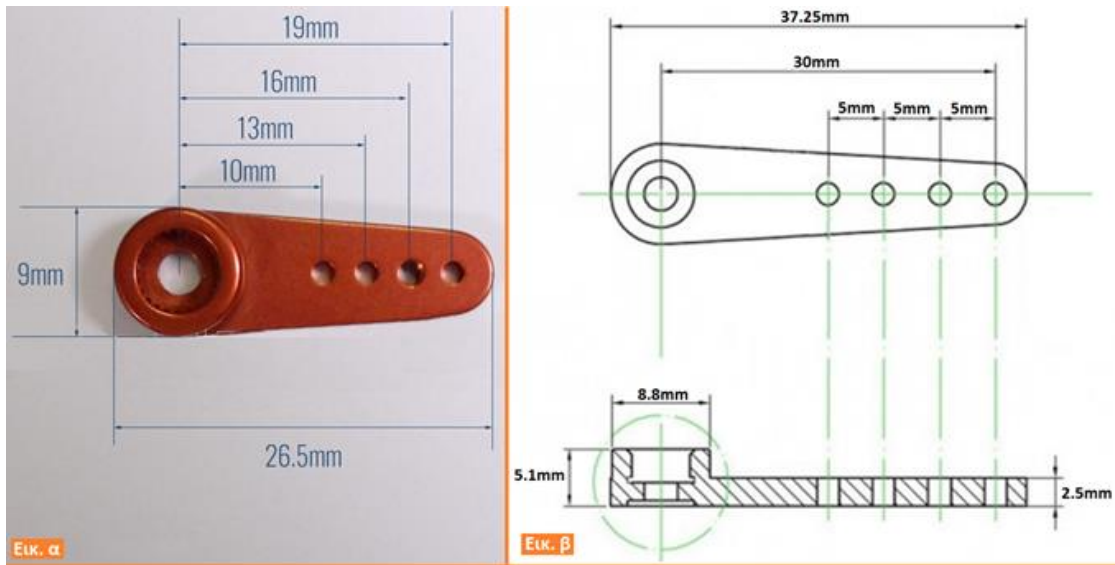
Μετριέται σε συσχέτιση του άξονα με τον μοχλό σε μια σχέση «ουγκιές*ίντσες» (*ounces*inches*), «δύναμη Νιούτον επί μέτρο» ($N*m$) ή «χιλιόγραμμα*εκατοστά» ($kg*cm$), αναλόγως την προτιμώμενη κατά περίπτωση μορφή έκφρασης της κλίμακας μέτρησης στο *SI*.

Όντας ουσιαστικά ένα μέτρο «δύναμης περιστροφής», η ροπή που αναπτύσσει ένας σερβοκινητήρας – αλλά και οποιοσδήποτε κινητήρας γενικότερα – καθορίζει το πόσο ισχυρός είναι. Ανάλογα με την εκάστοτε επιθυμητά οριζόμενη απόσταση που σχηματίζεται μεταξύ κάποιου σημείου του μοχλού και του άξονα, η ποσότητα δύναμης που μπορεί να ασκηθεί ποικίλει.

Χρησιμοποιώντας ως παράδειγμα τις δύο περιπτώσεις της κατωτέρω εικόνας, ένας σερβοκινητήρας $5kg*cm$ σημαίνει ότι ο σερβοκινητήρας αυτός μπορεί να υποστηρίξει αντικείμενο βάρους $5kg$ όταν αυτό τοποθετηθεί σε σημείο του μοχλού τέτοιο που να σχηματίζεται απόσταση $1cm$ μεταξύ του εν λόγω σημείου και του σημείου πάγιας θέσης του άξονα, ενώ μία απόσταση των $2cm$ από την θέση του άξονα καθιστά τον σερβοκινητήρα ικανό να υποστηρίξει βάρος έως και $2,5kg$ και ούτω καθεξής.



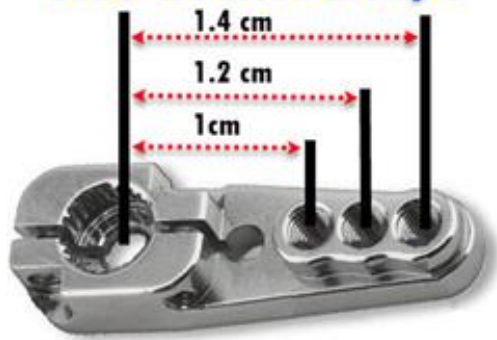
Εικόνα 2.40 Διαφορετικά σημεία επί του μοχλού δύνανται να ασκούν διαφορετικές δυνάμεις και άρα ο μοχλός να διαχειρίζεται κατ' αναλογία διαφορετικά βάρη



Εικ. α

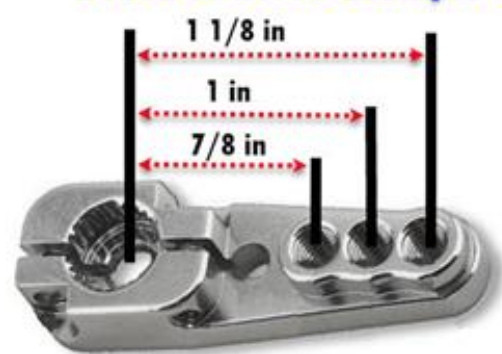
Εικ. β

20KG-CM Servo Example



Εικ. γ

240oz-in Servo Example



Εικόνα 2.41 Βάσει των χαρακτηριστικών κατασκευής τους, οι μοχλοί των περιστροφικών σερβοκινητήρων δύνανται να χρησιμοποιούνται σε πληθώρα εφαρμογών

Επίσης, τρία ακόμη μεγέθη που αναγράφονται στην λίστα των χαρακτηριστικών ενός σερβοκινητήρα και αποτελούν σημαντικές παραμέτρους για την αγορά και χρήση του, είναι:

- **η ονομαστική ροπή** (*rated torque*), δηλαδή η μέγιστη ροπή που αναπτύσσεται όταν ο σερβοκινητήρας περιστρέφεται με την ονομαστική του ταχύτητα
- **η ροπή στάσης** (*stall torque*), δηλαδή η ποσότητα ροπής που προκαλεί την διακοπή λειτουργίας του κινητήρα ή αλλιώς βάρος το οποίο δεν μπορεί πλέον να διαχειριστεί
- **η μέγιστη ροπή** (*peak torque*), δηλαδή η μέγιστη τιμή που μπορεί να φτάσει η ροπή για πολύ μικρές χρονικές περιόδους (*τάξη των msec*)

Επιπροσθέτως των ανωτέρω προδιαγραφών λαμβάνεται υπόψη και το μέγεθος του σερβοκινητήρα, καθώς όσο μεγαλύτερο είναι το μέγεθος του κινητήρα τόσο μεγαλύτερο και το μέγεθος του άξονα και κατά συνέπεια και η δύναμη που μπορεί να ασκηθεί με την προσθήκη κατάλληλου μοχλού.

Με αξιοποίηση των ανωτέρω γνώσεων, η επιλογή του κατάλληλου σερβοκινητήρα για επιθυμητές εφαρμογές γίνεται ευκολότερη.

2.5.1.1 Ψηφιακοί Σερβοκινητήρες Συνεχούς Ρεύματος

Για την έξοδο του υποσυστήματος δέκτη γίνεται επιλογή ενός ζεύγους ψηφιακών περιστροφικών σερβοκινητήρων μοντέλου TowerPro Digital MG995R με δυνατότητα περιστροφής του άξονα κατά 180° και ενός ζεύγους ψηφιακών περιστροφικών σερβοκινητήρων TowerPro Digital MG995 με δυνατότητα συνεχούς περιστροφής του άξονα κατά 360°, καθώς τέτοιοι τύποι μοντέλων σερβοκινητήρων διατίθενται στην αγορά σε χαμηλό κόστος και αφθονία.

Σκοπός των τεσσάρων αυτών συσκευών ως έξοδοι του συστήματος ηλεκτρονικού ασύρματου ελεγκτή είναι να δίδεται σε έναν χρήστη η δυνατότητα ελέγχου δύο διαφορετικών εφαρμογών στον ίδιο χώρο, καθώς ο σχεδιασμός κάθε ζεύγους σερβοκινητήρα θα είναι τέτοιος ώστε να προορίζεται για διαφορετική εφαρμογή.

Χαρακτηριστικά του μοντέλου MG995R (180°)

- Τάση λειτουργίας: 4.8-7.2V
- Ταχύτητα λειτουργίας: 0,20sec/60° (4,8V) ή 0,16sec/60° (6V)
- Ροπή στάσης: 8,5 kg*cm (4,8V) ή 10 kg*cm (6V)
- Περιστροφή άξονα: 180°
- Διαστάσεις: 40,7 x 19,7 x 42,9 mm
- Βάρος: 55,0g



Εικόνα 2.42 Μοντέλο ψηφιακού σερβοκινητήρα Tower Pro MG995R 180 Degree

Χαρακτηριστικά του μοντέλου MG995 (360°)

- Τάση λειτουργίας: 4.8 - 7.2V
- Ταχύτητα λειτουργίας: 0,20sec/60° (4,8V) ή 0,19sec/60° (6V)
- Ροπή στάσης: 13kg*cm (4,8V) ή 15kg*cm (6V)
- Περιστροφή άξονα: 360°
- Διαστάσεις: 40 x 19 x 43 mm / 54 x 43 x 20 mm
- Βάρος: 69,0g / 77,0g



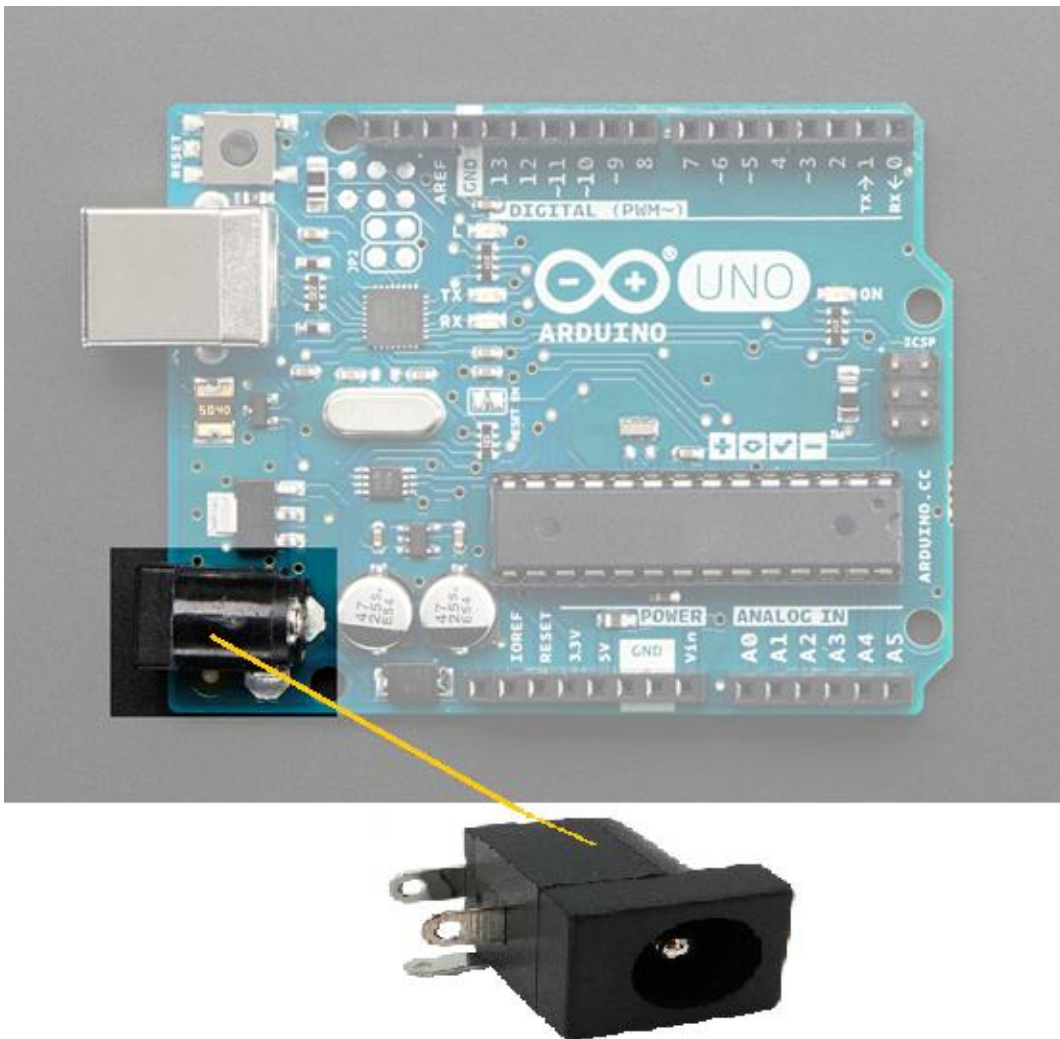
Εικόνα 2.43 Μοντέλο ψηφιακού σερβοκινητήρα Tower Pro MG995 360 Degree

2.6 Επιλογές Τροφοδοσίας

2.6.1 Τροφοδοσία Arduino UNO R3

2.6.1.1 Σύνδεση βύσματος Power Jack με μπαταρία

Οι τεχνικές προδιαγραφές για το βύσμα power jack που διαθέτει ο μικροελεγκτής Arduino UNO R3 (εσωτερική διάμετρος 2,1mm και εξωτερική διάμετρος 5,5mm) προσδίδουν στο βύσμα του UNO ένα μέγεθος αρκετά κοινό σε σχέση με τα μεγέθη των άκρων των καλωδίων που κυκλοφορούν στην αγορά και άρα η εύρεση αντίστοιχων τύπων καλωδίων για διασύνδεση καθίσταται αρκετά εύκολη.



Εικόνα 2.44 Βύσμα Power Jack ενός Arduino UNO

Για την τροφοδοσία του Arduino UNO R3 με μπαταρία και συγκεκριμένα με μπαταρία των 9V, γίνεται επιλογή του καλωδίου-συνδετήρα τύπου T με ενσωματωμένο βύσμα Barrel Jack (9V T-type Battery Connector to Barrel Jack) για διασύνδεση της μπαταρίας με το βύσμα Power Jack που διαθέτει ο μικροελεγκτής.



Εικόνα 2.45 Δύο καλώδια-συνδετήρες, διαφορετικού σχεδιασμού, για σύνδεση μπαταρίας των 9V με Arduino UNO

Διαθέτοντας τα ακόλουθα χαρακτηριστικά:

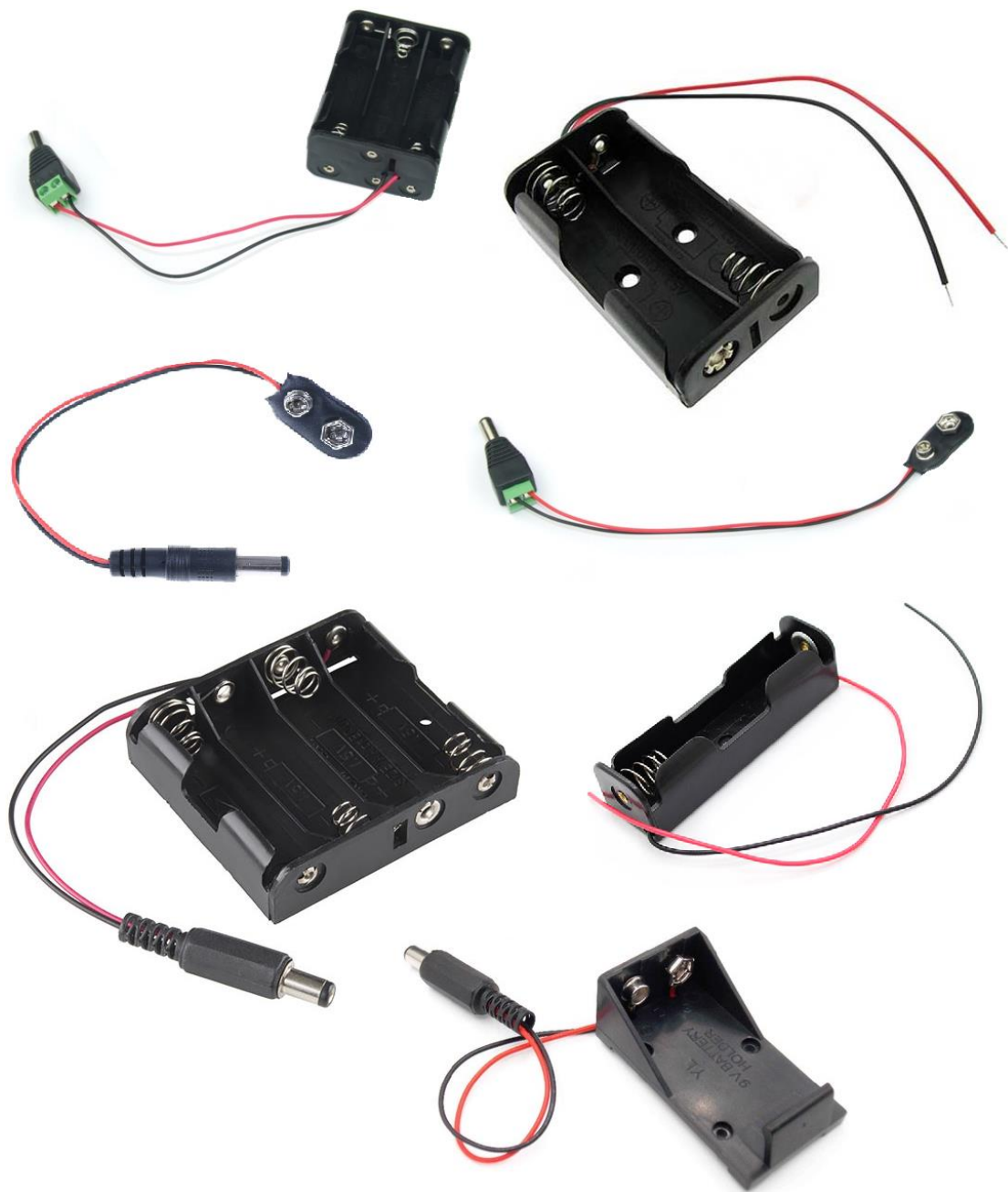
- Σκληρός πλαστικός σύνδεσμος τύπου T για σύνδεση με μπαταρία
- Καλώδιο μήκους ~15cm
 - απογυμνωμένα και κονσερβοποιημένα καλώδια μήκους 15cm: 1 κόκκινο και 1 μαύρο για εύκολη αναγνώριση πολικότητας κατά την σύνδεση με την μπαταρία
- Συνδετήρας 5.5x2.1mm για σύνδεση με το βύσμα power jack

με χρήση τέτοιου τύπου καλωδίου-συνδετήρα παρέχεται η δυνατότητα τροφοδοσίας του μικροελεγκτή Arduino UNO για αρκετά μεγάλο χρονικό διάστημα μέχρι την επόμενη αντικατάσταση της μπαταρίας.

Όσον αφορά την μπαταρία, επιλέγεται αυτή των 9V επειδή βάσει τεχνικών προδιαγραφών του Arduino το προτεινόμενο εύρος τάσης για εύρυθμη λειτουργία και ασφάλεια της ηλεκτρονικής πλακέτας είναι μεταξύ των 7V και 12V.



Εικόνα 2.46 Τυπική αλκαλική μπαταρία των 9V



Εικόνα 2.47 Διάφορα είδη καλωδίων-συνδετήρων με βύσμα Barrel Jack ή γυμνούς ακροδέκτες

ΕΠΙΣΗΜΑΝΣΗ – ΠΑΡΑΤΗΡΗΣΗ

Το βύσμα Power Jack παρέχει δύο τρόπους παροχής ενέργειας για τον μικροελεγκτή UNO, καθώς σε αυτό μπορεί να συνδεθεί είτε καλώδιο-συνδετήρας (*cable connector*) για σύνδεση με φορητό τροφοδοτικό όπως η 9V μπαταρία είτε κατάλληλο καλώδιο-προσαρμογέα (*cable adapter*) για τροφοδοσία από το δίκτυο της ΔΕΗ.

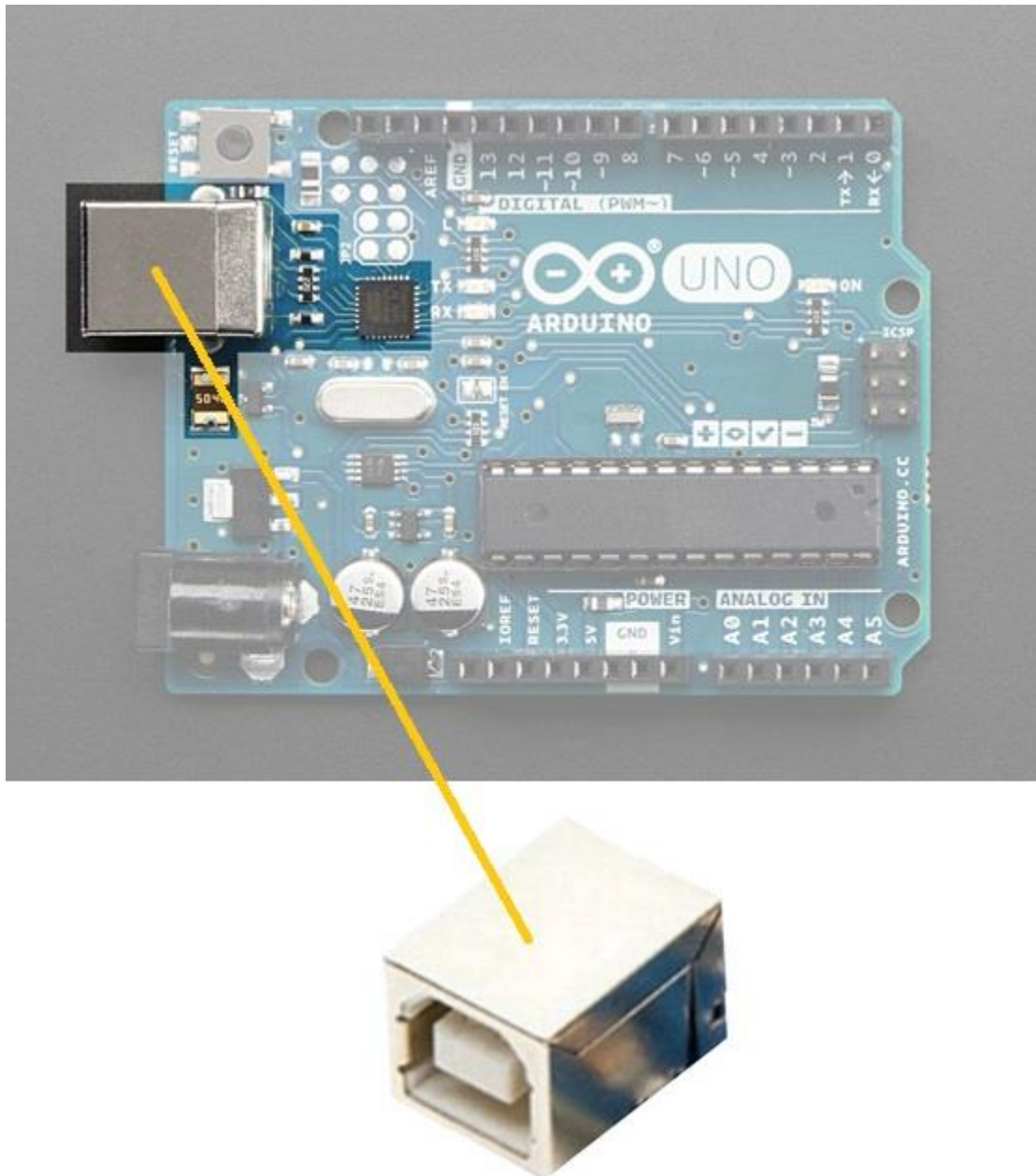
Ο δεύτερος τρόπος τροφοδοσίας από το δίκτυο της ΔΕΗ δεν επελέγη για τον σχεδιασμό του συστήματος ελέγχου και έτσι δεν περιλαμβάνεται περαιτέρω σχετική αναφορά περί αυτού στην Διπλωματική εργασία.



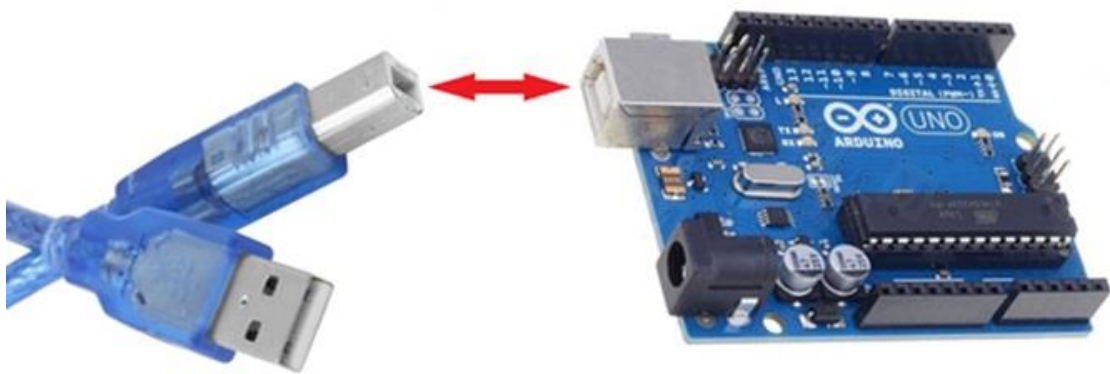
Εικόνα 2.48 Καλώδια-προσαρμογείς ή «αντάπτορες» (*cable adaptors*) με πρίζες διαφορετικού τύπου, για διασύνδεση του βύσματος Power Jack ενός Arduino UNO με αντίστοιχο τοπικό ηλεκτρικό δίκτυο.

2.6.1.2 Σύνδεση θύρας USB με Η/Υ

Όντας η πιο σημαντική θύρα του μικροελεγκτή Arduino UNO, η θύρα USB εκτός από την δυνατότητα προγραμματισμού του μικροελεγκτή όταν αυτός συνδέεται σε ηλεκτρονικό υπολογιστή παρέχει παράλληλα και ενέργεια κατά την σύνδεση.



Εικόνα 2.49 Θύρα καλωδίου USB μιας ηλεκτρονικής πλακέτας Arduino UNO



Εικόνα 2.50 Καλώδιο USB (άνω) και τρόπος διασύνδεσης (κάτω)

ΕΠΙΣΗΜΑΝΣΗ – ΠΑΡΑΤΗΡΗΣΗ

Η θύρα USB παρέχει έναν επιπλέον τρόπο παροχής ενέργειας για τον μικροελεγκτή UNO, καθώς σε αυτήν μπορεί να συνδεθεί καλώδιο USB (*USB cable*) σε συνδυασμό με προσαρμογέα USB (*USB wall adaptor*) για τροφοδοσία απευθείας από το δίκτυο της ΔΕΗ. Ο τρόπος αυτός επελέγη ως εναλλακτική τροφοδοσία για τον σχεδιασμό του συστήματος ελέγχου και σχετική ανάλυση περιλαμβάνεται στο τέλος του 3^{ου} κεφαλαίου της Διπλωματικής εργασίας.



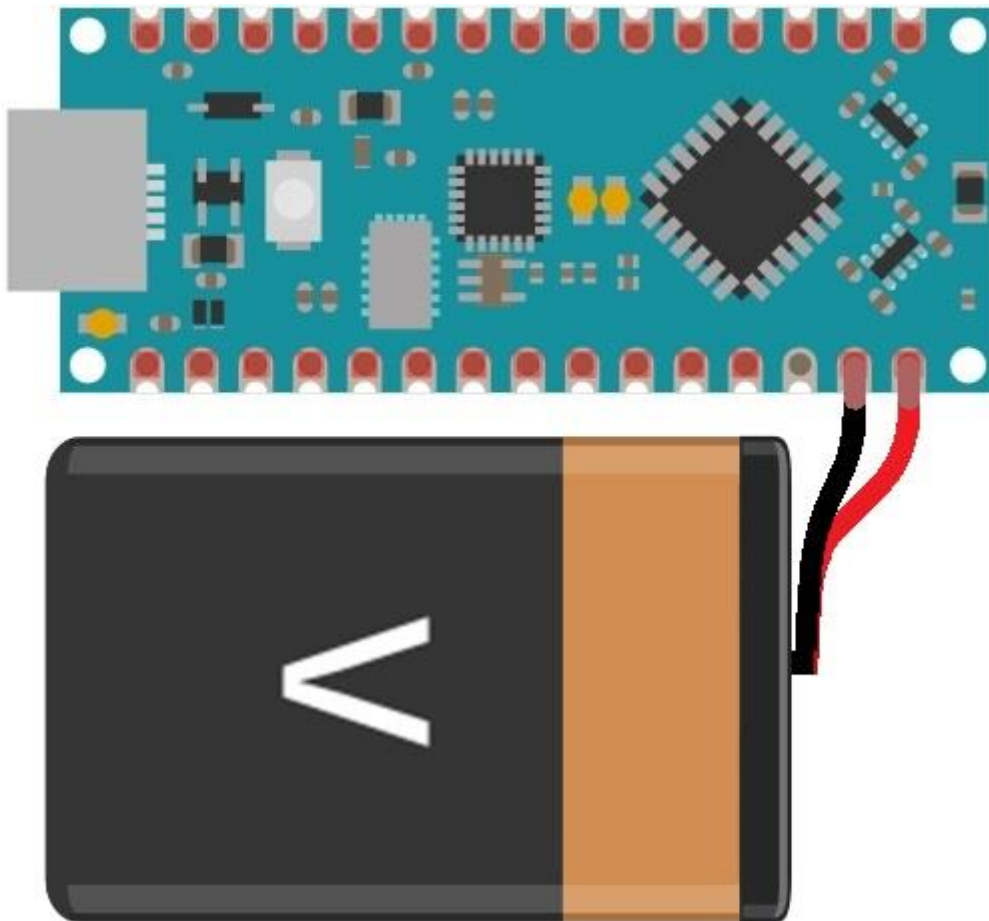
Εικόνα 2.51 Προσαρμογείς και καλώδια USB για διασύνδεση μεταξύ Arduino UNO και δικτύου της ΔΕΗ

2.6.2 Τροφοδοσία Arduino NANO EVERY

2.6.2.1 Σύνδεση Ακροδεκτών με μπαταρία

Λόγω του ότι ο μικροελεγκτής NANO EVERY δεν διαθέτει βύσμα power jack αλλά μόνο μια θύρα mini USB για διασύνδεση με ηλεκτρονικό υπολογιστή, επιλέγεται η τροφοδοσία σε αυτόν να πραγματοποιείται μέσω των ακροδεκτών VIN-GND με κατάλληλη αξιοποίηση μπαταρίας.

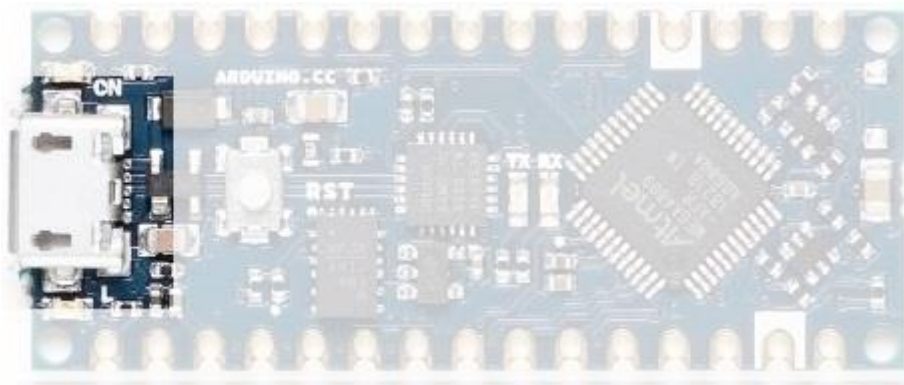
Ο εν λόγω τρόπος διασύνδεσης παρατίθεται στο 3^ο κεφάλαιο αλλά και σε υποενότητες που έπονται της παρούσης.



Εικόνα 2.52 Τροφοδοσία του Arduino NANO EVERY μέσω των ακροδεκτών VIN & GND με τάση συγκεκριμένης τιμής

2.6.2.2 Σύνδεση θύρας mini USB με Η/Υ

Ομοίως με την πλακέτα Arduino UNO, η θύρα **Mini-B USB** που διαθέτει η πλακέτα NANO χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή At-Mega4809 κατά την σύνδεση με ηλεκτρονικό υπολογιστή, παρέχοντας παράλληλα ενέργεια στην πλακέτα.



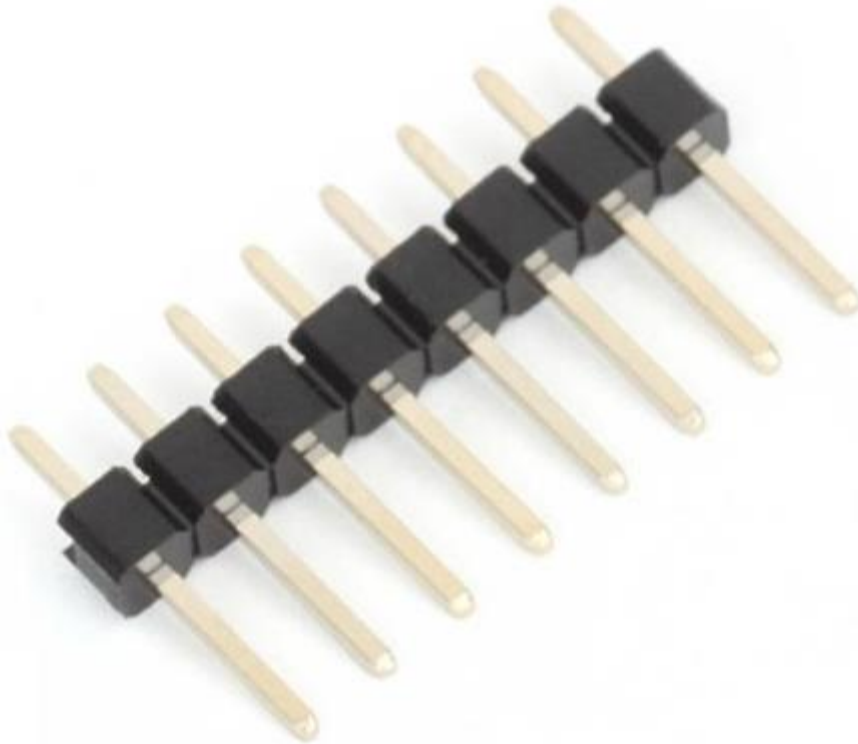
Εικόνα 2.53 Θύρα mini USB μιας ηλεκτρονικής πλακέτας Arduino NANO EVERY



Εικόνα 2.54 Ζεύγη καλωδίων με διαφορετικά άκρα (mini USB / USB)

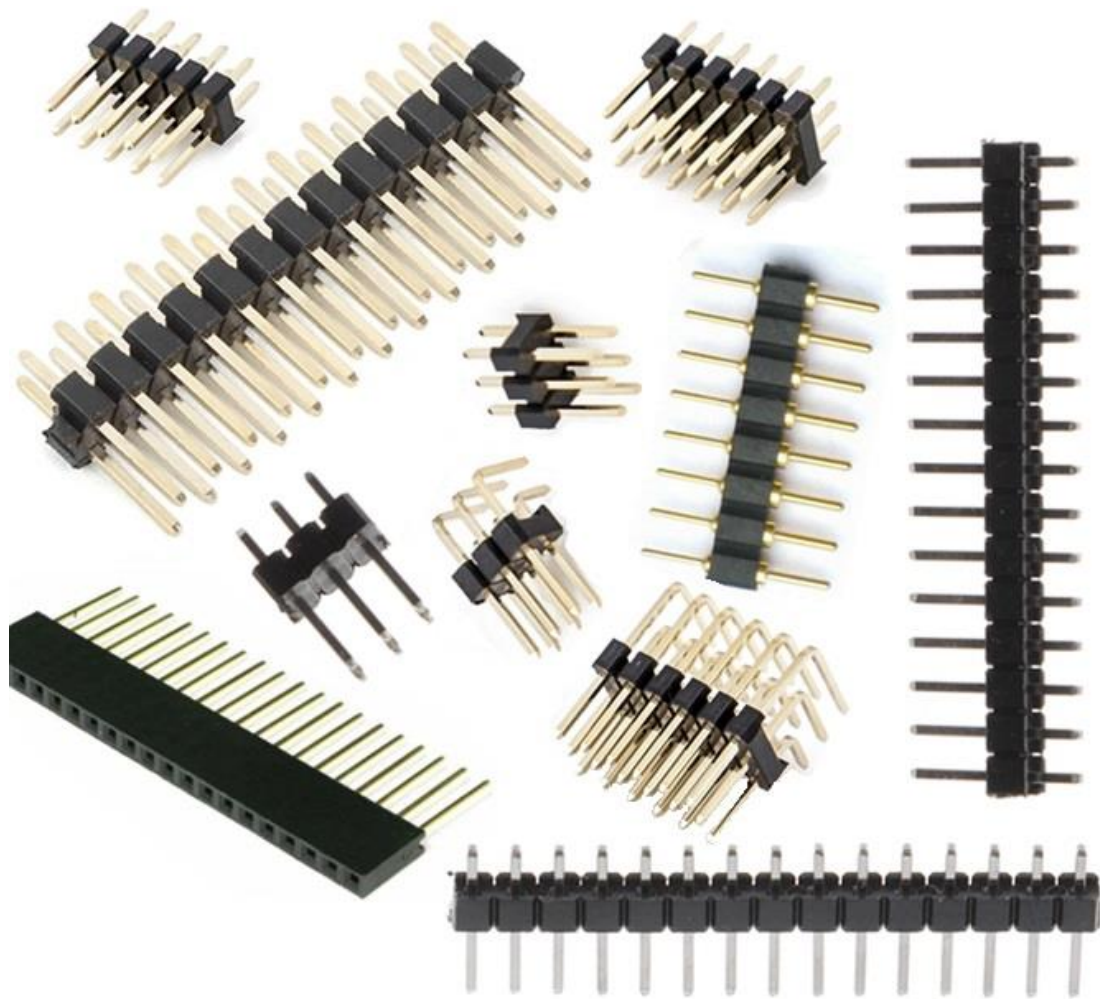
2.6.2.3 Κεφαλίδες Ακροδεκτών

Όντας ένας τύπος ηλεκτρικών συνδετήρων, οι **κεφαλίδες ή κεφαλές ακροδεκτών** (*headers ή pin headers*) χρησιμοποιούνται σε μεγάλο βαθμό στον τομέα της Ηλεκτρονικής και συναφών επιστημονικών πεδίων για την διασύνδεση μεταξύ δύο συσκευών ή εξαρτημάτων καθώς επιτρέπουν την διέλευση ρεύματος ή την μετάδοση ηλεκτρικού σήματος, αναλόγως την εκάστοτε εφαρμογή.

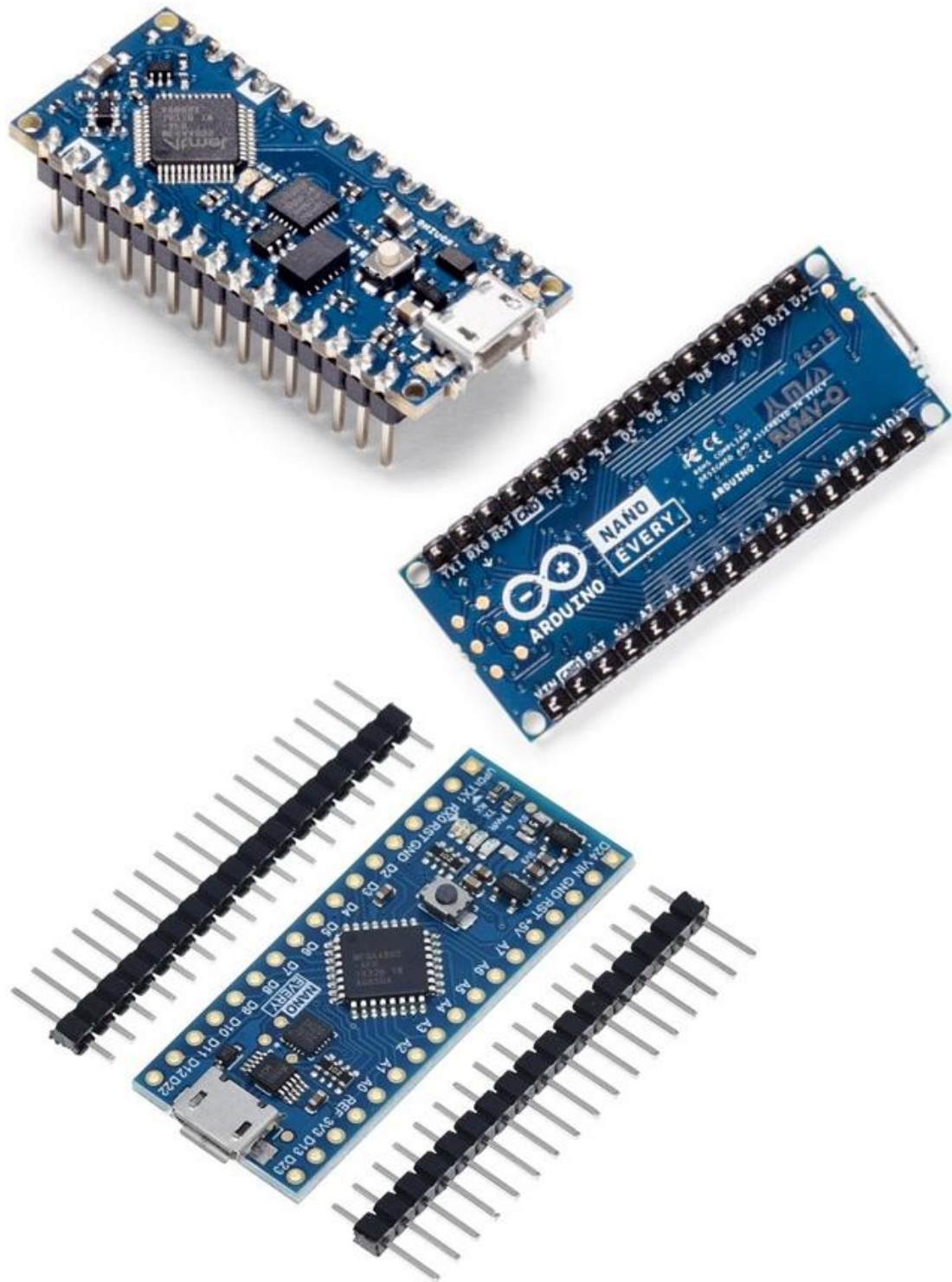


Εικόνα 2.55 Κεφαλές ακροδεκτών (pin headers) για διασύνδεση ενός Arduino με άλλα ηλεκτρονικά εξαρτήματα (π.χ. breadboards, ηλεκτρονικές πλακέτες, Arduino shields)

Η χρήση των ακροδεκτών κεφαλής επιλέγεται για δοκιμές και πιθανή ένταξη στον σχεδιασμό και υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή.



Εικόνα 2.56 Διάφοροι τύποι pin headers για διασύνδεση μεταξύ ηλεκτρονικών συσκευών και εξαρτημάτων



Εικόνα 2.57 Άνω και κάτω όψη ενός Arduino NANO EVERY με pin headers

2.6.3 Τροφοδοσία Πομποδεκτών NRF24L01

Η τροφοδοσία των δύο πομποδεκτών NRF24L01 πραγματοποιείται με διασύνδεσή τους με την αντίστοιχη ηλεκτρονική πλακέτα Arduino του εκάστοτε υποσυστήματος που απαρτίζουν το σύστημα ηλεκτρονικού ασύρματου ελεγκτή αλλά και με μονάδες βάσης (*NRF24L01 Base modules*) ειδικά διαμορφωμένες για σύνδεση με πομποδέκτες κατηγορίας NRF24L01 που παίζουν τον ρόλο προσαρμογέων (*adaptors*) ως εναλλακτική συνδεσμολογία, με αναλυτική περιγραφή επί των διαδικασιών αυτών να παρατίθεται στις υποενότητες του 3^{ου} κεφαλαίου.

2.6.4 Τροφοδοσία Σερβοκινητήρων

2.6.4.1 Σύνδεση Ακροδεκτών με μπαταρία

Βάσει χαρακτηριστικών κατασκευής, το μοντέλο σερβοκινητήρα *TowerPro* που επιλέχτηκε για την υλοποίηση του συστήματος ελέγχου λειτουργεί σε ένα εύρος τάσης μεταξύ 4,8-7,2V και ως εκ τούτου η ευνοϊκότερη φορητή πηγή τροφοδοσίας που προτείνεται θα πρέπει να δύναται να παρέχει στους σερβοκινητήρες τάση τιμής εντός των ορίων 5-6V.

Λαμβάνοντας όμως υπόψη το γεγονός ότι κατά την εκκίνησή τους οι σερβοκινητήρες καταναλώνουν ρεύμα μέγιστης έντασης, καθώς επίσης και ότι οι μπαταρίες και τα καλώδια-συνδετήρες που διατίθενται στην αγορά έχουν συγκεκριμένες τεχνικές προδιαγραφές (3V/4,8V/9V κλπ), κρίνεται απαραίτητη η δοκιμή διαφορετικών μεθόδων τροφοδοσίας για εύρεση της ευνοϊκότερης για το σύστημα ελέγχου.

Συγκεκριμένα, θα γίνουν οι εξής δοκιμές διασύνδεσης για τροφοδοσία, αρχικά με αξιοποίηση του ζεύγους σερβοκινητήρων περιστροφής 180°:

1. χρήση μίας μπαταρίας των 9V, ενός ρυθμιστή τάσης στα 5V και μιας πλακέτας δοκιμών, διασυνδεδεμένα με τρόπο τέτοιο ώστε έκαστος σερβοκινητήρας να λαμβάνει σταθερή τάση τροφοδοσίας της τάξεως των 5V
2. χρήση δύο μπαταριών των 6V εκάστη για άμεση διασύνδεση και παροχή τάσης 6V σε κάθε σερβοκινητήρα

και στην συνέχεια θα εφαρμοστούν αντίστοιχες δοκιμές διασύνδεσης και λειτουργίας για έλεγχο της τροφοδοσίας με αξιοποίηση του ζεύγους σερβοκινητήρων περιστροφής 360°.

Οι δοκιμές αυτές απαιτούν την προμήθεια δύο διαφορετικών τύπων καλωδίων-συνδετήρων για κάθε περίπτωση, έναν τύπο συνδετήρα που μπορεί να υποστηρίξει σύνδεση 9V μπαταρίας και δύο τύπους συνδετήρα που μπορούν να υποστηρίξουν προσφορά τάσης 6V έκαστος. Επιπλέον, για δυνατότητα διασύνδεσης με πλακέτα δοκιμών χωρίς κολλήσεις (*breadboard*) και παρεμβολή ρυθμιστή τάσης στα 5V, επιλέγεται η χρήση καλωδίων-συνδετήρων με γυμνά άκρα, ως φαίνονται στις εικόνες 2.58 και 2.59 της επόμενης σελίδας.

Για την τροφοδοσία του μικροελεγκτή UNO, επειδή διαθέτει βύσμα power jack για τροφοδοσία θα γίνουν δοκιμές αρχικά με χρήση καλωδίου barrel jack με 9V μπαταρία και έπειτα χρήση καλωδίου-συνδετήρα γυμνών άκρων ώστε το Arduino UNO να τροφοδοτείται

από τους ακροδέκτες του (*VIN-GND* ή εναλλακτικά *5V-GND*, αναλόγως του τρόπου συνδεσμολογίας).



Εικόνα 2.58 Καλώδια-συνδετήρες με γυμνά άκρα και καλώδια με βύσμα barrel jack, για τοποθέτηση μίας μπαταρίας των 9V

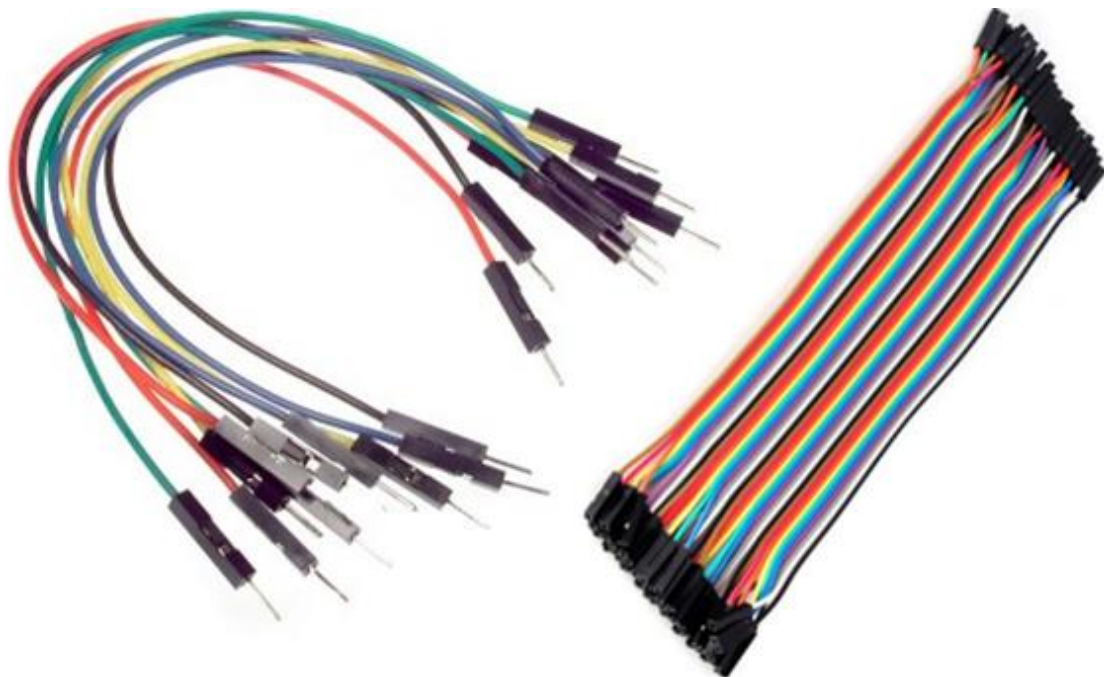


Εικόνα 2.59 Καλώδιο-συνδετήρας για τοποθέτηση τεσσάρων μπαταριών 4x1.5V AAA και σχηματισμό ενιαίας παροχής τάσης, είτε της τάξεως των 4,8V για επαναφορτιζόμενες μπαταρίες είτε της τάξεως των 6V για αλκαλικές μπαταρίες

2.7 Ηλεκτρικά Καλώδια Δοκιμών

Φυσικά από τον σχεδιασμό και την υλοποίηση του συστήματος ελέγχου δεν μπορούν να παραληφθούν τα μέσα φυσικής διασύνδεσης μεταξύ ηλεκτρονικών και ηλεκτρικών εξαρτημάτων, τα ηλεκτρικά καλώδια γνωστά ως καλώδια δοκιμών.

Τα **καλώδια δοκιμών** ή **καλώδια βραχυκύκλωσης** (*jump wires* ή *jumper wires*) είναι ηλεκτρικά καλώδια κατασκευασμένα με τρόπο τέτοιο που η διασύνδεση και αποσύνδεσή τους καθίσταται απλοϊκή, εύκολη και δίχως απαίτηση διαδικασιών συγκόλλησης ή αποκόλλησης στα άκρα των επαφών, καθώς διαθέτουν έναν συνδετήρα (*connector*) ή ακροδέκτη (*pin*) σε κάθε άκρο τους για την διασύνδεση μεταξύ εξαρτημάτων.



Εικόνα 2.60 Καλώδια δοκιμών με ακροδέκτες (αρσενικά) και καλώδια δοκιμών με σπές (θηλυκά), αντίστοιχα

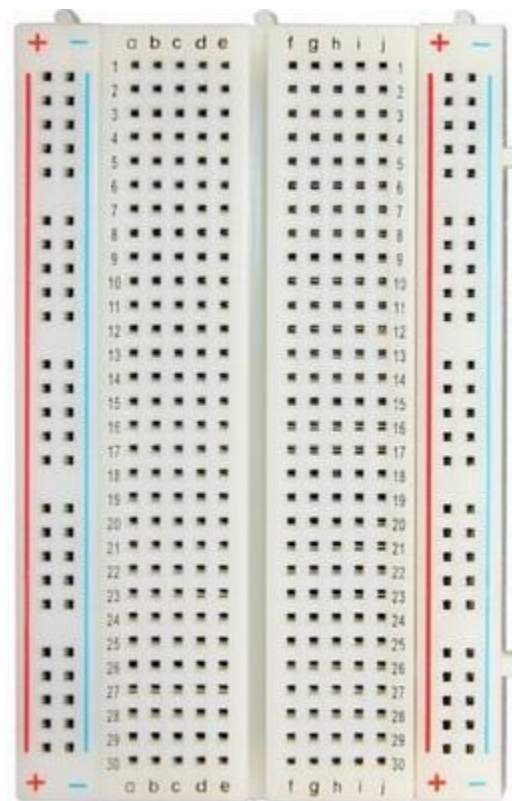
Υπάρχουν διάφοροι τύποι τέτοιων καλωδίων, με κάποια καλώδια να διαθέτουν ίδιους συνδετήρες ή ακροδέκτες στα άκρα τους ενώ σε άλλα καλώδια τα ίδια τους τα άκρα διαφέρουν.



Εικόνα 2.61 Διαφορετικοί τύποι καλωδίων δοκιμών για διαφορετικά εξαρτήματα και εφαρμογές

2.8 Πλακέτες διασύνδεσης χωρίς κολλήσεις

Με επιθυμία αποφυγής της αποκλειστικής διασύνδεσης των καλωδίων δοκιμών απευθείας στους ακροδέκτες και τους υποδοχείς των ηλεκτρονικών και ηλεκτρικών εξαρτημάτων, κρίνεται απαραίτητη η χρήση **πλακετών διασύνδεσης χωρίς κολλήσεις** (*breadboards*) – επίσης απλά γνωστές και ως **πλακέτες δοκιμών** ή **ράστερ** – και συγκεκριμένα η συμπερίληψη δύο τέτοιων πλακετών στον σχεδιασμό και υλοποίηση του συστήματος ελέγχου, μία για την διασύνδεση των μερών που θα απαρτίζουν το υποσύστημα πομπού και μία για το υποσύστημα δέκτη.



Εικόνα 2.62 Σύνθετος μοντέλο πλακέτας διασύνδεσης χωρίς κολλήσεις (*breadboard*)



Εικόνα 2.63 Διάφορα μοντέλα πλακετών διασύνδεσης χωρίς κολλήσεις

2.9 Δευτερεύουσες Συσκευές – Πρόσθετες Περιφερειακές Μονάδες

Ως έγινε αναφορά σε ανωτέρω υποενότητες, με τροποποίηση του σχεδιασμού και προγραμματισμού ενός ήδη υπάρχοντος υλοποιημένου συστήματος ελέγχου αυτό δύναται να αξιοποιείται για επιπλέον εφαρμογές από τις ήδη καθορισμένες, με αποτέλεσμα το σύστημα ελέγχου να αποκτά πληθώρα κατηγοριών στις οποίες δύναται να εφαρμοστεί, καλύπτοντας με αυτόν τον τρόπο ένα αρκετά μεγάλο εύρος αναγκών της καθημερινότητας των ανθρώπων.

Για τον λόγο αυτό γίνεται επιλογή πέντε ακόμη περιφερειακών μονάδων – συμβατών με τους μικροελεγκτές της οικογένειας Arduino – για υλοποίηση σχετικών παραδειγμάτων που επιβεβαιώνουν τις ανωτέρω σχετικές αναφορές, με τις εν λόγω μονάδες να περιγράφονται αναλυτικά στις υποενότητες που ακολουθούν.

2.9.1 Αισθητήρας Υπερήχων

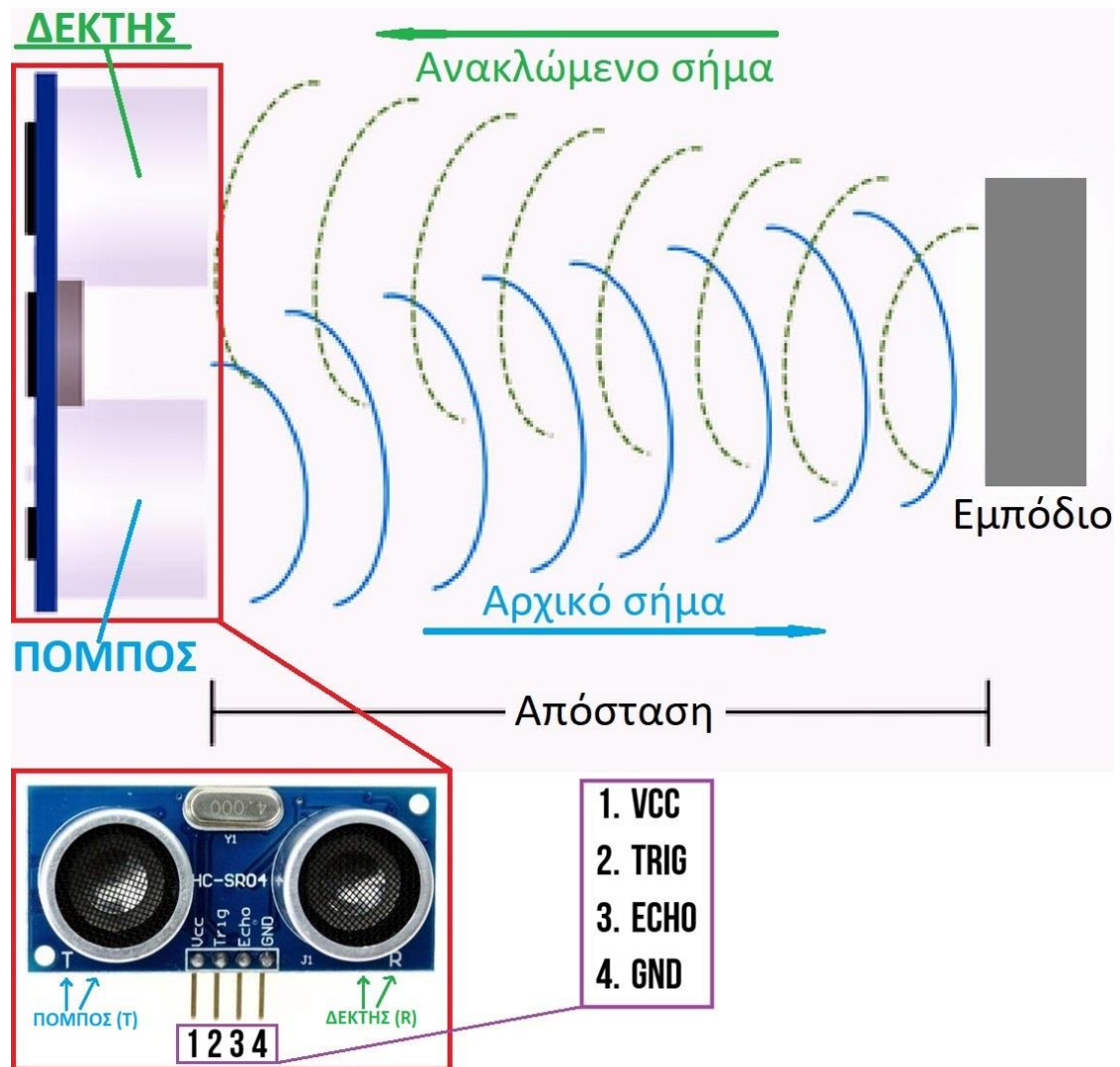
Η μονάδα **αισθητήρα υπερήχων HC-SR04** (*HC-SR04 ultrasonic sensor module*) αποτελεί μια ηλεκτρονική συσκευή που χρησιμοποιεί υπερήχους για ανέπαφη παρακολούθηση, μελέτη και υπολογισμό αποστάσεων σε συγκεκριμένο χώρο.

Χρησιμοποιείται για τον προσδιορισμό της απόστασης ενός αντικειμένου και άρα για τον έλεγχο της κίνησης στον χώρο με μέτρηση της κατεύθυνσης και της ταχύτητας του αντικειμένου.



Εικόνα 2.64 Αισθητήρας υπερήχων μοντέλου HC-SR04 (άνω) και απεικονίσεις του σε περιβάλλοντα προσομοίωσης (κάτω)

Σχεδιασμένη για μέτρηση αποστάσεων, η μονάδα αισθητήρα εκπέμπει υπερήχους στα 40kHz που ταξιδεύουν μέσω του αέρα και επιστρέφουν πίσω στην μονάδα όταν συναντήσουν κάποιο αντικείμενο ή εμπόδιο γενικότερα στην διαδρομή, με την εκάστοτε απόσταση να υπολογίζεται βάσει του χρόνου ταξιδιού και της ταχύτητας του ήχου.

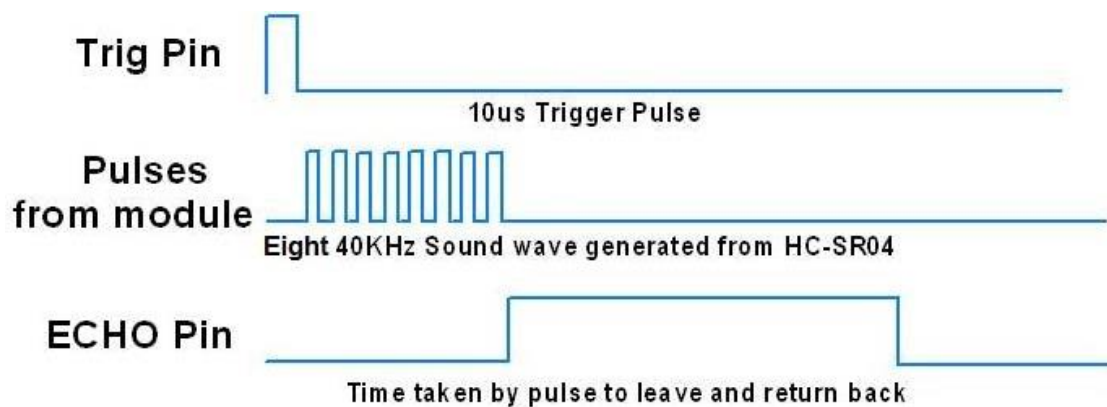


Εικόνα 2.65 Τρόπος λειτουργίας του αισθητήρα υπερήχων HC-SR04

Ακροδέκτες του αισθητήρα υπερήχων HC-SR04

1. **Vcc**: Ακροδέκτης τάσης τροφοδοσίας (5V)
2. **Trig**: Ακροδέκτης γραμμής σήματος εκπομπής για τον αισθητήρα-πομπό “Trigger”
3. **Echo**: Ακροδέκτης γραμμής σήματος λήψης (ηχούς) για τον αισθητήρα-δέκτη “Echo”
4. **Gnd**: Ακροδέκτης γείωσης (0V)

Για την δημιουργία υπερήχου απαιτείται ρύθμιση του αισθητήρα-πομπού *Trig (trigger)* σε κατάσταση “**High State**” και καθορισμένο χρόνο αποστολής σε μικροδευτερόλεπτα (μs). Έτσι εκπέμπεται από τον πομπό ένα ηχητικό κύμα που ταξιδεύει με την ταχύτητα του ήχου και λαμβάνεται από τον αισθητήρα δέκτη *Echo*, ο οποίος με την σειρά του εξάγει τον χρόνο ταξιδιού του ηχητικού κύματος σε μικροδευτερόλεπτα.

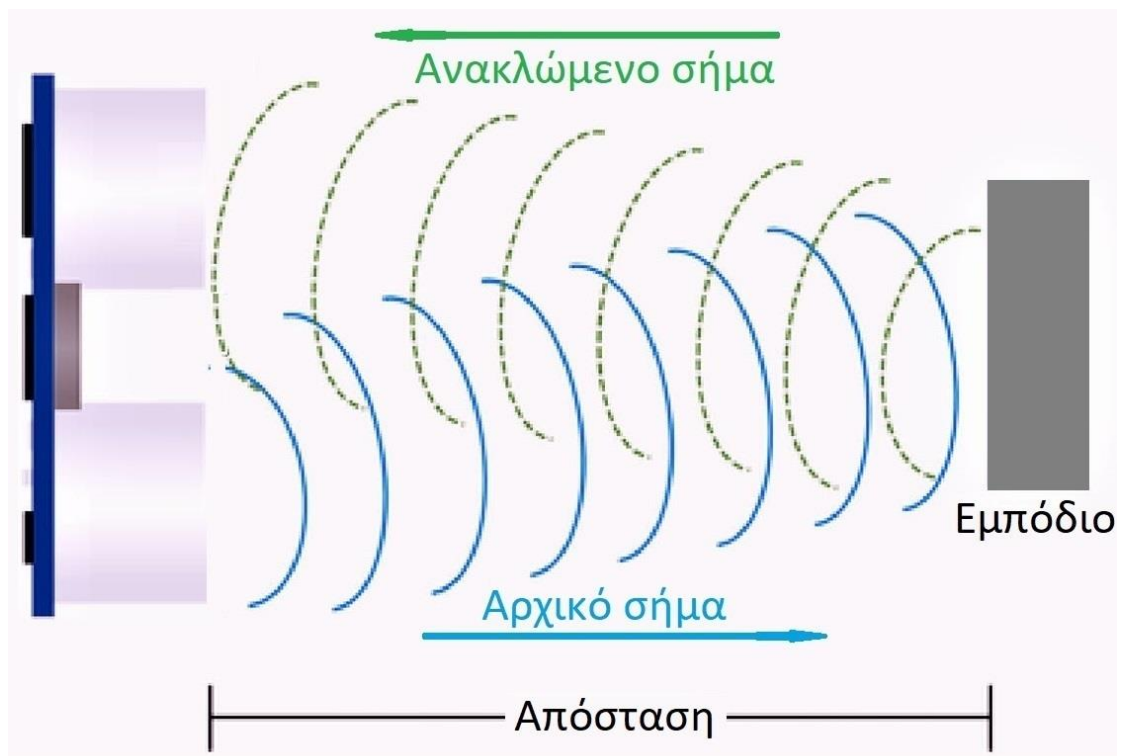


Εικόνα 2.66 Διάγραμμα χρονισμού του αισθητήρα υπερήχων HC-SR04 με ρύθμιση σε κατάσταση “High State” και χρόνο αποστολής 10 μs για εκπομπή από τον πομπό “Trig” ενός ηχητικού κύματος 8 κύκλων που θα ληφθεί από τον δέκτη “Echo”

Για παράδειγμα, εάν ένα αντικείμενο απέχει 20cm από τον αισθητήρα υπερήχων και η ταχύτητα του ήχου είναι 340 m/s ή αλλιώς 0,034 cm/μs, το ηχητικό κύμα θα ταξιδέψει περίπου ($20 / 0,034 =$) 588μs.

Ο ακροδέκτης Echo όμως θα λάβει το ανακλώμενο σήμα (ηχώ) σε διπλάσιο χρόνο λόγω του ότι το ηχητικό σήμα πρέπει να ταξιδέψει από τον αισθητήρα προς το αντικείμενο και ύστερα από το αντικείμενο πίσω στον αισθητήρα.

Επομένως για τον υπολογισμό της απόστασης σε cm απαιτείται πολλαπλασιασμός της τιμής του λαμβανόμενου χρόνου ταξιδιού από τον ακροδέκτη Echo με 0,034cm/μs και διαίρεση με το 2.



Ταχύτητα του ήχου = 0,034cm/μs.

Χρόνος = Απόσταση / Ταχύτητα ήχου =
= 20 cm / 0,034 cm/μs =
= 588μs.

Υπολογισμός απόστασης αισθητήρα από εμπόδιο (σε cm):

[588μs * 0,034cm/μs] / 2.

Εικόνα 2.67 Παράδειγμα υπολογισμού του χρονισμού λειτουργίας του αισθητήρα HC-SR04

2.9.2 Πιεζοηλεκτρικός Βομβητής / Buzzer

Όντας συσκευές ηχητικής σηματοδότησης, οι βομβητές (*buzzers / beepers*) απαντώνται σε συσκευές συναγερμού (*ζυπνητήρι, σειρήνα*), υπολογιστές, ρολόγια (*ψηφιακά ρολόγια χαλαζία*), χρονόμετρα, συσκευές ελέγχου και επιβεβαίωσης επιτυχούς εκτέλεσης εντολών εισόδου (*πάτημα πλήκτρων πληκτρολογίου και κόρνας οχήματος, λήψη πληροφοριών από αισθητήρα*) και γενικότερα σε συσκευές ηλεκτρονικής φύσεως.

Όσον αφορά τον σχεδιασμό και τον σκοπό χρήσης, ένας βομβητής μπορεί να είναι μηχανικός, ηλεκτρομηχανικός ή πιεζοηλεκτρικός.

Οι **πιεζοηλεκτρικοί βομβητές** (*piezoelectric buzzers*), πιεζο-βομβητές (*piezo-buzzers*) ή πιεζοηλεκτρικά ηχεία (*piezo speakers*) είναι συσκευές ηχητικής σηματοδότησης που δεδομένου των υλικών και τρόπου κατασκευής τους δύνανται να αξιοποιούν τον πιεζοηλεκτρισμό για την παραγωγή ήχου.



Εικόνα 2.68 Πιεζο-βομβητής ή αλλιώς πιεζο-ηχείο (άνω αριστερά) και απεικόνισή του σε περιβάλλον προσομοίωσης (κάτω δεξιά)

Τεχνικά χαρακτηριστικά ενός πιεζο-βομβητή

- **Ακροδέκτης +** : Τροφοδοσία του βομβητή, αναγνωρίζεται από το σύμβολο “+” ή τον ακροδέκτη με το μεγαλύτερο μήκος καλωδίου
- **Ακροδέκτης -** : Γείωση του βομβητή, αναγνωρίζεται από το σύμβολο “-” ή τον ακροδέκτη με το μικρότερο μήκος καλωδίου
- **Τάση λειτουργίας**: Αναλόγως μοντέλου (συνήθως 4-8V)
- **Ονομαστικό ρεύμα**: Αναλόγως μοντέλου (συνήθως < 30mA)
- **Τύπος ηχείου**: Συνεχές ηχητικό σήμα
- **Συχνότητα συντονισμού**: ~ 2.300Hz
- **Διαστάσεις**: 12mm διάμετρος / 18mm ύψος / 2g βάρος
- **Ευκολία διασύνδεσης με Arduino και πλακέτες δοκιμών**



Εικόνα 2.69 Διάφορα είδη πιεζο-βομβητών και πολικότητα ακροδεκτών τους

2.9.3 Δίοδοι Εκπομπής Φωτός / LEDs

Ως **δίοδος εκπομπής φωτός** (Light Emitting Diode) ή απλά **LED** χαρακτηρίζεται μια ημιαγώγιμη πηγή φωτός που εκπέμπει φως κατά το χρονικό διάστημα ροής ρεύματος στο εσωτερικό της.

Τα ηλεκτρόνια στον ημιαγωγό ανασυνδυάζονται με οπές ηλεκτρονίων, απελευθερώνοντας ενέργεια με τη μορφή φωτονίων. Το χρώμα του φωτός (που αντιστοιχεί στην ενέργεια των φωτονίων) καθορίζεται από την ενέργεια που απαιτείται για να διασχίσουν τα ηλεκτρόνια το διάκενο ζώνης (*band gap*) του ημιαγωγού.

Διαθέτοντας διάφορα σχήματα, χρώματα και μεγέθη ανά μοντέλο κατασκευής, για την παρούσα υλοποίηση γίνεται επιλογή μερικών LEDs διαφορετικού χρωματισμού για δήλωση καταστάσεων που αφορούν το ίδιο το σύστημα ελέγχου αλλά και εφαρμογές τις οποίες αυτό καλύπτει.



Εικόνα 2.70 Δίοδοι εκπομπής φωτός (LEDs) σε διάφορα χρώματα, σχήματα και μεγέθη

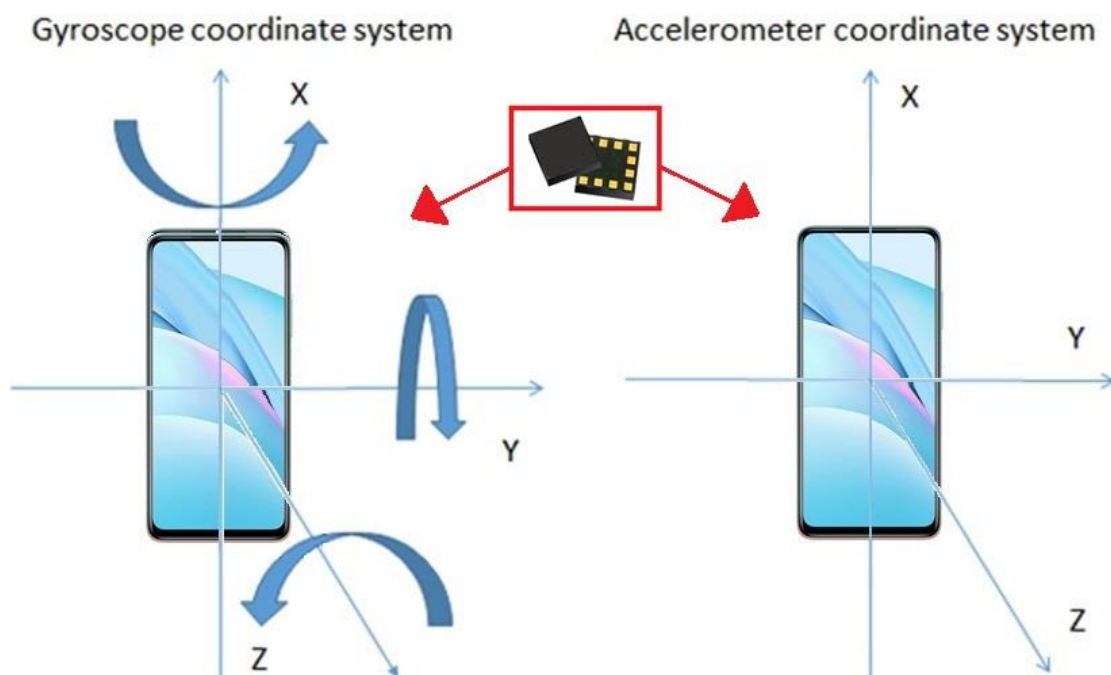
Ομοίως με τους πιεζο-βομβητές, τα LEDs διαθέτουν 2 ακροδέκτες διαφορετικού μήκους όπου αυτός με το μεγαλύτερο μήκος δέχεται την τάση για τροφοδοσία (V_{CC}) και ο ακροδέκτης με το μικρότερο μήκος αποτελεί την γείωση (GND).

2.9.4 Μονάδα Αισθητήρα Επιταχυνσιόμετρο-Γυροσκόπιο 3 Αξόνων

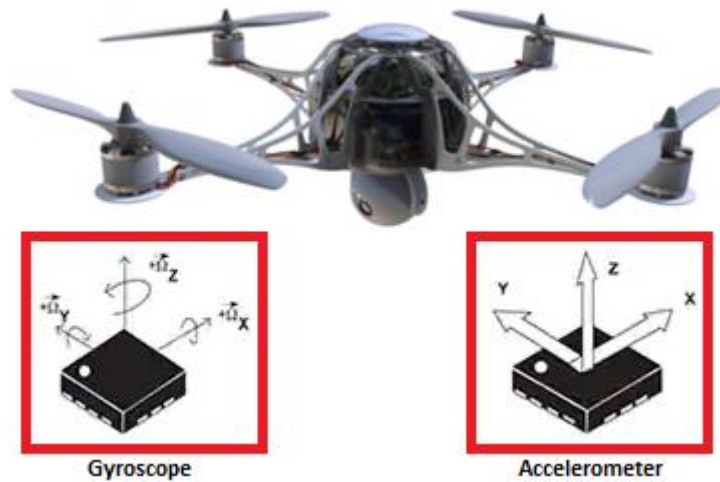
Υπάρχει πληθώρα συσκευών για τον προσδιορισμό της θέσης, του προσανατολισμού και του τρόπου μετακίνησης ενός αντικειμένου σε έναν χώρο, με ευρύτερα διαδεδομένες το γυροσκόπιο και το επιταχυνσιόμετρο, αντίστοιχα.

Ως **γυροσκόπιο** (*gyroscope*) ορίζεται η συσκευή που μετρά την ταχύτητα περιστροφής και τον προσανατολισμό ενός αντικειμένου με την πάροδο του χρόνου, κατά μήκος των 3 διαστάσεων X, Y & Z.

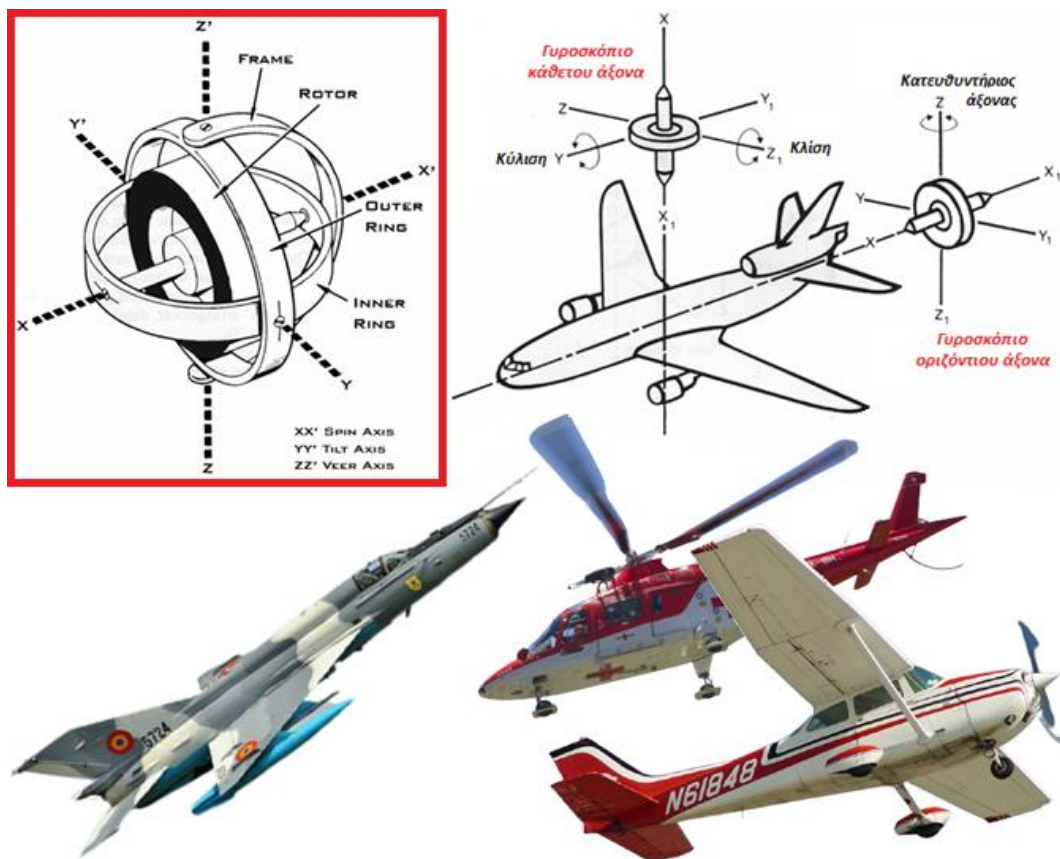
Το **επιταχυνσιόμετρο** (*accelerometer*) ή αλλιώς **μετρητής επιτάχυνσης** είναι μια συσκευή που μετρά δυνάμεις επιτάχυνσης, είτε αυτές είναι στατικές όπως η επιτάχυνση λόγω της βαρύτητας που εφαρμόζεται σε ένα αντικείμενο είτε είναι δυναμικές όπως η επιτάχυνση της κίνησης ή οι δονήσεις που βιώνει το αντικείμενο, κατά μήκος των αξόνων X, Y & Z.



Εικόνα 2.71 Τα γυροσκόπια & επιταχυνσιόμετρα δονούμενης δομής των smartphones ανιχνεύουν τον ρυθμό περιστροφής, τον προσανατολισμό και την κίνησή των με παρακολούθηση του τρόπου μετατόπισής τους στον τρισδιάστατο χώρο μέσω κατάλληλα σχεδιασμένων αισθητήρων.



Εικόνα 2.64 Τα γυροσκόπια & επιταχυνσιόμετρα στα drones χρησιμοποιούνται για τον προσδιορισμό της θέσης, του προσανατολισμού και της κίνησής τους κατά την πτήση.

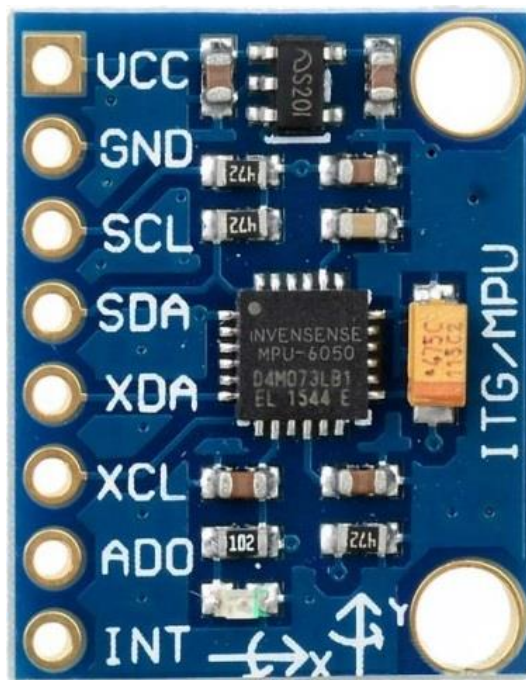


Εικόνα 2.72 Τα περιστροφικά γυροσκόπια των αεροσκαφών διατηρούνται σε ειδικούς κλωβούς με ηλεκτρικές επαφές & αισθητήρες μεταφέροντας πληροφορίες στον πιλότο & στα αδρανειακά συστήματα πλοήγησης για τον τρέχοντα προσανατολισμό του οχήματος, ενώ τα επιταχυνσιόμετρα λαμβάνουν τις δονήσεις που προκαλούνται σε αυτό και τις μετατρέπουν σε ηλεκτρικό σήμα που είναι ανάλογο της στιγμιαίας επιτάχυνσής του

Δεδομένου λοιπόν ότι το γυροσκόπιο μπορεί να ανιχνεύει, να αναλύει και να υπολογίζει την ταχύτητα περιστροφής και τον προσανατολισμό ενώ το επιταχυνσιόμετρο την επιτάχυνση, ο συνδυασμός των δύο σε μία ενιαία συσκευή μπορεί να παρέχει ακριβείς πληροφορίες για την ύπαρξη, κατάσταση και συμπεριφορά ενός αντικειμένου στον περιβάλλοντα χώρο.

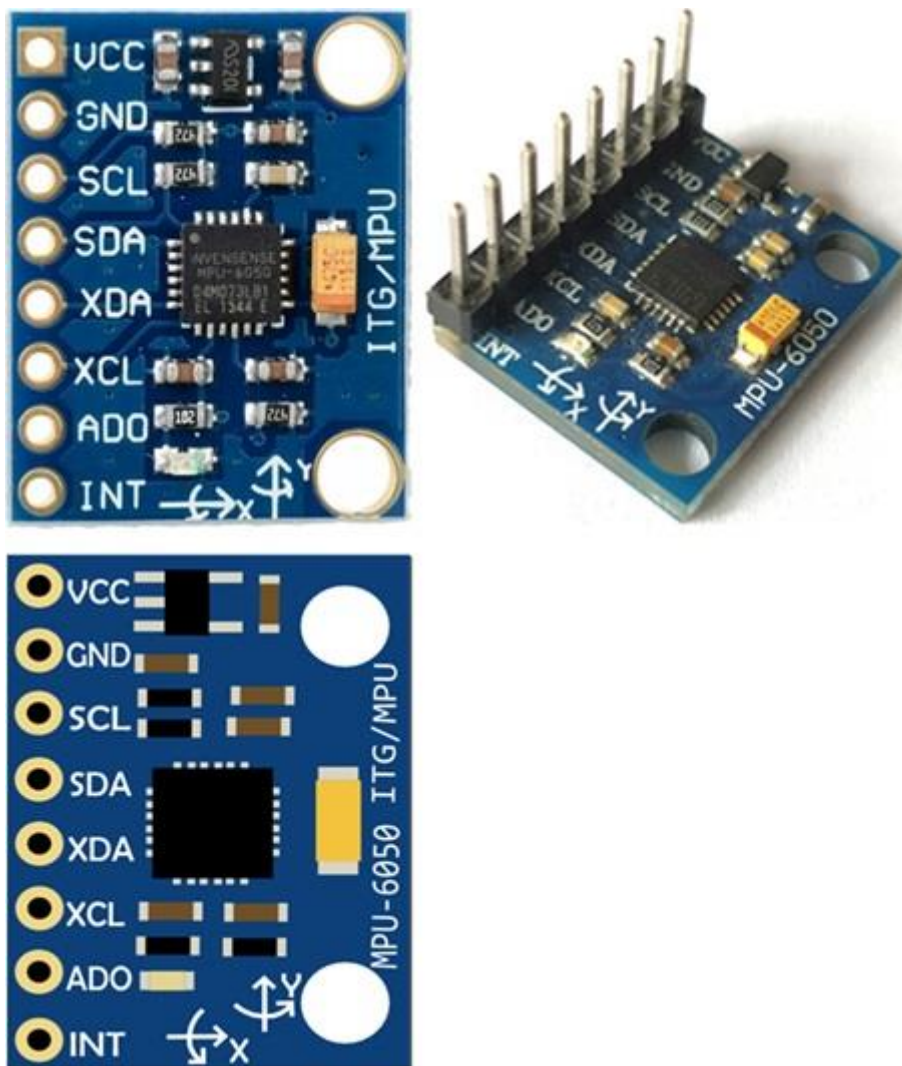
Μονάδα Αισθητήρα MPU6050

Η **μονάδα αισθητήρα MPU-6050** (*MPU-6050 sensor module*) περιέχει ένα γυροσκόπιο και ένα επιταχυνσιόμετρο στο ίδιο ολοκληρωμένο κύκλωμα. Συμπεριλαμβάνοντας έναν 16-bit μετατροπέα A/D για κάθε κανάλι, ο αισθητήρας δύναται να καταγράφει τα κανάλια X, Y & Z ταυτόχρονα, χρησιμοποιώντας το σειριακό σύστημα επικοινωνίας *I²C* (*Inter-Integrated Circuit*) για σύνδεση και επικοινωνία με μικροελεγκτές που υποστηρίζουν τέτοιον τύπο επικοινωνίας, όπως είναι τα Arduino.



Εικόνα 2.73 Επιταχυνσιόμετρο & Γυροσκόπιο 3 αξόνων μοντέλου IMU-MPU6050 (Accelerometer & Gyroscope 3 Axis IMU-MPU6050)

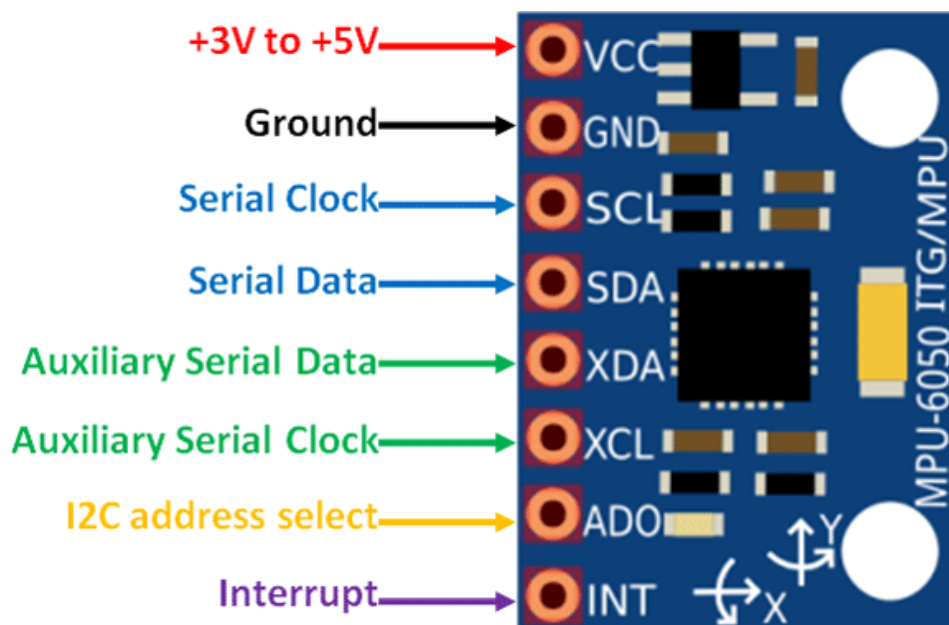
Με μέτρηση της επιτάχυνσης, της ταχύτητας, του προσανατολισμού, της μετατόπισης και πολλών άλλων παραμέτρων που σχετίζονται με την κίνηση, αλλά διαθέτοντας επιπλέον και έναν ενσωματωμένο αισθητήρα για μέτρηση της θερμοκρασίας από -40 έως 85°C με ακρίβεια $\pm 1^{\circ}\text{C}$, το ολοκληρωμένο κύκλωμα MPU6050 καθίσταται ως μια μονάδα αισθητήρα ικανή για ανάλυση και επεξεργασία της κίνησης ενός αντικειμένου ή συστήματος σε εσωτερικά και εξωτερικά περιβάλλοντα ποικίλων συνθηκών.







Εικόνα 2.74 Μονάδα αισθητήρα IMU-MPU6050 χωρίς & με κεφαλίδες ακροδεκτών (άνω) και απεικόνιση σε περιβάλλον προσομοίωσης (κάτω)

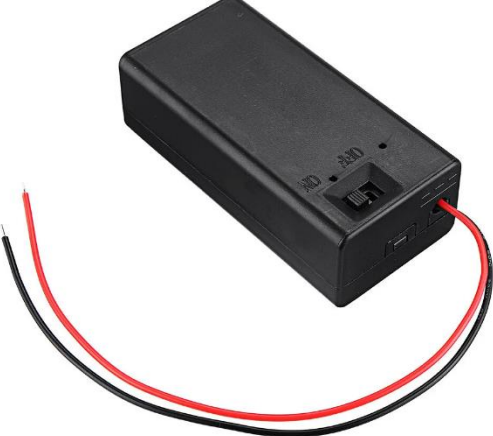


Ακροδέκτες της μονάδας MPU6050


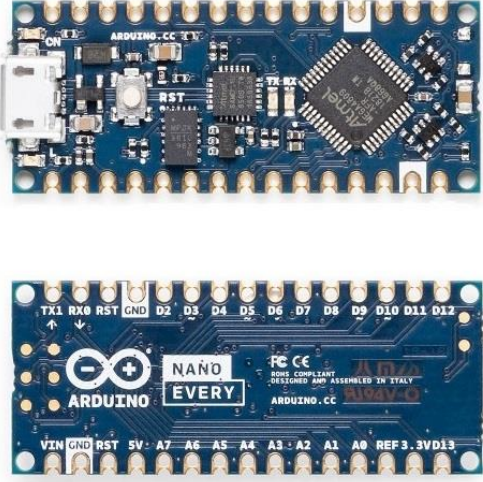

1. **VCC**: Τάση τροφοδοσίας 3-5V (είθισται η παροχή τάσης 5V)
2. **GND**: Γείωση
3. **Serial Clock (SCL)**: Χρησιμοποιείται για την παροχή παλμού ρολογιού για την επικοινωνία I2C
4. **Serial Data (SDA)**: Χρησιμοποιείται για τη μεταφορά δεδομένων με χρήση του πρωτοκόλλου επικοινωνίας I²C
5. **Auxiliary Serial Data (XDA)**; Βοηθητικός ακροδέκτης για διασύνδεση με άλλες μονάδες μέσω I²C (προαιρετικό)
6. **Auxiliary Serial Clock (XCL)**: Βοηθητικός ακροδέκτης για διασύνδεση με άλλες μονάδες μέσω I²C (προαιρετικό)
7. **AD0**: Εάν χρησιμοποιούνται περισσότερα από ένα MPU6050 σε έναν μόνο μικροελεγκτή, ο ακροδέκτης AD0 χρησιμοποιείται για διαφοροποίηση της διεύθυνσης εκάστου MPU6050
8. **Interrupt (INT)**: Ακροδέκτης διακοπής για υπόδειξη ότι τα δεδομένα είναι διαθέσιμα για ανάγνωση από τον μικροελεγκτή.









Εικόνα 2.75 MPU6050 Chip Pinout – Διάγραμμα διάταξης των ακροδεκτών του ολοκληρωμένου κυκλώματος MPU6050


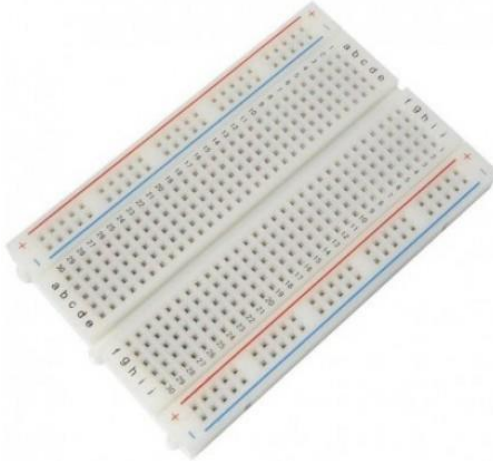

| | | | |
|------------------|--|----------------------|--|
| <p>4.</p> | <p>Πομποδέκτης NRF24L01 με ενσωματωμένη</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=NRF24L01+Transceiver+module+&btn=isch&ved=2ahUKEwJH7c7bisAhULaKOKHZE-C0AQ2:cCeqQIABAA&oeq=NRF24L01+Transceiver+module+&gs_lcp=CgNrbWcQAIDzL_VizL_WC:ImgAcAB4IAIAB1gBAIIBAJgBAKABAa0BC2d3cv13aXotaW1nwAFB&scient=img&ei=frYYImvElwkgWR_avABA&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>5.</p> | <p>Πομποδέκτης NRF24L01 με κεραία & τσιπ ενισχυτή RFX2401C</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=NRF24L01+Transceiver+module+with+SMA+Antenna+%26+RFX2401C+chip&btn=isch&ved=2ahUKEwTfGsuTAhWQ2aQKHUNGASIQ2:cCeqQIABAA&oeq=NRF24L01+Transceiver+module+with+SMA+Antenna+%26+RFX2401C+chip&gs_lcp=CgNrbWcQAICyZivZmDU22gAcAB4IAIABeQGIAYEBkaEDMC4smAEAoAFBqoEELZ3dJr_Xdiesi1nbWA0OE&scient=img&ei=Dhol_YLgKkC1kwXDIjWOAg&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>6.</p> | <p>Μονάδα Βάσης για Πομποδέκτη NRF24L01</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=nrf2401+base+module&ssrf=ALEkK03_bjnmajebUz:zesNkvsAX80:1626351813520&source=lms&btn=isch&sa=X&ved=2ahUKEwI-P7OIOXsAbYUhs0HHdAPCgoO_AUoAXoECAEOAw&bih=1254#imgrec=PalTWslMPFYhaM</p> | <p>2 τεμ.</p> |  |
| <p>7.</p> | <p>Μπαταριοθήκη μπαταρίας των 9V με διακόπτη ON/OFF και βύσμα Barrel Jack</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=9V+Battery+Holder+with+ON%2FOFF+switch+to+Barrel+Jack&btn=isch&ved=2ahUKEwZlaihqKTWAhUZkaQKHWLMBXoQ2:cCeqQIABAA&oeq=9V+Battery+Holder+with+ON%2FOFF+switch+to+Barrel+Jack&gs_lcp=CgNrbWcQAIDISiISmCOTWgAcAB4IAIAB1gBISIBAzAuMZgBAKABAa0BC2d3cv13aXotaW1nwAFB&scient=img&ei=cBaf_YNndCpmikgXpmJTQBW&bih=1253&biw=1435&hl=el</p> | <p>1 τεμ.</p> |  |


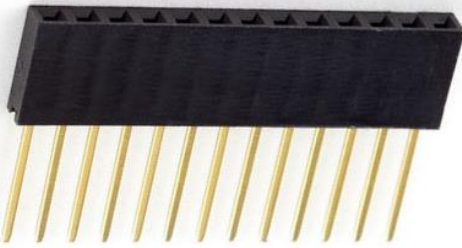

| | | | |
|------------|--|----------------------|--|
| <p>8.</p> | <p>Μπαταριοθήκη μπαταρίας των 9V με διακόπτη ON/OFF και γυμνούς ακροδέκτες</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=9V+Battery+Box+Pack+Holder+with+ON%2FOFF+switch%2C+no+plug&tbm=isch&ved=2ahUKEwiP8amE26ouAhXRIMUKHaUPCgcQ2:cCegQIABAA&oeq=9V+Battery+Box+Pack+Holder+with+ON%2FOFF+switch%2C+no+plug&gs_lcp=CgNpbWwQAIcYgAFYmIABYNGDAWAcAB4AIABcIqBcJI BAZAuMZgBAKABAAoBC2d3cyl3aXotaW1nwAFB&scient=img&ei=tiDwIYIW1L9HBIAaap6K4Bz&bih=1260&biw=1316&hl=el</p> | <p>1 τεμ.</p> |  |
| <p>9.</p> | <p>Κλιπ μπαταρίας 9V καλωδίου-συνδετήρα τύπου-T με βύσμα Barrel Jack</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=9V+T+Type+Battery+Snap+Connector+to+Barrel+Jack&tbm=isch&ved=2ahUKEwiGN5m2amAbWFL0wKHbBcHQ2-cCegQIABAA&oeq=9V+T+Type+Battery+Snap+Connector+to+Barrel+Jack&gs_lcp=CgNpbWwQAzIECCMOJLDwIQFY8JUBYZAWAcAB4AIAABZ4qBZ5IBAZAuMZgBAKABAAoBC2d3cyl3aXotaW1nwAFB&scient=img&ei=5zofYbpNIXZsAew1aWODw&bih=1260&biw=1316&hl=el#imgrec=Daf0Frzv-9Rm3M</p> | <p>1 τεμ.</p> |  |
| <p>10.</p> | <p>Κλιπ μπαταρίας 9V καλωδίου-συνδετήρα τύπου-T με γυμνούς ακροδέκτες</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=9V+T+Type+Battery+Snap+Connector+no+plug&tbm=isch&ved=2ahUKEwiXJGUpeXaAhUCgKOKHURDCPOO2-cCegQIABAA&oeq=9V+T+Type+Battery+Snap+Connector+no+plug&gs_lcp=CgNpbWwQAIcYgAFYmIABYNGDAWAcAB4AIABcIqBcJI BAZAuMZgBAKABAAoBC2d3cyl3aXotaW1nwAFB&scient=img&ei=rkhwYJipD4KAcXEHqGdW&bih=1260&biw=1316&hl=el#imgrec=HE5YPod-E-YeM</p> | <p>1 τεμ.</p> |  |


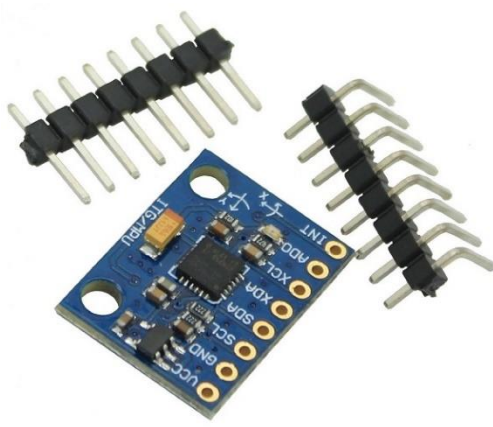
| | | | |
|-------------------|--|----------------------|--|
| <p>11.</p> | <p>Μπαταρία των 9V</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=%CE%9C%CF%80%CE%B1%CF%84%CE%B1%CF%81%CE%AF%CE%B1+%CE%91%CE%BB%CE%BA%CE%B1%CE%BB%CE%B9%CE%BA%CE%AE:9V+GP+ULTRA&btn=isch&ved=2ahUKEwi39-SVqKTwAhWLOwKH0J-CG0Q2-cCeqOIABAA&oeq=%CE%9C%CF%80%CE%B1%CF%84%CE%B1%CF%81%CE%AF%CE%B1+%CE%91%CE%BB%CE%BA%CE%B1%CE%BB%CE%B9%CE%BA%CE%AE:9V+GP+ULTRA&gs_lcp=CgNpbWcQA1CGQIBemDefGcCAB4A1ABLOGIAIjkaEDMC45mAEa0AEBAqEIZ3dJ_XdpeIjpbWAAOE&scient=img&ei=0RmLYPKL47zAeC_KHOBg&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>12.</p> | <p>Μικροελεγκτής Arduino Nano Every</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=Arduino+Nano+Every&btn=isch&ved=2ahUKEwJ0HrGp6TwAhVwOHhUppPAKoQ2-cCeqOIABAA&oeq=Arduino+Nano+Every&gs_lcp=CgNpbWcQA1C_OVI_OWc3OmgAcAB4A1ABIAIjBAJIBAJpBAKABAAoBC243ev13uXotaW1nwAEF&scient=img&ei=Kxmf.YMmOfsCi-pW56D0Ao&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>13.</p> | <p>Καλώδιο USB 2.0 A σε B (USB Cable 2.0 A to USB B micro - 1m για διασύνδεση με το Arduino Nano Every</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=USB+Cable+2.0+A+to+USB+B+micro+-+1m&btn=isch&ved=2ahUKEwIM56L-NpTWAhTmZKOKHZvVAmwO2-cCeqOIABAA&oeq=USB+Cable+2.0+A+to+USB+B+micro+-+1m&gs_lcp=CgNpbWcQA1CjcmxYooMMYPEDGAcAB4A1ABdopBdpIBAZAUMZpBAKABAAoBC2d3ev13uXotaW1nwAEF&scient=img&ei=LRLYIVdEvYk-Wb34vgBq&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |

| | | | |
|-------------------|--|----------------------|--|
| <p>14.</p> | <p>Μικροελεγκτής Arduino NANO V3 (V3.0)</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=%CE%9C%CE%B9%CE%BA%CF%81%CE%B7%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82-Arduino+NANO+V3+(V3.0)&rbm=isch&ved=2ahUKEwjs39H2aruAhVVuqOKHdsNBN4Q2-cCeqOIABAA&oeq=%CE%9C%CE%B9%CE%BA%CF%81%CE%B7%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82-Arduino+NANO+V3+(V3.0)&gs_lcp=CgNpbWcOAIcY6AdY2OgHYpPB2eAcAB4IAIBZ4eBZ5IBAZAuMZgBAKABAoBC2d3cy13aXotaW1nwAEB&scient=img&ei=NTsJYKzshXskvXhm5DwDO&bih=1260&biw=1316&hl=el#imgrec=noitR5sq9AzUM</p> | <p>2 τεμ.</p> |  |
| <p>15.</p> | <p>Καλώδιο Mini-B USB</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=%CE%9A%CE%B1%CE%BB%CF%8E%CE%B4%CE%B9%CE%BF-Mini-B+USB-&rbm=isch&ved=2ahUKEwidsSrZaruAbWL0ewKHQzmaAcUQ2-cCeqOIABAA&oeq=%CE%9A%CE%B1%CE%BB%CF%8E%CE%B4%CE%B9%CE%BF-Mini-B+USB-&gs_lcp=CgNpbWcOAIcXV1hXV2DmWmgAcAB4IAIBZogBZnIBAZAuMZgBAKABAoBC2d3cy13aXotaW1nwAEB&scient=img&ei=zsIVJ23FYvzaAel_zloDA&bih=1260&biw=1316&hl=el#imgrec=D40aMErejYBRDM</p> | <p>2 τεμ.</p> |  |
| <p>16.</p> | <p>Μπαταριοθήκη 4 x 1.5V μπαταριών AAA με γυμνούς ακροδέκτες</p> <p><u>Ενδεικτικό Google link:</u></p> <p>https://www.google.com/search?q=%CE%9C%CE%A0%CE%91%CE%A4%CE%91%CE%A1%CE%99%CE%9F%CE%98%CE%97%CE%9A%CE%97-4xAAA-Colour-black-150mm&rbm=isch&ved=2ahUKEwiv0uGkoeXsAhVUuqOKHZ5nC7cQ2-cCeqOIABAA&oeq=%CE%9C%CE%A0%CE%91%CE%A4%CE%91%CE%A1%CE%99%CE%9F%CE%98%CE%97%CE%9A%CE%97-4xAAA-Colour-black-150mm&gs_lcp=CgNpbWcOAZIECCMOJ1D70RY6-0EYIvBGgAcAB4IAIBcoBcpIBAZAuMZgBAKABAoBC2d3cy13aXotaW1nwAEB&scient=img&ei=akLwYlKKJNTskgWvez624Cw&bih=1260&biw=1316&hl=el#imgrec=Sa6iB85BjmOwM</p> | <p>2 τεμ.</p> |  |

| | | | |
|------------|---|---------------------|--|
| <p>20.</p> | <p>Σερβοκινητήρας Ψηφιακός MG995 συνεχούς περιστροφής 360 μοιρών με αξεσουάρ μοχλών</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Digital+Metal+Gear+Servo+Motor+MG995+360+Degree+&tbm=isch&ved=2ahUKEwi3nuX1pQTwAbUFP6QKHv7BWQ2-cCegQIABAA&oeq=Digital+Metal+Gear+Servo+Motor+MG995+360+Degree+&gs_lcp=CgNpbWwQAI1Cec1TMDGDMWwAcAB4A1ABAlcBAJBAJeBAAABAAoBC2d3cy13aXotaW1nwAEF&scient=img&ei=gRiLYPVVC4XukgX795cOBo&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>21.</p> | <p>Αρσενικά καλώδια Δοκιμών – Καλώδια Βραχυκύκλωσης <i>(Male to Male Jump wires)</i></p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=40pcs+Male+to+Male+Dupont+Wires+Cables+for+Arduino+(20cm)+(M/M)&hl=el&ssrf=AleKk02IVswk1YsAZaXWNIDbZ28EGZMR3w1619728383299&source=lnms&tbm=isch&sa=X&ved=2ahUKEw11Kv3psTgAbU1hv0HHW3iDEEO_A1toAXoECAEOAw&biw=1435&bih=1253</p> | <p>~100 καλώδια</p> |  |
| <p>22.</p> | <p>Θηλυκά καλώδια Δοκιμών – Καλώδια Βραχυκύκλωσης <i>(Female to Female Jump wires)</i></p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=40pcs+Female+to+Female+Dupont+Wires%2FCables+for+Arduino+(20cm)+(F%2FF)&tbm=isch&ved=2ahUKEw0pNCJpTgAbWGI60KHVhKCb002-cCegQIABAA&oeq=40pcs+Female+to+Female+Dupont+Wires%2FCables+for+Arduino+(20cm)+(F%2FF)&gs_lcp=CgNpbWwQAI1DhngYx64CYLSsAmgAcAB4A1A1B1fBtHBAzAuMZgBAAKABAAoBC2d3cy13aXotaW1nwAEF&scient=img&ei=nlYJdZG1avkwXWIK6Cw&bih=1253&biw=1435</p> | <p>~100 καλώδια</p> |  |

| | | | |
|------------|--|---|--|
| <p>23.</p> | <p>Καλώδιο επέκτασης Σερβοκινητήρα (Servo Extension Cable)</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Servo+Extension+Cable+&tbm=isch&ved=2ahUKEwiD86CHoiTwAhW0ggQKHUjQCccO2-cCgQIABAA&oeq=Servo+Extension+Cable+&gs_lcp=CgNpbWcOQ1DHGjG2C7HGzAcAB4IAIABAlB4IIBAJB4KABAAoBC2d3cv13oXoWlwwAEb&scilent=imz&ei=mhel.Y1O-GofFkaXloKadDg&bih=1253&biw=1435</p> | <p>2 τεμάχια των 30cm έκαστο</p> |  |
| <p>24.</p> | <p>Πλακέτα δοκιμών (Solderless Breadboard 400-Contacts)</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Mini+Solderless+Breadboard+400-Contacts+for+Arduino&csurf=ALeKk033ew1vbMml4DoOpSL6PaDzi76MjC.1619728212446&source=Inms&tbm=isch&sa=X&ved=2ahUKEwiTzvDlpaTwAhU9ePOHhBoQAK0Q_AUoAXoECAEQAw&biw=1435&bih=1253</p> | <p>2 τεμ.</p> |  |
| <p>25.</p> | <p>Πλακέτα δοκιμών χωρίς κολλήσεις (Breadboard)</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=BREADBOARD+47.3X83.7&csurf=ALeKk006uJ7evhEMUMoVeSHzQBwm7xmw.1626359129181&source=Inms&tbm=isch&sa=X&ved=2ahUKEwjzK_xo-XxAbX_hBHHWRGAoEO_AUoAuoECAEQBA&biw=1303&bih=1254&imrc=0faFp8-y-vxgmM</p> | <p>1 τεμ.</p> |  |

| | | | |
|------------|---|---------------|--|
| <p>26.</p> | <p>Ακιδοσειρές: Αρσενικές Κεφαλίδες Ακροδεκτών <i>(Male Pin Headers)</i></p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=%CE%91%CE%9A%CE%99%CE%94%CE%9F%CE%A3%CE%95%CE%99%CE%A1%CE%91+2X40-PINS+%CE%B1%CE%81%CE%83%CE%B5%CE%BD%CE%B9%CE%BA%CE%AC+%CE%B5%CE%85%CE%B8%CE%B5%CE%A9%CE%B1+2.54mm&btn=isch&ved=2ahUKEwimbH9oTwaAVCsoQKHZmhD40Q2-cCegQIABAA&oeq=%CE%91%CE%9A%CE%99%CE%94%CE%9F%CE%A3%CE%95%CE%99%CE%A1%CE%91+2X40-PINS+%CE%B1%CE%81%CE%83%CE%B5%CE%BD%CE%B9%CE%BA%CE%AC+%CE%B5%CE%85%CE%B8%CE%B5%CE%A9%CE%B1+2.54mm&gs_lcp=CgNpbWcQA1C30AVY9AFYOPSBWgAcAB4A1ABdYgBdZIBAzAuMzZgBAKABAAoBC2d3cy13aXotaW1nwAEB&schent=img&ei=bWLYPCzNMKMlwWZw7boDQ&bih=1253&biw=1435&hl=el</p> | <p>2 τεμ.</p> |  |
| <p>27.</p> | <p>Ακιδοσειρές: Θηλυκές Κεφαλίδες Ακροδεκτών <i>(Female Pin Headers)</i></p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Female+Pin+Headers&btn=isch&ved=2ahUKEwI3urDDo-XsAhVdeV0HHXvDA3Q2-cCegQIABAA&oeq=Female+Pin+Headers&gs_lcp=CgNpbWcQAzIECAAOE1DUvYFY1L4BYRHGAwAcAB4A1ABdYgBdZIBAzAuMzZgBAKABAAoBC2d3cy13aXotaW1nwAEB&schent=img&ei=FTWY1L0K92F9u8P-4aPvAY&bih=1253&biw=1435&hl=el#imgres=buJKBq8vY161M</p> | <p>2 τεμ.</p> |  |
| <p>28.</p> | <p>Αισθητήρας Υπερήχων HC-SR04 <i>(Ultrasonic Sensor HC-SR04 module)</i></p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=ElecFreaks+HC-SR04+Ultrasonic+Module+-Distance+Measuring+Transducer+sensor+For+Arduino&btn=isch&ved=2ahUKEwJc-atTMo6TwaAhWJ-aOKHfpcAokO2-cCegQIABAA&oeq=ElecFreaks+HC-SR04+Ultrasonic+Module+-Distance+Measuring+Transducer+sensor+For+Arduino&gs_lcp=CgNpbWcQA1C6_wYUv8CYM2BA2zAcAB4A1ABdYgBdZIBAzAuMzZgBAKABAAoBC2d3cy13aXotaW1nwAEB&schent=img&ei=BIWLYMLJOYnzkwX6uYnCA&bih=1253&biw=1435&hl=el</p> | <p>1 τεμ.</p> |  |

| | | | |
|------------|---|---------------|---|
| <p>29.</p> | <p>Πιεζοηλεκτρικός Βομβητής / Buzzer (<i>Piezoelectric Speaker / Buzzer</i>)</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Passive+Electronic+Piezoelectric+Piezo+Buzzer&btn=isch&ved=2ahUKFwi1Kvoo6TwAHXGgaOKH0sC_0Q2-cCeqQIABAA&coq=Passive+Electronic+Piezoelectric+Piezo+Buzzer&gs_lcp=CgNpbWwOAZlFCCMOJzOHCCMO6eIQ11DaugRYkcEYmVtBG6BcAB4IA1ABwGIAy_cKqEDMC4ymAEAoAEBqgELZ3dzLXdpej1pbWewAOzAAQe&sclem=img&ei=uhSLYLX1H8aDkgWj2a7oDw&bih=1253&biw=1435&hl=el</p> | <p>1 τεμ.</p> |  |
| <p>30.</p> | <p>Γυροσκόπιο και Επιταχυνσιόμετρο 3 Αξόνων IMU - MPU6050 (<i>Triple Axis Gyroscope & Accelerometer IMU - MPU6050</i>)</p> <p><u>Ενδεικτικό Google link:</u> https://www.google.com/search?q=Triple+Axis+Gyroscope+%26+Accelerometer+IMU+-MPU6050&btn=isch&ved=2ahUKFwi2zNDvoqTwAbUCG-wKHWWHC2MO2-cCeqQIABAA&coq=Triple+Axis+Gyroscope+%26+Accelerometer+IMU+-MPU6050&gs_lcp=CgNpbWwOAZlFCCMOJzOHCCMO6eIQ11DaugRYkcEYmVtBG6BcAB4IA1ABwGIAy_cKqEDMC4ymAEAoAEBqgELZ3dzLXdpej1pbWewAOzAAQe&sclem=img&ei=QsSLYPae4K2sAfIjoYB&bih=1253&biw=1435&hl=el</p> | <p>1 τεμ.</p> |  |

2.11 Συμπεράσματα

Με κατανόηση των χαρακτηριστικών, του σχεδιασμού και του τρόπου λειτουργίας της εκάστοτε ηλεκτρονικής μονάδας και εξαρτήματος που συμπεριλήφθηκε στο παρόν κεφάλαιο και κατά συνέπεια την ολοκλήρωση του συγκεντρωτικού πίνακα αυτών, καθίσταται πλέον δυνατός ο σχεδιασμός και η μετέπειτα υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή βάσει του πίνακα αυτού.

Κεφάλαιο 3:

Σχεδιασμός – Προγραμματισμός – Υλοποίηση

Το παρόν κεφάλαιο πραγματεύεται τον σχεδιασμό, τον προγραμματισμό και την υλοποίηση ενός **Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.)** προοριζόμενο για Βιομηχανικές Εφαρμογές. Για να καταστεί ευκολότερη η υλοποίηση αυτή, αρχικά θα δημιουργηθούν δύο υποσυστήματα: ένα **Υποσύστημα Πομπού (Transmitter Subsystem)** το οποίο εφεξής θα αναφέρεται ως **Κύκλωμα TS** και ένα **Υποσύστημα Δέκτη (Receiver Subsystem)** το οποίο θα αναφέρεται ως **Κύκλωμα RS**, για λόγους συντομίας.

Ξεκινώντας με το κύκλωμα TS, αρχικά θα πραγματοποιηθεί συνδεσμολογία μεταξύ αναλογικού μοχλού και μικροελεγκτή Arduino NANO EVERY για επαλήθευση της ορθής ανάγνωσης και λειτουργίας του μοχλού ως περιφερειακή συσκευή εισόδου για την ηλεκτρονική πλακέτα Arduino NANO και έπειτα συνδεσμολογία μεταξύ μικροελεγκτή NANO και δύο ψηφιακών σερβοκινητήρων για επαλήθευσή των ως περιφερειακές συσκευές εξόδου του NANO.

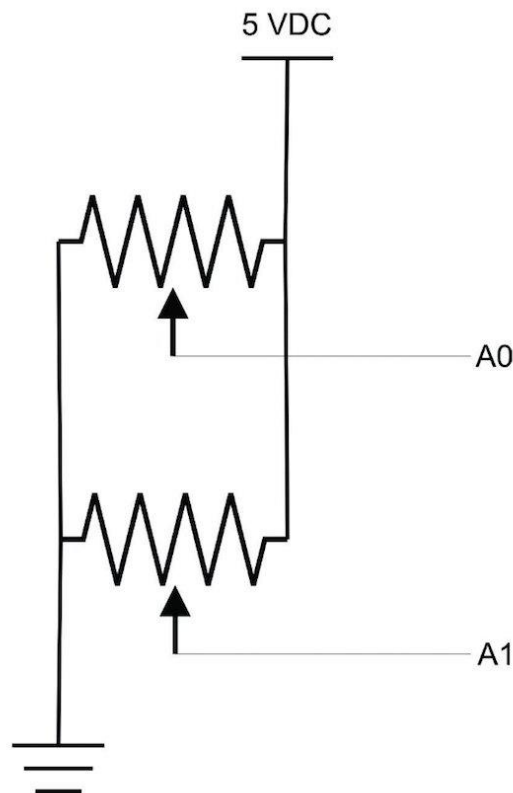
Στην συνέχεια, θα πραγματοποιηθεί συνδεσμολογία όλων των ανωτέρω ηλεκτρονικών και ηλεκτρικών εξαρτημάτων για την δημιουργία ενός ενιαίου ενσύρματου ηλεκτρονικού συστήματος ελέγχου, με σκοπό την ορθή και εύρυθμη λειτουργία του ως σύστημα απόκρισης μίας εισόδου με δύο εξόδους – θα έχει ουσιαστικά σχεδιαστεί και υλοποιηθεί ένα *Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή (Σ.Η.Ε.Ε.) για Βιομηχανικές Εφαρμογές*. Με περαιτέρω προσθήκη ενός επιπλέον αναλογικού μοχλού και δύο ακόμη σερβοκινητήρων, γίνεται έλεγχος της συμπεριφοράς του ενσύρματου ελεγκτή όταν σε αυτόν προστεθούν επιπλέον περιφερειακές μονάδες. Στο σημείο αυτό, το μόνο που απομένει είναι η ένταξη της μονάδας πομποδέκτη NRF24L01 στο κύκλωμα TS για προετοιμασία μετατροπής του από ενσύρματο σε ασύρματο.

Το κύκλωμα RS σχεδιάζεται και υλοποιείται κατ' ανάλογο τρόπο, ώστε καταληκτικά να επιτευχθεί ένα *Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) για Βιομηχανικές Εφαρμογές* που είναι και το ζητούμενο της Διπλωματικής εργασίας.

3.1 Υποσύστημα Πομπού (Κύκλωμα TS)

3.1.1 Σύνδεση Χειριστηρίου & Arduino NANO EVERY

Όπως έγινε αναφορά στην υποενότητα 2.2.1 περί των ποτενσιομέτρων των αναλογικών μοχλών, με κατάλληλη σύνδεση των ακροδεκτών του μοχλού σε ακροδέκτες του μικροελεγκτή NANO, με ολίσθηση του μοχλού από τον χρήστη και την αντίστοιχη αυξομείωση των ποτενσιομέτρων δημιουργείται και ένας διαφορετικός διαιρέτης τάσης κάθε φορά, με αποτέλεσμα την ανάγνωση της εκάστοτε παραγόμενης τάσης από το σύστημα του μοχλού και οδήγησή της στην είσοδο του μικροελεγκτή.



Εικόνα 3.1 Σχέση αντιστοιχίας μεταξύ των ακροδεκτών A0 & A1 ενός Arduino και των δύο ποτενσιομέτρων ενός αναλογικού μοχλού 2 αξόνων, όταν ο αναλογικός μοχλός ολισθαίνει στους άξονες X & Y αντίστοιχα

Η συνδεσμολογία μεταξύ μοχλού και μικροελεγκτή NANO είναι αρκετά εύκολη, απλή και πραγματοποιείται ως εξής:

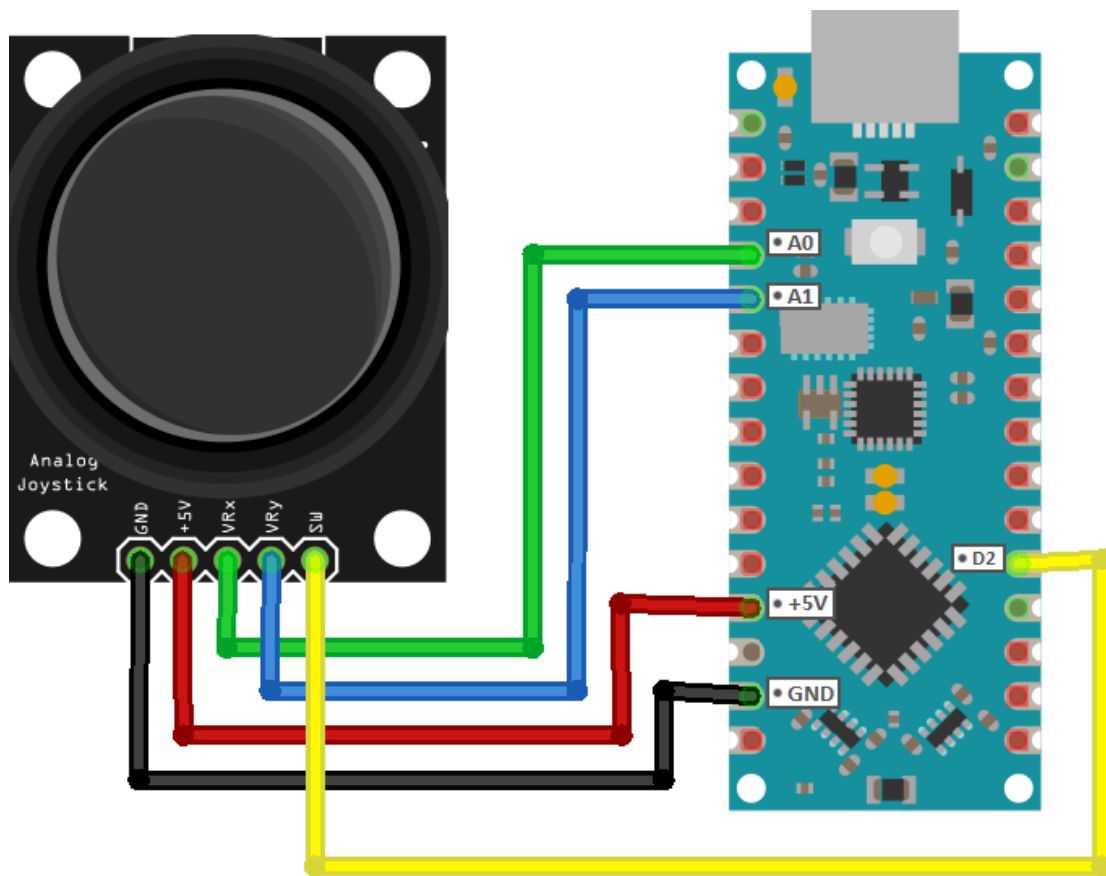
ΠΙΝΑΚΑΣ 3_1

| Ακροδέκτες μικροελεγκτή (NANO) | Ακροδέκτες αναλογικού μοχλού |
|--------------------------------|--|
| 5V (analog pin 5V) | +5V |
| GND (analog pin GND) | GND |
| A0 (analog pin 0) | VR _X (ηλεκτρικό σήμα X άξονα) |
| A1 (analog pin 1) | VR _Y (ηλεκτρικό σήμα Y άξονα) |
| D2 (digital pin 2) | SW (Διακόπτης μπουτόν) |

Όπως προαναφέρθηκε στην υποενότητα 2.2.1 οι ακραίες τιμές των δύο ποτενσιομέτρων ενός αναλογικού μοχλού είναι η ελάχιστη τιμή “0” και η μέγιστη τιμή “ R_{max} ” που μπορεί να λάβει βάσει κατασκευής του (π.χ. αν $R_{max}=10k\Omega$). Όπως όμως είναι φυσικό, δεδομένου ότι η θεωρία πάντα διαφέρει από την πράξη, οι αναμενόμενες διαφορές δυναμικού (τάσεις) στα άκρα των ποτενσιομέτρων κυμαίνονται από πολύ κοντά στην τιμή 0V έως πολύ κοντά στην μέγιστη τιμή V_{max} βάσει κατασκευής και απωλειών επί των μεταβλητών αντιστάσεων. Για λόγους οπτικής ευκολίας στην διάκριση των καλωδίων κατά τον σχεδιασμό, καλώδια διαφορετικού χρώματος χρησιμοποιούνται για έκαστην σύνδεση ακροδεκτών μεταξύ μοχλού και Arduino NANO, ως φαίνεται στον **πίνακα 3_2** και στην **εικόνα 3.2** που ακολουθούν.

ΠΙΝΑΚΑΣ 3_2

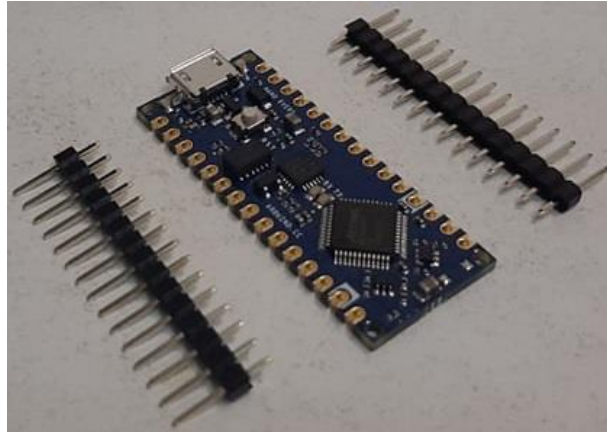
| Ακροδέκτες μικροελεγκτή (NANO) | Ακροδέκτες αναλογικού μοχλού | Χρώμα καλωδίων σύνδεσης |
|--------------------------------|---|-------------------------|
| 5V (analog pin 5V) | +5V | Κόκκινο |
| GND (analog pin GND) | GND | Μαύρο |
| A0 (analog pin 0) | VR _X (ηλεκτρικό σήμα X άξονα) | Πράσινο |
| A1 (analog pin 1) | VR _Y (ηλεκτρικό σήμα Y άξονα) | Μπλε |
| D2 (digital pin 2) | SW (Διακόπτης μπουτόν) | Κίτρινο |



Εικόνα 3.2 Συνδεσμολογία αναλογικού μοχλού 2 αξόνων και μικροελεγκτή Arduino NANO EVERY, σε περιβάλλον προσομοίωσης

Η ηλεκτρονική πλακέτα Arduino NANO EVERY δεν διαθέτει ενσωματωμένες κεφαλίδες ακροδεκτών (*pin headers*) από κατασκευής της, παρά μόνο οπές υποδοχής.

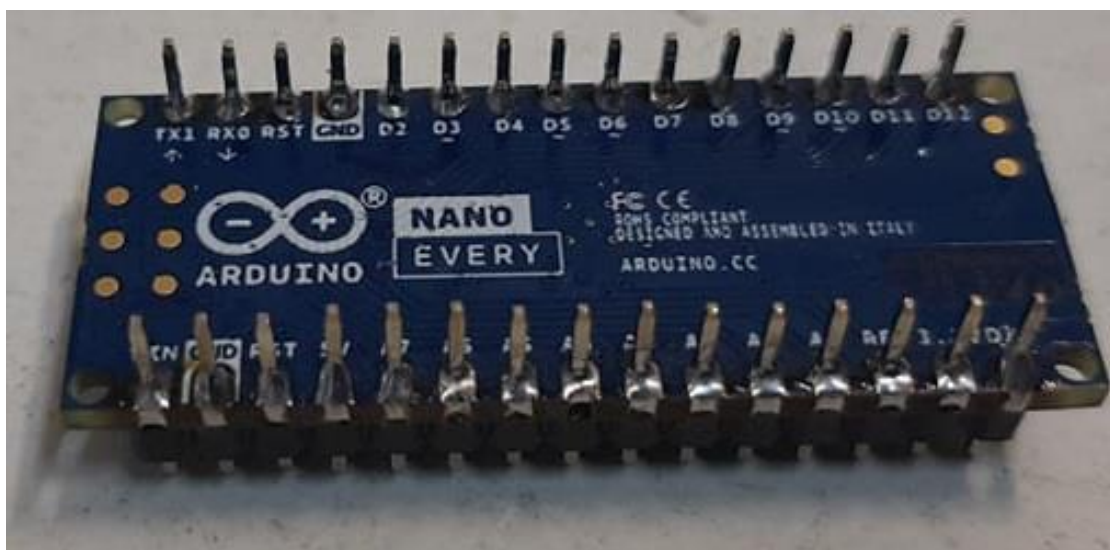
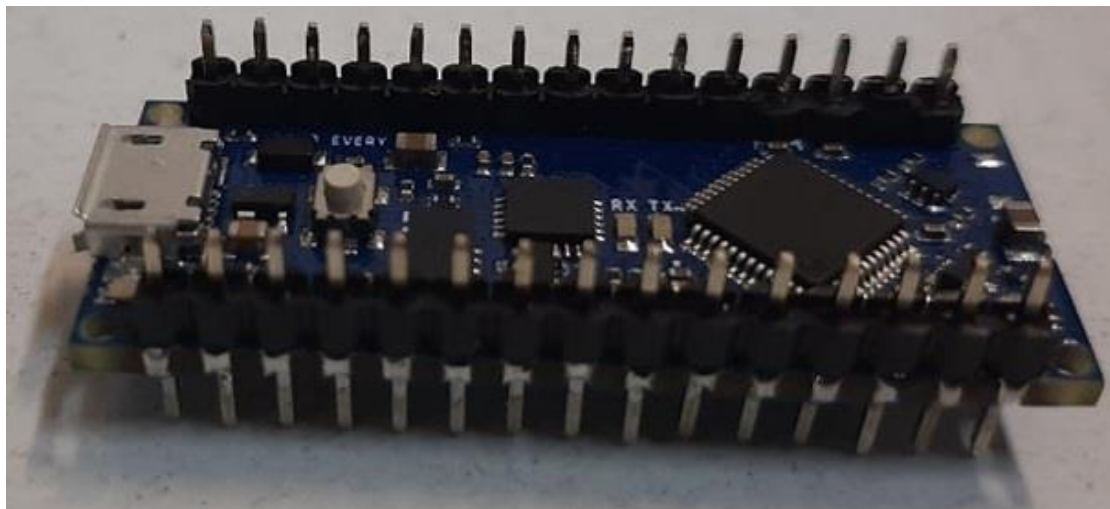
Για τον λόγο αυτό συνοδεύεται από 2 ακιδοσειρές αρσενικών κεφαλίδων (*male pinheaders*) που επιδέχονται μεθόδους συγκόλλησης επί των οπών ώστε η πλακέτα Arduino NANO EVERY να καθίσταται συμβατή με τον τύπο ακροδεκτών που φέρουν οι περισσότερες περιφερειακές μονάδες και ηλεκτρονικές συσκευές που κυκλοφορούν στην αγορά.



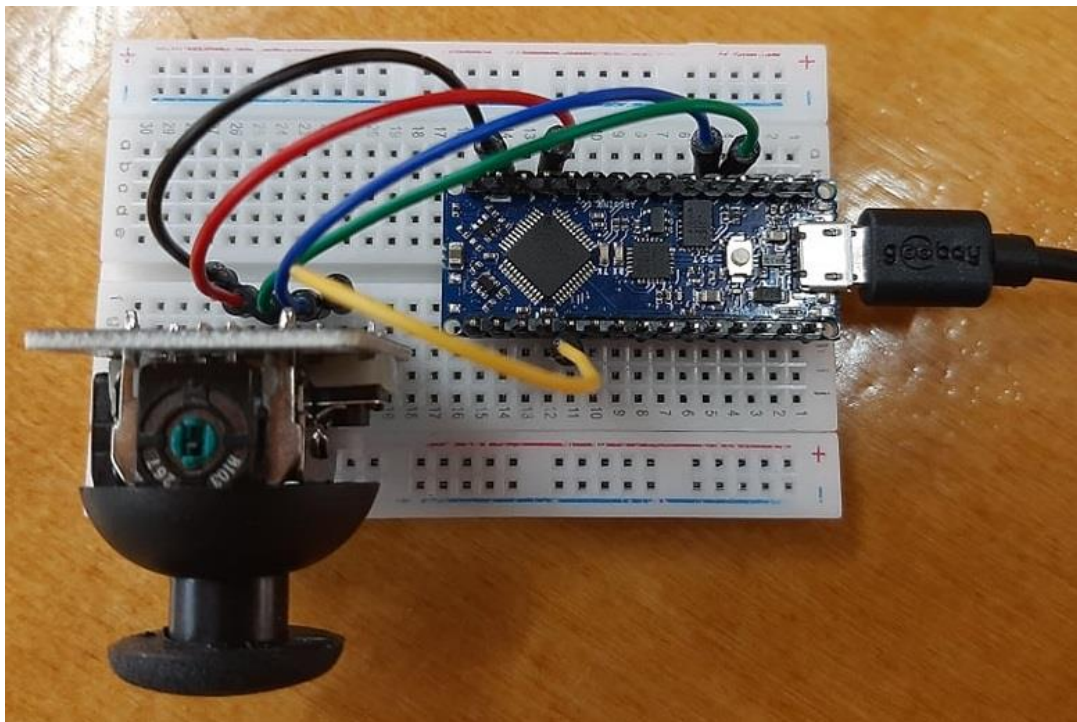
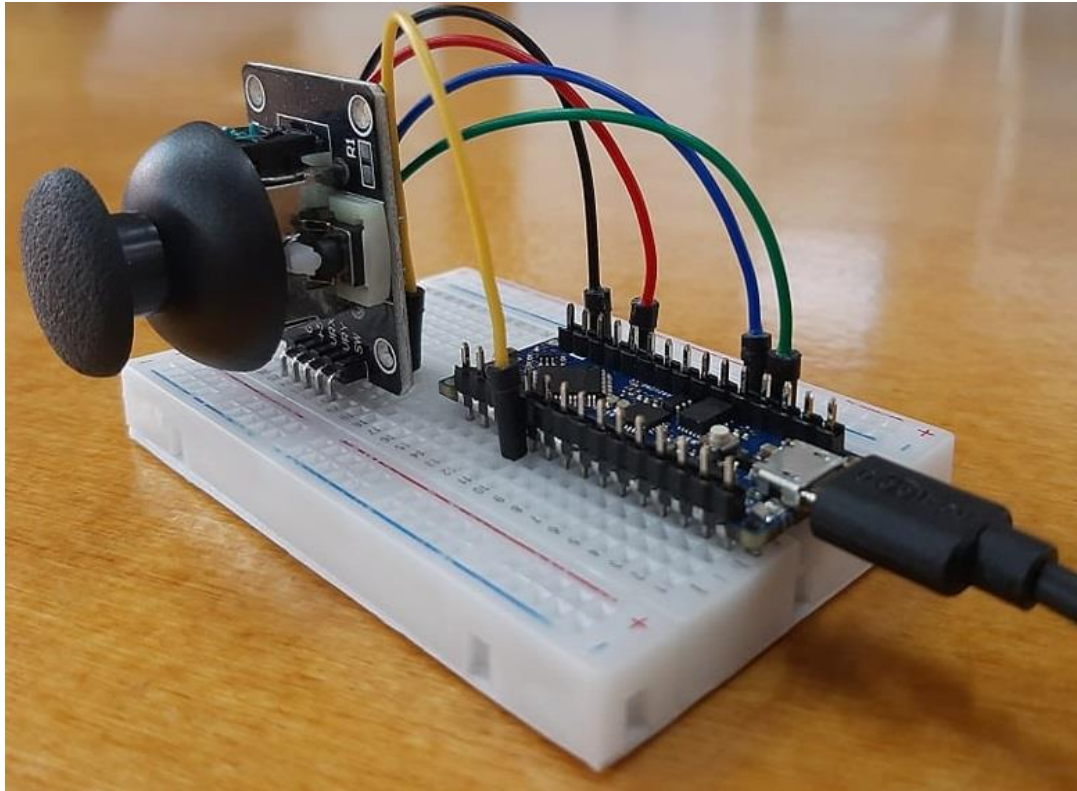
Εικόνα 3.3 Ηλεκτρονική πλακέτα Arduino NANO EVERY και 2 ακιδοσειρές τύπου αρσενικών κεφαλίδων (*male pinheaders*)



Εικόνα 3.4 Διαδικασία συγκόλλησης ακιδοσειρών επί των οπών υποδοχής της ηλεκτρονικής πλακέτας Arduino NANO EVERY σε πάγκο εργασίας



Εικόνα 3.5 Αποτέλεσμα συγκόλλησης ακιδοσειρών επί των οπών υποδοχής της ηλεκτρονικής πλακέτας Arduino NANO EVERY

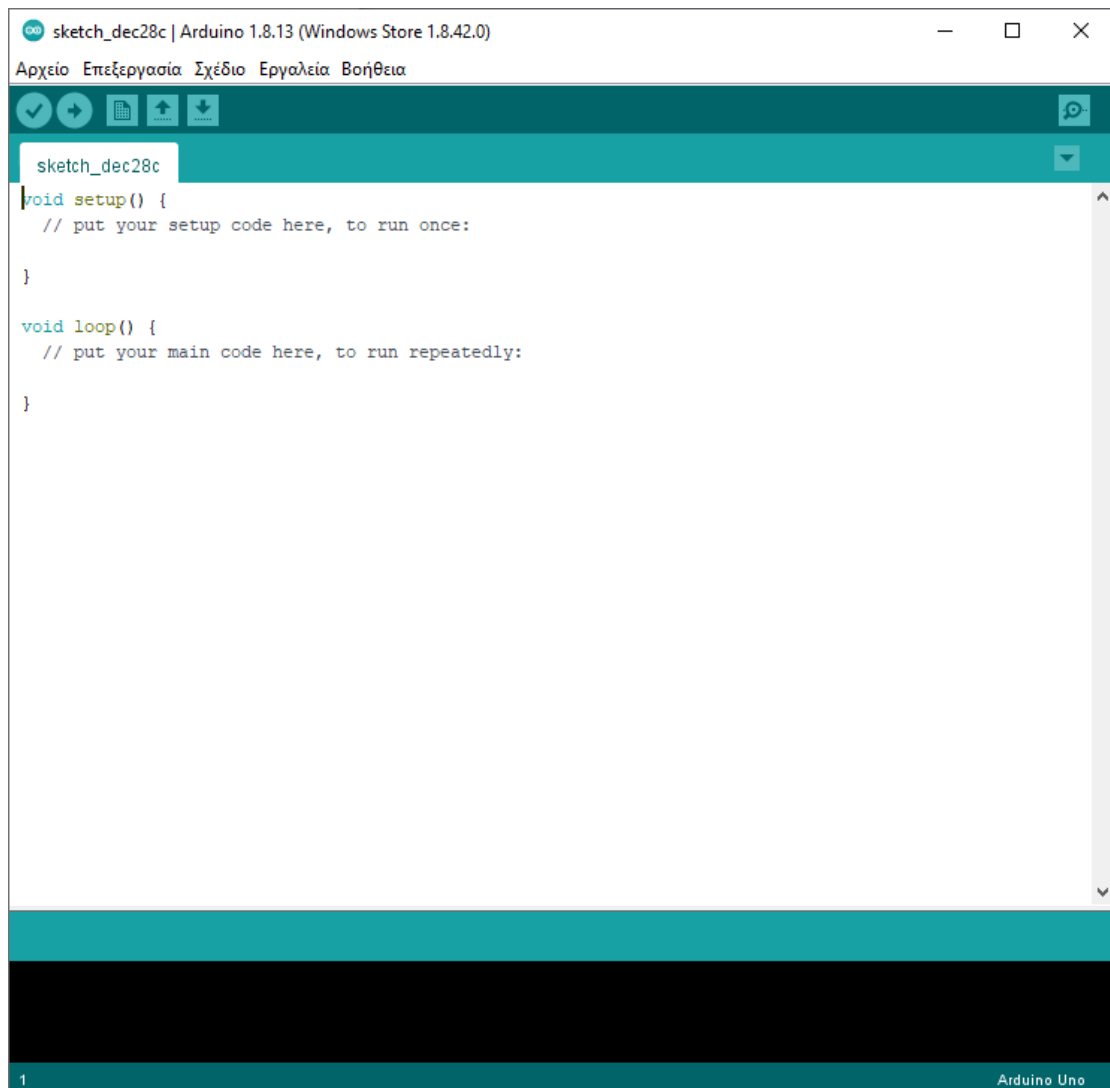


Εικόνα 3.6 Συνδεσμολογία αναλογικού μοχλού 2 αξόνων και μικροελεγκτή Arduino NANO EVERY, υλοποιημένη σε επαγγελματικό εργαστήριο

3.1.1.1 Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας

Ρύθμιση παραμέτρων του περιβάλλοντος *Arduino IDE*

Έχοντας πραγματοποιήσει την ανωτέρω συνδεσμολογία, γίνεται μετάβαση στο ολοκληρωμένο περιβάλλον ανάπτυξης (*Integrated Development Environment – IDE*) του **Arduino**, γνωστό με το ακρώνυμο “*Arduino IDE*”, για παρατήρηση και μελέτη της συμπεριφοράς του αναλογικού μοχλού με χρήση του μικροελεγκτή Arduino NANO EVERY:



```
sketch_dec28c | Arduino 1.8.13 (Windows Store 1.8.42.0)
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
sketch_dec28c
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

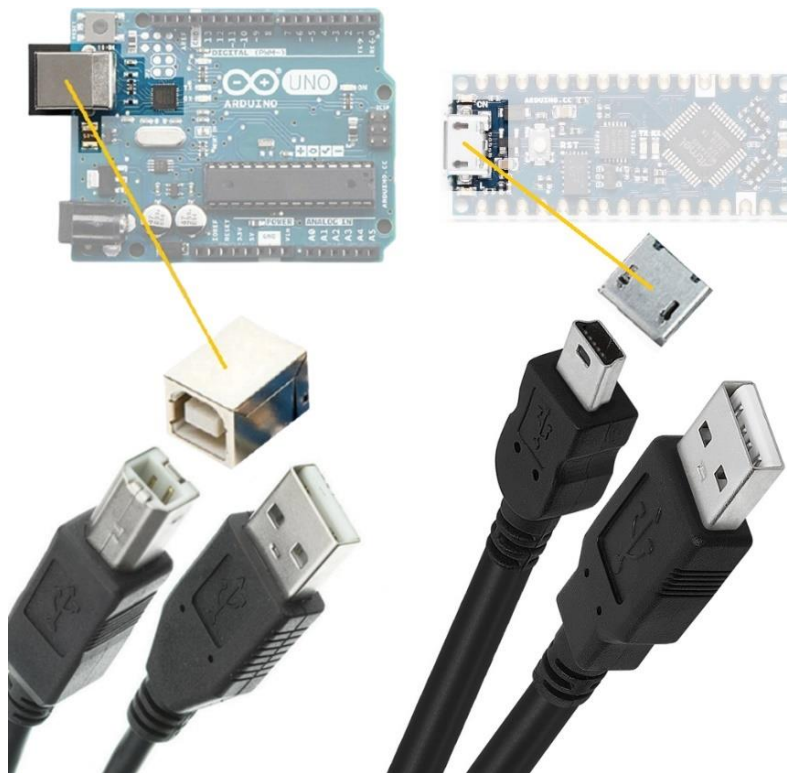
1 Arduino Uno
```

Εικόνα 3.7 Αρχική κατάσταση παραθύρου και κώδικα του ολοκληρωμένου περιβάλλοντος ανάπτυξης “Arduino IDE”

Το *Arduino IDE* είναι μια εφαρμογή για πολλαπλές πλατφόρμες (*Windows, macOS, Linux*) που αναπτύχθηκε από την εταιρεία Arduino, συντάσσεται με συναρτήσεις των γλωσσών προγραμματισμού C και C++ και χρησιμοποιείται για την μεταγλώττιση και την εγγραφή προγραμμάτων σε ηλεκτρονικές πλακέτες της οικογένειας Arduino, αλλά και σε άλλες πλακέτες ανάπτυξης λογισμικού μη συμβατές με Arduino όταν γίνεται συμπερίληψη πρόσθετων μεταγλωττιστών και εργαλείων προγραμματισμού άλλων εταιρειών.

Με σύνδεση του ανωτέρω υλοποιημένου κυκλώματος «*Μοχλός/Arduino NANO*» με έναν Η/Υ μέσω της θύρας USB του Arduino NANO, κατάλληλη ρύθμιση του περιβάλλοντος IDE για ορθή επικοινωνία του κυκλώματος με τον Η/Υ, προγραμματισμό του μικροελεγκτή και εκτέλεση του προγράμματος, όταν ο χρήστης ολισθαίνει τον αναλογικό μοχλό στον X και Y άξονα αντίστοιχα, οι αλλαγές στην τάση των δύο ποτενσιομέτρων εκφράζονται με την βοήθεια του περιβάλλοντος *Arduino IDE* με συγκεκριμένα μηνύματα που εμφανίζονται στην οθόνη του Η/Υ.

Για επικοινωνία μίας ηλεκτρονικής πλακέτας Arduino με έναν Η/Υ, γίνεται αρχικά χρήση ενός καλωδίου τύπου USB για σύνδεση του Arduino με τον υπολογιστή.

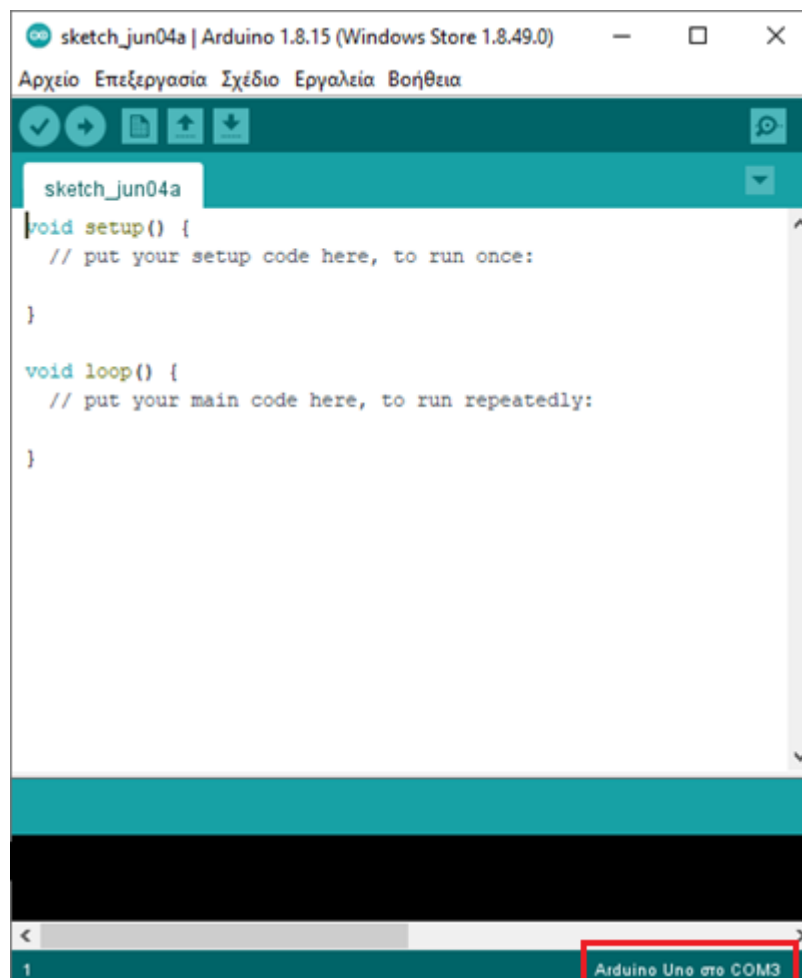


Εικόνα 3.8 Θύρες τύπου USB για διασύνδεση μοντέλων Arduino με Η/Υ

Με απόκτηση από το Διαδίκτυο και εγκατάσταση σε ηλεκτρονικό υπολογιστή του αρχείου εφαρμογής *arduino.exe* (έκδοση 1.8.13 και μετέπειτα ενημέρωση σε 1.8.15), μεταφερόμαστε στο εν λόγω περιβάλλον ανάπτυξης ανοίγοντας το σχετικό αρχείο συντόμευσης εφαρμογής.

Στο περιβάλλον αυτό, όπως γίνεται αντιληπτό και από την εικόνα 3.9, υπάρχουν δύο γραμμές κώδικα – δύο βασικές εντολές, ουσιαστικά – με μία γραμμή από σχόλια να ακολουθεί την κάθε εντολή.

Αυτή είναι η αρχική μορφή του ολοκληρωμένου περιβάλλοντος ανάπτυξης. Η πρώτη ενέργεια εδώ είναι να γνωστοποιήσουμε στο περιβάλλον IDE το μοντέλο Arduino που επιθυμούμε να χρησιμοποιήσουμε. Παρατηρούμε στο κάτω δεξιό μέρος του παραθύρου την ένδειξη “*Arduino Uno στο COM3*». Αυτό σημαίνει ότι ως αρχική ρύθμιση έχει ορισθεί η θύρα COM3 για σύνδεση του Η/Υ με το μοντέλο Arduino UNO.



Εικόνα 3.9 Ολοκληρωμένο περιβάλλον ανάπτυξης Arduino IDE

Με μετάβαση στην καρτέλα «*Εργαλεία*» της γραμμής εργαλείων του *IDE* ως εξής:

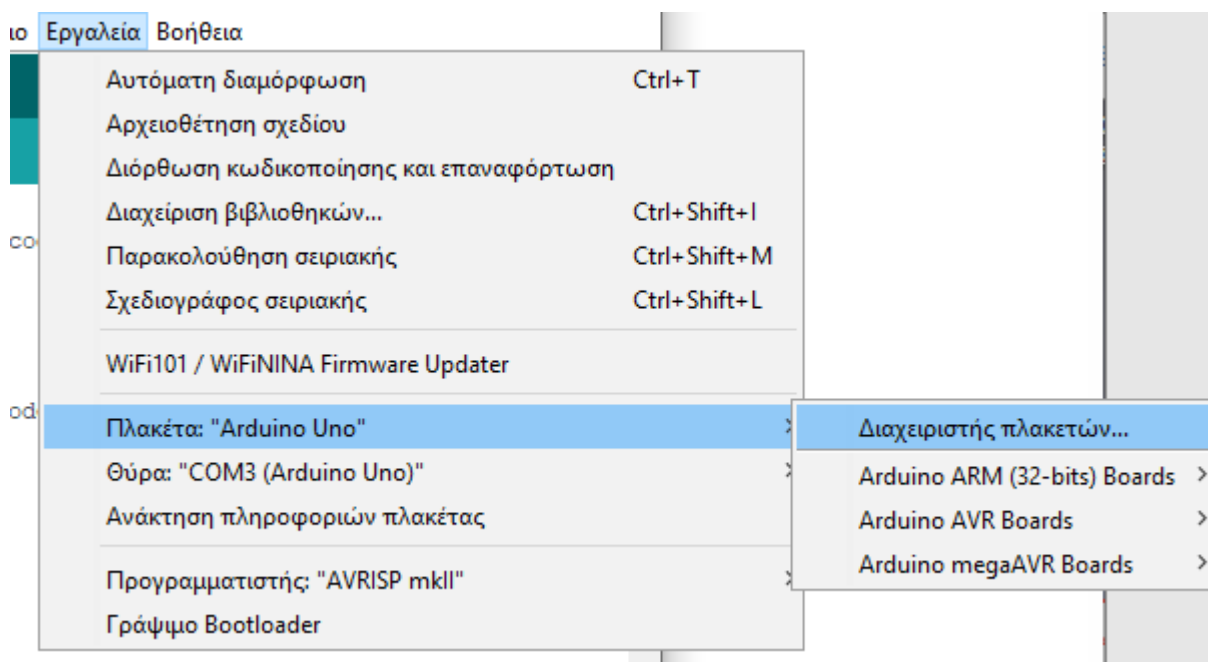
«*Εργαλεία/Πλακέτα*»

και

«*Εργαλεία/Θύρα*»

μπορούμε να αλλάξουμε τις ανωτέρω ρυθμίσεις εάν επιθυμούμε την χρήση διαφορετικού μοντέλου ηλεκτρονικής πλακέτας της οικογένειας Arduino.

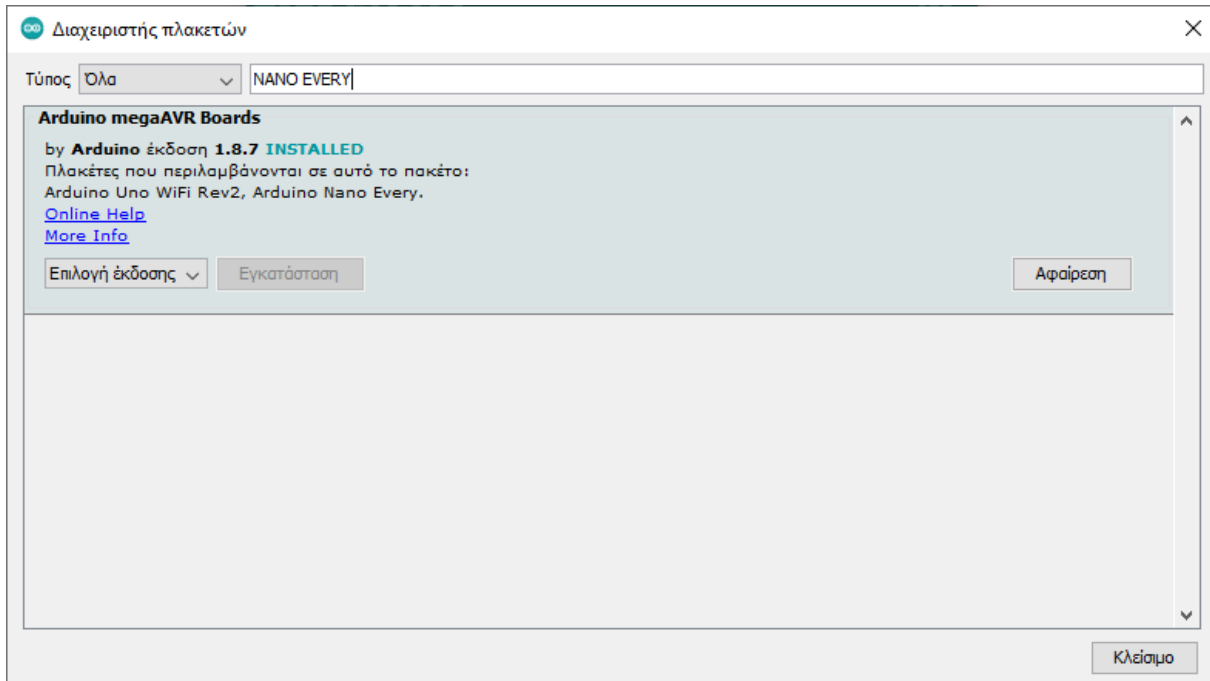
Για τον προγραμματισμό και έλεγχο του υλοποιημένου κυκλώματος «*Μοχλός/Arduino NANO*» συγκεκριμένα, εκτελούμε την διαδικασία «*Εργαλεία/Πλακέτα*» και παρατηρούμε ότι στην υπάρχουσα λίστα δεν περιλαμβάνεται ο μικροελεγκτής Arduino NANO EVERY, οπότε με εκτέλεση της λειτουργίας «*Εργαλεία/Πλακέτα/Διαχειριστής Πλακετών*»,



Εικόνα 3.10 Ρύθμιση επιλογής μοντέλου ηλεκτρονικής πλακέτας Arduino

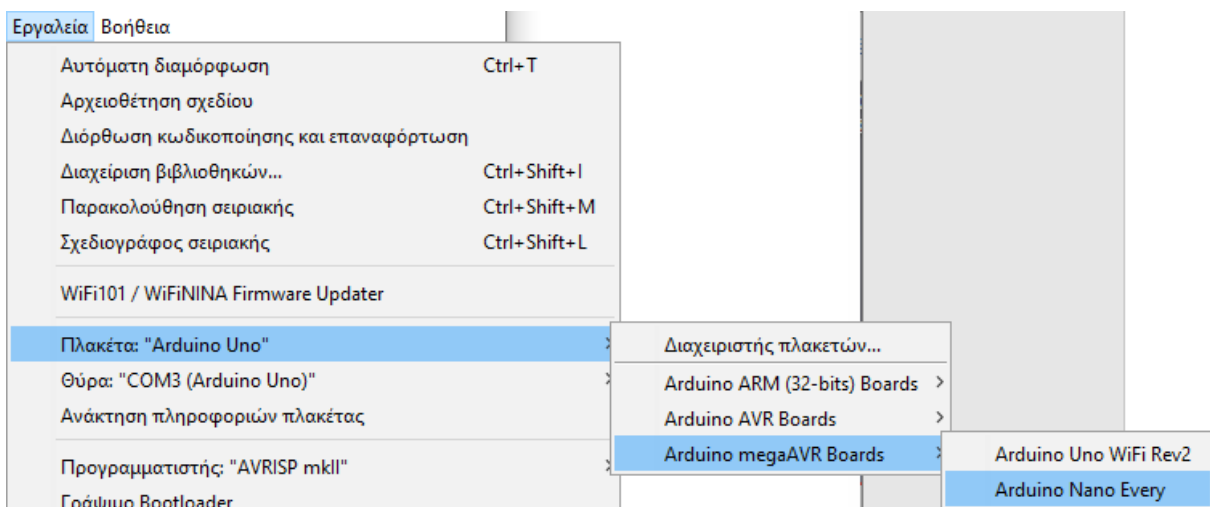
εμφανίζεται η καρτέλα διαχείρισης πλακετών μέσω της οποίας πραγματοποιούμε αναζήτηση της επιθυμητής ηλεκτρονικής πλακέτας, ως φαίνεται στην ανωτέρω εικόνα 3.10.

Με εύρεση και επιτυχή εγκατάσταση του πακέτου «*Arduino megaAVR Boards*» που περιλαμβάνει την πλακέτα *Arduino Nano Every*, εμφανίζεται το ακόλουθο παράθυρο της εικόνας 3.11 στην οθόνη του υπολογιστή:



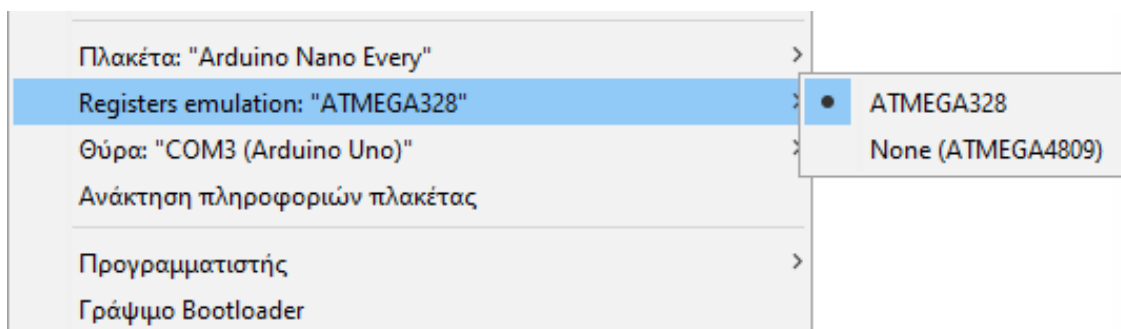
Εικόνα 3.11 Ένδειξη επιτυχούς εγκατάστασης ρυθμίσεων πακέτου της ηλεκτρονικής πλακέτας Arduino μοντέλου NANO EVERY

Με επιπλέον εκτέλεση της διαδικασίας «*Εργαλεία/Πλακέτα/Arduino megaAVR Boards/Arduino Nano Every*» (εικ. 12):



Εικόνα 3.12 Ρύθμιση επιλογής μοντέλου πλακέτας Arduino

καθίσταται διαθέσιμη και η πρόσθετη επιλογή «Εργαλεία/Registers emulation» (εικ. 3.13):



Εικόνα 3.13 Εξομοίωση καταχωρητών (Registers emulation) – Ρύθμιση επιλογής εξομοιωτή μνήμης για την πλακέτα Arduino NANO EVERY

Το Arduino NANO EVERY αποτελεί την ολοκαίνουργια έκδοση της σειράς Arduino NANO ως μια ισοδύναμη εξέλιξη του, με ισχυρότερο επεξεργαστή (ATmega4809) μεγαλύτερης μνήμη RAM. Ο μικροελεγκτής της πλακέτας Arduino NANO EVERY λειτουργεί στα 5V και είναι πλήρως συμβατός με τα αυθεντικά σχέδια της σειράς Arduino NANO, οι ακροδέκτες είναι ακριβώς οι ίδιοι και ως αποτέλεσμα καθίσταται εφικτή η αντικατάσταση οποιασδήποτε πλακέτας έκδοσης Arduino NANO με το νέο μοντέλο Arduino NANO EVERY. Από πλευράς λογισμικού, δεδομένου ότι η πλακέτα EVERY διαθέτει μονάδα μικροελεγκτή ATmega4809 σε αντίθεση με όλες τις άλλες εκδόσεις Arduino NANO που διαθέτουν ATmega328, ενδέχεται να υπάρχουν θέματα με βιβλιοθήκες που δεν έχουν σχεδιαστεί για να διαχειρίζονται τον συγκεκριμένο μικροελεγκτή.

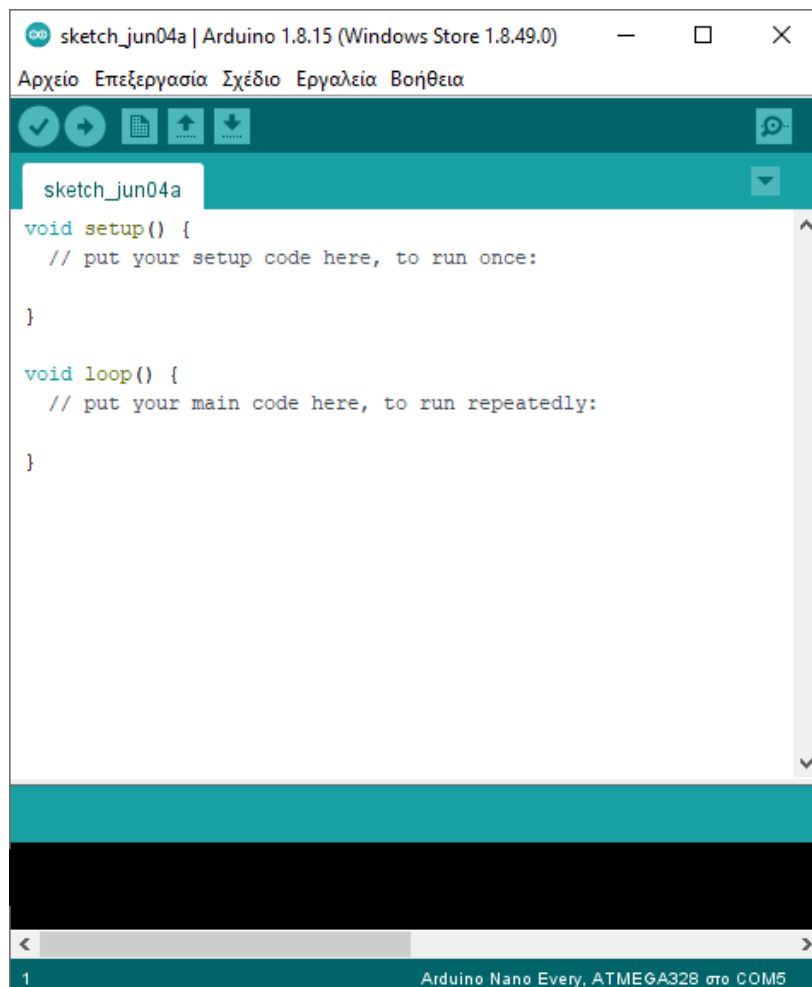
Για αυτές τις περιπτώσεις, μπορεί να χρησιμοποιηθεί η λειτουργία "Registers Emulation", για μίμηση των ATmega328 καταχωρητών στον υπάρχον ATmega4809 κατά την μεταγλώττιση (compilation process).

| Επεξεργαστής | Arduino NANO | Arduino NANO EVERY |
|------------------------|--------------|--------------------|
| Τύπος | Atmega328 | Atmega4809 |
| Ταχύτητα (Clock speed) | 16MHz | 20MHz |
| Διεπαφή με Η/Υ | USB type | USB type |
| Τάση λειτουργίας | 5V | 5V |
| Μνήμη Flash | 32KB | 48KB |
| EEPROM | 1KB | 256byte |
| SRAM | 2KB | 6KB |

Με περαιτέρω εκτέλεση της διαδικασίας «Εργαλεία/Θύρα/COM5» για επιλογή θύρας (εικ. 3.14) προκύπτουν οι επιθυμητές και απαραίτητες ρυθμίσεις, όπως αυτές παρουσιάζονται ως ένδειξη στο κάτω δεξιό άκρο του παραθύρου *Arduino IDE* (εικ. 3.15), θέτοντας πλέον τον μικροελεγκτή *Arduino NANO EVERY* έτοιμο για επικοινωνία και προγραμματισμό μέσω *H/Y*.



Εικόνα 3.14 Ρύθμιση επιλογής θύρας τρέχουσας ηλεκτρονικής πλακέτας *Arduino*



Εικόνα 3.15 Περιβάλλον *Arduino IDE* ρυθμισμένο για επικοινωνία *H/Y* με το *Arduino NANO EVERY*, εξομοιωτή καταχωρητών στο *ATMEGA328* και θύρα *COM5*

Προγραμματιστικός κώδικας ελέγχου λειτουργίας αναλογικού μοχλού

Με διασύνδεση μεταξύ μικροελεγκτή Arduino NANO EVERY και αναλογικού μοχλού 2 αξόνων, ο κώδικας που αναπτύχθηκε και παρατίθεται κατωτέρω χρησιμοποιείται για την ανάγνωση των τιμών/εντολών εισόδου από ολίσθηση του μοχλού με την βοήθεια του μικροελεγκτή.

```
/*
ΚΩΔΙΚΑΣ ΓΙΑ ΑΝΑΓΝΩΣΗ ΤΩΝ ΤΙΜΩΝ ΤΑΣΗΣ ΤΩΝ 2 ΠΟΤΕΝΣΙΟΜΕΤΡΩΝ ΜΕ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΩΝ ΤΙΜΩΝ X, Y ΚΑΙ SW
ΜΕΣΩ ΤΟΥ SERIAL MONITOR ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ARDUINO IDE
*/

const int joystick_VRX = A0;
const int joystick_VRY = A1;

void setup() {
  Serial.begin(9600);          // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP);   // Set pin connections as 'joystick_SW = 2'
}
void loop() {
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2);             // Read the digital input of the Switch / Push button

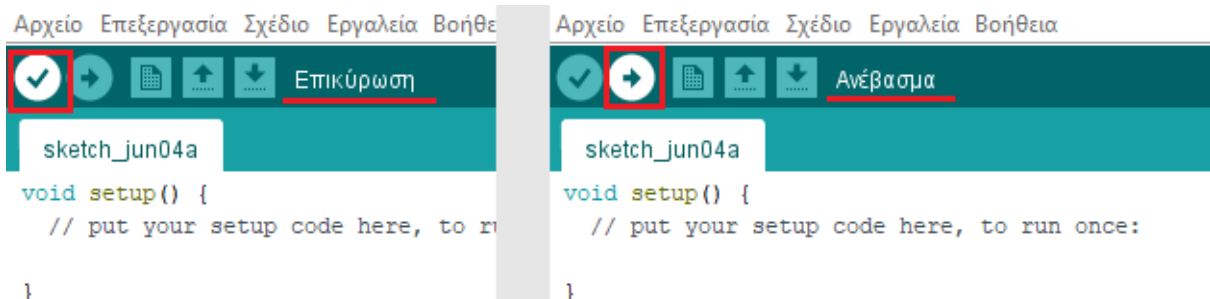
  // Print the results on the screen using the Serial Monitor
  Serial.print("X_Voltage = ");
  Serial.print(X_ADC_Value);
  Serial.print("\t");
  Serial.print("Y_Voltage = ");
  Serial.print(Y_ADC_Value);
  Serial.print("\t");
  Serial.print("SW_State = ");
  Serial.println(SW_State);
  delay(500);
}
```

ΚΩΔΙΚΑΣ_1 Κώδικας ανάγνωσης και εμφάνισης των τιμών των 2 ποτενσιομέτρων του αναλογικού μοχλού, εκφρασμένες σε bits

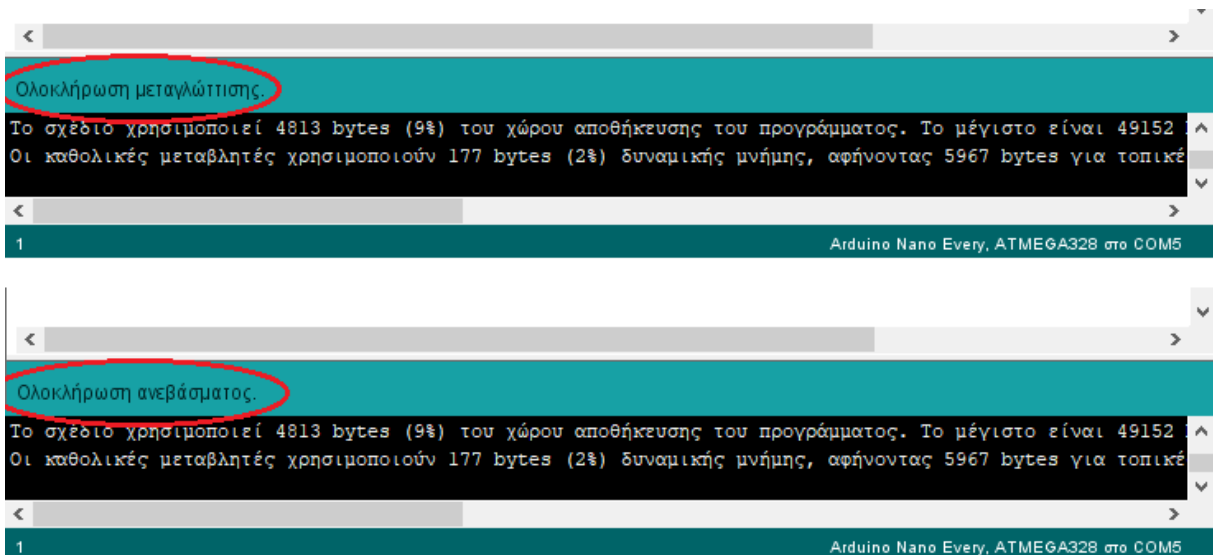
Στον εν λόγω κώδικα γίνεται αξιοποίηση και της επιπλέον λειτουργίας του μοχλού ως διακόπτης μπουτόν (*switch*). Ο μοχλός από κατασκευής του είναι σχεδιασμένος έτσι ώστε με ανάγνωση της τιμής ή αλλιώς κατάστασής του ως switch να φέρει την τιμή “1” όταν δεν χρησιμοποιείται.

Καθαρά, λοιπόν, από επιλογή του προγραμματιστή, γίνεται χρήση της εντολής “SW_State = !digitalRead(2);” ώστε η καθορισμένη κατάσταση του μοχλού να αντιστρέφεται και να ορίζεται ως “0” όταν δεν χρησιμοποιείται ως switch.

Με χρήση των ενεργειών «**Επικύρωση**» και «**Ανεβάσμα**» που βρίσκονται στην γραμμή εργασιών του παραθύρου του *Arduino IDE* (εικόνες 3.16 & 3.17), ο εν λόγω κώδικας εγκαθίσταται επιτυχώς στον μικροελεγκτή NANO EVERY.



Εικόνα 3.16 Λειτουργία Επικύρωσης και Ανεβάσματος, αντίστοιχα



Εικόνα 3.17 Ενημέρωση επιτυχούς Επικύρωσης και Ανεβάσματος, αντίστοιχα

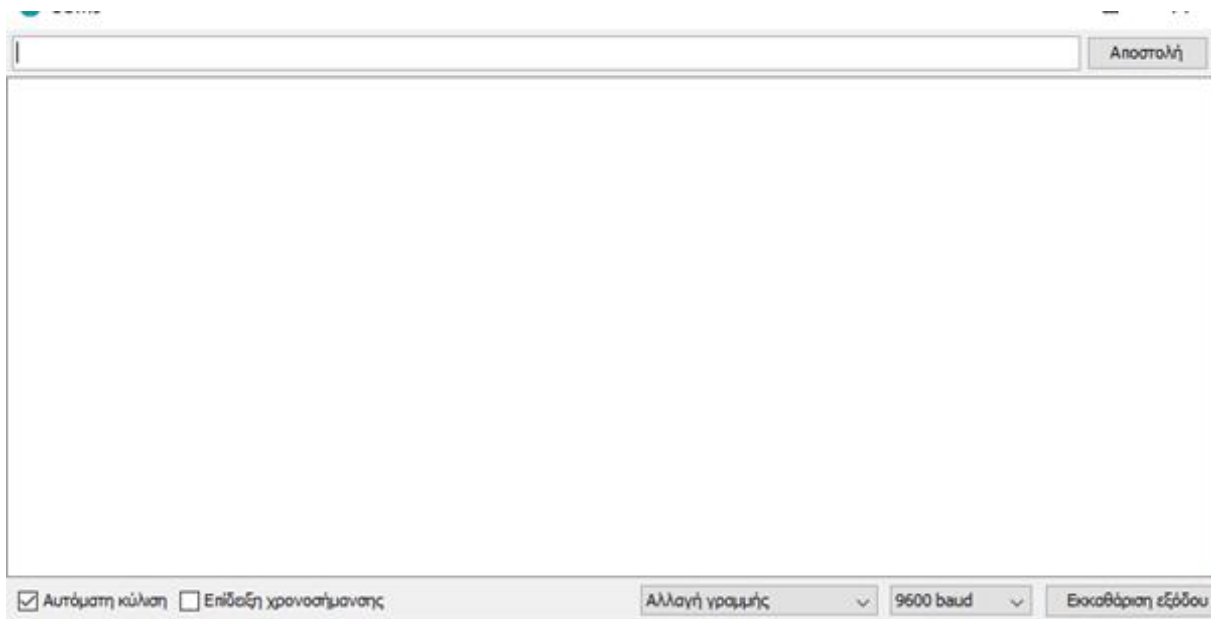
Στο σημείο αυτό πραγματοποιείται έλεγχος για την ορθή συνδεσμολογία και λειτουργία του κυκλώματος, όπου βάσει του κώδικα θα πρέπει με ολίσθηση στον άξονα των X – συγκεκριμένα με αλλαγή της αρχικής θέσης του μοχλού αρχικά προς τα δεξιά και έπειτα προς τα αριστερά– να παρατηρείται η σταδιακή αύξηση και μετέπειτα μείωση της παραγόμενης διαφοράς δυναμικού στα άκρα εκάστου ποτενσιομέτρου (σταδιακή μετάβαση τιμής από $\frac{1}{2} V_{max}$ της θέσης ηρεμίας σε V_{max} της ακραίας δεξιάς μετατόπισης και έπειτα σε $0V$ της ακραίας αριστερής μετατόπισης και επαναφορά στην θέση ηρεμίας του μοχλού με τιμή τάσης $\frac{1}{2} V_{max}$).

Μετά το επιτυχές «ανέβασμα» του κώδικα στον μικροελεγκτή Arduino, με χρήση της επιλογής «**Παρακολούθηση σειριακής**» που βρίσκεται επίσης στην γραμμή εργαλείων:



Εικόνα 3.18 Εικονίδιο λειτουργίας παρακολούθησης Σειριακής Θύρας (Serial Monitor)

ανοίγει ένα νέο παράθυρο, με την μορφή που παρουσιάζεται στην εικόνα της επόμενης σελίδας:



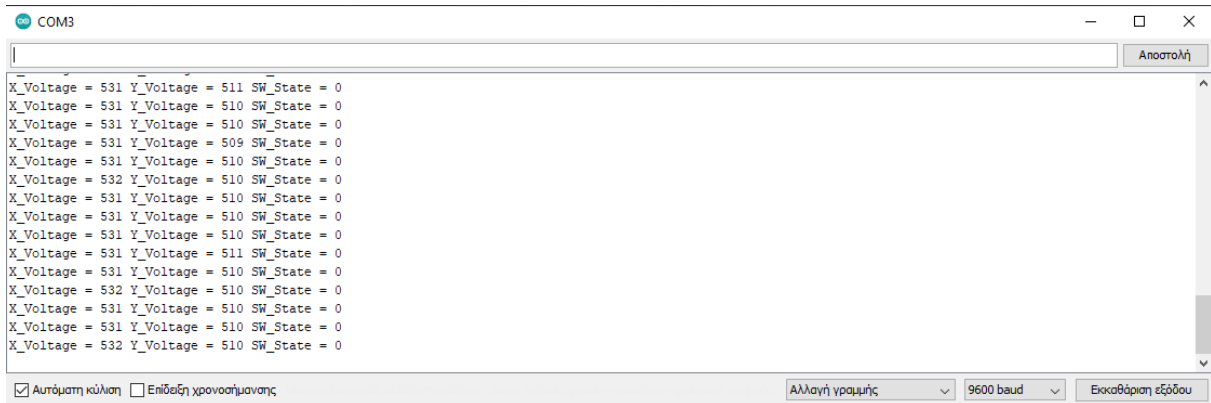
Εικόνα 3.19 Αρχική κατάσταση του παραθύρου γνωστού ως Σειριακή Οθόνη Παρακολούθησης (Serial Monitor)

Το παράθυρο της εικόνας 3.19 ονομάζεται **Σειριακή Οθόνη Παρακολούθησης** (*Serial Monitor*), αποτελεί μέρος του λογισμικού *Arduino IDE* και ο σκοπός ύπαρξής του είναι να επιτρέπεται η αποστολή και λήψη μηνυμάτων από έναν υπολογιστή σε μια πλακέτα *Arduino*, μέσω του καλωδίου *USB*.

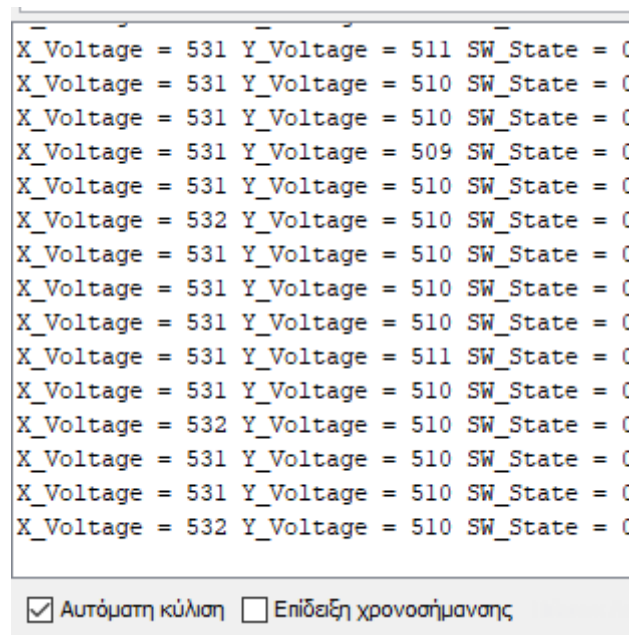
Με χρήση του *Serial Monitor* του *Arduino IDE* παρατηρείται πληθώρα αποτελεσμάτων περί της συμπεριφοράς του αναλογικού μοχλού.

Αρχικά γίνεται χρήση του προγραμματιστικού κώδικα που παρατέθηκε προηγουμένως (*ΚΩΔΙΚΑΣ_1*) για ανάγνωση των σημάτων που προσφέρονται στον μικροελεγκτή *NANO* από τον αναλογικό μοχλό.

Στις δύο κατωτέρω εικόνες παρουσιάζεται η κατάσταση του μοχλού όταν αυτός βρίσκεται σε αδράνεια, με τις τιμές των X και Y να αντιστοιχούν στην μεσαία θέση (τιμή κοντά στο 512) και ο διακόπτης μπουτόν (SW) να λαμβάνει την τιμή '0':



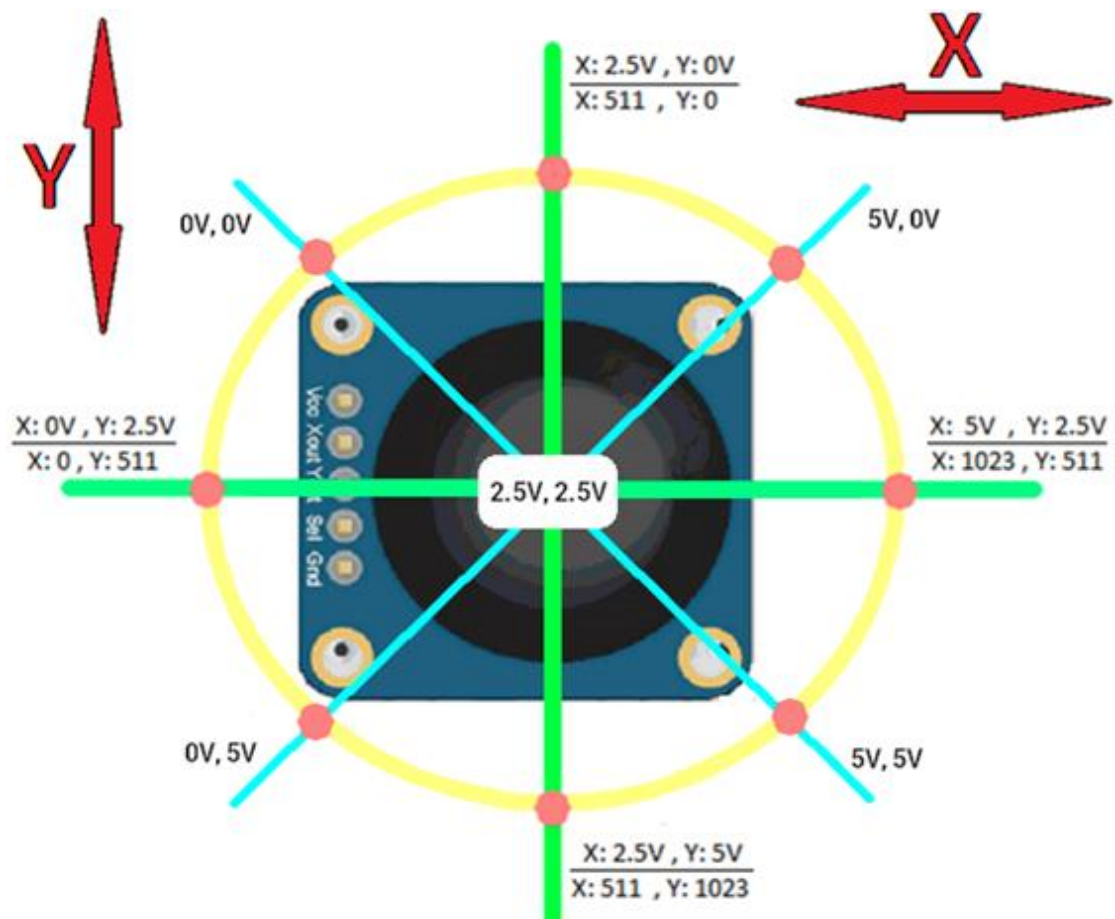
Εικόνα 3.20 Σειριακή οθόνη παρακολούθησης με εμφάνιση των τιμών τάσης των 2 ποτενσιομέτρων του αναλογικού μοχλού σε αδράνεια ($X \approx 512$, $Y \approx 512$, $SW=0$)



Εικόνα 3.20 – Zoom in Τιμές των 2 ποτενσιομέτρων του αναλογικού μοχλού, εκφρασμένες σε bits από την ADC ανάλυση, όταν ο μοχλός βρίσκεται σε αδράνεια ($X \approx 512$, $Y \approx 512$, $SW=0$)

Μια σημαντική λεπτομέρεια που εξηγείται με μεγαλύτερη ευκρίνεια επί του προγραμματιστικού κώδικα παρά με μια απλή θεωρητική αναφορά είναι το γεγονός ότι ο μικροελεγκτής Arduino NANO EVERY –αλλά και γενικότερα οποιαδήποτε ηλεκτρονική πλακέτα της οικογένειας Arduino– διαθέτει ADC ανάλυση των 10bit, έτσι οι τιμές εκάστου αναλογικού καναλιού κυμαίνονται μεταξύ του εύρους τιμών 0-1023.

Ουσιαστικά το «σημείο ηρεμίας» του μοχλού –η κάθετη επί της βάσεως θέση– διαθέτει τιμή κοντά στην τιμή 511 τόσο για τον X όσο και για τον Y άξονα, με ελάχιστη τιμή το 0 και μέγιστη το 1023, ως αποτυπώνεται αναλυτικότερα και στην εικόνα που ακολουθεί. Αυτός είναι και ο λόγος τροποποίησης του κώδικα No1 και ανάπτυξη ενός νέου (ΚΩΔΙΚΑΣ_2) που κάνει χρήση των μεταβλητών “X_Volt” και “Y_Volt” για μετατροπή των τιμών από μονάδες μέτρησης bits σε Volts ώστε να επιτυγχάνεται καλύτερος καθορισμός, ευκρίνεια και ανάγνωση των τιμών των εισόδων για έκαστο άξονα από έναν χρήστη.



Εικόνα 3.21 Διακύμανση τιμών τάσης των δύο ποτενσιομέτρων ενός αναλογικού μοχλού με ολίσθησή του στους άξονες X και Y, αντίστοιχα

Από προσαρμογή του προηγούμενου κώδικα, με ένταξη των προαναφερόμενων μεταβλητών συναρτήσεων *X_Volt* και *Y_Volt* για εμφάνιση των τιμών των 2 ποτενσιομέτρων σε μορφή τάσης εύρους 0-5V με την τιμή 2,5V να ορίζεται ως το σημείο ηρεμίας του μοχλού, προκύπτει ο εξής νέος κώδικας προγραμματισμού:

```
/*
ΑΝΑΓΝΩΣΗ ΤΩΝ ΤΙΜΩΝ ΤΑΣΗΣ ΤΩΝ 2 ΠΟΤΕΝΣΙΟΜΕΤΡΩΝ ΜΕ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΩΝ ΤΙΜΩΝ X, Y ΚΑΙ SW
ΜΕΣΩ ΤΟΥ SERIAL MONITOR ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ARDUINO IDE ΚΑΙ ΕΚΦΡΑΣΗ ΤΟΥΣ ΣΕ VOLTS
*/

const int joystick_VRX = A0;
const int joystick_VRY = A1;

void setup() {
  Serial.begin(9600);          // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP);   // Set pin connections as 'joystick_SW = 2'
}

void loop() {
  int X_ADC_Value = analogRead(joystick_VRX);    // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY);    // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2);                // Read the digital input of the Switch

  float X_Volt = ( ( X_ADC_Value * 5.0 ) / 1023 ); // Convert digital value (bits) to Voltage
  float Y_Volt = ( ( Y_ADC_Value * 5.0 ) / 1023 ); // Convert digital value (bits) to Voltage

  // Print the results on the screen using the Serial Monitor
  Serial.print("X_Voltage = ");
  Serial.print(X_Volt);
  Serial.print("\t");
  Serial.print("Y_Voltage = ");
  Serial.print(Y_Volt);
  Serial.print("\t");
  Serial.print("SW_State = ");
  Serial.println(SW_State);
  delay(500);
}
```

ΚΩΔΙΚΑΣ_2 Κώδικας ανάγνωσης και εμφάνισης των τιμών των 2 ποτενσιομέτρων του αναλογικού μοχλού, εκφρασμένες σε Volts

Στον εν λόγω κώδικα δεν χρειάστηκε τροποποίηση περί της έκφρασης της κατάστασης του μοχλού ως διακόπτης μπουτόν (SW) καθώς λαμβάνει εξαρχής τις επιθυμητές τιμές “0” ή “1”.

Με εκτέλεση του κώδικα No2, το Serial Monitor εμφανίζει τα ακόλουθα αποτελέσματα:

```
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 0
```

Αυτόματη κύλιση Επίδειξη χρονοσήμανσης

Εικόνα 3.22 Τιμές των 2 ποτενσιομέτρων και του διακόπτη μπουτόν ενός αναλογικού μοχλού σε αδράνεια

Στην ανωτέρω εικόνα παρουσιάζονται οι τιμές των δύο ποτενσιομέτρων X και Y με τιμή $\approx 2,5\text{Volt}$ και ο διακόπτης μπουτόν $SW=0$, όταν ο αναλογικός μοχλός βρίσκεται σε κατάσταση ηρεμίας (αδράνεια).

Βάσει της ανωτέρω εικόνας 3.22, ολισθαίνοντας τον μοχλό από την θέση ηρεμίας κατά τον οριζόντιο άξονα X αρχικά προς τα δεξιά (εικ. 3.23) και έπειτα προς τα αριστερά (εικ. 3.24), παρατηρείται η σταδιακή αύξηση και μετέπειτα μείωση της παραγόμενης διαφοράς δυναμικού (σταδιακή μετάβαση τιμής από $\frac{1}{2}V_{max} \approx 2,5V$ της θέσης ηρεμίας στην μέγιστη τιμή $X_Voltage=5V$ της ακραίας δεξιάς μετατόπισης, έπειτα στην ελάχιστη τιμή $X_Voltage=0V$ της ακραίας αριστερής μετατόπισης και επαναφορά στην θέση ηρεμίας του μοχλού με τιμή τάσης $\frac{1}{2}V_{max}$).

| | | |
|------------------|------------------|--------------|
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 5.00 | Y_Voltage = 2.46 | SW_State = 0 |

Εικόνα 3.23 Μετατόπιση του μοχλού κατά τον X άξονα και συγκεκριμένα στην ακραία δεξιά θέση ($X_Voltage_{(max)} = 5$)

| | | |
|------------------|------------------|--------------|
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |
| X_Voltage = 0.00 | Y_Voltage = 2.49 | SW_State = 0 |

Εικόνα 3.24 Μετατόπιση του μοχλού κατά τον X άξονα και συγκεκριμένα στην ακραία αριστερή θέση ($X_Voltage_{(min)} = 0$)

Επίσης, βάσει του ΚΩΔΙΚΑ_2, ο διακόπτης μπουτόν (SW) δύναται να λαμβάνει τιμή '1' όταν ο μοχλός πιέζεται παρατεταμένα στην εν λόγω θέση, αλλιώς αποκτά την τιμή '0':

| | | |
|------------------|------------------|--------------|
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |
| X_Voltage = 2.60 | Y_Voltage = 2.49 | SW_State = 1 |

Εικόνα 3.27 Παρατεταμένο πάτημα του μοχλού για διατήρηση της λειτουργίας του ως διακόπτη μπουτόν μετάβαση από την κατάσταση ηρεμίας SW_State = 0 στην ενεργή κατάσταση SW_State = 1)

Για καθαρά πειραματικό σκοπό, γίνεται προσαρμογή του προγραμματιστικού κώδικα Νο2 με αποτέλεσμα την δημιουργία του κώδικα Νο3, ώστε οι τιμές X και Y να εμφανίζονται μόνο με την λειτουργία του μοχλού ως switch στην ενεργή κατάσταση 1:

```

SW_State = 0
SW_State = 0
SW_State = 0
SW_State = 0
SW_State = 0
SW_State = 0
SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1

```

Εικόνα 3.28 Πατώντας και κρατώντας πατημένο τον μοχλό, η τιμή SW_State μεταβαίνει από την τιμή '0' στην τιμή '1' και μόνο τότε γίνεται ανάγνωση των τιμών μέσω του Serial Monitor

Με κρατημένο τον μοχλό για διατήρηση της κατάστασης SW_State=1, ο μοχλός δύναται επίσης να ολισθαίνει κατά τους άξονες X και Y:

```

SW_State = 0
X_Voltage = 2.60      Y_Voltage = 2.49      SW_State = 1
SW_State = 0
SW_State = 0
X_Voltage = 2.60      Y_Voltage = 0.77      SW_State = 1
SW_State = 0
X_Voltage = 5.00      Y_Voltage = 0.00      SW_State = 1
SW_State = 0
X_Voltage = 2.85      Y_Voltage = 5.00      SW_State = 1
SW_State = 0
X_Voltage = 0.00      Y_Voltage = 2.49      SW_State = 1
SW_State = 0
X_Voltage = 5.00      Y_Voltage = 2.50      SW_State = 1
SW_State = 0
SW_State = 0

```

Εικόνα 3.29 Η μετατόπιση του μοχλού στους X και Y άξονες γίνεται εμφανής στο Serial Monitor μόνο με διατήρηση της κατάσταση SW_State=1


```

/*
ΔΙΑΒΑΣΜΑ ΤΩΝ ΤΙΜΩΝ ΤΩΝ X ΚΑΙ Y ΜΕ ΠΑΡΑΤΕΤΑΜΕΝΟ ΠΑΤΗΜΑ ΤΟΥ SW
ΚΑΙ ΕΜΦΑΝΙΣΗ ΤΟΥΣ ΜΕΣΩ ΤΟΥ SERIAL MONITOR ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ARDUINO IDE
*/

const int joystick_VRX = A0;
const int joystick_VRY = A1;

void setup() {
  Serial.begin(9600);          // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP);   // Set pin connections as 'joystick_SW = 2'
}
void loop() {
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int SW_State = digitalRead(2);             // Read the digital input of the Switch

  if (SW_State == HIGH) { // If the switch is pressed
    // Print the X & Y values on the screen using the Serial Monitor
    float X_Volt = ( ( X_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
    float Y_Volt = ( ( Y_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
    Serial.print("X_Voltage = ");
    Serial.print(X_Volt);
    Serial.print("\t");
    Serial.print("Y_Voltage = ");
    Serial.print(Y_Volt);
  }

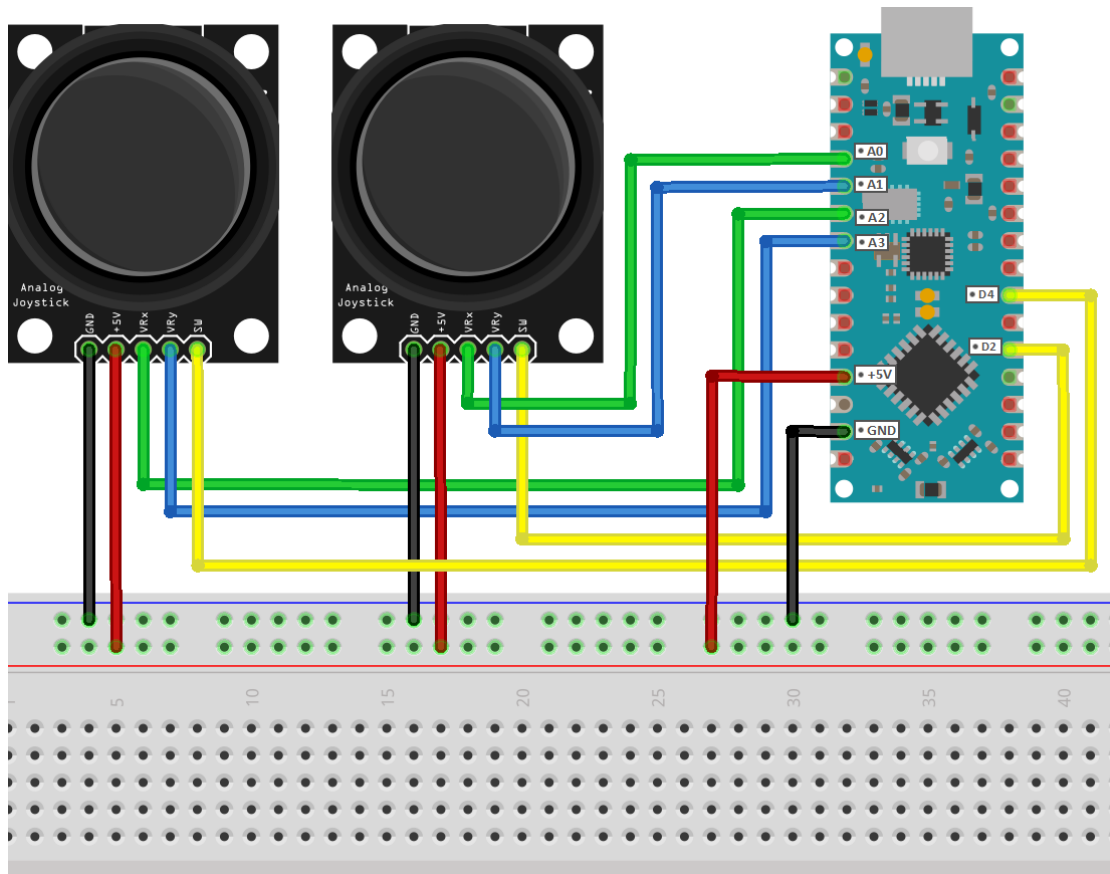
  // Print the state of the switch using the Serial Monitor
  Serial.print("\t");
  Serial.print("SW_State = ");
  Serial.println(SW_State);
  delay(500);
}

```

ΚΩΔΙΚΑΣ_3 Κώδικας ανάγνωσης των τιμών του αναλογικού μοχλού μόνο στις περιπτώσεις που αυτός αξιοποιείται ως διακόπτης μπουτόν (SW_State=1 / SW PRESSED)

Προσθήκη 2^{ου} αναλογικού μοχλού και αντίστοιχος έλεγχος λειτουργίας

Με προσθήκη ενός επιπλέον αναλογικού μοχλού 2 αξόνων στο ήδη υπάρχον κύκλωμα προκύπτει το σχηματικό διάγραμμα της εικόνας 3.30.



Εικόνα 3.30 Σχηματικό διάγραμμα διασύνδεσης δύο αναλογικών χειριστηρίων με τον μικροελεγκτή Arduino NANO EVERY σε περιβάλλον προσομοίωσης

Στο εν λόγω κύκλωμα γίνεται χρήση των X, Y και SW ακροδεκτών του 2^{ου} μοχλού με τρόπο αντίστοιχο με αυτόν του 1^{ου}, με τον σχετικό κώδικα προγραμματισμού να παρατίθεται στην επόμενη σελίδα.

```

/*
ΑΝΑΓΝΩΣΗ ΤΩΝ ΤΙΜΩΝ ΤΑΣΗΣ ΤΩΝ ΠΟΤΕΝΣΙΟΜΕΤΡΩΝ ΤΩΝ ΔΥΟ ΑΝΑΛΟΓΙΚΩΝ ΜΟΧΛΩΝ ΜΕ ΠΑΡΑΚΟΛΟΥΘΗΣΗ
ΤΩΝ ΤΙΜΩΝ X, Y, SW ΚΑΙ X_2, Y_2, SW_2 ΜΕΣΩ ΤΟΥ SERIAL MONITOR ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ARDUINO IDE
*/

const int joystick_VRX = A0;
const int joystick_VRY = A1;

const int joystick_VRX_2 = A2;
const int joystick_VRY_2 = A3;

void setup() {
  Serial.begin(9600);          // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP);   // Set pin connections as 'joystick_SW = 2'
  pinMode(4, INPUT_PULLUP);   // Set pin connections as 'joystick_SW_2 = 4'
}

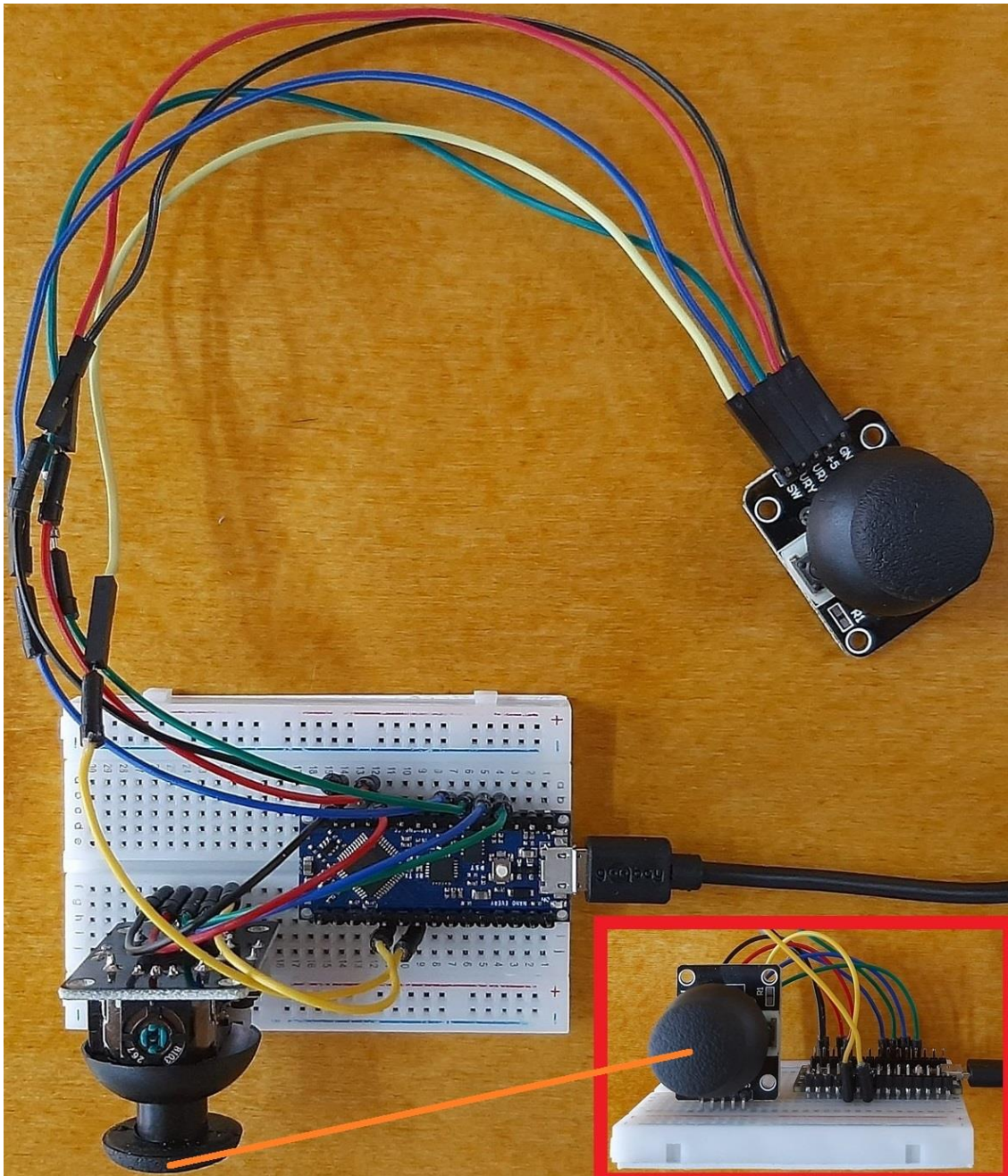
void loop() {
  int X_ADC_Value = analogRead(joystick_VRX);          // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY);          // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2);                      // Read the digital input of Switch
  float X_Volt = ( ( X_ADC_Value * 5.0 ) / 1023 );    // Convert digital value to Voltage
  float Y_Volt = ( ( Y_ADC_Value * 5.0 ) / 1023 );    // Convert digital value to Voltage

  int X_2_ADC_Value = analogRead(joystick_VRX_2);     // Read the analog/mechanical input of X_2-axis
  int Y_2_ADC_Value = analogRead(joystick_VRY_2);     // Read the analog/mechanical input of Y_2-axis
  int SW_2_State = !digitalRead(4);                   // Read the digital input of Switch_2
  float X_2_Volt = ( ( X_2_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
  float Y_2_Volt = ( ( Y_2_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage

  // Print the results on the screen using the Serial Monitor
  Serial.print("X_Voltage = ");
  Serial.print(X_Volt);
  Serial.print("\t");
  Serial.print("X2_Voltage = ");
  Serial.println(X_2_Volt);
  Serial.print("Y_Voltage = ");
  Serial.print(Y_Volt);
  Serial.print("\t");
  Serial.print("Y2_Voltage = ");
  Serial.println(Y_2_Volt);
  Serial.print("SW_State = ");
  Serial.print(SW_State);
  Serial.print("\t\t");
  Serial.print("SW2_State = ");
  Serial.println(SW_2_State);
  Serial.println("");
  delay(1000);
}

```

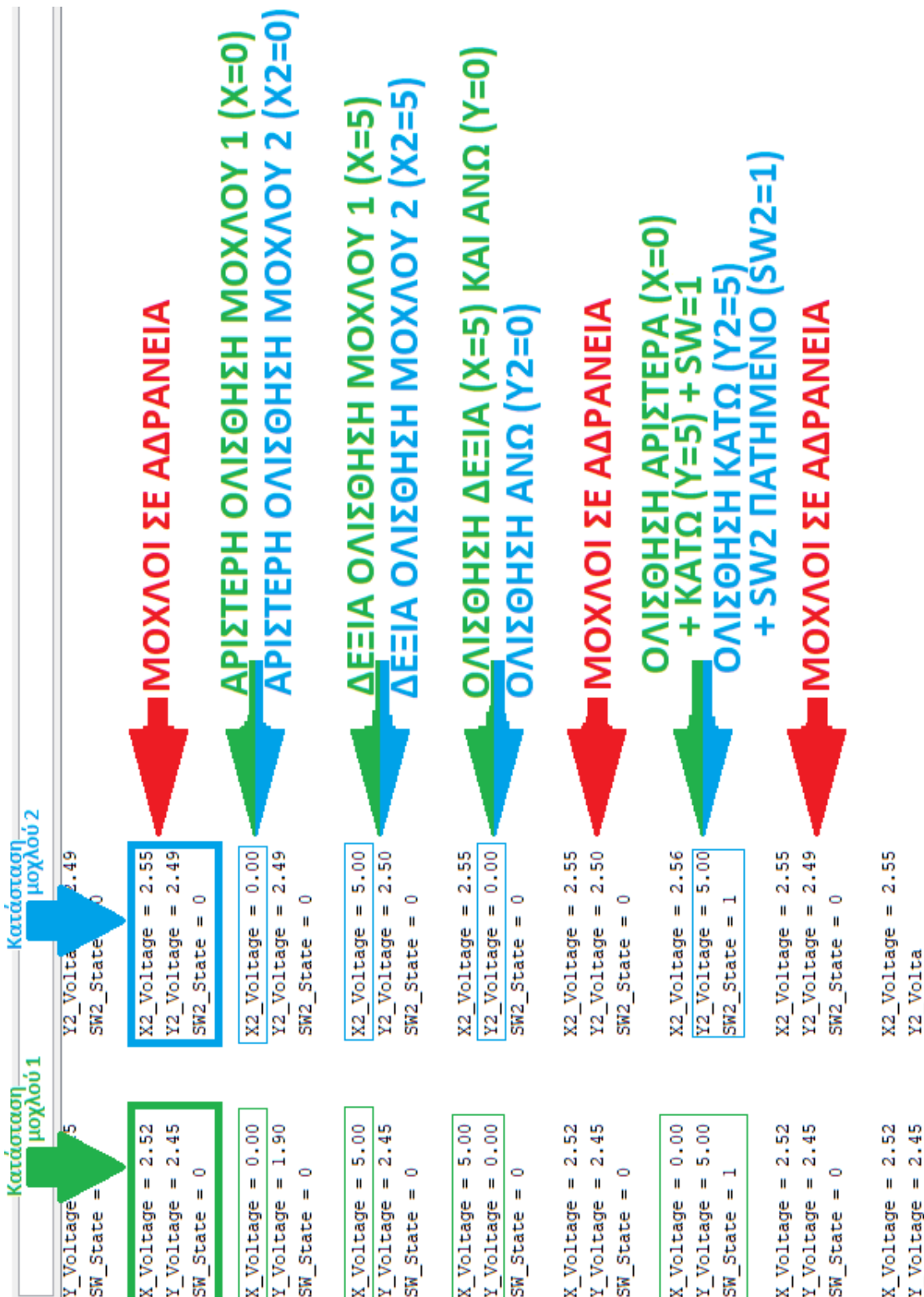
ΚΩΔΙΚΑΣ_4 Κώδικας για ανάγνωση των τιμών των ποτενσιομέτρων των 2 αναλογικών μοχλών και των καταστάσεών τους ως διακόπτες μπουτόν



Εικόνα 3.31 Εναλλακτική διασύνδεση δύο αναλογικών μοχλών με τον μικροελεγκτή Arduino NANO EVERY, υλοποιημένη σε επαγγελματικό εργαστήριο

| | |
|------------------|-------------------|
| Y_Voltage = 2.45 | Y2_Voltage = 2.49 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 2.52 | X2_Voltage = 2.55 |
| Y_Voltage = 2.45 | Y2_Voltage = 2.49 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 0.00 | X2_Voltage = 0.00 |
| Y_Voltage = 1.90 | Y2_Voltage = 2.49 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 5.00 | X2_Voltage = 5.00 |
| Y_Voltage = 2.45 | Y2_Voltage = 2.50 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 5.00 | X2_Voltage = 2.55 |
| Y_Voltage = 0.00 | Y2_Voltage = 0.00 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 2.52 | X2_Voltage = 2.55 |
| Y_Voltage = 2.45 | Y2_Voltage = 2.50 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 0.00 | X2_Voltage = 2.56 |
| Y_Voltage = 5.00 | Y2_Voltage = 5.00 |
| SW_State = 1 | SW2_State = 1 |
| X_Voltage = 2.52 | X2_Voltage = 2.55 |
| Y_Voltage = 2.45 | Y2_Voltage = 2.49 |
| SW_State = 0 | SW2_State = 0 |
| X_Voltage = 2.52 | X2_Voltage = 2.55 |
| Y_Voltage = 2.45 | Y2_Voltage = 2.49 |

Εικόνα 3.32 Με ολίσθηση και πάτημα –ή μη– των δύο αναλογικών μοχλών, οι τιμές τάσης των ποτενσιομέτρων αλλά και οι διακόπτες μπουτόν μεταβάλλονται



Εικόνα 3.33 Σχολιασμός αλλαγής καταστάσεων των τιμών των 2 αναλογικών χειριστηρίων

3.1.1.2 Προσθήκη 2 Σερβοκινητήρων για Έλεγχο Απόκρισης I/O ως Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή (Σ.Η.Ε.Ε.) για Βιομηχανικές Εφαρμογές

Με επιτυχή ανάγνωση της κίνησης των αναλογικών μοχλών από τον μικροελεγκτή και άρα επαλήθευση της μεταξύ τους ορθής συνδεσμολογίας, προστίθενται στην συνέχεια δύο σερβοκινητήρες στο υπάρχον κύκλωμα για έλεγχο απόκρισης της εξόδου (σερβοκινητήρες) ως προς την είσοδο (μοχλός) αρχικά με ενσύρματο τρόπο.

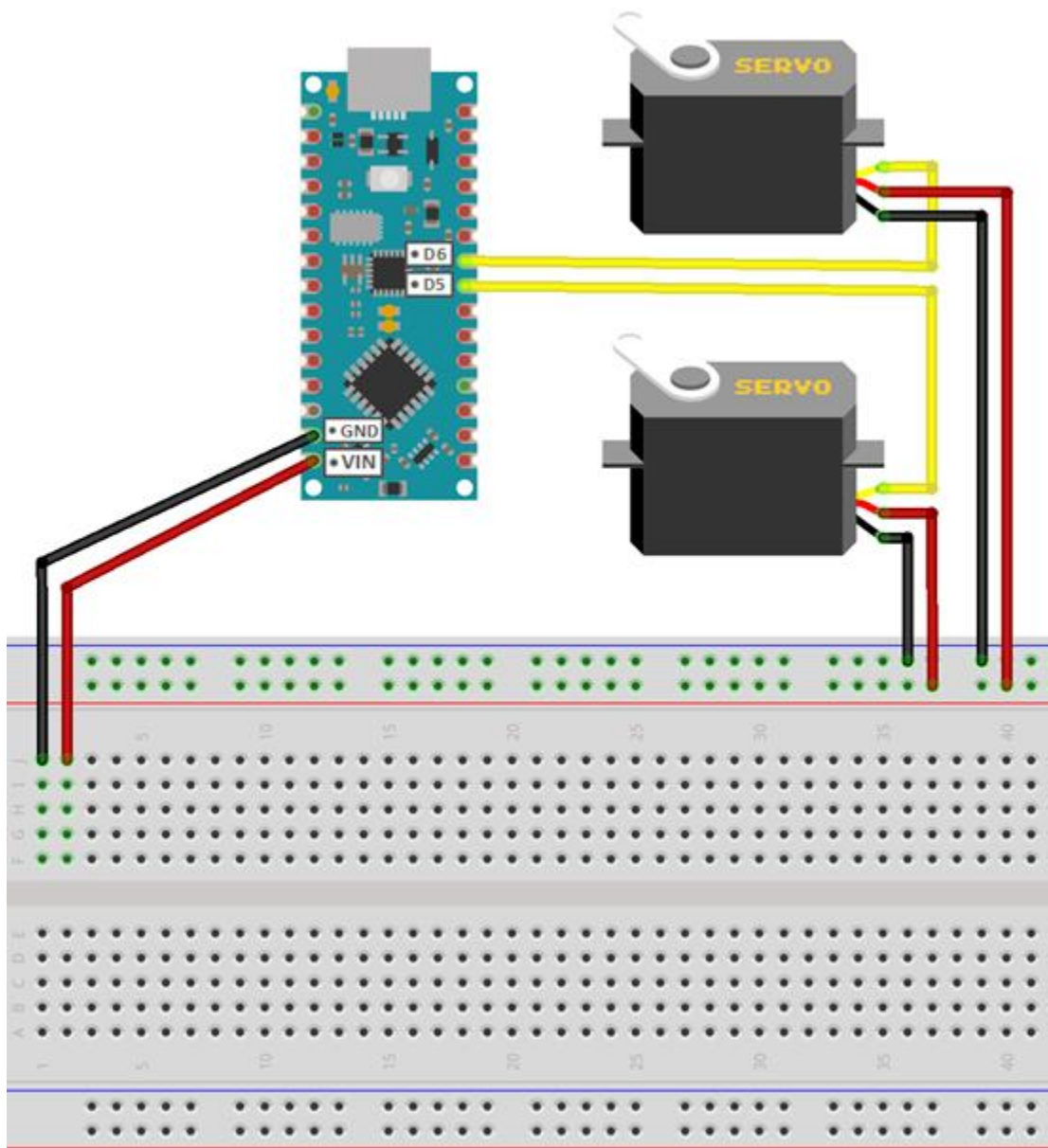
Ένας τέτοιος τύπος κυκλώματος θα μπορούσε κάλλιστα να ονομαστεί ως **Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές** (*Electronic Wired Controller System for Industrial Applications*) καθώς σε αυτήν την φάση της εργασίας γίνεται χρήση ενός εξ' ολοκλήρου ενσύρματου τρόπου διασύνδεσης και επικοινωνίας μεταξύ των μερών του, αντί ασύρματου που είναι το αντικείμενο της Διπλωματικής.

Η επίτευξη της επιθυμητής συμπεριφοράς εισόδου/εξόδου της συνδεσμολογίας αυτής σημαίνει ότι επιτυγχάνεται ο έλεγχος μιας ή και περισσότερων εφαρμογών από έναν χρήστη μέσω του συστήματος ηλεκτρονικού ενσύρματου ελεγκτή, αφήνοντας πλέον ως τελικό βήμα προς ολοκλήρωση της εργασίας την πραγματοποίηση απομακρυσμένου ελέγχου χρησιμοποιώντας ασύρματη μέθοδο επικοινωνίας.

Προτού γίνει ο έλεγχος του συστήματος απόκρισης I/O, επιλέγεται η δοκιμή σύνταξης κώδικα για μελέτη του τρόπου λειτουργίας των δύο σερβοκινητήρων δίχως την ύπαρξη αναλογικού μοχλού, με την εν λόγω συνδεσμολογία να παρουσιάζεται στον πίνακα 3_3 που ακολουθεί:

ΠΙΝΑΚΑΣ 3_3

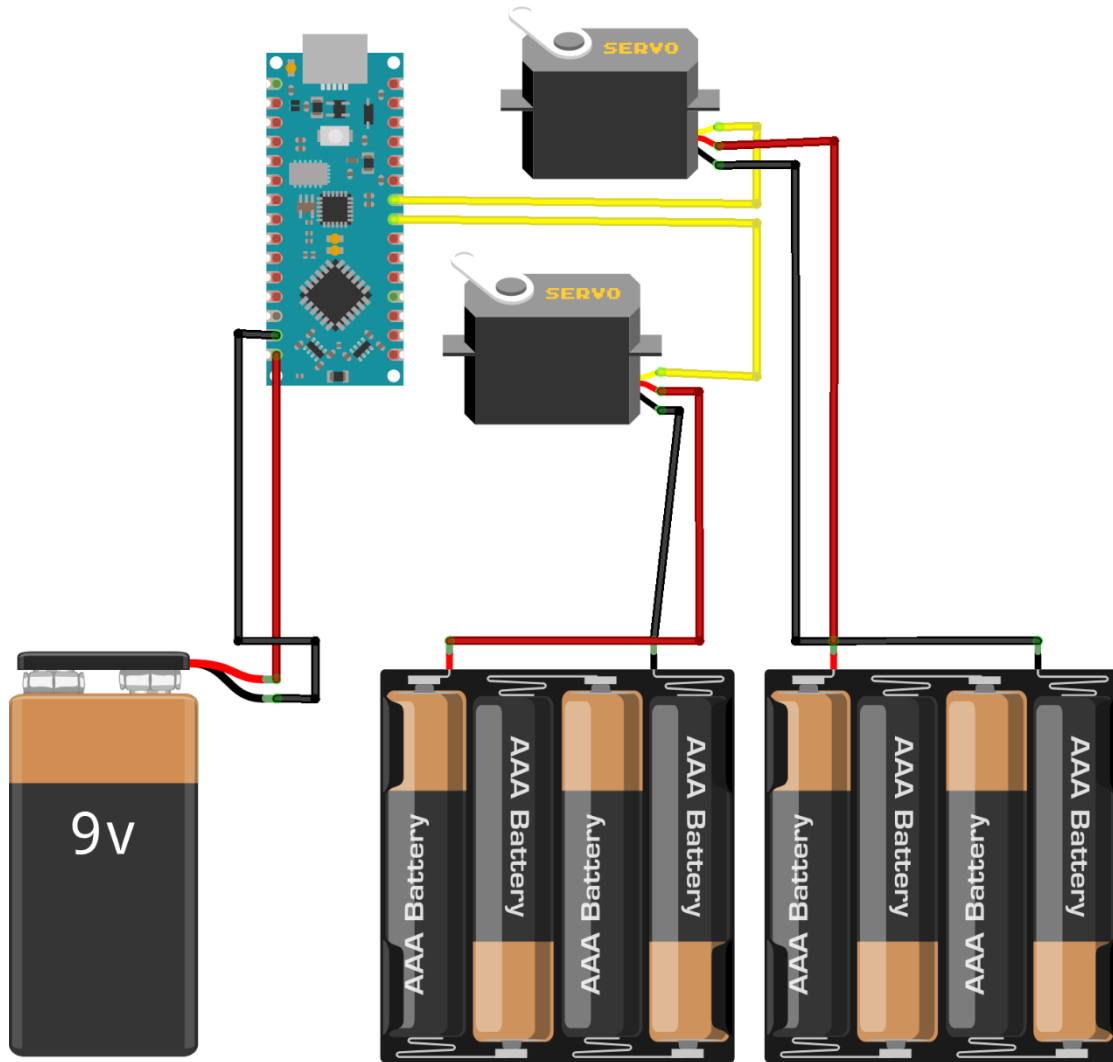
| Ακροδέκτες μικροελεγκτή (NANO) | Σερβοκινητήρας (1 ^{ος}) | Σερβοκινητήρας (2 ^{ος}) |
|--------------------------------|-----------------------------------|-----------------------------------|
| 5V (analog pin 5V) | VCC | VCC |
| GND (analog pin GND) | GND | GND |
| 5 (digital pin 5) | Signal line (PWM) | — |
| 6 (digital pin 6) | — | Signal line (PWM) |



Εικόνα 3.34 Συνδεσμολογία του Arduino Nano Every με 2 σερβοκινητήρες μοντέλου MG995R περιστροφής 180 μοιρών

Με ένταξη μιας πλακέτας δοκιμών (*breadboard*) στην ανωτέρω συνδεσμολογία, η παροχή ενέργειας δύναται να πραγματοποιείται με διάφορους τρόπους, ήτοι από μία ή περισσότερες εξωτερικές πηγές τροφοδοσίας και κατάλληλη αξιοποίηση των κοινών δυναμικών που παρέχει η πλακέτα δοκιμών ή με άμεση παροχή ενέργειας από ξεχωριστές εξωτερικές πηγές με απευθείας διασύνδεσή τους με εκάστη ηλεκτρονική μονάδα.

Στην προκείμενη περίπτωση γίνεται χρήση μίας μπαταριοθήκης παροχής τάσεως 9V για τροφοδοσία του Arduino NANO EVERY και μία μπαταριοθήκη παροχής 6V για κάθε σερβοκινητήρα, με την εν λόγω μέθοδο τροφοδοσίας να απεικονίζεται στην εικόνα 3.35.



Εικόνα 3.35 Συνδεσμολογία του Arduino Nano Every με 2 σερβοκινητήρες MG995R, μία μπαταριοθήκη παροχής 9V και δύο παροχής 6V

Η μπαταριοθήκη των 9V συνδέεται απευθείας στους ακροδέκτες “VIN-GND” του NANO EVERY και έτσι η ηλεκτρονική πλακέτα τροφοδοτείται απρόσκοπτα με τάση 5V καθώς διαθέτει εσωτερικό ρυθμιστή τάσης στα 5V, με έκαστο σερβοκινητήρα να τροφοδοτείται με τάση 6V από μπαταριοθήκες που φέρουν από 4 μπαταρίες των 1,5V εκάστη.

Ο αντίστοιχος προγραμματιστικός κώδικας για έλεγχο της ορθής λειτουργίας της ανωτέρω συνδεσμολογίας παρατίθεται ακολούθως:

```
/*
ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΗΣ ΘΕΣΗΣ/ΚΛΙΣΗΣ ΤΩΝ 2 ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ.
ΣΤΗΝ ΣΥΝΕΧΕΙΑ ΣΤΑΔΙΑΚΗ ΟΛΙΣΘΗΣΗ ΤΟΥ ΣΕΡΒΟΚΙΝΗΤΗΡΑ 1 (SIGNAL LINE 5 PWM)
ΑΡΧΙΚΑ ΑΠΟ 0 ΕΩΣ 180 ΜΟΙΡΕΣ ΚΑΙ ΕΠΕΙΤΑ ΕΠΑΝΑΦΟΡΑ ΣΤΗΝ ΑΡΧΙΚΗ ΤΟΥ ΘΕΣΗ
ΜΕ ΟΛΙΣΘΗΣΗ ΑΠΟ 180 ΕΩΣ 0 ΜΟΙΡΕΣ.
Ο ΣΕΡΒΟΚΙΝΗΤΗΡΑΣ 2 (SIGNAL LINE 6 PWM) ΟΛΙΣΘΑΙΝΕΙ ΜΟΝΟ ΣΤΙΣ ΑΚΡΑΙΕΣ ΘΕΣΕΙΣ,
ΔΕΝ ΕΚΤΕΛΕΙ ΣΤΑΔΙΑΚΗ/ΒΗΜΑΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΤΟΥ ΑΞΟΝΑ ΜΕΤΑΞΥ ΤΟΥ ΕΥΡΟΥΣ 0-180
ΚΑΙ 180-0 ΜΟΙΡΩΝ.
*/

#include <Servo.h>      // Add the Servo library
Servo myServo1;        // Define 'myServo1' as Servo
Servo myServo2;        // Define 'myServo2' as Servo2

int angleV1 = 0;       // Initialize a variable to store Servo1 position/angle
int angleV2 = 0;       // Initialize a variable to store Servo2 position/angle

void setup() {
  myServo1.write(180);  // Initialize Servo1 position/angle to max-point (180 degrees)
  myServo2.write(0);    // Initialize Servo2 position/angle to min-point (0 degrees)
  myServo1.attach(5);   // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6);   // Attaches Servo2 Signal Line to pin 6 (PWM)
}

void loop() {
  // for the first servo: myServo1
  for (angleV1 = 0; angleV1 <= 180; angleV1 +=1) {
    myServo1.write(angleV1); // Use the variable to rotate the Servo Motor 1
                                // and change its position according to the variable value
                                // from 0 to 180 degrees
    delay(20); // waits 15ms for the servo to reach the corresponding angle
  }

  for (angleV1 = 180; angleV1 >= 0; angleV1 -=1) {
    myServo1.write(angleV1); // Use the variable to rotate the Servo Motor 1
                                // and change its position according to the variable value
                                // from 180 to 0 degrees
    delay(20); // waits 15ms for the servo to reach the corresponding angle
  }

  // for the second servo: myServo2
  for (angleV2 = 0; angleV2 <= 180; angleV2 +=1) {
    if (angleV2 = 180); { // If angleV2 = 180 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
                                // and change its position according to the variable value
                                // from 0 to 180 degrees
    }
    delay(2500); // waits 15ms for the servo to reach the corresponding angle
  }

  for (angleV2 = 180; angleV2 >= 0; angleV2 -=1) {
    if (angleV2 = 0); { // If angleV2 = 0 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
                                // and change its position according to the variable value
                                // from 180 to 0 degrees
    }
    delay(1000); // waits 15ms for the servo to reach the corresponding angle
  }
}
}
```

ΚΩΔΙΚΑΣ_5 Λειτουργία 2 σερβοκινητήρων με χρήση προγραμματιστικού κώδικα

Με χρήση του κώδικα Νο5 επιτυγχάνεται ο έλεγχος 2 ψηφιακών σερβοκινητήρων δυνατότητας περιστροφής 180° χωρίς το κύκλωμα να διαθέτει μονάδες παροχής εντολών εισόδου (π.χ. μοχλός).

Αρχικά γίνεται συμπερίληψη της σχετικής βιβλιοθήκης απαραίτητης για επικοινωνία του μικροελεγκτή Arduino με τους σερβοκινητήρες με χρήση της εντολής “*include<Servo.h>*” και προσδιορισμός των σερβοκινητήρων από αυτόν με χρήση της “*Servo myServo;*”.

Έπειτα μέσω της εντολής “*myServo.write();*” μέσα στην συνάρτηση “*void setup()*” επιτυγχάνεται η επαναφορά του άξονα των 2 σερβοκινητήρων σε συγκεκριμένη θέση/κλίση με κάθε νέα ενεργοποίηση του κυκλώματος μετά από παύση της λειτουργίας του – συγκεκριμένα ο Servo1 θα «αρχικοποιείται» με μετάβαση στην θέση των 180° ενώ ο Servo2 στην θέση των 0°.

Με την βοήθεια της “*myServo.attach;*” καθορίζεται η διέγερση (*signal line*) εκάστου ακροδέκτη με τον αντίστοιχο επιθυμητό ακροδέκτη του Arduino NANO EVERY.

Τέλος, στην συνάρτηση “*void loop()*” περιλαμβάνεται το κομμάτι κώδικα που είναι υπεύθυνο για την ολίσθηση του άξονα των 2 σερβοκινητήρων σε ένα εύρος τιμών 0-180°, αποτελούμενο από 4 βρόχους επανάληψης “*for*”, 4 εντολές καθυστέρησης “*delay*”, 4 εντολές “*myServo.write();*” (2 για κάθε σερβοκινητήρα) και 2 συνθήκες “*if*”.

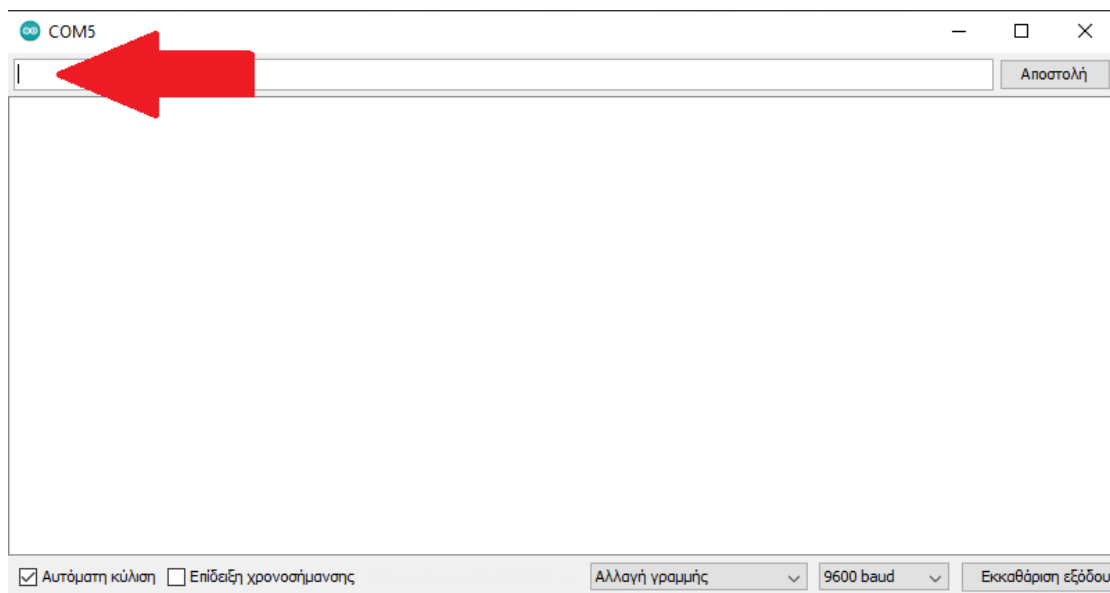
Στην συνάρτηση αυτή, για τον 1° σερβοκινητήρα (*Servo1*) αναπτύσσεται ένας βρόχος επανάληψης “*for*” στον οποίο περικλείεται η εντολή “*myServo1.write(angleV1);*” η οποία είναι υπεύθυνη για την ολίσθηση του άξονα του σερβοκινητήρα. Με αύξηση της τιμής της μεταβλητής “*angleV1*”, ξεκινώντας από την τιμή 0 και καταλήγοντας στην τιμή 180 μοιρών λόγω εκτέλεσης του πρώτου βρόχου επανάληψης, η εντολή “*myServo1.write(angleV1);*” θα προκαλεί ολίσθηση στον άξονα του σερβοκινητήρα (*Servo1*) για κάθε μια τιμή σε κάθε επανάληψη ώσπου να έχει μετακινηθεί σε όλες τις θέσεις από 0° έως 180°. Ακριβώς με τον ίδιο τρόπο λειτουργεί και ο 2^{ος} βρόχος αλλά για τιμές ολίσθησης από 180-0°.

Για τον 2° σερβοκινητήρα (*Servo2*), λόγω ύπαρξης της συνθήκης “*if*” στον 3° και 4° βρόχο επανάληψης, ο άξονάς του θα ολισθαίνει μόνο στις θέσεις που αντιστοιχούν στις τιμές 0° ή 180° και σε καμία άλλη του εύρους 0-180°.

Παροχή εντολών με χρήση του Serial Monitor

Μια επιπλέον δυνατότητα του Serial Monitor, εκτός από την μεταφορά μηνυμάτων μεταξύ Η/Υ και μικροελεγκτή Arduino, είναι η παροχή εντολών μέσω της δικής του «**Γραμμής Εντολών**», για παράδειγμα από έναν χρήστη μέσω ενός πληκτρολογίου.

Στο παράδειγμα που ακολουθεί, αντί για ανάπτυξη κατάλληλου προγραμματιστικού κώδικα, γίνεται χρήση της γραμμής εντολών (εικ. 3.36) του Serial Monitor του Arduino IDE για έλεγχο ορθής λειτουργίας ενός σερβοκινητήρα με εντολές παρεχόμενες από τον χρήστη μέσω του πληκτρολογίου του υπολογιστή.



Εικόνα 3.36 Χρήση της Γραμμής εντολών του Serial Monitor για παροχή εντολών εισόδου από το πληκτρολόγιο

```

/*
ΕΛΕΓΧΟΣ ΕΝΟΣ MG995R ΣΕΡΒΟΚΙΝΗΤΗΡΑ ΜΕ ΧΡΗΣΗ ΠΛΗΚΤΡΟΛΟΓΙΟΥ ΓΙΑ ΠΑΡΟΧΗ ΕΝΤΟΛΩΝ ΕΙΣΟΔΟΥ
ΜΕΣΩ ΤΗΣ ΔΥΝΑΤΟΤΗΤΑΣ ΠΟΥ ΠΡΟΣΦΕΡΕΙ ΤΟ SERIAL MONITOR.
*/

#include <Servo.h>
Servo myservo;
int angleV1 = 0;    // Initialize a variable to store Servo position/angle

void setup() {
  Serial.begin(9600); // Define baud rate for serial communication
  while (!Serial);
  Serial.println("-----");
  Serial.println("Circuit operation test");
  delay(1000);
  Serial.println("-----");
  myservo.attach(5); // Attaches Servo Signal Line to pin 5 (PWM)
  Serial.println("calibrating Servo Motor 1, lever position/angle at 0-->90-->180-->0...");
  for(angleV1 = 0; angleV1 <= 180; angleV1 += 1)
    myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);
  myservo.write(0);
  delay(1000);
  Serial.println("Servo Motor 1 calibrated");
  Serial.println("-----");
  Serial.println("<<Circuit is ready>>");
  Serial.println("Command input available, press any number from range 1-180 to perform action");
  Serial.println("-----");
}

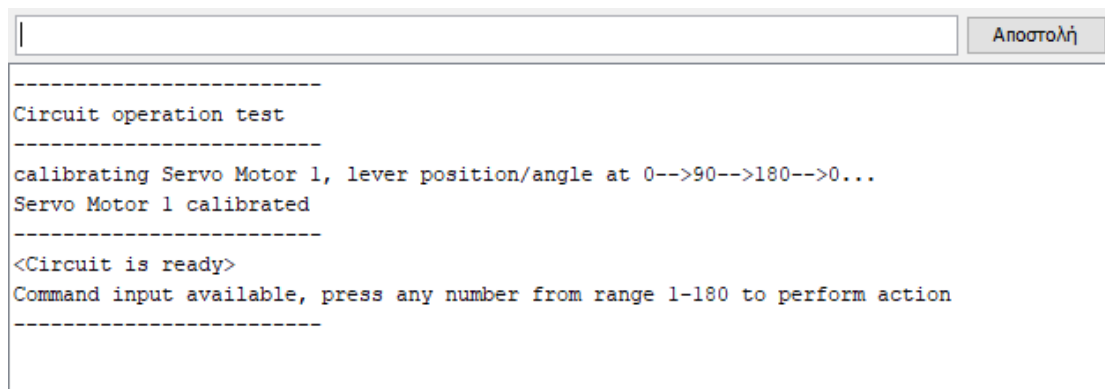
void loop() {
  for(angleV1 = 0; angleV1 <= 180; angleV1 += 1)
  if (Serial.available()) { // If serial monitor is available, execute the below command
    int state = Serial.parseInt(); // Sets "state" to number that has been sent to serial monitor
    // The "parseInt" reads every number sent to the serial monitor.
    // It can read only numbers, not letters etc.

    if (state >= 1 && state <= 180) {
      Serial.print(">");
      Serial.println(state);
      Serial.print("turning servo to ");
      Serial.print(state);
      Serial.println(" degrees");
      myservo.write(state);
    }
  }
}
}

```

ΚΩΔΙΚΑΣ_6 Κώδικας ελέγχου σερβοκινητήρα από το πληκτρολόγιο
με χρήση της Γραμμής Εντολών του Serial Monitor

Εκτελώντας τον κώδικα με χρήση του Serial Monitor, γίνεται αμέσως αντιληπτός ο τρόπος λειτουργίας του κυκλώματος. Αρχικά πραγματοποιείται βαθμονόμηση (*calibration*) του σερβοκινητήρα με σταδιακή ολίσθηση στις μοίρες 0→90→180 και κατάληξη πάλι στις 0°.

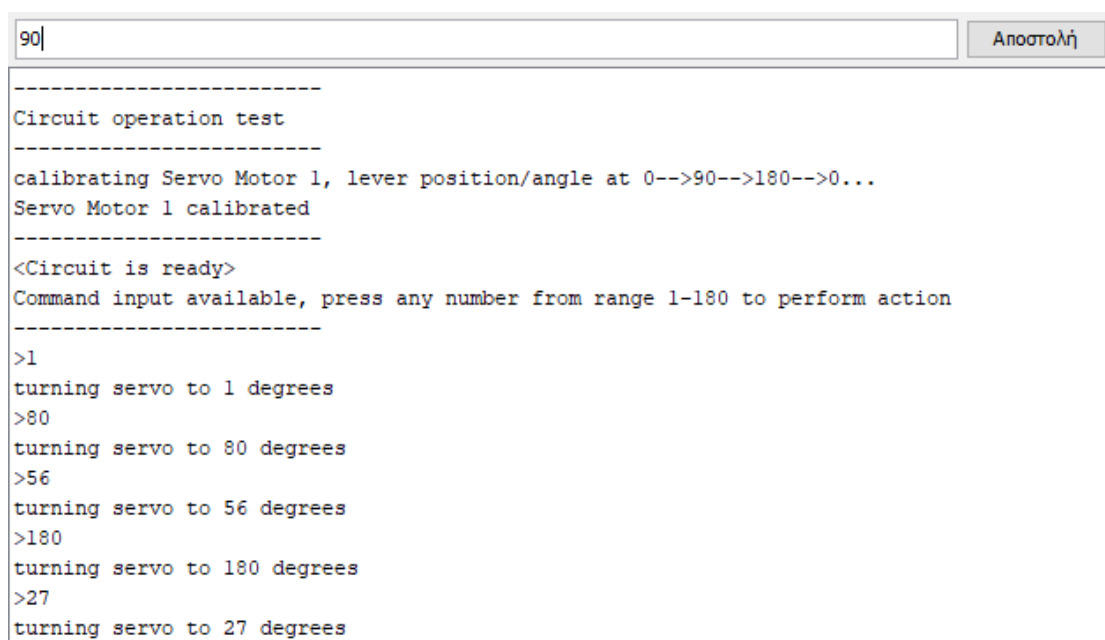


```
-----
Circuit operation test
-----
calibrating Servo Motor 1, lever position/angle at 0-->90-->180-->0...
Servo Motor 1 calibrated
-----
<Circuit is ready>
Command input available, press any number from range 1-180 to perform action
-----
```

Εικόνα 3.37 Αρχική κατάσταση και αναμονή του κυκλώματος για παροχή εντολών από το πληκτρολόγιο μέσω της Γραμμής εντολών του Serial Monitor

Πληκτρολογώντας έναν θετικό ακέραιο αριθμό –που εκλαμβάνεται από το κύκλωμα ως τιμή σε μοίρες– και πατώντας την επιλογή «Αποστολή» ή απλά το πλήκτρο «Enter», παρουσιάζεται στο Serial Monitor η δοθείσα τιμή και ακριβώς από κάτω ένα μήνυμα που δηλώνει την θέση ολίσθησης του σερβοκινητήρα στην εν λόγω τιμή.

Δίνοντας συνεχώς τιμές, ο σερβοκινητήρας θα ολισθαίνει αντιστοίχως – με παροχή μη θετικών ακέραιων τιμών το κύκλωμα δεν θα λειτουργεί σωστά καθώς δεν έχει ρυθμιστεί αναλόγως.



```
-----
Circuit operation test
-----
calibrating Servo Motor 1, lever position/angle at 0-->90-->180-->0...
Servo Motor 1 calibrated
-----
<Circuit is ready>
Command input available, press any number from range 1-180 to perform action
-----
>1
turning servo to 1 degrees
>80
turning servo to 80 degrees
>56
turning servo to 56 degrees
>180
turning servo to 180 degrees
>27
turning servo to 27 degrees
```

Εικόνα 3.38 Χρήση της Γραμμής εντολών για διάφορες θετικές ακέραιες τιμές

Ως συνέχεια του προηγούμενου παραδείγματος, γίνεται χρήση της γραμμής εντολών του Serial Monitor για έλεγχο ορθής λειτουργίας δύο σερβοκινητήρων με εντολές παρεχόμενες από τον χρήστη μέσω του πληκτρολογίου του υπολογιστή, με τον σχετικό κώδικα να παρατίθεται κατωτέρω:

```

/*
ΕΛΕΓΧΟΣ ΔΥΟ MG995R ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ ΜΕ ΧΡΗΣΗ ΠΛΗΚΤΡΟΛΟΓΙΟΥ ΓΙΑ ΠΑΡΟΧΗ ΕΝΤΟΛΩΝ ΕΙΣΟΔΟΥ
ΜΕΣΩ ΤΗΣ ΔΥΝΑΤΟΤΗΤΑΣ ΠΟΥ ΠΡΟΣΦΕΡΕΙ ΤΟ SERIAL MONITOR.
*/

#include <Servo.h>      // Add the Servo library
Servo myServo1;        // Define 'myServo1' as Servo1
Servo myServo2;        // Define 'myServo2' as Servo2
int angleV = 0;
int angleV1 = 0;       // Initialize variable to store Servo1 position/angle
int angleV2 = 0;       // Initialize variable to store Servo2 position/angle

void setup() {
  Serial.begin(9600);   // Define baud rate for serial communication
  while (!Serial);
  Serial.println("-----");
  Serial.println("Circuit operation test");
  delay(1000);
  Serial.println("-----");

  myServo1.attach(5);   // Attaches Servo1 Signal Line to pin 5 (PWM)
  Serial.println("calibrating Servo Motor 1, lever position/angle at 0-->90-->180-->0...");
  for(angleV1 = 0; angleV1 <= 180; angleV1 += 1)
    myServo1.write(0);
    delay(1000);
  myServo1.write(90);
  delay(1000);
  myServo1.write(180);
  delay(1000);
  myServo1.write(0);
  delay(1000);
  Serial.println("Servo Motor 1 calibrated");
  Serial.println("-----");

  myServo2.attach(6);   // Attaches Servo2 Signal Line to pin 6 (PWM)
  Serial.println("calibrating Servo Motor 2, lever position/angle at 0-->90-->180-->0...");
  for(angleV2 = 0; angleV2 <= 180; angleV2 += 1)
    myServo2.write(0);
    delay(1000);
  myServo2.write(90);
  delay(1000);
  myServo2.write(180);
  delay(1000);
  myServo2.write(0);
  delay(1000);
  Serial.println("Servo Motor 2 calibrated");
  Serial.println("-----");

  Serial.println("Arduino is ready");
  Serial.println("<Circuit is ready>");
  Serial.println("Command input available, press any number from range 1-180 to perform action on the 2 Servos");
  Serial.println("-----");
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

void loop() {
  for(angleV = 0; angleV <= 180; angleV += 1)
  if (Serial.available()) { // If serial monitor is available, execute the below command
    int state = Serial.parseInt(); // Sets "state" to number that has been sent to serial monitor
    // The "parseInt" reads every number sent to the serial monitor.
    // It can read only numbers, not letters etc.

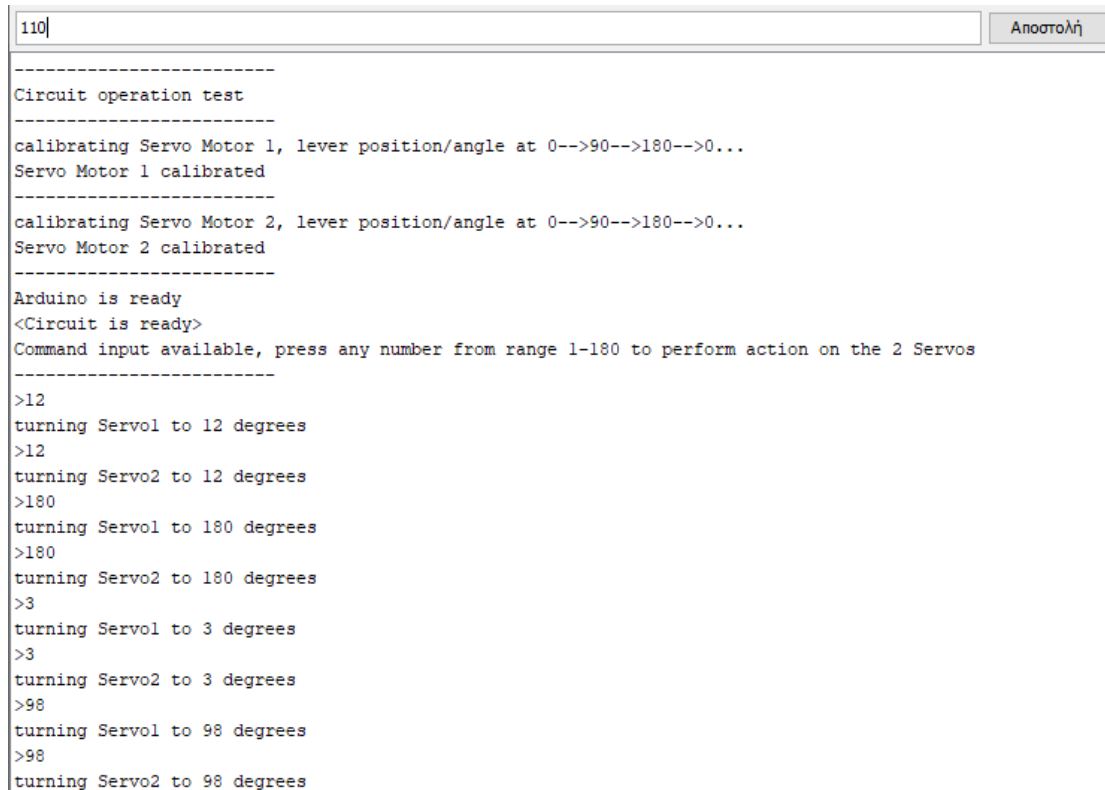
    int state1 = state;
    int state2 = state;
    if (state >= 1 && state <= 180) {
      Serial.print(">");
      Serial.println(state1);
      Serial.print("turning Servo1 to ");
      Serial.print(state1);
      Serial.println(" degrees");
      myServo1.write(state1);

      Serial.print(">");
      Serial.println(state2);
      Serial.print("turning Servo2 to ");
      Serial.print(state2);
      Serial.println(" degrees");
      myServo2.write(state2);
    }
  }
}

```

ΚΩΔΙΚΑΣ_7 Κώδικας ελέγχου 2 σερβοκινητήρων από το πληκτρολόγιο με χρήση της Γραμμής Εντολών του Serial Monitor

Ο προγραμματιστικός κώδικας Νο7 καθιστά τους 2 σερβοκινητήρες ελεγχόμενους από έναν χρήστη μέσω ενός πληκτρολογίου, όταν οι εντολές παρέχονται με χρήση του Serial Monitor. Με πληκτρολόγηση ενός θετικού ακεραίου οι σερβοκινητήρες λαμβάνουν και οι δύο την τιμή αυτή και ολισθαίνουν αντιστοίχως.

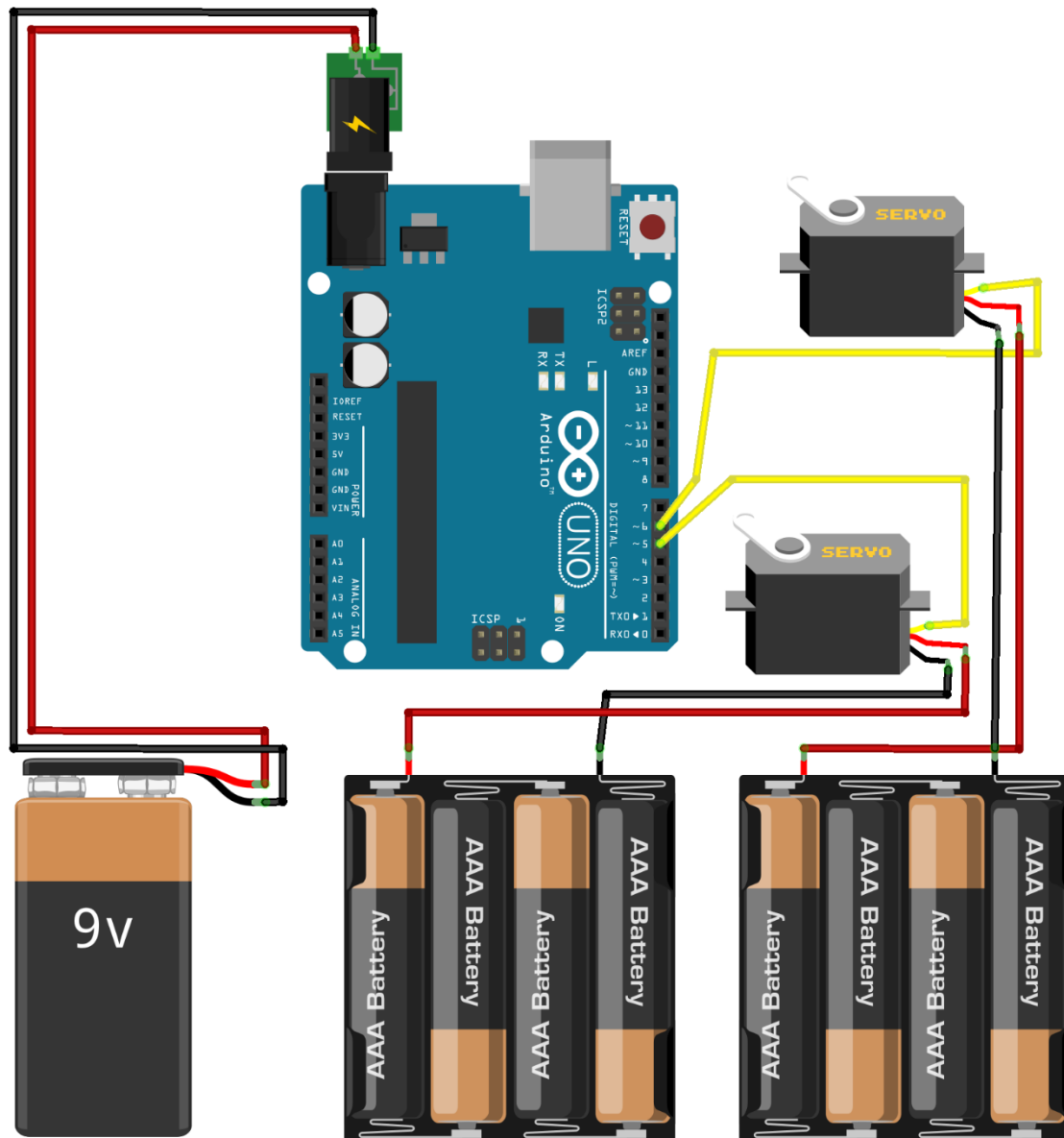


```
110| Αποστολή
-----
Circuit operation test
-----
calibrating Servo Motor 1, lever position/angle at 0-->90-->180-->0...
Servo Motor 1 calibrated
-----
calibrating Servo Motor 2, lever position/angle at 0-->90-->180-->0...
Servo Motor 2 calibrated
-----
Arduino is ready
<Circuit is ready>
Command input available, press any number from range 1-180 to perform action on the 2 Servos
-----
>12
turning Servo1 to 12 degrees
>12
turning Servo2 to 12 degrees
>180
turning Servo1 to 180 degrees
>180
turning Servo2 to 180 degrees
>3
turning Servo1 to 3 degrees
>3
turning Servo2 to 3 degrees
>98
turning Servo1 to 98 degrees
>98
turning Servo2 to 98 degrees
```

Εικόνα 3.39 Χρήση της Γραμμής εντολών για ταυτόχρονο έλεγχο 2 σερβοκινητήρων μέσω πληκτρολογίου με χρήση του Serial Monitor

Εναλλακτική δοκιμαστική συνδεσμολογία

Καθώς οι σερβοκινητήρες αποτελούν μέρος του Κυκλώματος RS, εξετάζεται επίσης η συνδεσμολογία μεταξύ μικροελεγκτή Arduino και σερβοκινητήρων με χρήση του μοντέλου Arduino UNO R3 (εικ. 3.40).



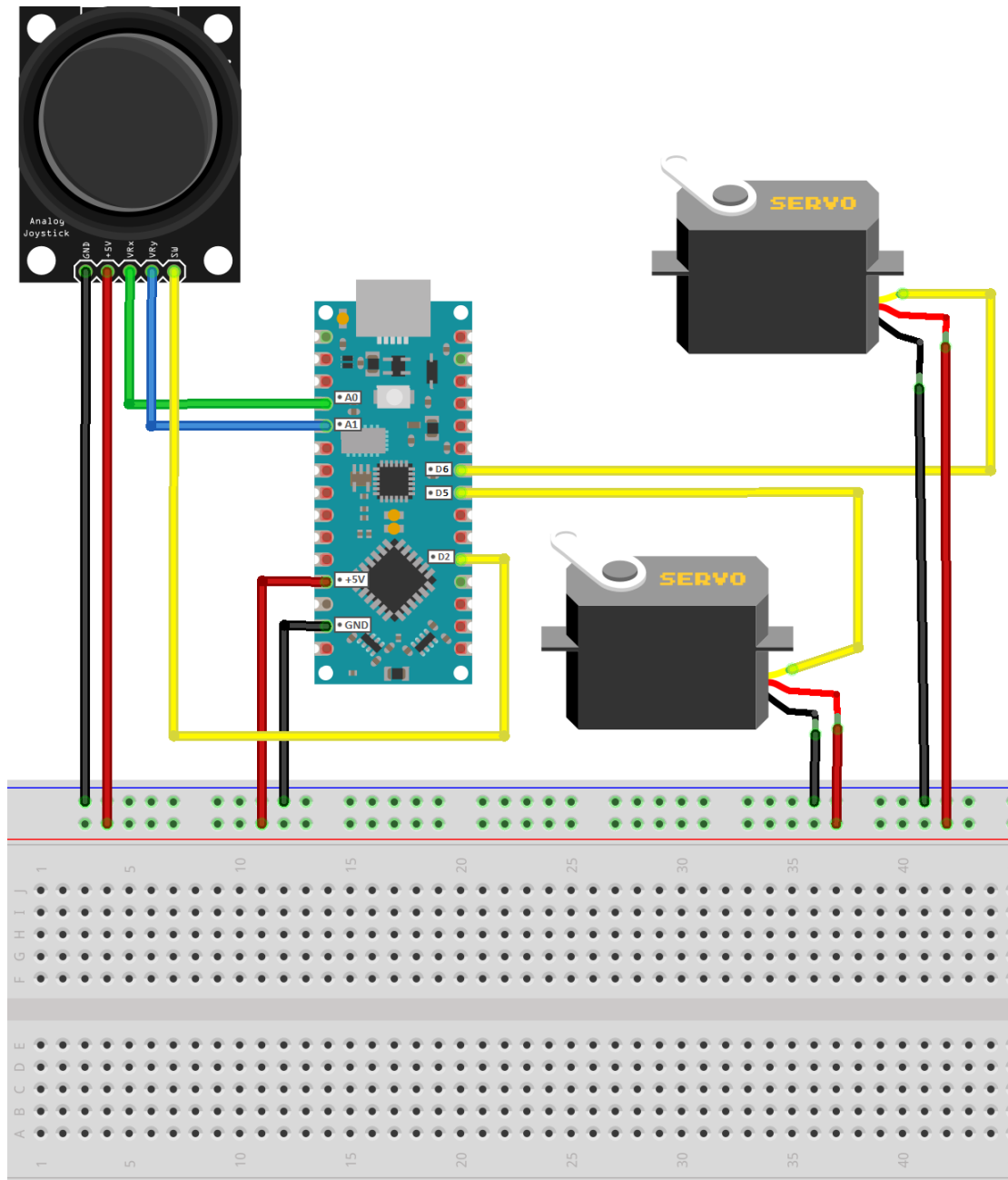
Εικόνα 3.40 Συνδεσμολογία του Arduino UNO R3 με 2 σερβοκινητήρες MG995R, μία μπαταριοθήκη τροφοδοσίας 9V και δύο τροφοδοσίες 6V

Ο κώδικας που χρησιμοποιείται για την λειτουργία του ανωτέρω κυκλώματος είναι ίδιος με τον κώδικα του κυκλώματος της εικόνας 3.35 (ΚΩΔΙΚΑΣ_5) που περιλαμβάνει τον μικροελεγκτή NANO EVERY.

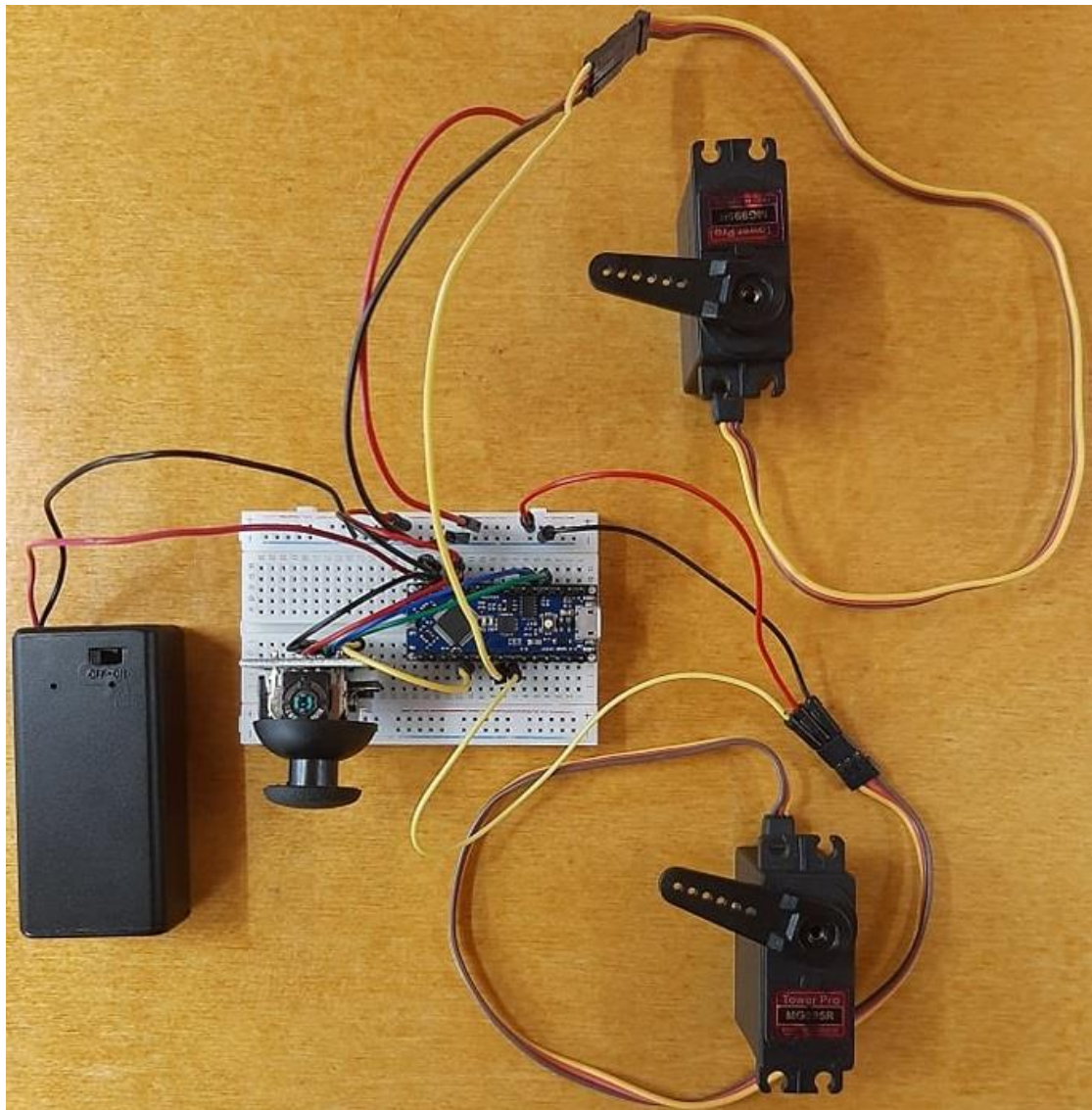
Λόγω αλλαγής του μοντέλου μικροελεγκτή από NANO σε UNO, η μόνη ενέργεια που απαιτείται ώστε να λειτουργεί σωστά το κύκλωμα της εικόνας 3.40 είναι η ρύθμιση των παραμέτρων του περιβάλλοντος *Arduino IDE* ώστε να δύναται ο Η/Υ να επικοινωνεί επιτυχώς με το Arduino UNO R3.

Τα βήματα ρύθμισης των σχετικών παραμέτρων εδώ είναι τα ίδια με τα βήματα που περιγράφηκαν αναλυτικά στην υποενότητα 3.1.1.1.

Με το πέρας του ελέγχου για ορθή λειτουργία και συμπεριφορά των δύο ψηφιακών σερβοκινητήρων, προσθέτουμε στο κύκλωμα της εικόνας 3.34 έναν αναλογικό μοχλό χρησιμοποιώντας την συνδεσμολογία που αναφέρεται στον πίνακα 3_1 ή στον πίνακα 3_2 της υποενότητας 3.1.1 και προκύπτει το ακόλουθο κύκλωμα:



Εικόνα 3.41 Σχεδιασμός ενός Συστήματος Ηλεκτρονικού Ενσύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές ως ενσύρματο σύστημα απόκρισης εισόδου/εξόδου χωρίς πηγές τροφοδοσίας, υλοποιημένο σε περιβάλλον προσομοίωσης



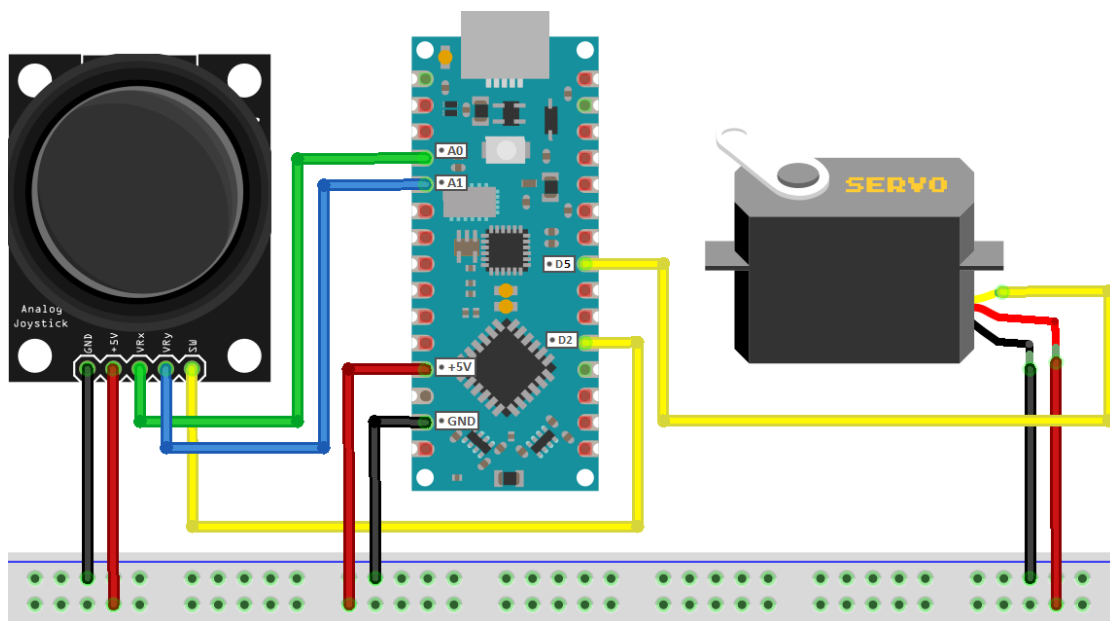
***Εικόνα 3.42** Σχεδιασμός ενός Συστήματος Ηλεκτρονικού Ενσύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές ως ενσύρματο σύστημα απόκρισης εισόδου/εξόδου χωρίς πηγές τροφοδοσίας, υλοποιημένο σε επαγγελματικό εργαστήριο*

Στην εικόνα 3.41 παρουσιάζεται ο τρόπος συνδεσμολογίας του εν λόγω ενσύρματου συστήματος ελέγχου απόκρισης μιας εισόδου εντολών από έναν αναλογικό μοχλό με δύο εξόδους από δύο ψηφιακούς σερβοκινητήρες.

Ως ηλεκτρικές συσκευές που μπορούν να χρησιμοποιηθούν για την περιστροφή αντικειμένων ή εξαρτημάτων (π.χ. ρομποτικός βραχίονας, μεταλλικός περιστροφικός άξονας, μηχανικά γρανάζια κλπ) με ακρίβεια, οι δύο σερβοκινητήρες δύνανται να συνδυάζονται με διάφορα υλικά και εξαρτήματα για την πραγματοποίηση πληθώρας εφαρμογών.

Στην προαναφερόμενη εικόνα, αρχικά δεν περιλαμβάνεται συνδεσμολογία εξωτερικών πηγών τροφοδοσίας για παροχή ενέργειες στο κύκλωμα, καθώς αρχικά θα γίνει χρήση του καλωδίου USB για διασύνδεση με Η/Υ και αξιοποίηση του περιβάλλοντος Arduino IDE για ανάπτυξη κώδικα προγραμματισμού και μελέτη της συμπεριφοράς των δύο ψηφιακών σερβοκινητήρων βάσει εντολών εισόδου παρεχόμενων από τον αναλογικό μοχλό.

Η υλοποίηση πραγματοποιείται σταδιακά, σε πρώτο στάδιο με διασύνδεση μοχλού, μικροελεγκτή NANO και ενός σερβοκινητήρα MG995R με δυνατότητα περιστροφής 180°, για την δημιουργία μιας απλής μορφής κυκλώματος απόκρισης μιας εξόδου από μία είσοδο.



Εικόνα 3.43 Σχεδιασμός ενός συστήματος απόκρισης μιας εξόδου από μια είσοδο

Προγραμματίζοντας κατάλληλα τον μικροελεγκτή NANO, μελετάται η συμπεριφορά του σερβοκινητήρα όταν παρέχονται στην ηλεκτρονική πλακέτα Arduino NANO εντολές από τον μοχλό, με τον σχετικό κώδικα του ανωτέρω κυκλώματος να παρατίθεται ακολούθως.

```
#include <Servo.h>      // Add the servo library
Servo myServo;         // Define 'myServo' as Servo

int angleV1 = 0;       // Initialize variable to store Servo position/angle

const int joystick_VRX = A0; // Attaches joystick X-axis to Arduino pin A0
const int joystick_VRY = A1; // Attaches joystick Y-axis to Arduino pin A1

void setup() {
  Serial.begin(9600);      // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 2'

  myServo.write(180);      // Initialize Servo position/angle to max-point (180 degrees)
  myServo.attach(5);       // Attaches Servo Signal Line to pin 5 (PWM)
}

void loop() {
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2); // Read the digital input of the Switch

  float X_Volt = ( ( X_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
  float Y_Volt = ( ( Y_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage

  Serial.print("X_Voltage = ");
  Serial.print(X_Volt);
  Serial.print("\t");
  Serial.print("Y_Voltage = ");
  Serial.print(Y_Volt);
  Serial.print("\t");
  Serial.print("SW_State = ");
  Serial.println(SW_State);
  delay(100);

  // Set the analog/mechanical input of X-axis as degrees of range 0-180
  angleV1 = map(X_ADC_Value, 0, 1023, 0, 180);
  myServo.write(angleV1);

  delay(100); // waits 100ms for the servo to reach the corresponding angle
}
```

ΚΩΔΙΚΑΣ_8 Κώδικας για έλεγχο ενός σερβοκινητήρα από έναν αναλογικό μοχλό

Στη συνέχεια προστίθεται στο κύκλωμα ένας ακόμη σερβοκινητήρας ίδιου μοντέλου και ο ανωτέρω κώδικας τροποποιείται με τρόπο τέτοιον ώστε το κύκλωμα που προκύπτει να αποτελεί ένα κύκλωμα συστήματος απόκρισης μιας εισόδου με δύο εξόδους, ως ορίζεται σχηματικά και σχεδιαστικά στην εικόνα 3.41.

Ο εν λόγω τροποποιημένος κώδικας παρουσιάζεται κατωτέρω.

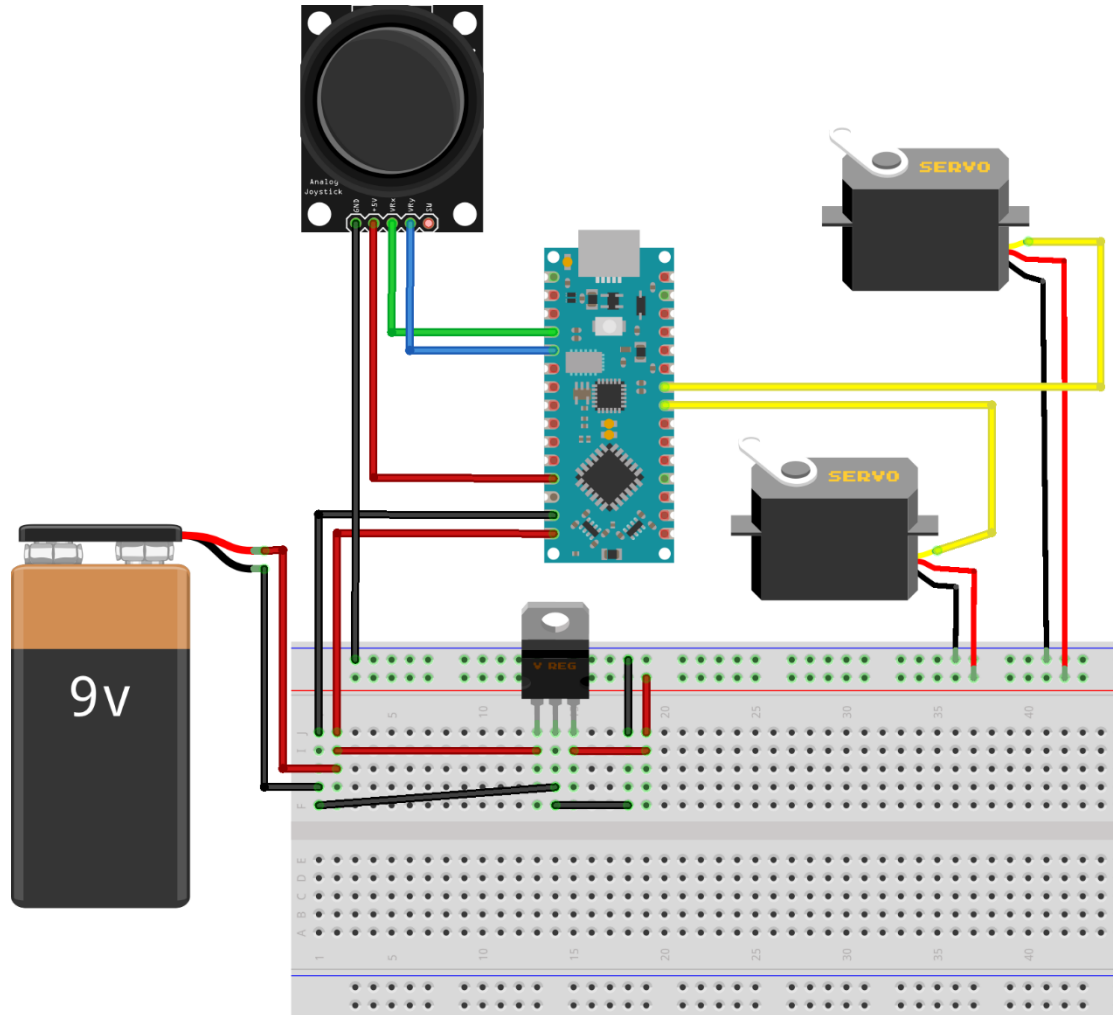
```
#include <Servo.h>          // Add the servo library
Servo myServo1;           // Define 'myServo1' as Servo1
Servo myServo2;           // Define 'myServo2' as Servo2
int angleV1 = 0;          // Initialize variable to store Servo1 position/angle
int angleV2 = 0;          // Initialize variable to store Servo2 position/angle
const int joystick_VRX = A0; // Attaches joystick X-axis to Arduino pin A0
const int joystick_VRY = A1; // Attaches joystick Y-axis to Arduino pin A1
void setup() {
  Serial.begin(9600);      // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 2'
  myServo1.write(180);     // Initialize Servo1 position/angle to max-point (180 degrees)
  myServo2.write(0);       // Initialize Servo2 position/angle to min-point (0 degrees)
  myServo1.attach(5);      // attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6);      // attaches Servo2 Signal Line to pin 6 (PWM)
}
void loop() {
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2);           // Read the digital input of the Switch
  float X_Volt = ( ( X_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
  float Y_Volt = ( ( Y_ADC_Value * 5.0 ) / 1023 ); // Convert digital value to Voltage
  Serial.print("X_Voltage = ");
  Serial.print(X_Volt);
  Serial.print("\t");
  Serial.print("Y_Voltage = ");
  Serial.print(Y_Volt);
  Serial.print("\t");
  Serial.print("SW_State = ");
  Serial.println(SW_State);
  delay(100);
  // Set the analog/mechanical input of X-axis as degrees of range 0-180
  angleV1 = map(X_ADC_Value, 0, 1023, 0, 180);
  myServo1.write(angleV1);
  // Set the analog/mechanical input of Y-axis as degrees of range 0-180
  angleV2 = map(Y_ADC_Value, 0, 1023, 0, 180);
  myServo2.write(angleV2);
  delay(100); // waits 100ms for the servo to reach the corresponding angle
}
```

ΚΩΔΙΚΑΣ_9 Κώδικας για έλεγχο δύο σερβοκινητήρων από έναν αναλογικό μοχλό

Λόγω του ότι παρέχεται αρκετά μεγάλη ευελιξία στον τρόπο διασύνδεσης των ηλεκτρονικών πλακετών Arduino με περιφερειακές μονάδες λόγω του ιδιαίτερου σχεδιασμού τους, η χρήση ή μη μιας πλακέτας δοκιμών «*breadboard*» καθιστά το κύκλωμα ικανό να λειτουργεί εξίσου αποτελεσματικά όταν σε αυτό προστίθενται νέα ηλεκτρονικά στοιχεία.

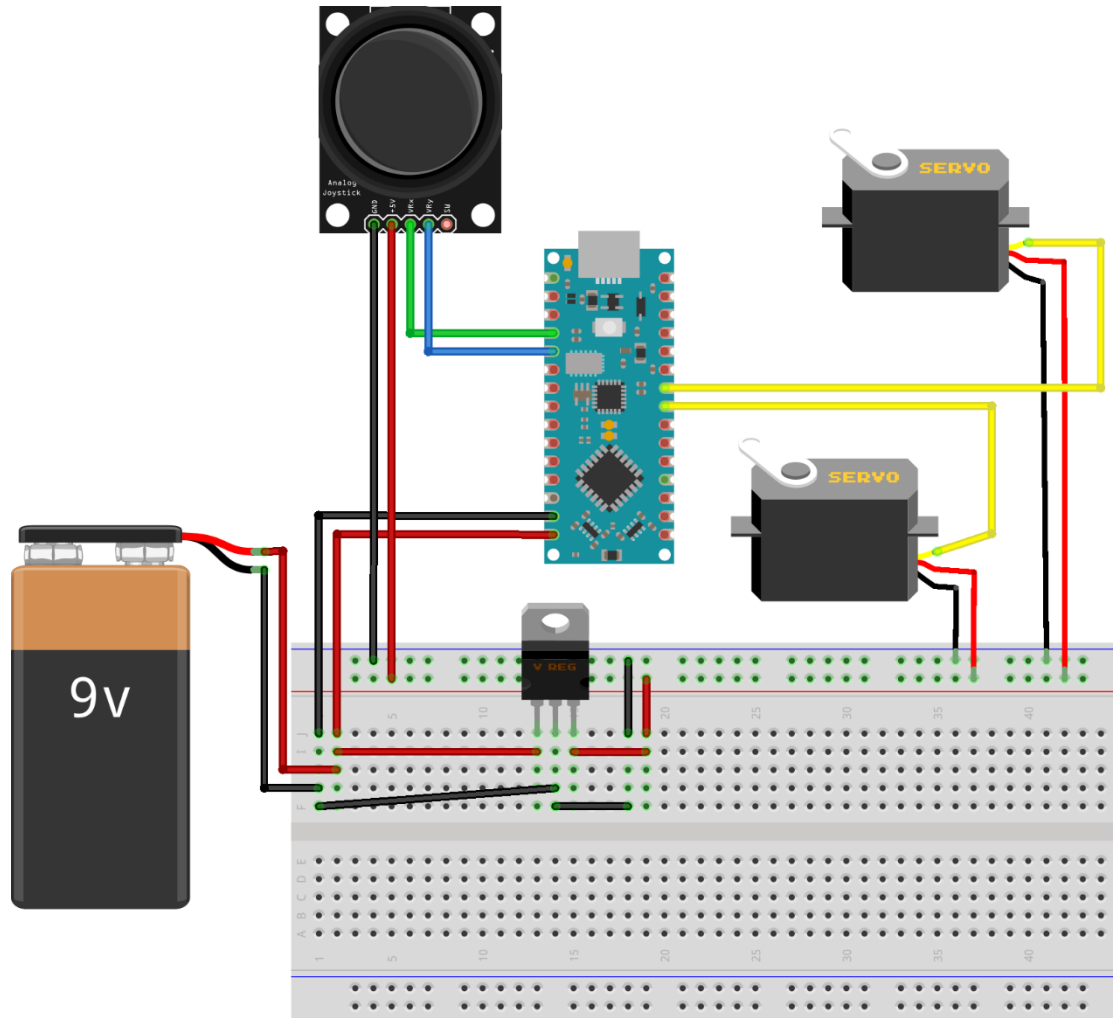
Η ευελιξία αυτή επεκτείνεται όπως είναι φυσικό και στον τρόπο με τον οποίο ένας μικροελεγκτής Arduino δέχεται και προσφέρει ενέργεια, όντας μέρος ενός κυκλώματος, με μερικές μεθόδους τροφοδοσίας για το κύκλωμα της εικόνας 3.41 να παρουσιάζονται μέσω τεσσάρων χαρακτηριστικών παραδειγμάτων στις εικόνες που ακολουθούν.

1. Τροφοδοσία του Σ.Η.Ε.Ε με χρήση μπαταρίας 9V και ρυθμιστή τάσης στα 5V



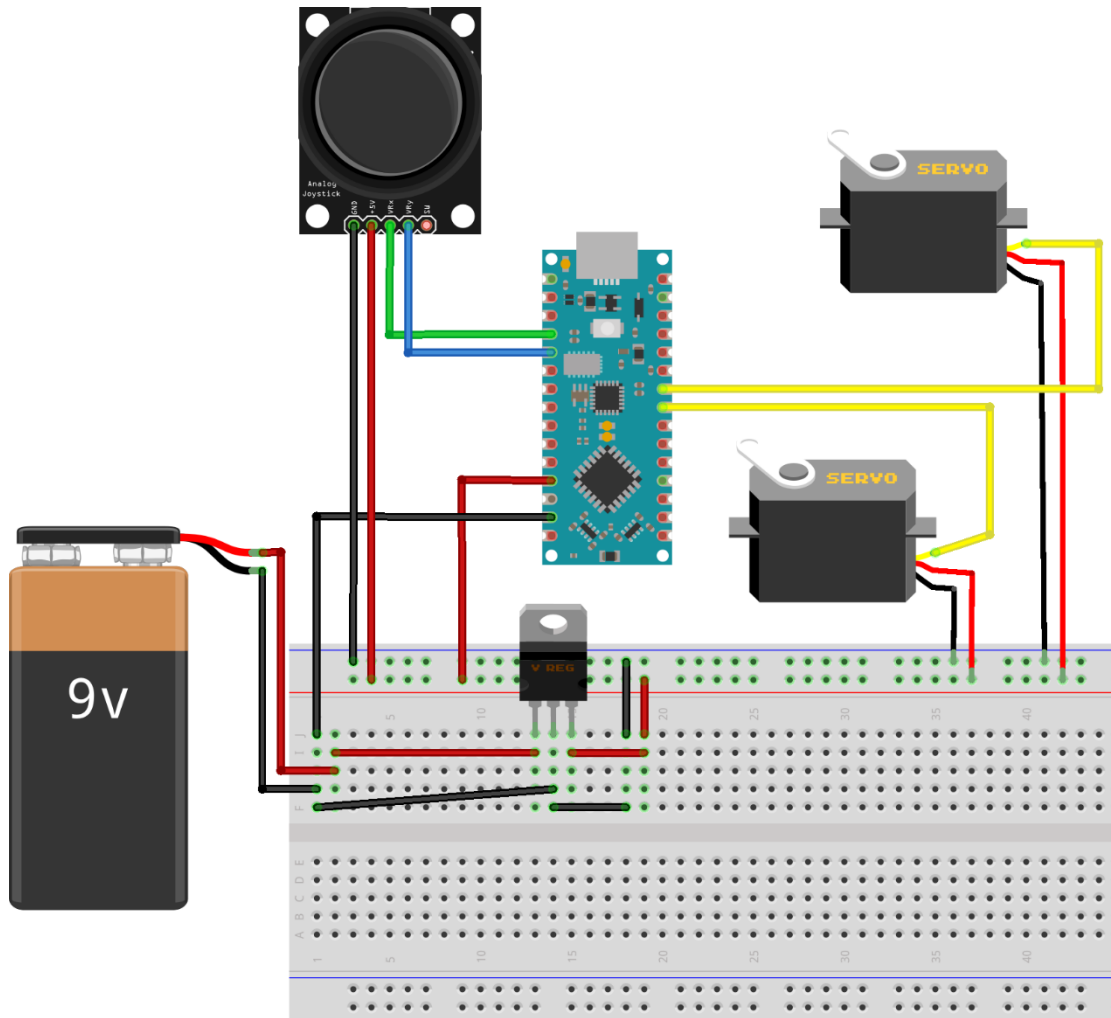
Εικόνα 3.44 Τροφοδοσία του συστήματος ηλεκτρονικού ενσύρματου ελεγκτή (Σ.Η.Ε.Ε.) με χρήση μπαταρίας των 9V, για απευθείας σύνδεση και τροφοδοσία του Arduino μέσω των pins VIN-GND, απευθείας διασύνδεση μεταξύ ακροδεκτών θετικού δυναμικού αναλογικού μοχλού & Arduino (pins VCC & 5V) για τροφοδοσία μοχλού με τάση 5V από το Arduino και ρυθμιστή τάσης στα 5V για προσφορά 5V στους δύο σερβοκινητήρες

2. Τροφοδοσία του Σ.Η.Ε.Ε με χρήση μπαταρίας 9V και ρυθμιστή τάσης στα 5V με εναλλακτική συνδεσμολογία



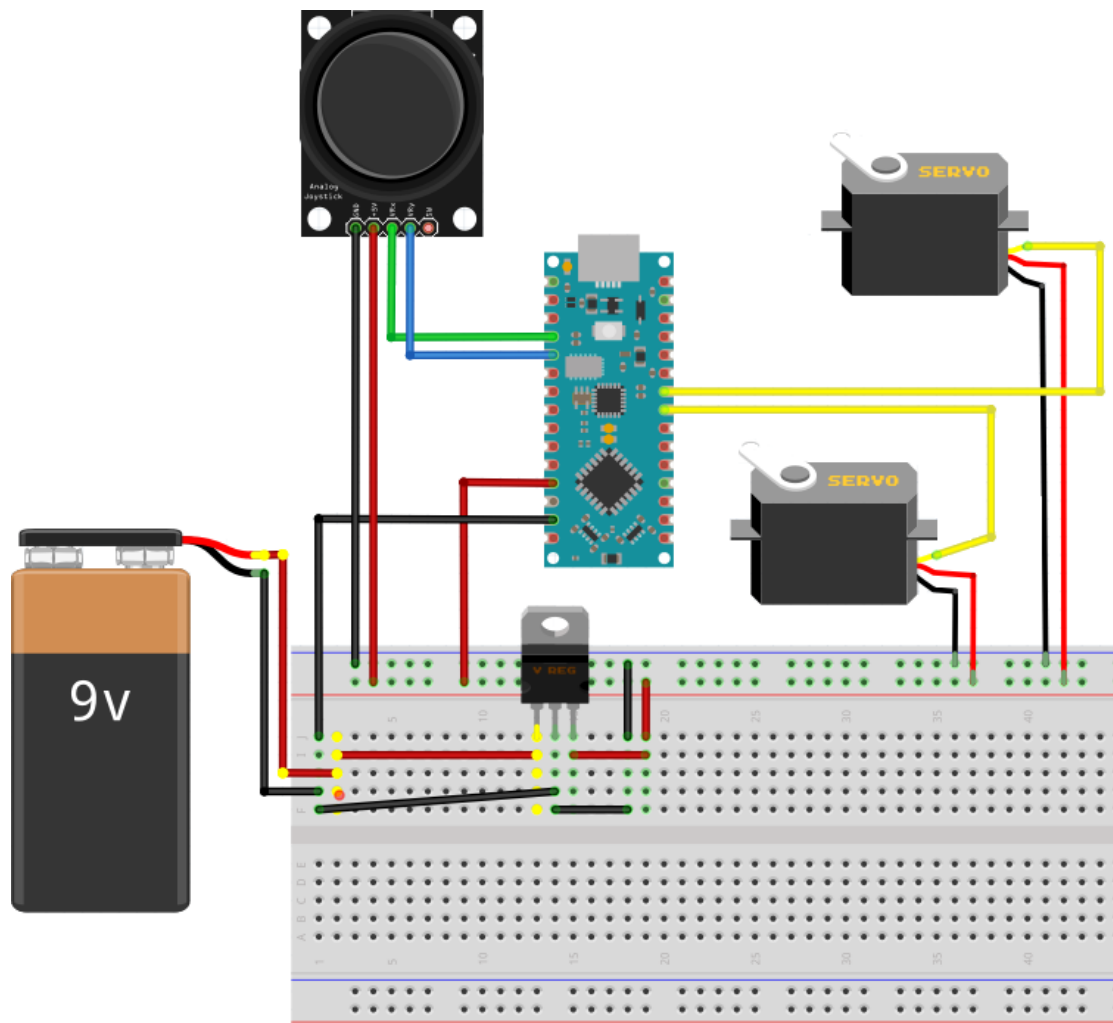
Εικόνα 3.45 Τροφοδοσία του Σ.Η.Ε.Ε με χρήση μπαταρίας των 9V, για απευθείας διασύνδεση και τροφοδοσία του μικροελεγκτή NANO (pins VIN-GND) και ρυθμιστή τάσης στα 5V για προσφορά τάσης 5V σε μοχλό και σερβοκινητήρες

3. Τροφοδοσία του Σ.Η.Ε.Ε με χρήση μπαταρίας 9V και ρυθμιστή τάσης στα 5V με πρόσθετη εναλλακτική συνδεσμολογία

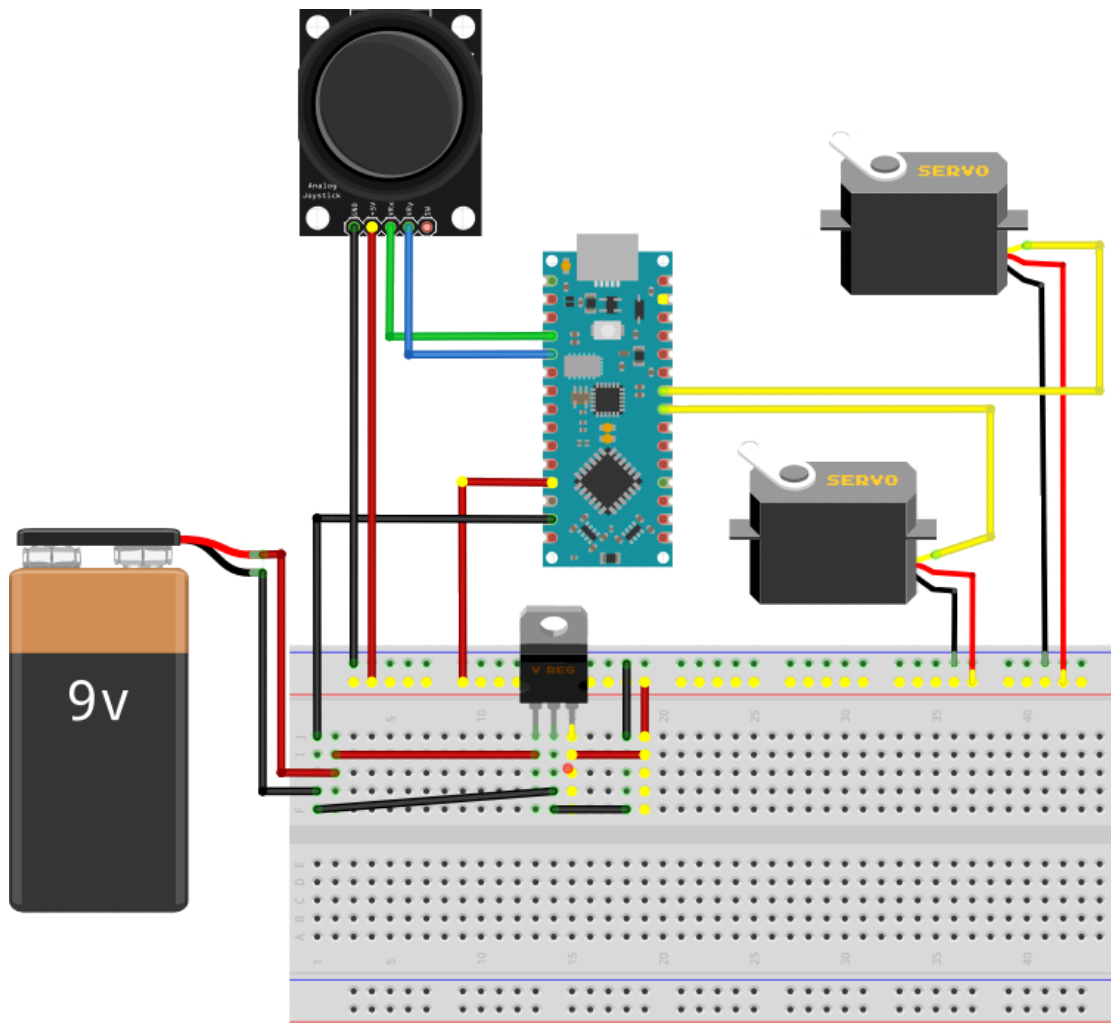


Εικόνα 3.46 Τροφοδοσία του Σ.Η.Ε.Ε. με χρήση μπαταρίας των 9V και ρυθμιστή τάσης στα 5V για προσφορά 5V τάσης σε μοχλό, μικροελεγκτή NANO (pins 5V-GND) και στους δύο ψηφιακούς σερβοκινητήρες μοντέλου MG995R περιστροφής 180°

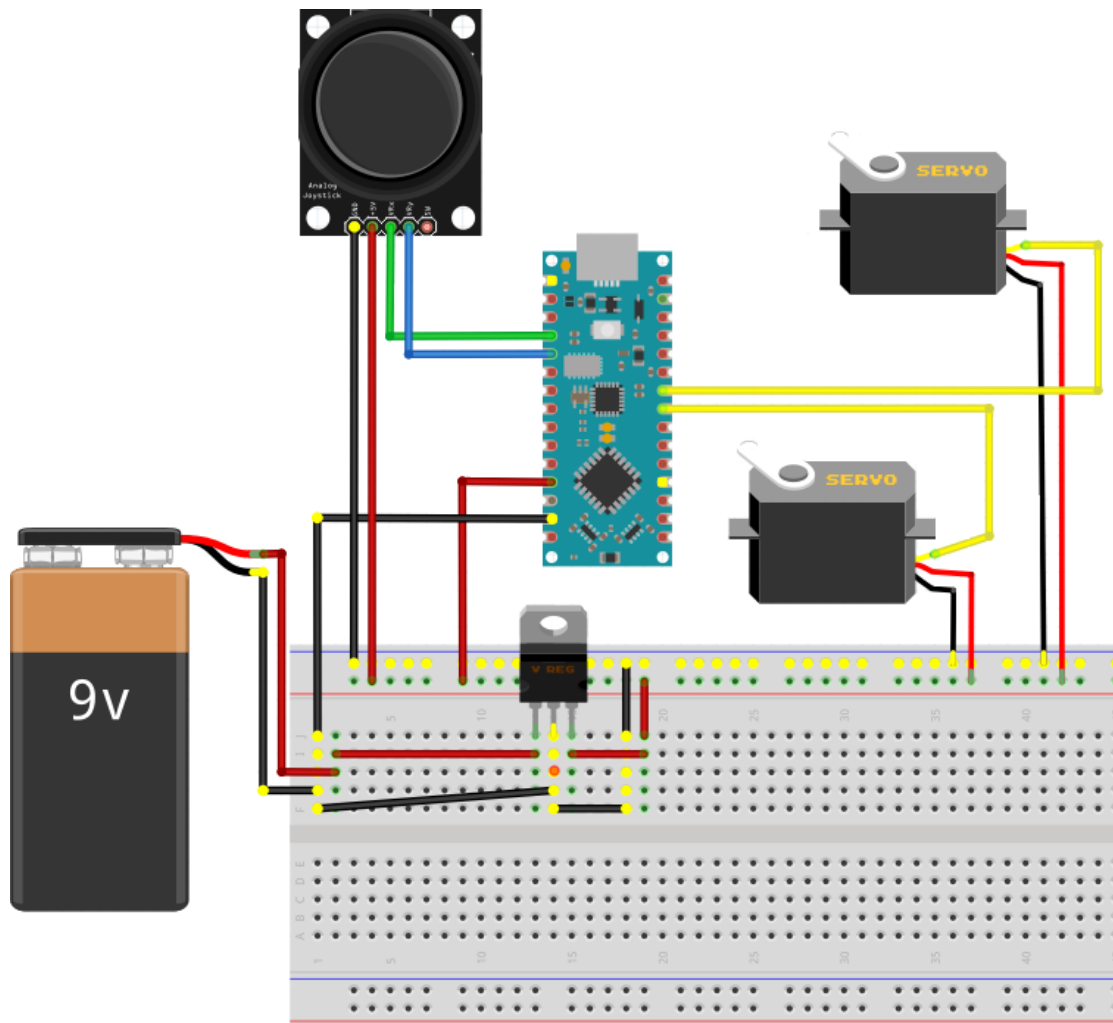
Με χρήση αυτού του τρόπου διασύνδεσης και βάσει των κατωτέρω τριών εικόνων (3.46.α, 3.46.β και 3.46.γ) γίνεται αντιληπτή η ύπαρξη και μεταφορά δυναμικού –λόγω παροχής τάσης από την 9V μπαταρία– σε συγκεκριμένα σημεία επί των καλωδίων του κυκλώματος και αντίστοιχες τιμές (σε Volt) των σημείων αυτών που φέρουν δυναμικό.



Εικόνα 3.46.α Σημεία δυναμικού VCC στο κύκλωμα (κίτρινες κουκίδες) λόγω ύπαρξης τάσης τροφοδοσίας από μπαταρία των 9V. Τα σημεία αυτά φέρουν όλα τιμή δυναμικού +9V

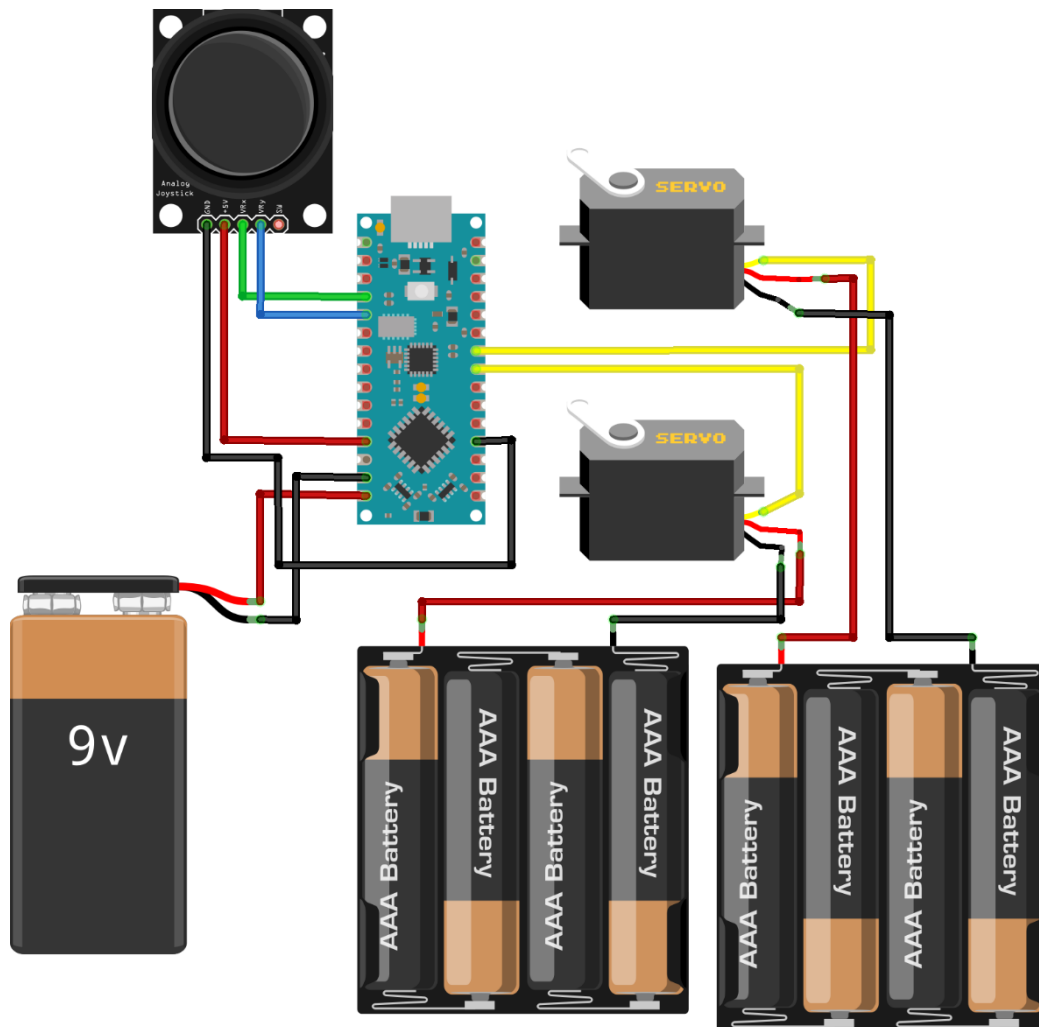


Εικόνα 3.46.β Σημεία δυναμικού V στο κύκλωμα (κίτρινες κουκίδες) λόγω διαμεσολάβησης ρυθμιστή τάσης στα 5V. Τα εν λόγω σημεία φέρουν όλα τους δυναμικό τιμής 5V



Εικόνα 3.46.γ Σημεία κοινού δυναμικού 0V (GND) στο κύκλωμα (κίτρινες κουκίδες) λόγω ύπαρξης τάσης (διαφοράς δυναμικού) τροφοδοσίας από την 9V μπαταρία. Τα σημεία αυτά φέρουν δυναμικό τιμής 0V (γείωση)

4. Τροφοδοσία του Σ.Η.Ε.Ε με χρήση μίας μπαταρίας 9V και δύο μπαταριοθηκών 6V



Εικόνα 3.47 Τροφοδοσία του συστήματος με χρήση μίας μπαταρίας 9V για παροχή ενέργειας στο Arduino NANO μέσω των pins “VIN-GND”, ενώ από τους ακροδέκτες “5V-GND” του μικροελεγκτή Arduino προσφέρεται τάση 5V στον αναλογικό μοχλό. Οι δύο μπαταριοθήκες δύνανται να παρέχουν στους σερβοκινητήρες τάση 6V

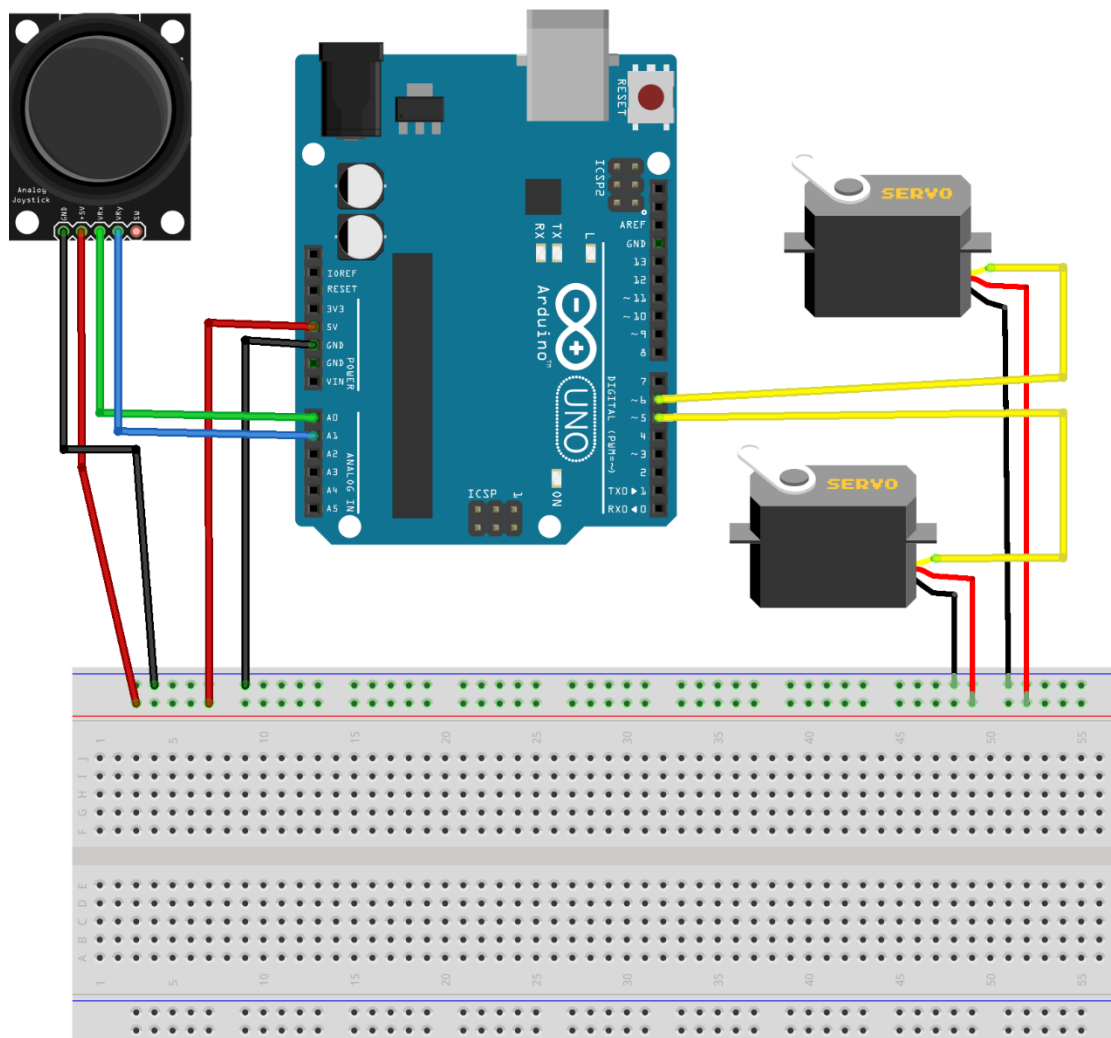
Βάσει των τεχνικών προδιαγραφών των 2 σερβοκινητήρων, η παροχή τάσης στα 6V τους καθιστά ικανούς να ασκούν μεγαλύτερη δύναμη περιστροφής, υποστηρίζοντας περισσότερο βάρος επί του μοχλού του άξονά τους και άρα λειτουργώντας αποδοτικότερα.

Εναλλακτικά, στην ανωτέρω συνδεσμολογία θα μπορούσε να γίνει αντικατάσταση των δύο μπαταριοθηκών με μία μπαταριοθήκη και μία πλακέτα δοκιμών για κοινή τροφοδοσία των 2 σερβοκινητήρων.

Τροφοδοσία του Σ.Η.Ε με χρήση της ηλεκτρονικής πλακέτας Arduino UNO R3

Στο σημείο αυτό, εκτός των ανωτέρω τρόπων διασύνδεσης, θα μπορούσε κάλλιστα να χρησιμοποιηθεί το μοντέλο *Arduino UNO R3* αντί του *NANO EVERY* για τον σχεδιασμό και την υλοποίηση του *Συστήματος Ηλεκτρονικού Ενσύρματου Ελεγκτή*, με κατάλληλη προσαρμογή των ρυθμίσεων (πλακέτα, θύρα κτλ) του ολοκληρωμένου περιβάλλοντος ανάπτυξης *Arduino IDE* για προγραμματισμό του Arduino UNO.

Για του λόγου το αληθές και για καθαρά πειραματικό σκοπό, η υλοποίηση του Σ.Η.Ε. με χρήση του Arduino UNO πραγματοποιείται και παρατίθεται κατωτέρω:



Εικόνα 3.48 Σχεδιασμός ενός συστήματος ηλεκτρονικού ενσύρματου ελεγκτή για Βιομηχανικές Εφαρμογές ως ενσύρματο σύστημα απόκρισης εισόδου/εξόδου, χωρίς πηγές τροφοδοσίας, με χρήση του Arduino UNO R3

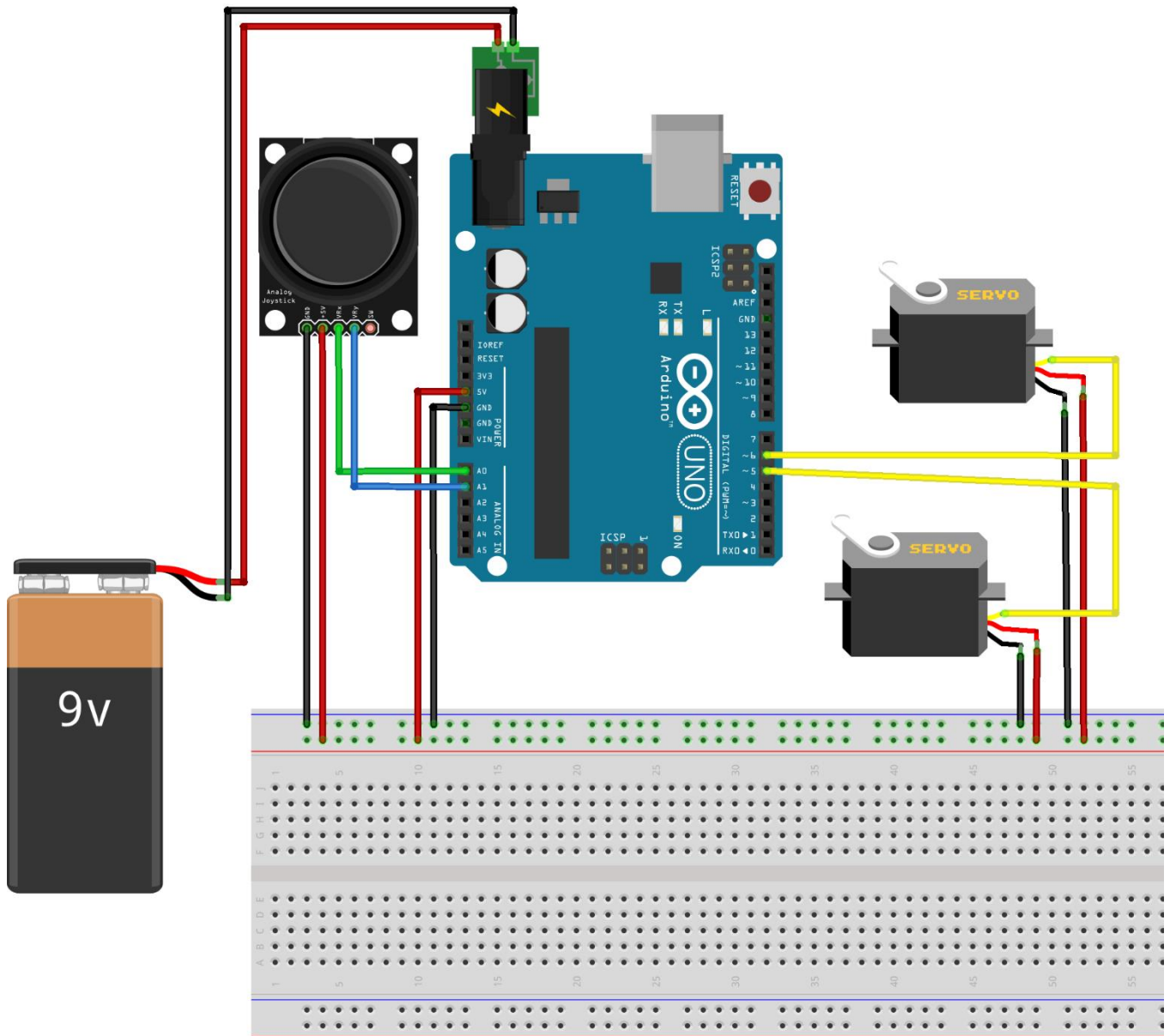
Στην εικόνα 3.48 παρουσιάζεται ο τρόπος συνδεσμολογίας του εν λόγω ενσύρματου συστήματος ελέγχου απόκρισης μιας εισόδου εντολών από αναλογικό μοχλό με δύο εξόδους από δύο ψηφιακούς σερβοκινητήρες. Η διαφορά εδώ βρίσκεται στο μοντέλο μικροελεγκτή που απαρτίζει το σύστημα.

Ο σχεδιασμός του κυκλώματος της εικόνας εικόνα 3.48 δεν περιλαμβάνει συνδεσμολογία εξωτερικών πηγών για παροχή ενέργειες στο κύκλωμα, καθώς γίνεται χρήση της θύρας USB για προγραμματισμό του μικροελεγκτή UNO R3 που παρέχει παράλληλα και τροφοδοσία στην ηλεκτρονική πλακέτα.

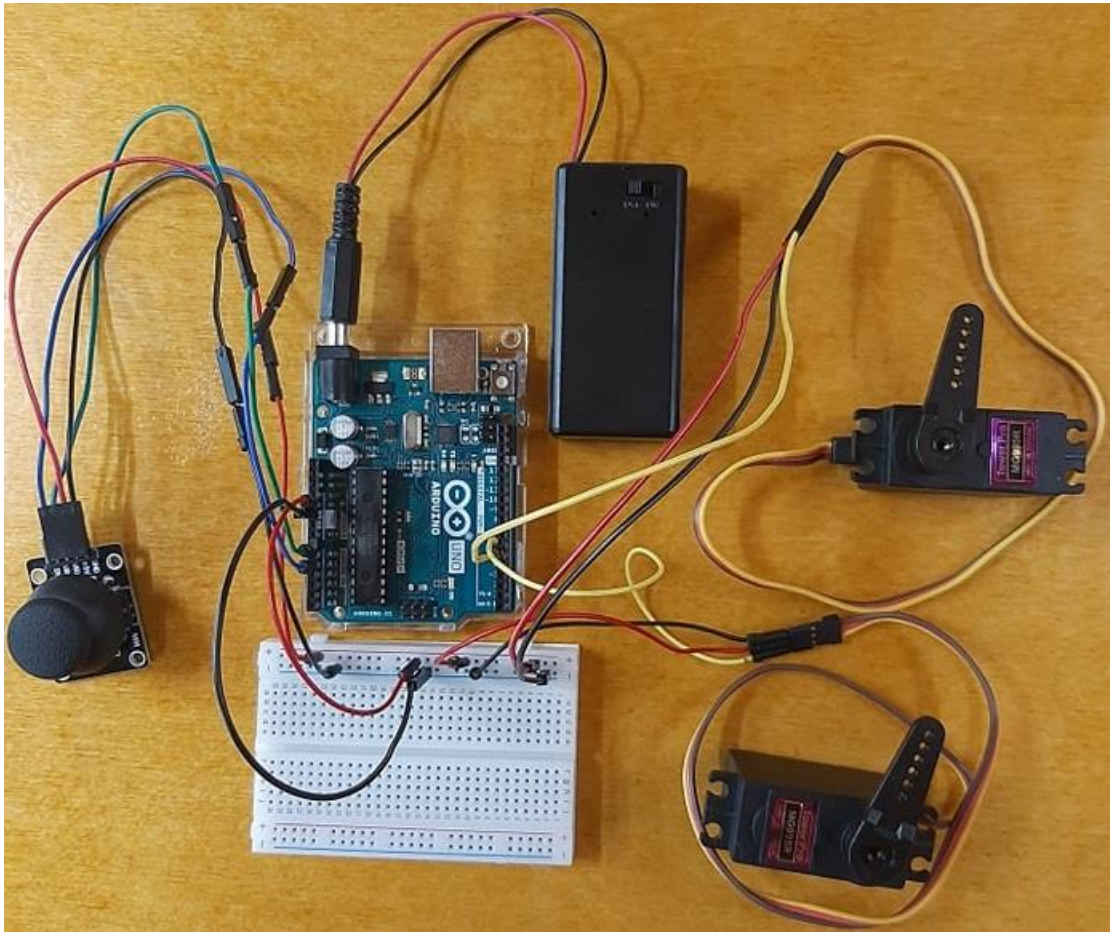
Έτσι, με την προσθήκη των δύο σερβοκινητήρων οι ακροδέκτες “VCC-GND” για την τροφοδοσία του χειριστηρίου συνδέονται με τους ακροδέκτες “5V-GND” του Arduino UNO μέσω μιας πλακέτας δοκιμών –αντί να πραγματοποιηθεί απευθείας σύνδεση με καλώδια– καθώς ο επιθυμητός τρόπος τροφοδοσίας που επιλέγεται στο παρόν κύκλωμα είναι η παροχή ενέργειας σε αναλογικό μοχλό και σερβοκινητήρες αποκλειστικά από την πλακέτα του μικροελεγκτή UNO και όχι από εξωτερικές πηγές τροφοδοσίας.

Ομοίως με την ηλεκτρονική πλακέτα Arduino NANO EVERY, η πλακέτα Arduino UNO R3 λαμβάνει και προσφέρει ενέργεια με παρόμοιο τρόπο, με μερικές μεθόδους παροχής ενέργειας για το κύκλωμα της εικόνα 3.48 να παρουσιάζονται μέσω πέντε χαρακτηριστικών παραδειγμάτων στις εικόνες που ακολουθούν.

1. Παροχή ενέργειας με χρήση μόνο μιας μπαταρίας 9V

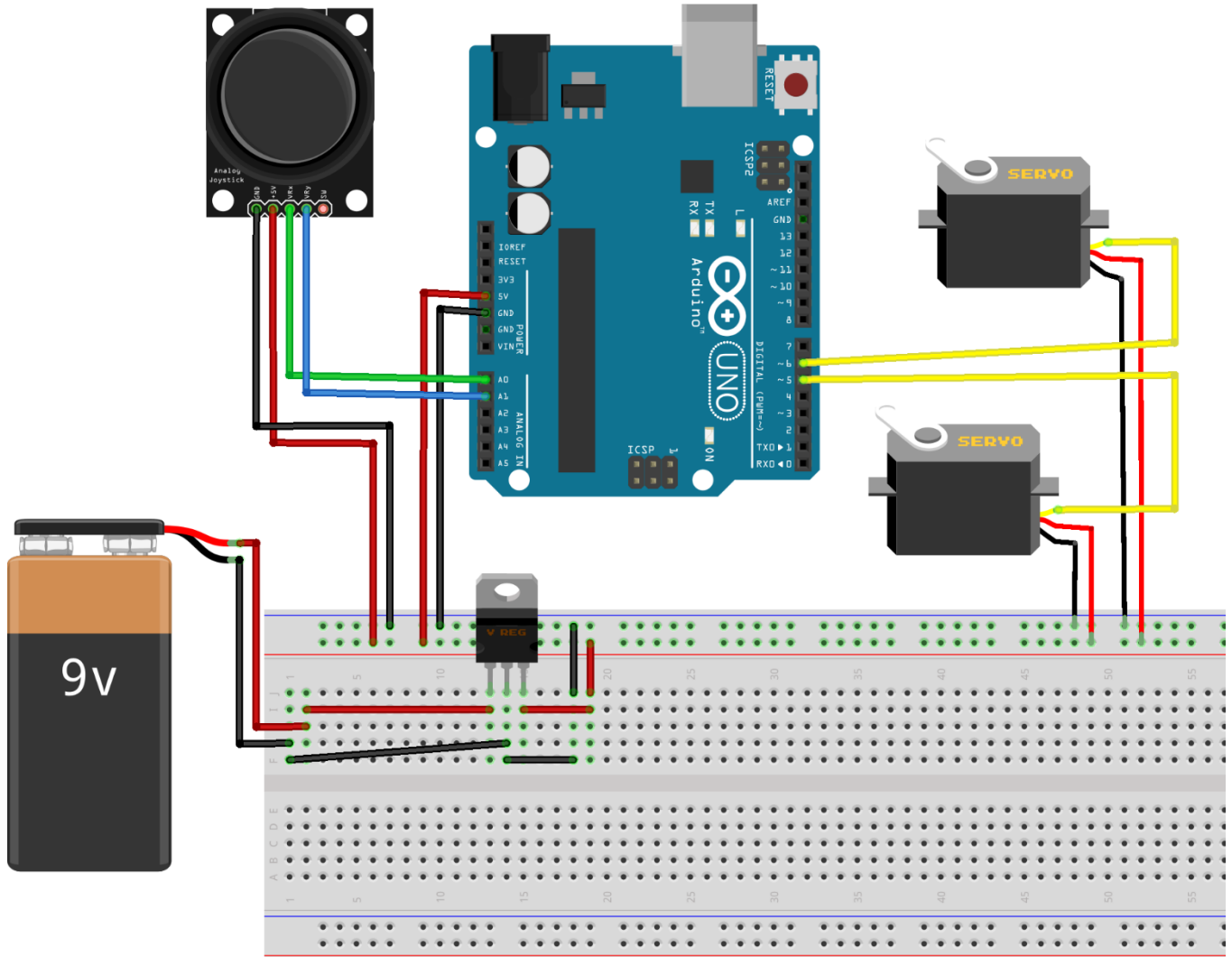


Εικόνα 3.49 Σχεδιασμός ενός Συστήματος Ηλεκτρονικού Ενσύρματου Ελεγκτή (Σ.Η.Ε.Ε.) για Βιομηχανικές Εφαρμογές ως ενσύρματο σύστημα απόκρισης εισόδου/εξόδου, με μία πηγή τροφοδοσίας, υλοποιημένο σε περιβάλλον προσομοίωσης



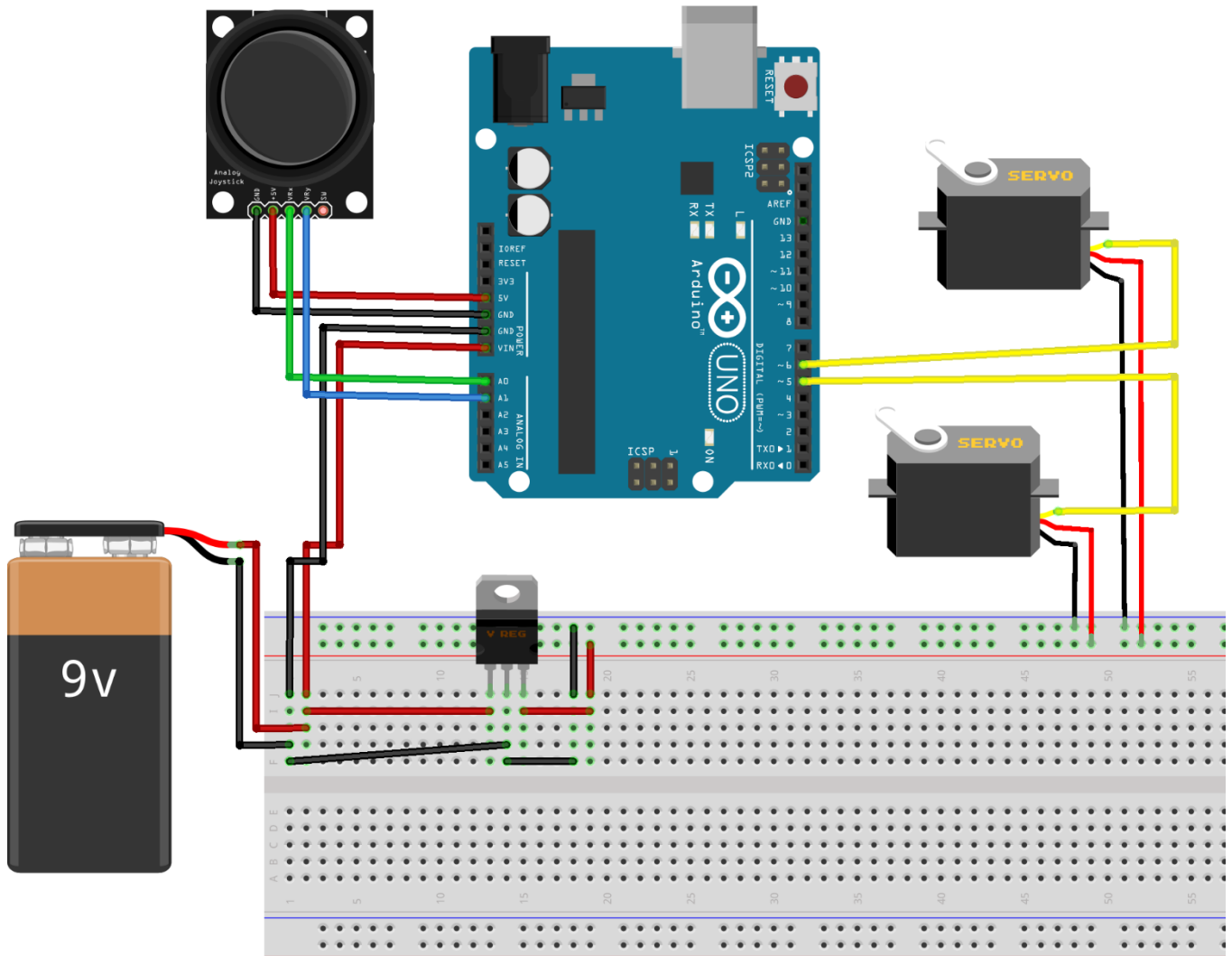
Εικόνα 3.50 Σχεδιασμός ενός Συστήματος Ηλεκτρονικού Ενσύρματου Ελεγκτή (Σ.Η.Ε.Ε.) για Βιομηχανικές Εφαρμογές, ως ενσύρματο σύστημα απόκρισης εισόδου/εξόδου με μία πηγή τροφοδοσίας από μπαταρία των 9V, υλοποιημένο σε επαγγελματικό εργαστήριο

2. Παροχή ενέργειας με χρήση μπαταρίας 9V και ρυθμιστή τάσης 5V



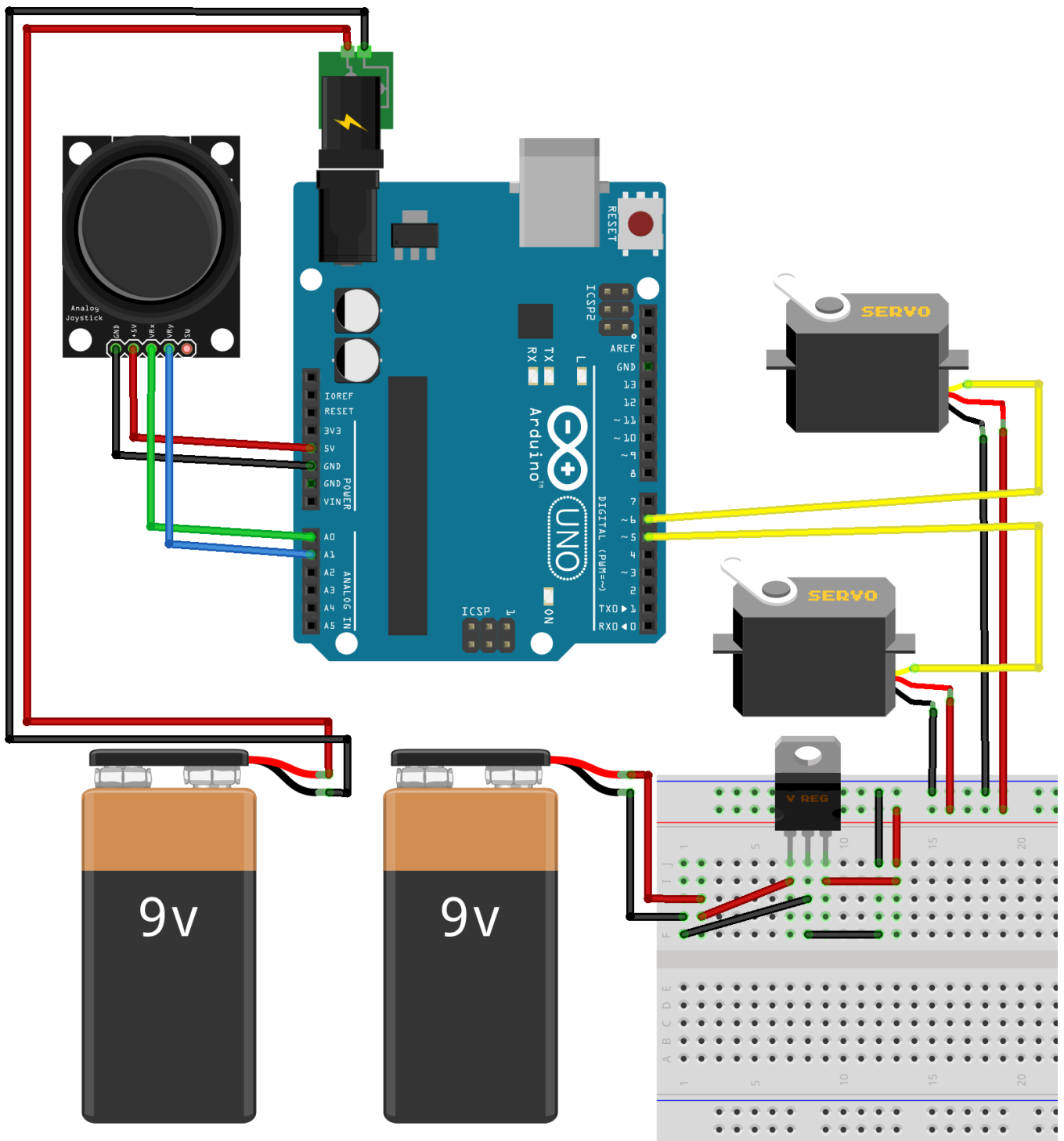
Εικόνα 3.51 Τροφοδοσία του Σ.Η.Ε.Ε. με χρήση μπαταρίας των 9V και ρυθμιστή τάσης στα 5V για προσφορά 5V σε αναλογικό μοχλό, μικροελεγκτή UNO (pins 5V-GND) και στους δύο σερβοκινητήρες

3. Παροχή ενέργειας με χρήση μπαταρίας 9V και ρυθμιστή τάσης 5V με εναλλακτική συνδεσμολογία



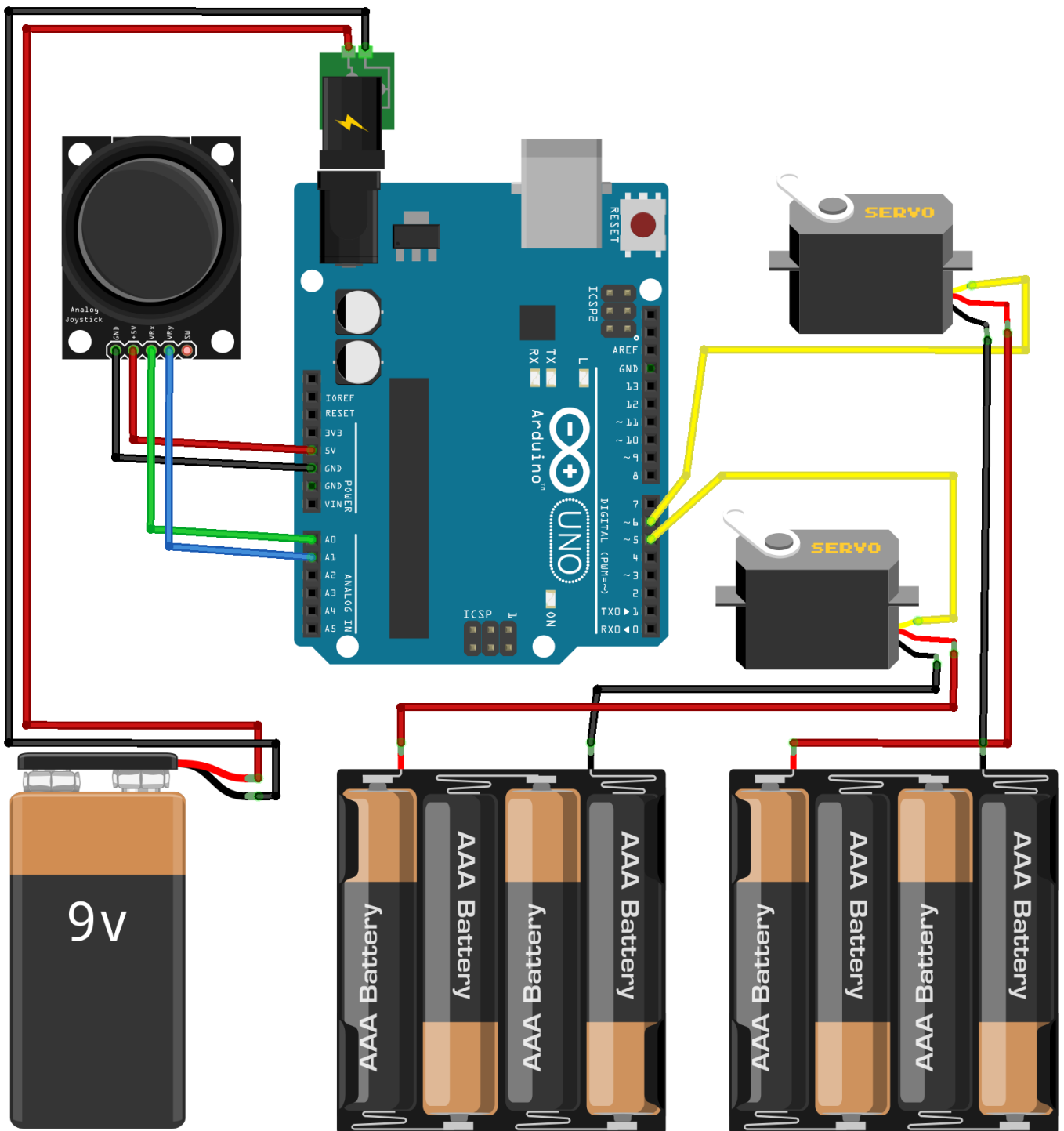
Εικόνα 3.52 Τροφοδοσία του Σ.Η.Ε.Ε. με χρήση μπαταρίας των 9V για παροχή 9V απευθείας στον μικροελεγκτή UNO (pins VIN-GND) και ρυθμιστή τάσης στα 5V για προσφορά 5V στους δύο σερβοκινητήρες. Ο αναλογικός μοχλός τροφοδοτείται με τάση 5V απευθείας από τον μικροελεγκτή UNO (pins 5V-GND)

4. Παροχή ενέργειας με χρήση 2 μπαταριών 9V και ρυθμιστή τάσης 5V



Εικόνα 3.53 Τροφοδοσία του Σ.Η.Ε.Ε. με χρήση δύο μπαταριών των 9V. Η μία μπαταρία παρέχει άμεσα 9V στον μικροελεγκτή UNO μέσω του “power jack” βύσματός του και κατά συνέπεια μέσω των pins “5V-GND” του UNO προσφέρεται τάση 5V στον αναλογικό μοχλό. Η δεύτερη μπαταρία των 9V παρέχει μέσω του ρυθμιστή τάσης 5V στους δύο σερβοκινητήρες

5. Παροχή ενέργειας από μία μπαταρία 9V και δύο μπαταριοθήκες 6V



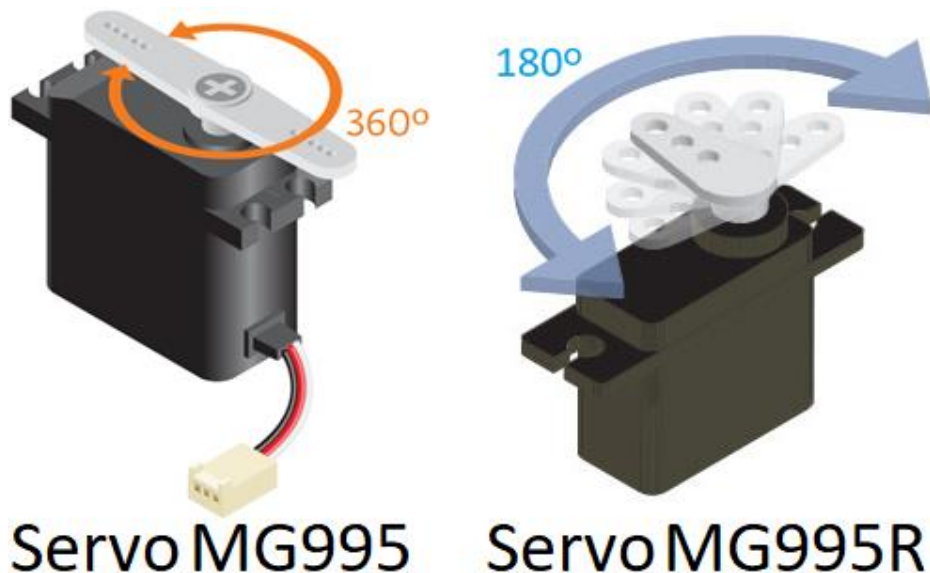
Εικόνα 3.54 Τροφοδοσία του Σ.Η.Ε. με χρήση μίας μπαταρίας των 9V και δύο μπαταριοθηκών 6V. Η 9V μπαταρία παρέχει άμεσα 9V στο Arduino UNO μέσω του “power jack” βύσματος και κατά συνέπεια μέσω των pins “5V-GND” του UNO προσφέρεται τάση 5V στον μοχλό. Οι μπαταριοθήκες δύνανται να παρέχουν απευθείας 6V στους σερβοκινητήρες

Κατ' αντιστοιχία του προγραμματιστικού κώδικα επαλήθευσης (ΚΩΔΙΚΑΣ_9) που αναπτύχθηκε για το σύστημα ηλεκτρονικού ενσύρματου ελεγκτή της εικόνας 3.41 και περιλαμβάνει τον μικροελεγκτή Arduino NANO EVERY, ο ίδιος κώδικας αξιοποιείται και για προγραμματισμό του Σ.Η.Ε.Ε. που τώρα διαθέτει μικροελεγκτή Arduino UNO R3, με κατάλληλη τροποποίηση των ρυθμίσεων του περιβάλλοντος Arduino IDE για επικοινωνία μεταξύ Arduino UNO και H/Y, όπως προαναφέρθηκε και σε προηγούμενες υποενότητες.

3.1.1.2.1 Έλεγχος Απόκρισης του Σ.Η.Ε.Ε. με ένταξη περαιτέρω περιφερειακών συσκευών I/O

Με περαιτέρω προσθήκη ενός ακόμη αναλογικού μοχλού και δύο νέων μοντέλων σερβοκινητήρων, γίνεται έλεγχος της συμπεριφοράς του ενσύρματου ελεγκτή ως σύστημα απόκρισης δύο εισόδων (*Inputs*) με τέσσερις εξόδους (*Outputs*).

Αρχικά γίνεται έλεγχος ορθής λειτουργίας των δύο νέων μοντέλων ψηφιακού σερβοκινητήρα MG995 με δυνατότητα περιστροφής 360°, σε αντίθεση με το προηγούμενο ζεύγος σερβοκινητήρων μοντέλου MG995R με δυνατότητα περιστροφής του άξονα κατά 180°.

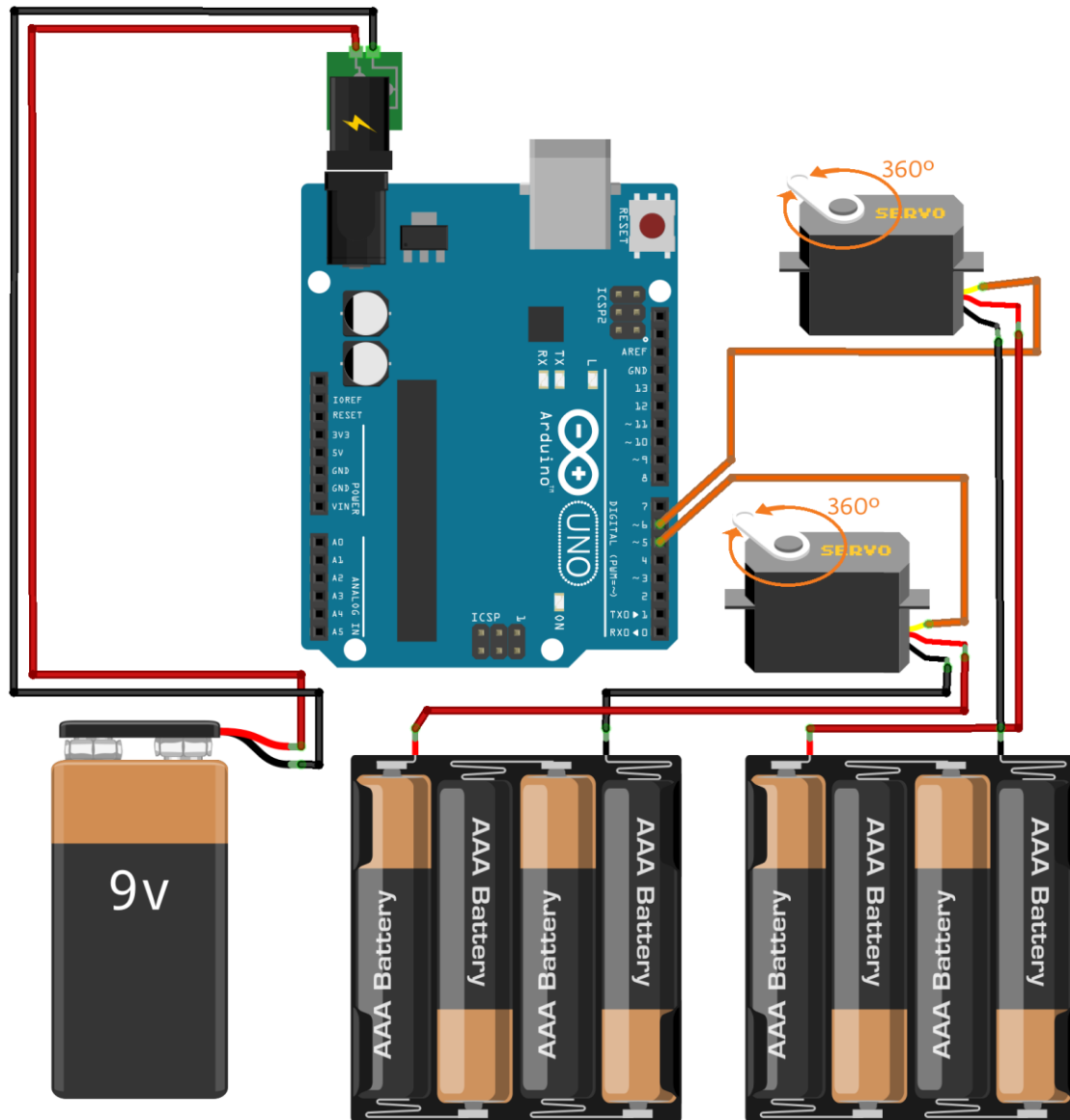


Εικόνα 3.55 Σερβοκινητήρες διαφορετικής δυνατότητας περιστροφής

Έπειτα πραγματοποιείται συνδεσμολογία μεταξύ του μικροελεγκτή Arduino UNO R3 και των τεσσάρων ψηφιακών σερβοκινητήρων για μελέτη της συμπεριφοράς και λειτουργίας αυτών ως ένα ενιαίο κύκλωμα.

Με το πέρας του επιτυχούς αυτού ελέγχου γίνεται προσθήκη των δύο αναλογικών μοχλών για δημιουργία του κυκλώματος του Σ.Η.Ε.Ε. με πρόσθετες περιφερειακές συσκευές I/O και πραγματοποίηση του επιθυμητού ελέγχου απόκρισης, με το ζεύγος των σερβοκινητήρων περιστροφής 180° να ανταποκρίνεται σε εντολές του ενός μοχλού και το ζεύγος των σερβοκινητήρων περιστροφής 360° σε εντολές του άλλου.

Για το παράδειγμα ελέγχου ορθής λειτουργίας του κυκλώματος και χρήση του ζεύγους σερβοκινητήρων 360° ως έξοδό του, επιλέγεται ως σχεδιαστική βάση ο τρόπος παροχής ενέργειας που παρουσιάζεται στην εικόνα 3.54, με απαλοιφή του αναλογικού μοχλού φυσικά.



Εικόνα 3.56 Συνδεσμολογία του Arduino UNO R3 με 2 σερβοκινητήρες MG995, μία μπαταριοθήκη τροφοδοσίας 9V και δύο τροφοδοσίες 6V

Ο κώδικας που αναπτύχθηκε για τον έλεγχο του παραπάνω κυκλώματος είναι ο εξής:

```
/*
ΕΛΕΓΧΟΣ ΑΠΟΚΡΙΣΗΣ ΔΥΟ ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ MG995 ΜΕ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΟ
ΤΡΟΠΟ - ΧΡΗΣΗ ΚΩΔΙΚΑ - ΑΝΤΙ ΠΑΡΟΧΗΣ ΕΝΤΟΛΩΝ ΕΙΣΟΔΟΥ ΑΠΟ ΣΥΣΚΕΥΗ
*/

#include <Servo.h> // Add the servo library

Servo myServo1;    // Define 'myServo1' as Servo1
Servo myServo2;    // Define 'myServo2' as Servo2

int angleV1 = 0;   // Initialize variable to store Servo1 rotation speed to max clockwise
int angleV2 = 180; // Initialize variable to store Servo2 rotation speed to max anti-clockwise

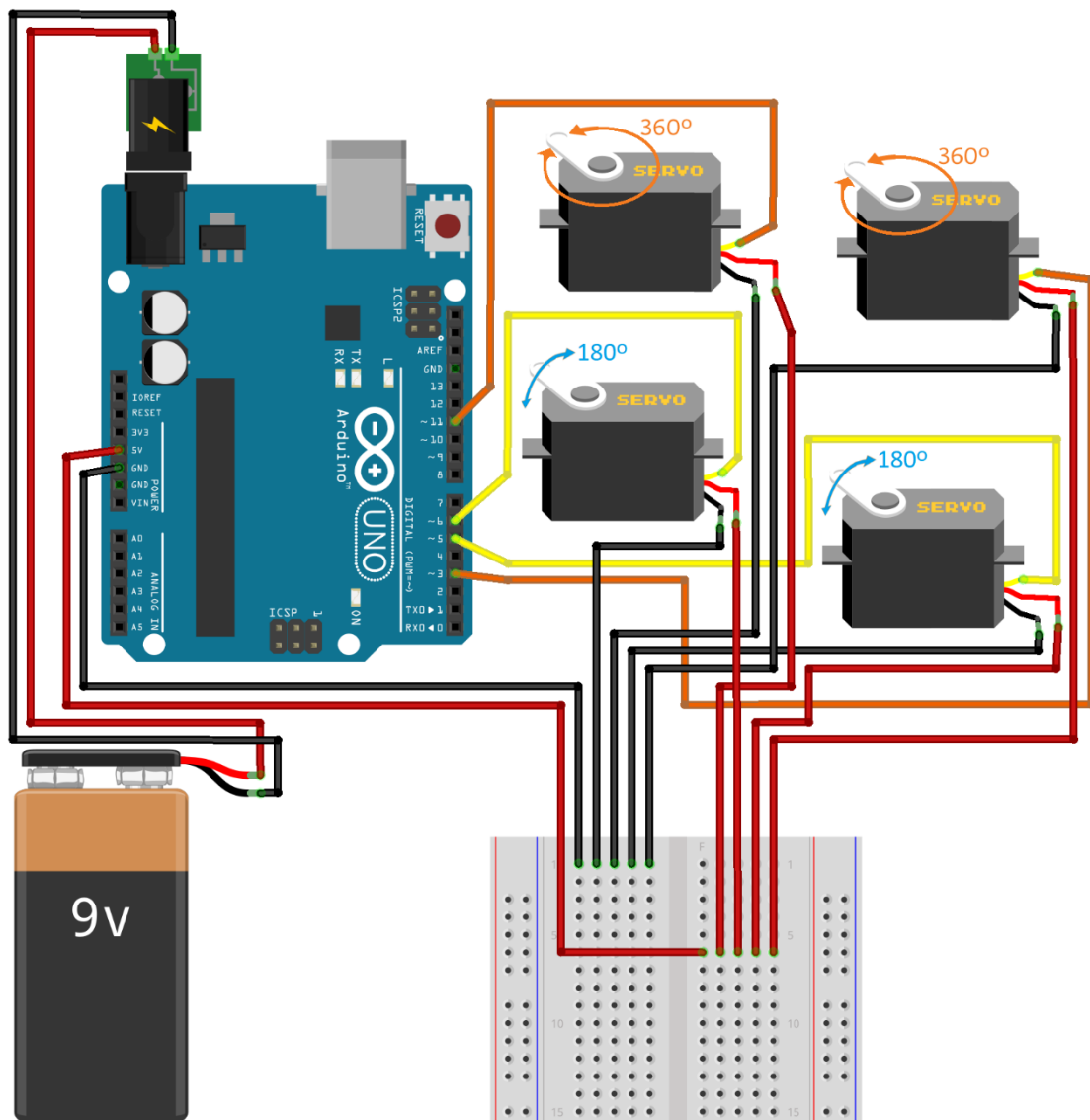
void setup() {
  myServo1.attach(5); // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // Attaches Servo2 Signal Line to pin 6 (PWM)
}

void loop() {
  // for the first servo: myServo1
  myServo1.write(0); // Use the variable to rotate the Servo Motor 1 continously
                    // with the "0 degrees" value that represents the
                    // max clockwise rotation speed of the Servo
  delay(3000);      // waits 3000ms for the next command to be executed

  // for the second servo: myServo2
  myServo2.write(180); // Use the variable to rotate the Servo Motor 2 continously
                      // with the "180 degrees" value that represents the
                      // max anti-clockwise rotation speed of the Servo
  //delay(15);      // waits 15ms for the next command to be executed
}
```

ΚΩΔΙΚΑΣ_10 Έλεγχος απόκρισης 2 σερβοκινητήρων με χρήση προγραμματιστικού κώδικα

Έπειτα, με αντικατάσταση των δύο μπαταριοθηκών τροφοδοσίας 6V με μια πλακέτα δοκιμών και προσθήκη του ζεύγους σερβοκινητήρων 180° (MG995R) ως επιπλέον έξοδος για το υπάρχον κύκλωμα, η πλακέτα Arduino UNO R3 τροφοδοτείται με τάση 9V μέσω του βύσματος barrel jack και έτσι με την σειρά της παρέχει στους 4 σερβοκινητήρες τάση 5V με κατάλληλη αξιοποίηση του ακροδέκτη 5V που διαθέτει και της πλακέτας δοκιμών, ως φαίνεται στην κάτωθι εικόνα.



Εικόνα 3.57 Συνδεσμολογία του Arduino UNO R3 με 2 σερβοκινητήρες MG995R, 2 σερβοκινητήρες MG995, μία πλακέτα δοκιμών και μία μπαταρία των 9V

Ο προγραμματιστικός κώδικας για το κύκλωμα των 4 εξόδων (σερβοκινητήρων) είναι ο εξής:

```
/*
ΕΛΕΓΧΟΣ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ 4 ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ (2 MG99R & 2 MG995).

- Servo1 (MG995R): ΟΛΙΣΘΑΙΝΕΙ ΣΤΑΔΙΑΚΑ ΜΕΤΑΞΥ ΟΛΩΝ ΤΩΝ ΤΙΜΩΝ ΤΟΥ ΕΥΡΟΥΣ 0-180 ΜΟΙΡΩΝ, ΛΟΓΩ ΤΗΣ
ΕΚΤΕΛΕΣΗΣ ΤΩΝ 2 ΒΡΟΧΩΝ ΕΠΑΝΑΛΗΨΗΣ ΓΙΑ ΤΙΜΕΣ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ "angleV1" ΑΠΟ 0-180 & ΑΠΟ 180-0 ΜΟΙΡΕΣ,
ΑΝΤΙΣΤΟΙΧΑ.

- Servo2 (MG995R): ΜΕΤΑΒΑΣΗ ΜΟΝΟ ΣΤΙΣ ΑΚΡΑΙΕΣ ΘΕΣΕΙΣ 0 Η 180 ΜΟΙΡΕΣ, ΛΟΓΩ ΥΠΑΡΧΗΣ ΜΙΑΣ ΣΥΝΘΗΚΗΣ
"if" ΣΤΟΥΣ 2 ΒΡΟΧΟΥΣ ΕΠΑΝΑΛΗΨΗΣ ΓΙΑ ΟΡΙΘΜΕΤΗΣ ΤΩΝ ΤΙΜΩΝ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ "angleV2".

- Servo3 (MG995): ΣΥΝΕΧΗΣ ΔΕΞΙΟΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΙΚΗ ΚΙΝΗΣΗ 360 ΜΟΙΡΩΝ ΜΕ ΤΗΝ ΜΕΓΙΣΤΗ ΔΥΝΑΤΗ ΤΑΧΥΤΗΤΑ,
ΛΕΙΤΟΥΡΓΙΑ ΠΟΥ ΧΑΡΑΚΤΗΡΙΖΕΤΑΙ ΑΠΟ ΤΗΝ ΤΙΜΗ 0 (ΣΕ ΜΟΙΡΕΣ).

- Servo4 (MG995): ΣΥΝΕΧΗΣ ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΙΚΗ ΚΙΝΗΣΗ 360 ΜΟΙΡΩΝ ΜΕ ΤΗΝ ΜΕΓΙΣΤΗ ΔΥΝΑΤΗ ΤΑΧΥΤΗΤΑ,
ΛΕΙΤΟΥΡΓΙΑ ΠΟΥ ΧΑΡΑΚΤΗΡΙΖΕΤΑΙ ΑΠΟ ΤΗΝ ΤΙΜΗ 180 (ΣΕ ΜΟΙΡΕΣ).
*/

#include <Servo.h> // Add the servo library
Servo myServo1; // Define 'myServo1' as Servo1
Servo myServo2; // Define 'myServo2' as Servo2
Servo myServo3; // Define 'myServo3' as Servo3
Servo myServo4; // Define 'myServo4' as Servo4
int angleV1 = 0; // Initialize variable to store servol position/angle
int angleV2 = 0; // Initialize variable to store servo2 position/angle

void setup() {
  myServo1.write(180); // Initialize Servo1 position/angle to max-point (180 degrees)
  myServo2.write(0); // Initialize Servo2 position/angle to min-point (0 degrees)
  myServo1.attach(5); // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // Attaches Servo2 Signal Line to pin 6 (PWM)
  myServo3.attach(3); // Attaches Servo3 Signal Line to pin 3 (PWM)
  myServo4.attach(11); // Attaches Servo4 Signal Line to pin 11 (PWM)
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

void loop() {
  // for the first servo: myServo1
  for (angleV1 = 0; angleV1 <= 180; angleV1 +=1) {
    myServo1.write(angleV1);    // Use the variable to rotate the Servo Motor 1
                                // and change its position according to the variable value
                                // from 0 to 180 degrees
    delay(20); // waits 20ms for the servo to reach the corresponding angle
  }
  for (angleV1 = 180; angleV1 >= 0; angleV1 -=1) {
    myServo1.write(angleV1);    // Use the variable to rotate the Servo Motor 1
                                // and change its position according to the variable value
                                // from 180 to 0 degrees
    delay(20); // waits 20ms for the servo to reach the corresponding angle
  }

  // for the second servo: myServo2
  for (angleV2 = 0; angleV2 <= 180; angleV2 +=1) {
    if (angleV2 = 180); {      // if angleV2 = 180 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
                                // and change its position according to the variable value
                                // from 0 to 180 degrees
      delay(2500); // waits 2500ms for the servo to reach the corresponding angle
    }
  }
  for (angleV2 = 180; angleV2 >= 0; angleV2 -=1) {
    if (angleV2 = 0); {      // if angleV2 = 0 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
                                // and change its position according to the variable value
                                // from 180 to 0 degrees
      delay(1000); // waits 1000ms for the servo to reach the corresponding angle
    }
  }

  // for the third & fourth servo: myServo3 & myServo4
  myServo3.write(0); // Use the variable to rotate the Servo Motor 3 continuously
                    // with the "0 degrees" value that represents the
                    // max clockwise rotation speed of the Servo
  delay(3000); // waits 3000ms for the next command to be executed

  myServo4.write(180); // Use the variable to rotate the Servo Motor 4 continuously
                      // with the "180 degrees" value that represents the
                      // max anti-clockwise rotation speed of the Servo
  delay(15); // waits 15ms for the next command to be executed
}

```

**ΚΩΔΙΚΑΣ_11 Έλεγχος ορθής λειτουργίας και απόκρισης του κυκλώματος
δύο σερβοκινητήρων δίχως αναλογικό μοχλό**

Στον κώδικα Νο11 γίνεται απαλοιφή των καθυστερήσεων (εντολές καθυστέρησης *delay*) για περαιτέρω μελέτη της συμπεριφοράς των τεσσάρων εξόδων, με τον νέο τροποποιημένο κώδικα Νο12 να ακολουθεί στην επόμενη σελίδα και την σχετική εξήγηση της συμπεριφοράς των 4 σερβοκινητήρων να παρατίθεται ως μέρος των σχολίων του κώδικα.

```

/*
ΕΛΕΓΧΟΣ ΣΥΜΠΕΡΙΦΟΡΑΣ 4 ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ (2 MG99R & 2 MG995) ΧΩΡΙΣ ΚΑΘΥΣΤΕΡΗΣΕΙΣ (delay).

- Servo1 (MG995R): ΟΛΙΓΘΑΙΝΕΙ ΜΟΝΟ ΜΕΤΑΞΥ ΤΩΝ ΜΕΣΑΙΩΝ ΤΙΜΩΝ ΤΟΥ ΕΥΡΟΥΣ 0-180 ΜΟΙΡΩΝ,
ΛΟΓΩ ΤΗΣ ΕΚΤΕΛΕΣΗΣ ΤΩΝ 2 ΒΡΟΧΩΝ ΕΠΑΝΑΛΗΨΗΣ ΧΩΡΙΣ ΚΑΘΥΣΤΕΡΗΣΕΙΣ ΓΙΑ ΤΙΜΕΣ ΤΗΣ
ΜΕΤΑΒΛΗΤΗΣ "angleV1" ΑΠΟ 0-180 & ΑΠΟ 180-0 ΜΟΙΡΕΣ.

- Servo2 (MG995R): ΠΑΡΑΜΟΝΗ ΣΕ ΑΔΡΑΝΕΙΑ, ΚΑΘΩΣ ΛΟΓΩ ΥΠΑΡΞΗΣ ΤΩΝ 2 ΒΡΟΧΩΝ ΕΠΑΝΑΛΗΨΗΣ
ΧΩΡΙΣ ΚΑΘΥΣΤΕΡΗΣΕΙΣ Η ΜΕΤΑΒΛΗΤΗ "angleV2" ΔΕΝ ΦΤΑΝΕΙ ΠΟΤΕ ΤΙΣ ΑΚΡΑΙΕΣ ΤΙΜΕΣ
0 ΚΑΙ 180 ΜΟΙΡΕΣ.

- Servo3 (MG995): ΣΥΝΕΧΗΣ ΔΕΞΙΟΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΗ 360 ΜΟΙΡΩΝ ΜΕ ΤΗΝ ΜΕΓΙΣΤΗ ΔΥΝΑΤΗ ΤΑΧΥΤΗΤΑ,
ΛΕΙΤΟΥΡΓΙΑ ΠΟΥ ΧΑΡΑΚΤΗΡΙΖΕΤΑΙ ΑΠΟ ΤΗΝ ΤΙΜΗ 0 (ΣΕ ΜΟΙΡΕΣ).

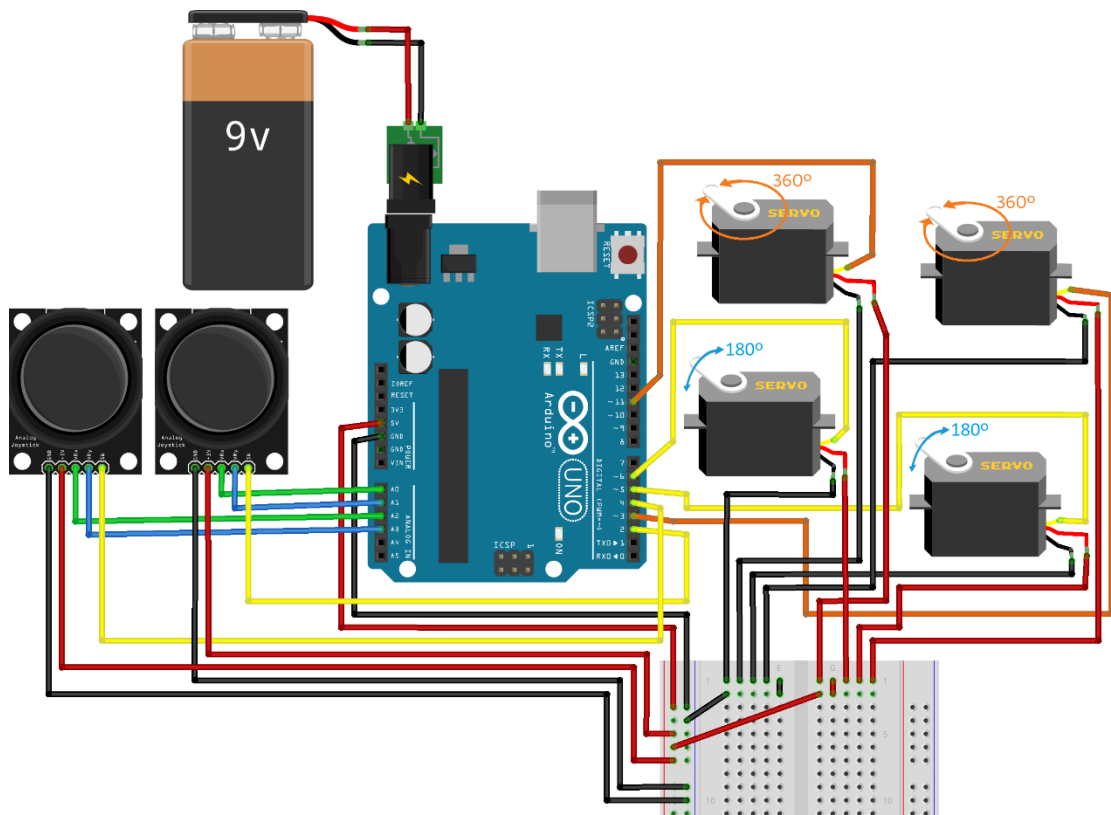
- Servo4 (MG995): ΣΥΝΕΧΗΣ ΑΡΙΣΤΕΡΟΣΤΡΟΦΗ ΠΕΡΙΣΤΡΟΦΗ 360 ΜΟΙΡΩΝ ΜΕ ΤΗΝ ΜΕΓΙΣΤΗ ΔΥΝΑΤΗ ΤΑΧΥΤΗΤΑ,
ΛΕΙΤΟΥΡΓΙΑ ΠΟΥ ΧΑΡΑΚΤΗΡΙΖΕΤΑΙ ΑΠΟ ΤΗΝ ΤΙΜΗ 180 (ΣΕ ΜΟΙΡΕΣ).
*/
#include <Servo.h> // Add the servo library
Servo myServo1; // Define 'myServo1' as Servo1
Servo myServo2; // Define 'myServo2' as Servo2
Servo myServo3; // Define 'myServo3' as Servo3
Servo myServo4; // Define 'myServo4' as Servo4
int angleV1 = 0; // Initialize variable to store servo1 position/angle
int angleV2 = 0; // Initialize variable to store servo2 position/angle
void setup() {
  myServo1.write(180); // Initialize Servo1 position/angle to max-point (180 degrees)
  myServo2.write(0); // Initialize Servo2 position/angle to min-point (0 degrees)
  myServo1.attach(5); // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // Attaches Servo2 Signal Line to pin 6 (PWM)
  myServo3.attach(3); // Attaches Servo3 Signal Line to pin 3 (PWM)
  myServo4.attach(11); // Attaches Servo4 Signal Line to pin 11 (PWM)
}
void loop() {
  // for the first servo: myServo1
  for (angleV1 = 0; angleV1 <= 180; angleV1 +=1) {
    myServo1.write(angleV1); // Use the variable to rotate the Servo Motor 1
  }
  for (angleV1 = 180; angleV1 >= 0; angleV1 -=1) {
    myServo1.write(angleV1); // Use the variable to rotate the Servo Motor 1
  }
  // for the second servo: myServo2
  for (angleV2 = 0; angleV2 <= 180; angleV2 +=1) {
    if (angleV2 = 180); { // if angleV2 = 180 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
      //delay(2500); // waits 2500ms for the servo to reach the corresponding angle
    }
  }
  for (angleV2 = 180; angleV2 >= 0; angleV2 -=1) {
    if (angleV2 = 0); { // if angleV2 = 0 degrees
      myServo2.write(angleV2); // Use the variable to rotate the Servo Motor 2
    }
  }
}
// for the third & fourth servo: myServo3 & myServo4
myServo3.write(0); // Use the variable to rotate the Servo Motor 3 continuously
myServo4.write(180); // Use the variable to rotate the Servo Motor 4 continuously
}

```

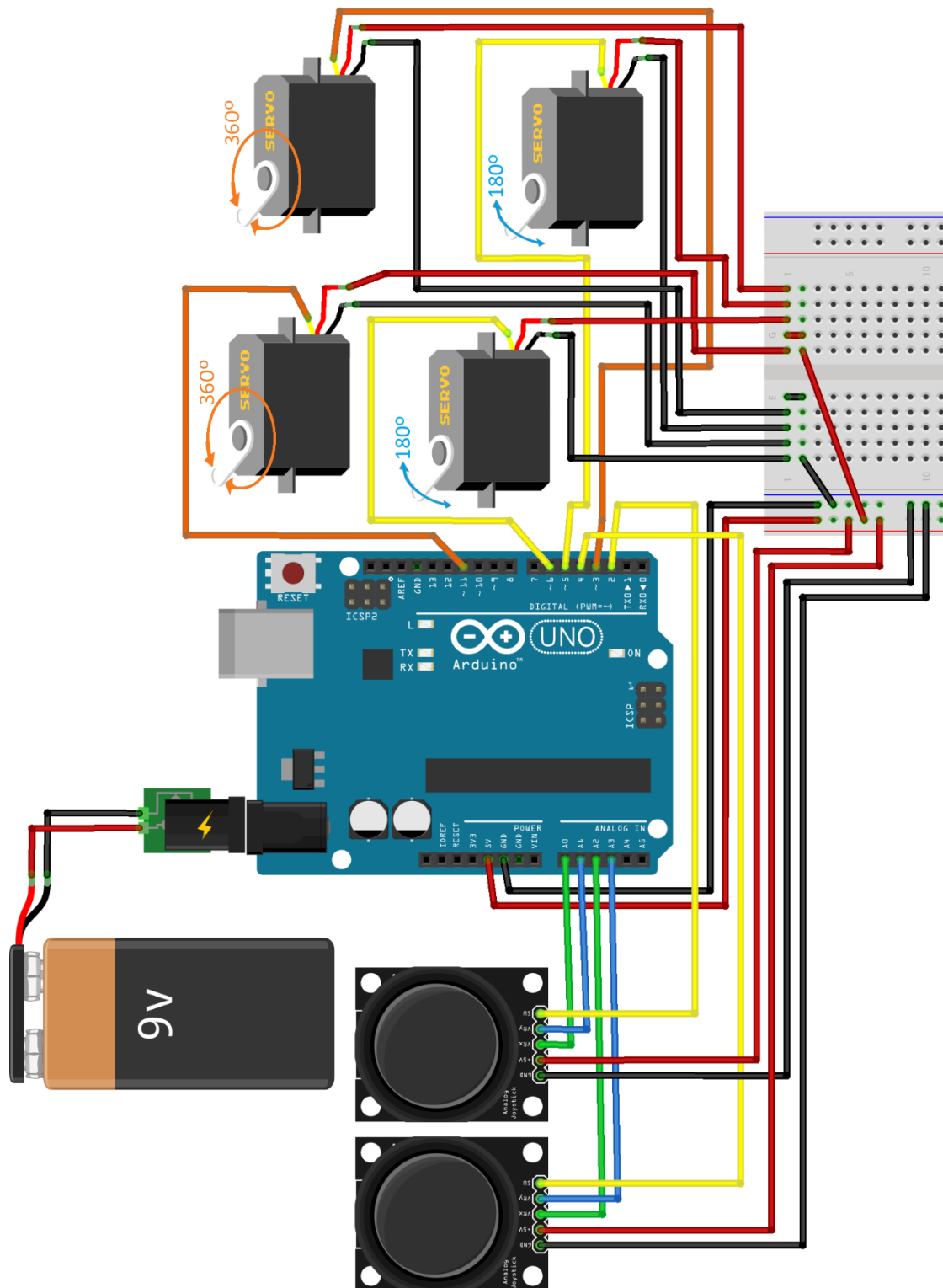
ΚΩΔΙΚΑΣ_12 Έλεγχος ορθής λειτουργίας και απόκρισης του κυκλώματος δύο σερβοκινητήρων δίχως αναλογικό μοχλό και εντολές delay

Στο σημείο αυτό, γίνεται ένταξη των δύο αναλογικών μοχλών στο υλοποιημένο κύκλωμα της εικόνας 3.57 και τροποποίηση του αντίστοιχου κώδικα (ΚΩΔΙΚΑΣ_11) του κυκλώματος για συμπερίληψη των δύο μοχλών.

Ως αποτέλεσμα, προκύπτει ένα Σ.Η.Ε.Ε. με πρόσθετες μονάδες I/O, το οποίο παρουσιάζεται στο σχηματικό διάγραμμα της εικόνας 3.58, με τον τροποποιημένο κώδικα (ΚΩΔΙΚΑΣ_13) να ακολουθεί του διαγράμματος.



Εικόνα 3.58 Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή 2 εισόδων και 4 εξόδων σε σχηματικό διάγραμμα υλοποιημένο σε περιβάλλον προσομοίωσης



Εικόνα 3.58 – Πλάγια όψη Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή 2 εισόδων και 4 εξόδων σε σχηματικό διάγραμμα υλοποιημένο σε περιβάλλον προσομοίωσης

```

#include <Servo.h> // Add the servo library

Servo myServo1; // Define 'myServo1' as Servo1
Servo myServo2; // Define 'myServo2' as Servo2
Servo myServo3; // Define 'myServo3' as Servo3
Servo myServo4; // Define 'myServo4' as Servo4

int angleV1 = 0; // Initialize variable to store Servo1 position/angle
int angleV2 = 0; // Initialize variable to store Servo2 position/angle
int angleV3 = 0; // Initialize variable to store Servo3 rotational speed
int angleV4 = 0; // Initialize variable to store Servo4 rotational speed

const int joystick_VRX = A0; // Connect the X-axis potentiometer to Arduino pin A0
const int joystick_VRY = A1; // Connect the Y-axis potentiometer to Arduino pin A1
const int joystick_VRX_2 = A2; // Connect the X_2-axis potentiometer to Arduino pin A2
const int joystick_VRY_2 = A3; // Connect the Y_2-axis potentiometer to Arduino pin A3

void setup() {
  Serial.begin(9600); // Define baud rate for serial communication
  pinMode(2, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 2'
  pinMode(4, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 4'

  myServo1.write(90); // Initialize Servo1 position/angle to mid-point (90 degrees)
  myServo2.write(90); // Initialize Servo2 position/angle to mid-point (90 degrees)
  myServo1.attach(5); // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // Attaches Servo2 Signal Line to pin 6 (PWM)

  myServo3.write(90); // Initialize Servo3 rotation speed to zero (90 degrees)
  myServo4.write(90); // Initialize Servo4 rotation speed to zero (90 degrees)
  myServo3.attach(3); // Attaches Servo3 Signal Line to pin 3 (PWM)
  myServo4.attach(11); // Attaches Servo4 Signal Line to pin 11 (PWM)
}

void loop() {
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int SW_State = !digitalRead(2); // Read the digital input of the Switch

  int X_2_ADC_Value = analogRead(joystick_VRX_2); // Read the analog/mechanical input of X_2-axis
  int Y_2_ADC_Value = analogRead(joystick_VRY_2); // Read the analog/mechanical input of Y_2-axis
  int SW_2_State = !digitalRead(4); // Read the digital input of the Switch_2

  // for the first servo: myServo1
  angleV1 = map(X_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of X-axis as degrees of range 0-180
  myServo1.write(angleV1);

  delay(100); // waits 100ms for the next command to be executed

  // for the second servo: myServo2
  angleV2 = map(Y_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of Y-axis as degrees of range 0-180
  myServo2.write(angleV2);

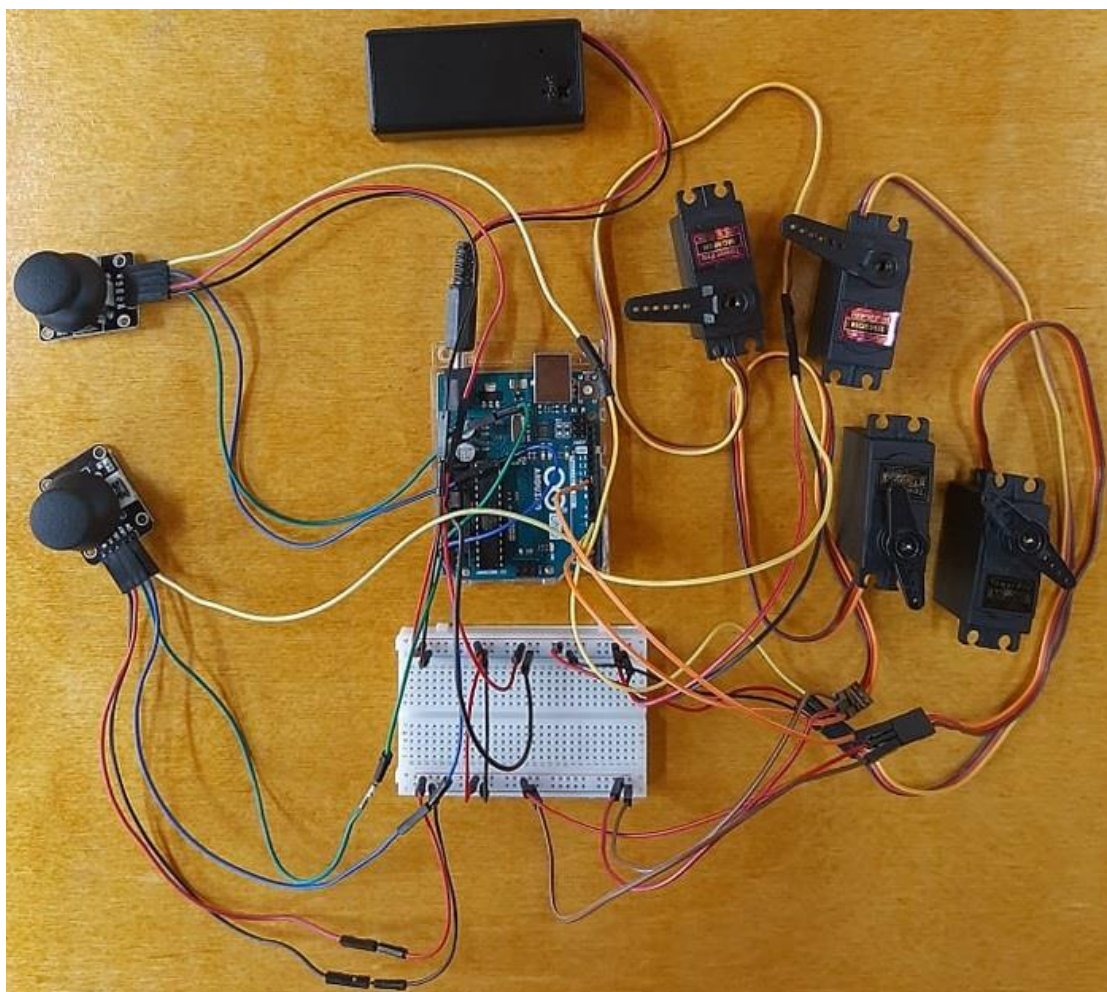
  // for the third servo: myServo3
  angleV3 = map(X_2_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of X_2-axis as degrees of range 0-180
  myServo3.write(angleV3); // use the variable to rotate the Servo Motor 3 continuously
  // "0 degrees" --> max clockwise rotation speed
  // max "180 degrees" --> max anticlockwise rotation speed

  delay(100); // waits 100ms for the next command to be executed

  // for the fourth servo: myServo4
  angleV4 = map(Y_2_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of Y_2-axis as degrees of range 0-180
  myServo4.write(angleV4); // use the variable to rotate the Servo Motor 4 continuously
  // "0 degrees" --> max clockwise rotation speed
  // max "180 degrees" --> max anticlockwise rotation speed
}

```

ΚΩΔΙΚΑΣ_13 Έλεγχος απόκρισης του Σ.Η.Ε.Ε ως σύστημα απόκρισης 2 εισόδων με 4 εξόδους

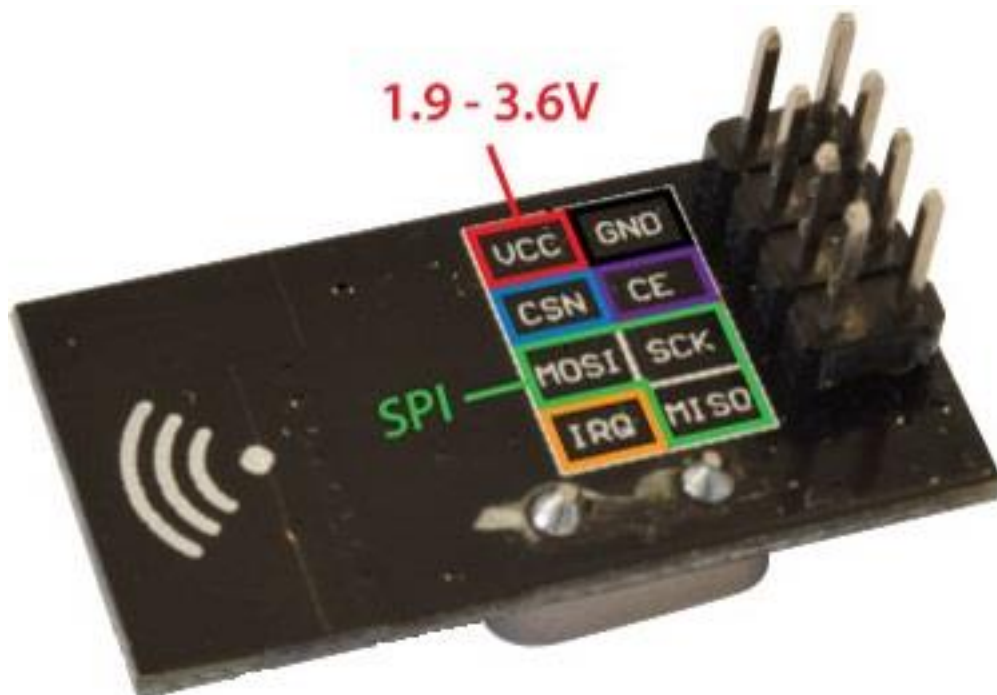


Εικόνα 3.59 Σύστημα Ηλεκτρονικού Ενσύρματου Ελεγκτή 2 εισόδων και 4 εξόδων, υλοποιημένο σε επαγγελματικό εργαστήριο

3.1.2 Σύνδεση πομποδέκτη NRF24L01 & Arduino NANO EVERY

Ως έγινε αναφορά και στην υποενότητα 2.4.1, λαμβάνοντας υπόψη τις προδιαγραφές κατασκευής για τους ακροδέκτες του πομποδέκτη NRF24L01:

- οι **VCC** και **GND** ακροδέκτες χρησιμοποιούνται για την τροφοδοσία του πομποδέκτη
- οι **MOSI**, **SCK** και **MISO** ακροδέκτες χρησιμεύουν για SPI επικοινωνία και άρα για σύνδεσή των με τους αντίστοιχους ακροδέκτες των Arduino NANO και UNO
- οι **CSN** και **CE** ακροδέκτες συνδέονται σε οποιονδήποτε ψηφιακό ακροδέκτη των NANO και UNO, θέτουν τον πομποδέκτη RF σε καταστάσεις “αναμονής/ενεργής” λειτουργίας (*standby/active mode*) και εκτελούν εναλλαγή μεταξύ λειτουργίας “μετάδοσης/εντολής” (*transmit/command mode*)
- ο **IRQ** είναι ακροδέκτης διακοπής προοριζόμενος για προγραμματισμένες διακοπές του NRF.



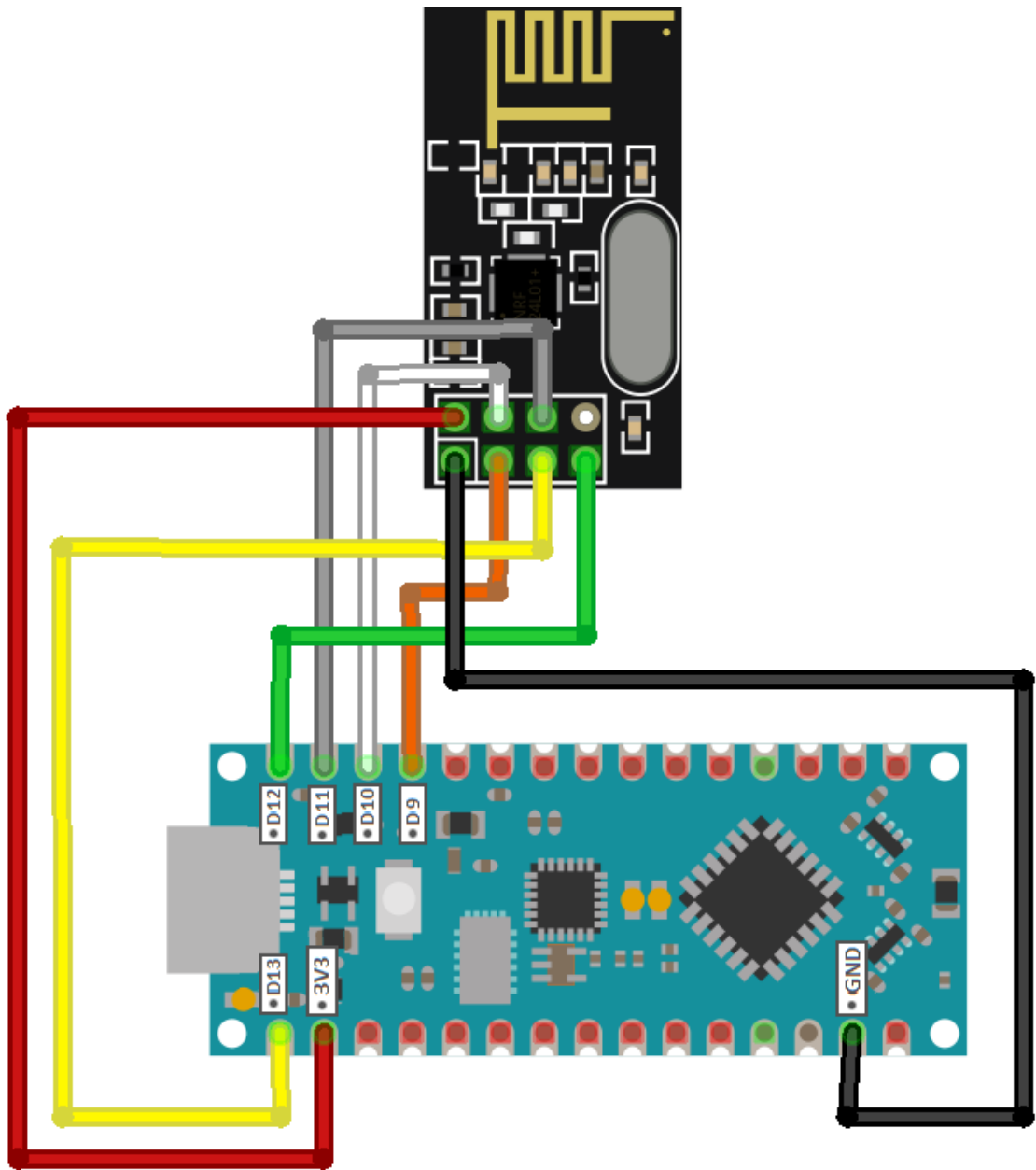
Εικόνα 3.60 Ακροδέκτες (pins) του πομποδέκτη NRF24L01

Βάσει των ανωτέρω, η συνδεσμολογία μεταξύ NRF24L01 και Arduino NANO παρουσιάζεται στον πίνακα 3_4 που ακολουθεί, με τον διαφορετικό χρωματισμό στα κελιά του πίνακα να σηματοδοτούν το χρώμα των αντίστοιχων καλωδίων που ενώνουν κατάλληλα τους ακροδέκτες.

ΠΙΝΑΚΑΣ 3_4

| | | | | | | | |
|--|-------------|------------|------------|-----------|-------------|-------------|------------|
| Ακροδέκτες πομποδέκτη NRF24L01 | VCC | GND | CSN | CE | MOSI | MISO | SCK |
| Ακροδέκτες Arduino NANO EVERY | 3.3V | GND | D10 | D9 | D11 | D12 | D13 |

Με επίτευξη της συνδεσμολογίας σύμφωνα με τα οριζόμενα του πίνακα 3_4, η διασύνδεση του πομποδέκτη NRF24L01 με την πλακέτα μικροελεγκτή Arduino NANO παρατίθεται στην εικόνα 3.61 της επόμενης σελίδας.



Εικόνα 3.61 Σύνδεση του πομποδέκτη NRF24L01 με τον μικροελεγκτή Arduino NANO EVERY

3.1.2.1 Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας

Στην παρούσα υποενότητα παρατίθεται ο κώδικας προγραμματισμού που αναπτύχθηκε για την επίτευξη της ασύρματης επικοινωνίας μεταξύ των Υποσυστημάτων Πομπού (*κύκλωμα TS*) και Δέκτη (*κύκλωμα RS*) και αφορά το κομμάτι του κυκλώματος TS.

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 10); // CE, CSN

//address through which 2 modules communicate.
const byte address[6] = "00001";

void setup()
{
  Serial.begin(9600);
  radio.begin();

  //set the address
  radio.openWritingPipe(address);

  //Set module as transmitter
  radio.stopListening();
}
void loop() {
  //Send message to receiver
  const char text[] = "Wireless messaging test !!!";
  radio.write(&text, sizeof(text));
  Serial.println(text);
  Serial.print("\n");
  delay(1000);
}
```

ΚΩΔΙΚΑΣ_14-1 Κώδικας λειτουργίας του κυκλώματος TS
ως Υποσύστημα Πομπού μονόδρομης επικοινωνίας


```
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!  
Wireless messaging test !!!
```

Εικόνα 3.62 Εμφάνιση σχετικού μηνύματος που αποστέλει το κύκλωμα TS στο κύκλωμα RS

3.2 Υποσύστημα Δέκτη (Κύκλωμα RS)

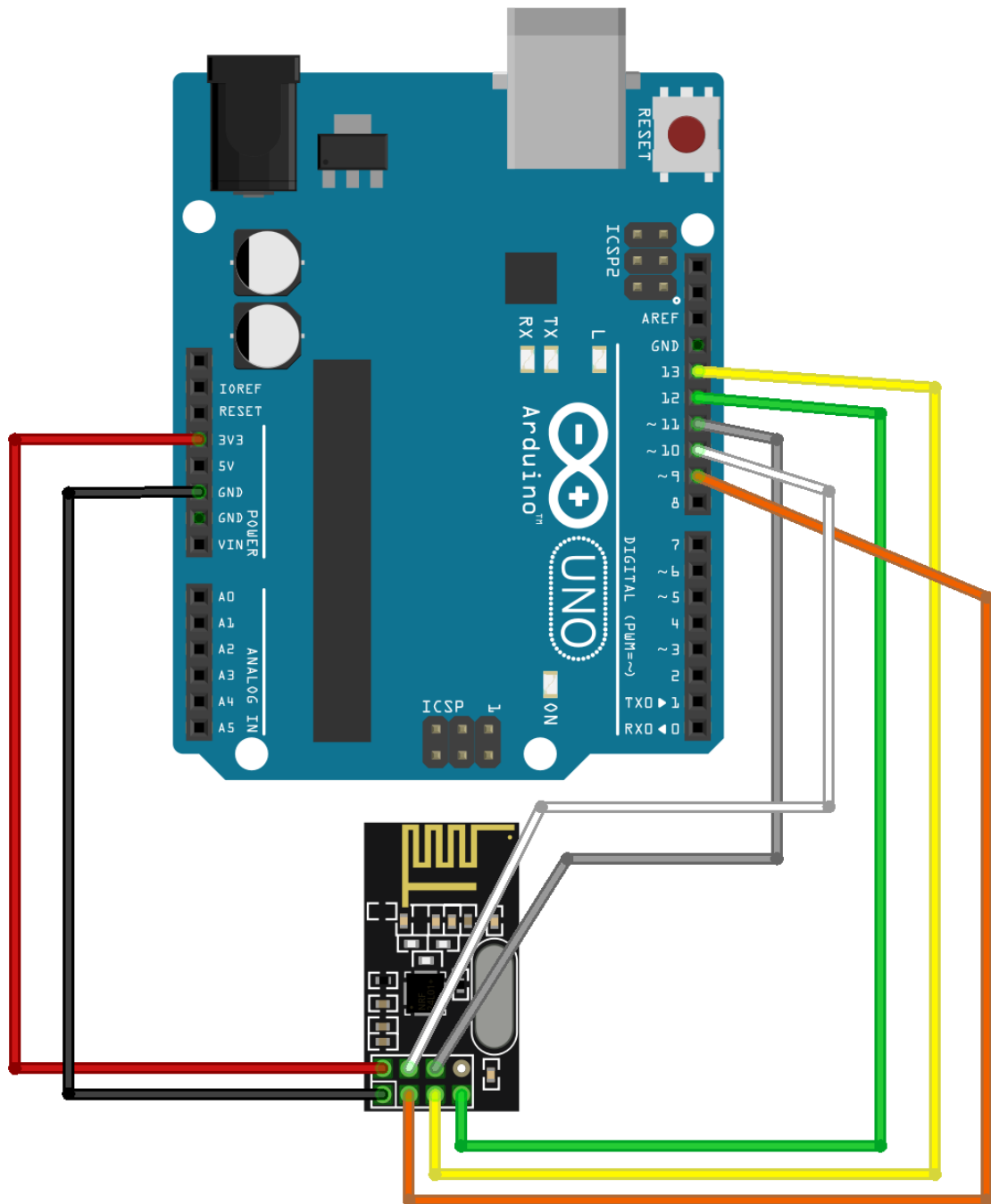
Εφόσον η ηλεκτρονική πλακέτα Arduino UNO R3 συνδέεται με τον πομποδέκτη NRF24L01 κατά τρόπο ίδιο με την πλακέτα NANO EVERY, ως έγινε και αναφορά στο κεφάλαιο 2 καθότι τα δύο μοντέλα μικροελεγκτών διαθέτουν τους ίδιους ακροδέκτες (το NANO περιλαμβάνει 2 επιπλέον pins), η συνδεσμολογία μεταξύ NRF24L01 και UNO παρουσιάζεται στην ακόλουθη υποενότητα.

3.2.1 Σύνδεση πομποδέκτη NRF24L01 & Arduino UNO R3

Ο τρόπος διασύνδεσης μεταξύ ακροδεκτών NRF24L01 και Arduino UNO παρουσιάζεται στον πίνακα 3_5 με χρωματισμό κελιών αντίστοιχο του χρώματος εκάστου καλωδίου διασύνδεσης ακροδεκτών, με την συνδεσμολογία καθ' αυτή να παρατίθεται στην εικόνα 3.62 της επόμενης σελίδας.

ΠΙΝΑΚΑΣ 3_5

| | | | | | | | |
|--------------------------------------|------|-----|-----|----|------|------|-----|
| Ακροδέκτες πομποδέκτη NRF24L01 | VCC | GND | CSN | CE | MOSI | MISO | SCK |
| Ακροδέκτες Arduino UNO R3 | 3.3V | GND | D10 | D9 | D11 | D12 | D13 |



Εικόνα 3.63 Σύνδεση του πομποδέκτη NRF24L01 με τον μικροελεγκτή Arduino UNO R3

3.2.1.1 Ανάπτυξη Κώδικα Επαλήθευσης Συνδεσμολογίας

Στην παρούσα υποενότητα παρατίθεται ο κώδικας προγραμματισμού που αναπτύχθηκε για την επίτευξη της ασύρματης επικοινωνίας μεταξύ των Υποσυστημάτων Πομπού (*κύκλωμα TS*) και Δέκτη (*κύκλωμα RS*) και αφορά το κομμάτι του κυκλώματος RS.

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 10); // CE, CSN

//address through which 2 modules communicate.
const byte address[6] = "00001";

void setup() {
  while (!Serial);
  Serial.begin(9600);

  radio.begin();

  //set the address
  radio.openReadingPipe(0, address);

  //Set module as receiver
  radio.startListening();
}

void loop() {
  //Read the data if available in buffer
  if (radio.available()) {
    Serial.print("The received message is: ");
    char text[32] = {0};
    radio.read(&text, sizeof(text));
    Serial.println(text);
    Serial.print("\n");
  }
  delay(1000);
}
```

*ΚΩΔΙΚΑΣ_15-1 Κώδικας λειτουργίας του κυκλώματος RS
ως Υποσύστημα Δέκτη μονόδρομης επικοινωνίας*

Κάνοντας χρήση του Serial Monitor (εικ. 3.63) για τον Δέκτη, παρατηρούμε ότι παραλαμβάνει ορθώς το επιθυμητό μήνυμα και άρα η ασύρματη επικοινωνία έχει πραγματοποιηθεί επιτυχώς.

```
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!  
The received message is: Wireless messaging test !!!
```

Εικόνα 3.64 Εμφάνιση σχετικού μηνύματος που δέχεται το κύκλωμα RS από το κύκλωμα TS

Καθώς όμως οι ανωτέρω προγραμματιστικοί κώδικες υποστηρίζουν μονόπλευρη επικοινωνία, στο σημείο αυτό γίνεται τροποποίηση εκάστου κώδικα για δημιουργία αμφίδρομης επικοινωνίας η οποία αποτελεί απαραίτητη προϋπόθεση για την μετέπειτα υλοποίηση του συστήματος ηλεκτρονικού ασύρματου ελεγκτή.

Στις επόμενες δύο σελίδες παρουσιάζονται οι κώδικες 14-2 & 15-2 οι οποίοι αποτελούν τροποποίηση των 14-1 & 15-1, αντίστοιχα, για μετατροπή της μονόδρομης ασύρματης επικοινωνίας μεταξύ των δύο Υποσυστημάτων σε αμφίδρομη.

Κώδικας Κυκλώματος TS για αμφίδρομη επικοινωνία

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(7, 8); // CE, CSN

//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};

void setup() {
  Serial.begin(9600);
  radio.begin();

  //set the address
  radio.openWritingPipe(addresses[1]); // 00002
  radio.openReadingPipe(1, addresses[0]); // 00001
}

void loop() {
  //Set module as transmitter
  radio.stopListening();
  //Send message to receiver
  Serial.print("The TS Circuit sends: ");
  const char text[] = "Wireless messaging test!";
  Serial.print(text);
  Serial.println("\n");
  radio.write(&text, sizeof(text));
  delay(200);

  //Set module as receiver
  radio.startListening();
  while (!radio.available());
  char text_back[32] = {0};
  Serial.print("So, the RS Circuit responds: ");
  radio.read(&text_back, sizeof(text_back));
  Serial.println(text_back);
  Serial.println("\n");
  delay(200);
}
```

ΚΩΔΙΚΑΣ_14-2 Κώδικας λειτουργίας του κυκλώματος TS
ως Υποσύστημα Πομπού αμφίδρομης επικοινωνίας

Κώδικας Κυκλώματος RS για αμφίδρομη επικοινωνία

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(7, 8); // CE, CSN

//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};

void setup() {
  Serial.begin(9600);
  radio.begin();
  //set the address
  radio.openWritingPipe(addresses[0]); // 00001
  radio.openReadingPipe(1, addresses[1]); // 00002
}

void loop() {
  //Set module as receiver
  radio.startListening();

  //Read the data if available in buffer
  if (radio.available()) {
    while (radio.available()) {
      Serial.print("Message from the TS Circuit: ");
      char text[32] = {0};
      radio.read(&text, sizeof(text));
      Serial.print(text);
      Serial.print("\n");
    }
    delay(200);
    //Set module as transmitter
    radio.stopListening();
    Serial.print("As a reply, the RS Circuit sends back: ");
    const char text_back[] = "The test was a success!!";
    radio.write(&text_back, sizeof(text_back));
    Serial.print(text_back);
    Serial.println("\n");
  }
  delay(200);
}
```

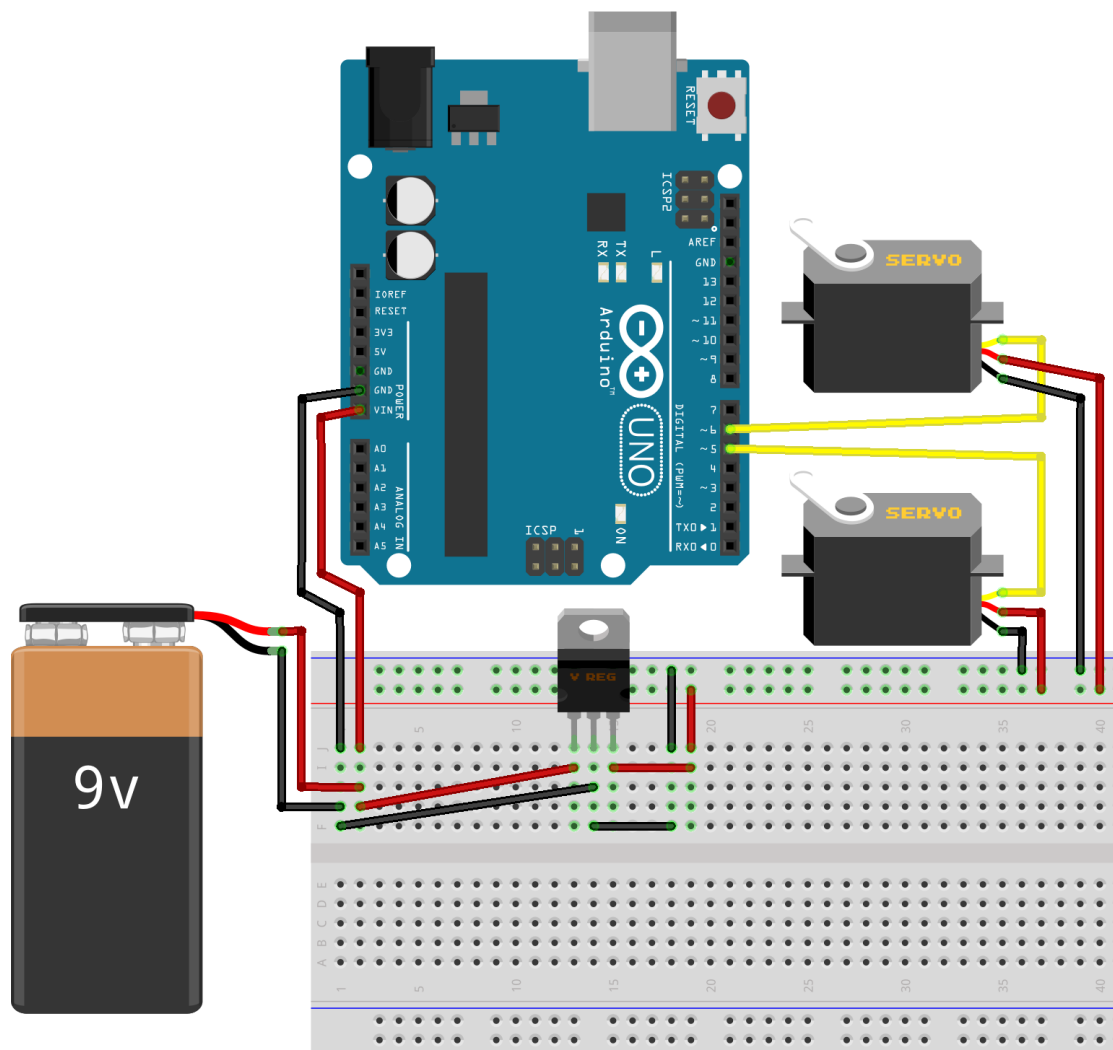
ΚΩΔΙΚΑΣ_15-2 Κώδικας λειτουργίας του κυκλώματος RS
ως Υποσύστημα Δέκτη αμφίδρομης επικοινωνίας

3.2.2 Σύνδεση 2 Σερβοκινητήρων & Arduino UNO R3

Λόγω της ιδιαίτερης ευελιξίας των κυκλωμάτων ως προς τον τρόπο διασύνδεσης μεταξύ των μερών που τα απαρτίζουν, παρέχεται αρκετά μεγάλη ελευθερία επιλογής στο κομμάτι του σχεδιασμού και της μετέπειτα υλοποίησης.

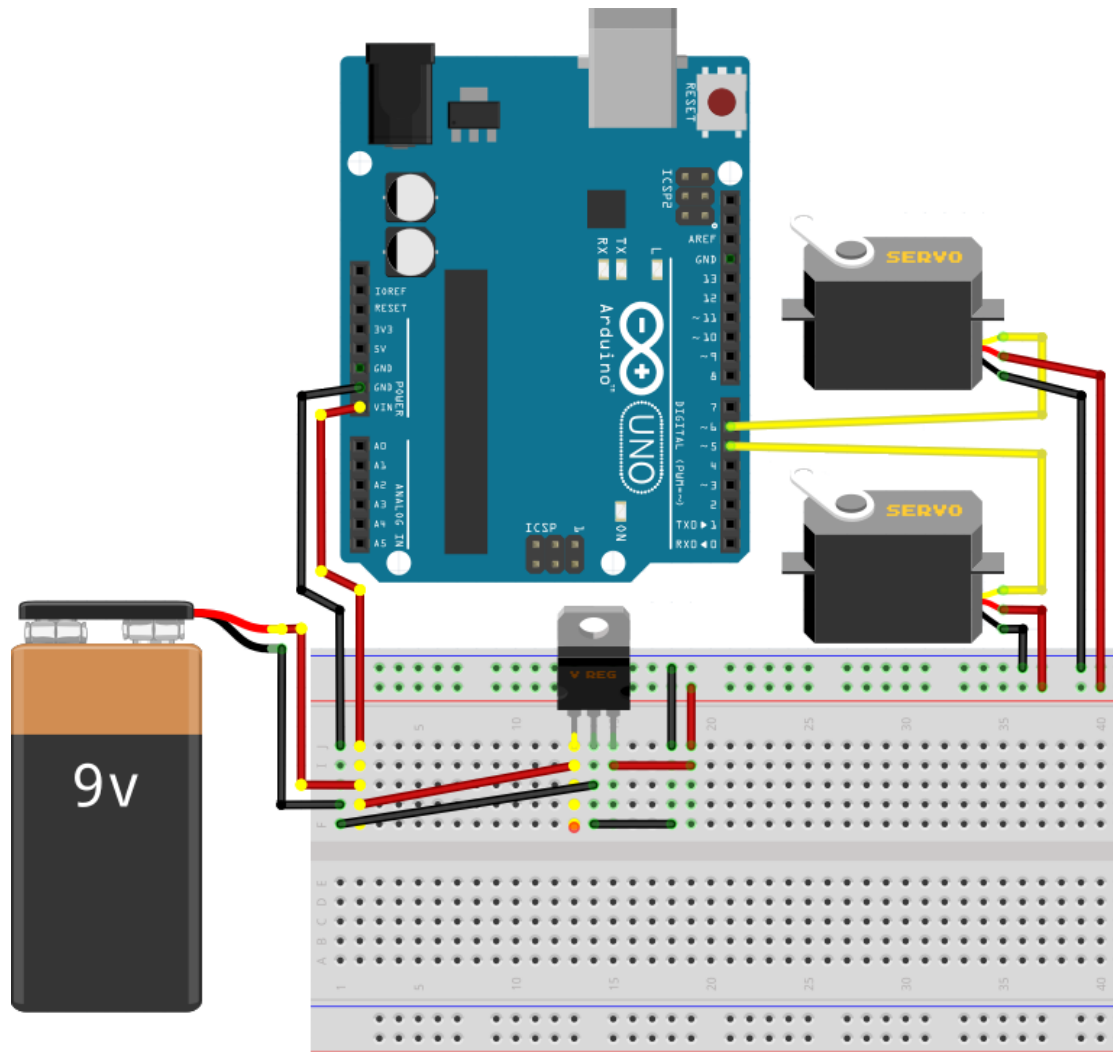
Για παράδειγμα, στο παρόν κύκλωμα θα μπορούσε να γίνει χρήση μιας μπαταρίας των 9V για απευθείας τροφοδοσία του Arduino UNO, όπως επίσης και δύο μπαταριοθηκών με δυνατότητα παροχής 6V εκάστη για απευθείας διασύνδεσή των με τους σερβοκινητήρες.

Αντ' αυτού, έγινε χρήση μίας και μόνο μπαταρίας των 9V και ενός ρυθμιστή τάσης (*voltage regulator*) στα 5V, όπου με κατάλληλη συνδεσμολογία (εικ. 3.65) παρέχεται φορητή τροφοδοσία στο κύκλωμα από μια και μόνο πηγή.

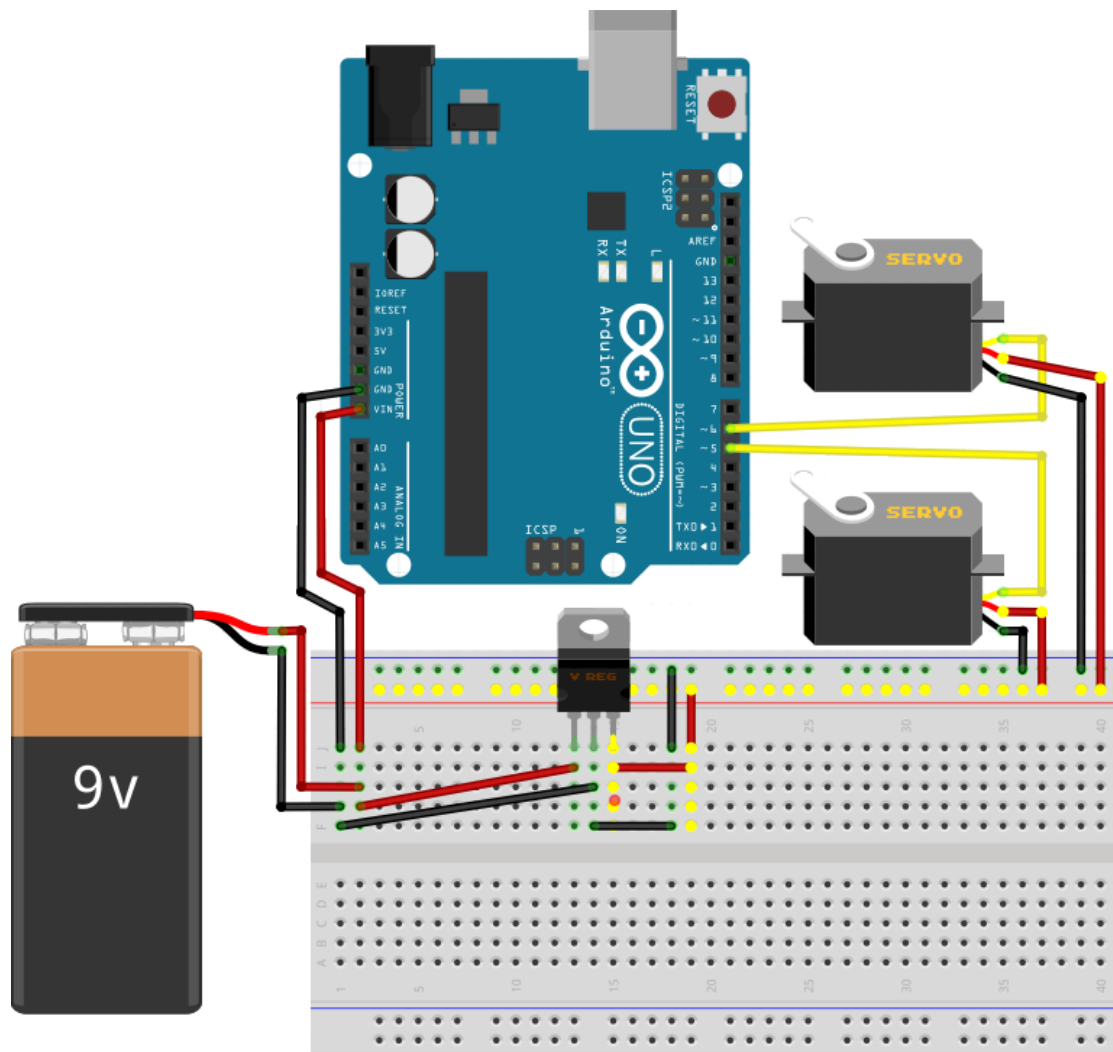


Εικόνα 3.65 Διασύνδεση του Arduino UNO R3 με τους 2 σερβοκινητήρες

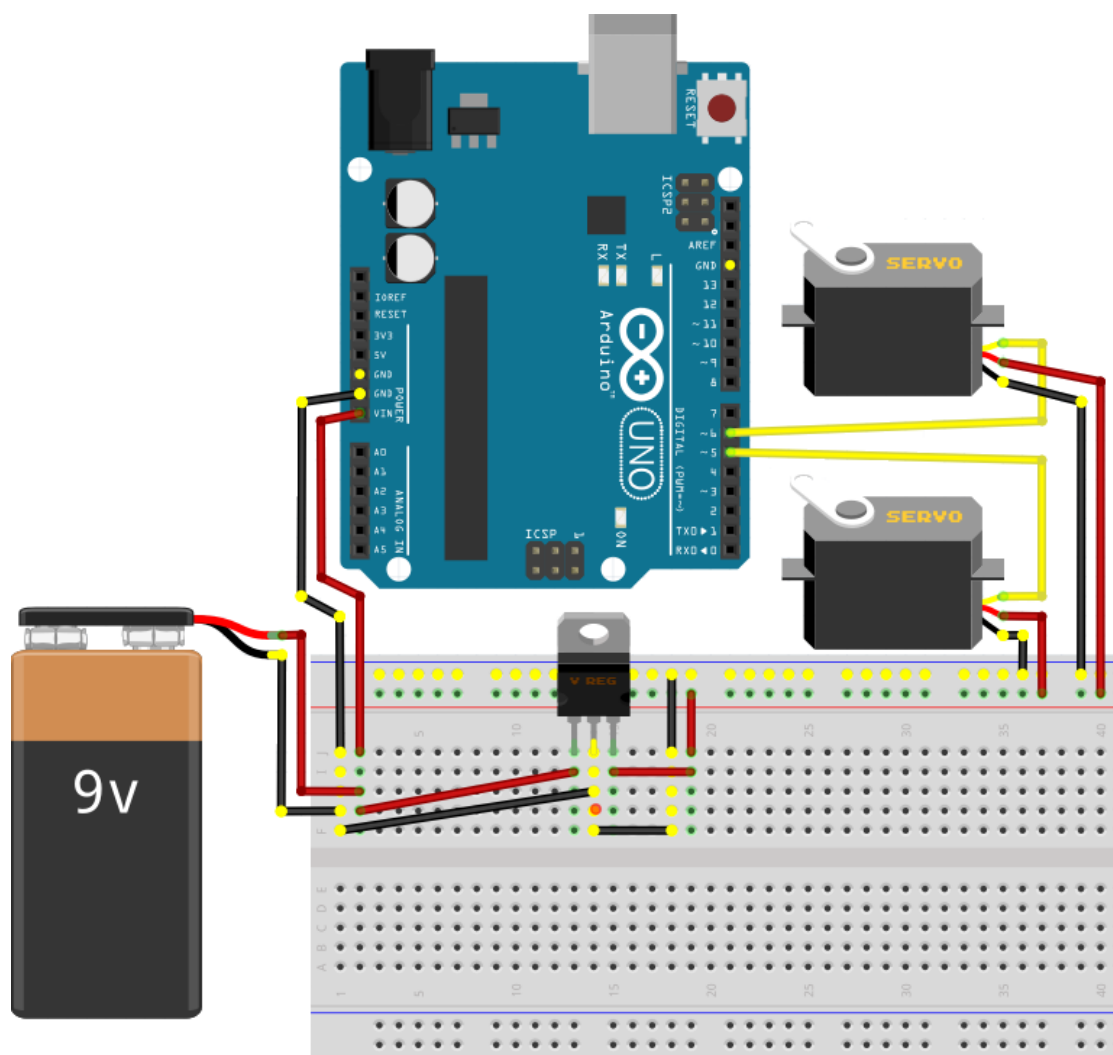
Με χρήση αυτού του τρόπου διασύνδεσης και βάσει των κατωτέρω τριών εικόνων (3.66, 3.67 και 3.68) γίνεται αντιληπτή η ύπαρξη κοινού δυναμικού από παροχή τάσης της 9V μπαταρία σε συγκεκριμένα σημεία του κυκλώματος και αντίστοιχες τιμές (σε Volt) των σημείων αυτών που φέρουν δυναμικό.



Εικόνα 3.66 Σημεία δυναμικού VCC στο κύκλωμα (κίτρινες κουκίδες) λόγω ύπαρξης τάσης τροφοδοσίας από μπαταρία των 9V. Τα σημεία αυτά φέρουν όλα τιμή δυναμικού 9V



Εικόνα 3.67 Σημεία δυναμικού V στο κύκλωμα (κίτρινες κουκίδες) λόγω διαμεσολάβησης ρυθμιστή τάσης των 5V μεταξύ διασύνδεσης της 9V μπαταρίας και των 2 σερβοκινητήρων. Τα εν λόγω σημεία φέρουν όλα τους δυναμικό τιμής +5V



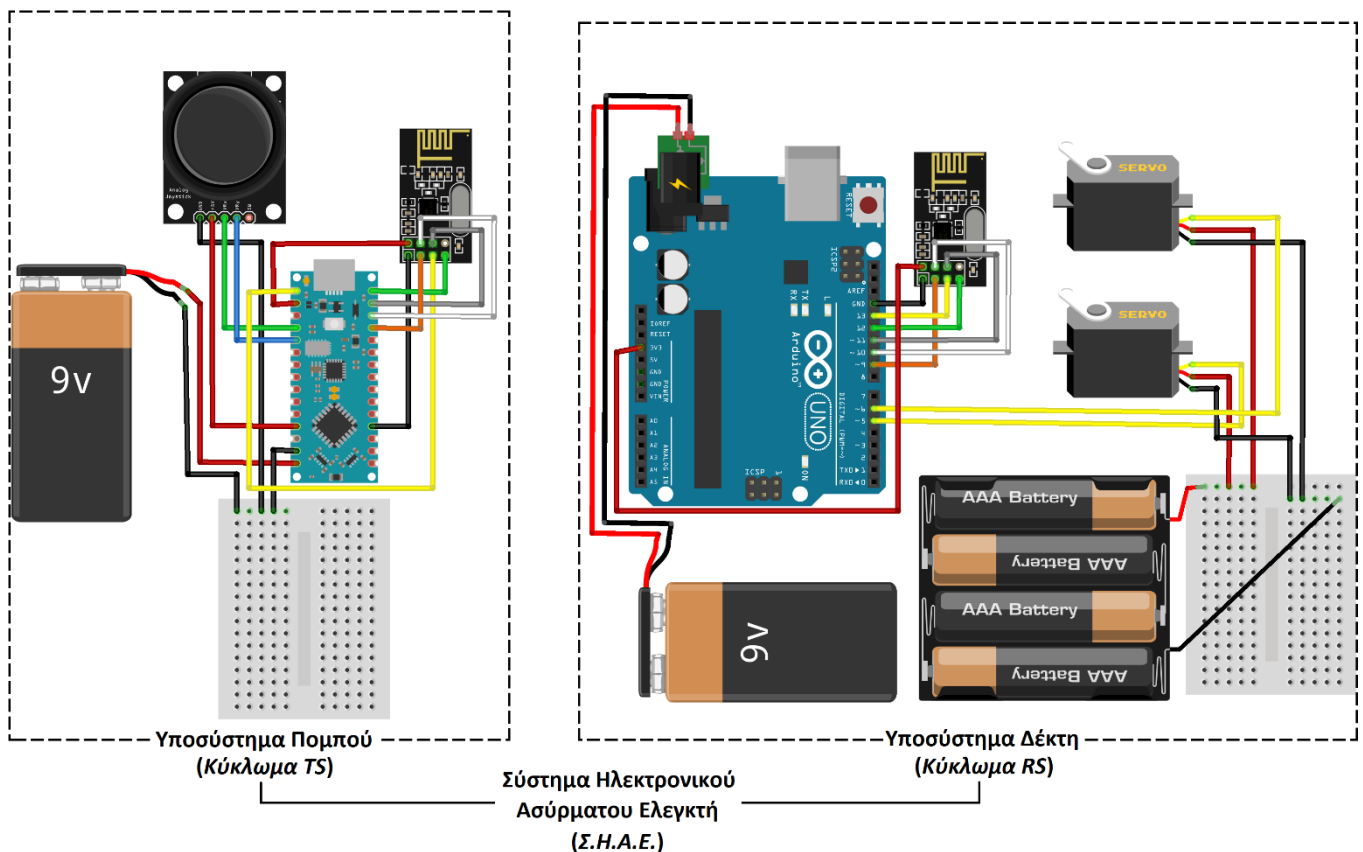
Εικόνα 3.68 Σημεία δυναμικού 0V (GND) στο κύκλωμα (κίτρινες κουκίδες) λόγω ύπαρξης τάσης από την 9V μπαταρία. Τα σημεία αυτά φέρουν δυναμικό τιμής 0V (γείωση)

Ο προγραμματιστικός κώδικας για την λειτουργία του κυκλώματος της ανωτέρω συνδεσμολογίας είναι ο ίδιος που χρησιμοποιήθηκε στην υποενότητα 3.1.1.2 και συγκεκριμένα για την περίπτωση του ελέγχου απόκρισης 2 σερβοκινητήρων με εντολές από τον μικροελεγκτή Arduino χωρίς την ύπαρξη αναλογικού μοχλού.

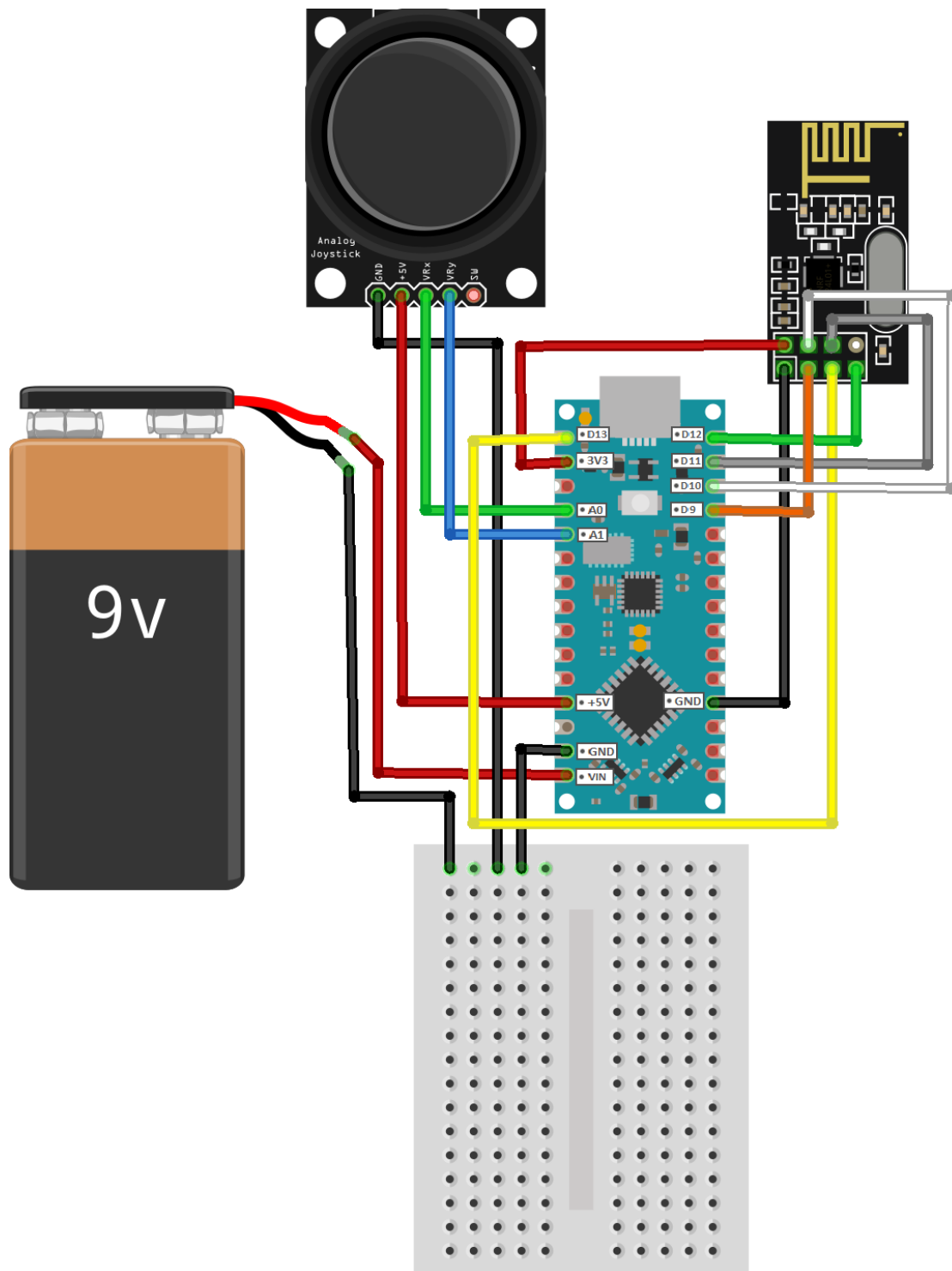
Με την ολοκλήρωση των απαραίτητων συνδέσεων, δοκιμών και ελέγχων να πραγματοποιούνται επιτυχώς στις ως άνω δύο ενότητες του παρόντος κεφαλαίου, στην 3^η και τελευταία ενότητα που ακολουθεί εφαρμόζεται ο τελικός σχεδιασμός, ο προγραμματισμός και η υλοποίηση του Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) ως σύστημα απόκρισης εισόδου/εξόδου για Βιομηχανικές Εφαρμογές.

3.3 Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) για Βιομηχανικές Εφαρμογές

Λαμβάνοντας υπόψη όλες τις διασυνδέσεις, τις παραμέτρους και τις διαδικασίες που πραγματοποιήθηκαν ως τώρα και εφαρμόζοντάς τες ξανά για την δημιουργία του ζητούμενου ηλεκτρονικού κυκλώματος στην ολοκληρωμένη του πλέον μορφή, προκύπτει το σχεδιαστικά ολοκληρωμένο κύκλωμα του Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) το οποίο παρουσιάζεται αυτούσιο στην κάτωθι εικόνα 3.69 αλλά και ως δύο ξεχωριστά υποσυστήματα (κύκλωμα TS & RS) στις αμέσως δύο επόμενες σελίδες (εικ. 3.70 & 3.71) για αναλυτικότερη απεικόνιση των μερών που το απαρτίζουν.

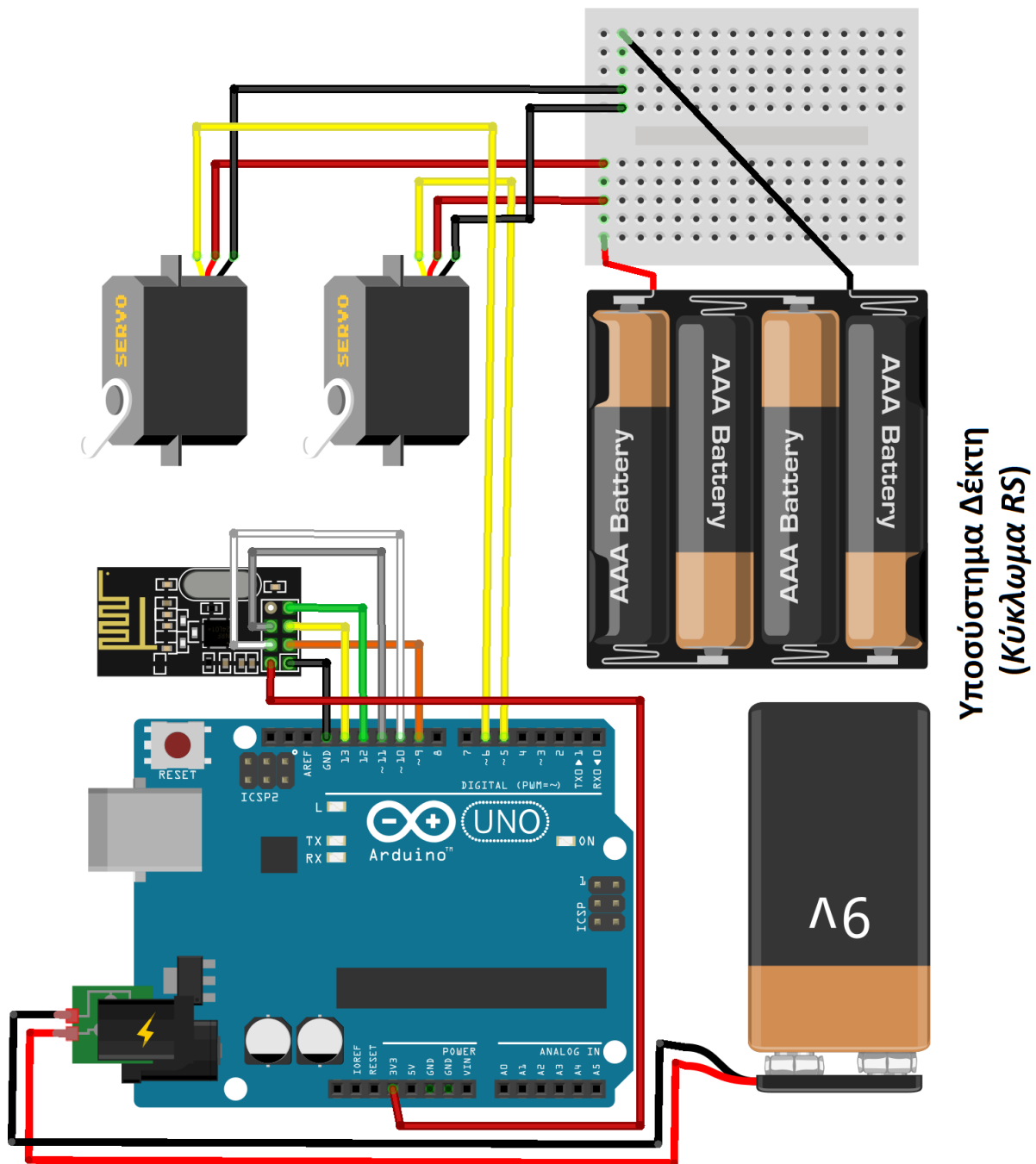


Εικόνα 3.69 Ολοκληρωμένη σχεδίαση του Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές



Υποσύστημα Πομπού (Κύκλωμα TS)

Εικόνα 3.70 Κύκλωμα TS του συστήματος ηλεκτρονικού ασύρματου ελεγκτή



Εικόνα 3.71 Κύκλωμα RS του συστήματος ηλεκτρονικού ασύρματου ελεγκτή

Παρά το ότι ο σχεδιασμός του Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή έχει ολοκληρωθεί, για να αποκτήσει ζωή, λειτουργικότητα και χρησιμότητα, απαιτείται δημιουργία και εγκατάσταση σε αυτό κατάλληλου προγραμματιστικού κώδικα.

Η λειτουργία του ανωτέρω Σ.Η.Α.Ε. είναι η ασύρματη αποστολή πληροφοριών από το Υποσύστημα Πομπού στο Υποσύστημα Δέκτη.

Αρχικά το *Υποσύστημα Πομπού* ή αλλιώς *Κύκλωμα TS* αποστέλλει την μηχανική κίνηση των 2 αναλογικών μοχλών του –εκφρασμένη σε μοίρες– στο *Υποσύστημα Δέκτη* ή αλλιώς *Κύκλωμα RS* με αποτέλεσμα την περιστροφική κίνηση του άξονα των 2 ψηφιακών σερβοκινητήρων κατά τιμή μοιρών ίδια με αυτή που έλαβε ο Δέκτης από τον Πομπό.

Η έκφραση της μηχανικής κίνησης των μοχλών μετατρέπεται με κατάλληλο προγραμματισμό του μικροελεγκτή Arduino NANO EVERY σε μοίρες ώστε η αντιστοίχιση της ολίσθησης των μοχλών σε μοίρες με την ολίσθηση των σερβοκινητήρων στις μοίρες αυτές να καθίσταται τόσο απλοϊκή στον χειρισμό του Σ.Η.Α.Ε. όσο και ευανάγνωστη μέσω του Serial Monitor, από τον χρήστη.

Στις υποενότητες 3.3.1 & 3.3.2 που ακολουθούν παρουσιάζονται λεπτομερώς οι κώδικες προγραμματισμού που αναπτύχθηκαν για τα κυκλώματα TS & RS και στην υποενότητα 3.3.3 πραγματοποιείται έλεγχος ορθής λειτουργίας του Σ.Η.Α.Ε. ως ασύρματο σύστημα απόκρισης I/O.

3.3.1 Ανάπτυξη Κώδικα για το Κύκλωμα TS

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 10); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "00001";

// include an analog joystick of 2-axis
const int joystick_VRX = A0; // connect the X-axis potentiometer to Arduino pin A0
const int joystick_VRY = A1; // connect the Y-axis potentiometer to Arduino pin A1

void setup() {
  Serial.begin(9600);
  radio.begin();
  //set the address
  radio.openWritingPipe(address);
}

void loop() {
  //Set module as transmitter
  radio.stopListening();

  // calculation process of joystick values
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis

  float X_Volt = ( X_ADC_Value * 5.0 ) / 1023 ; // Convert digital value to Voltage
  float Y_Volt = ( Y_ADC_Value * 5.0 ) / 1023 ; // Convert digital value to Voltage
  delay(100);

  // variables to store joystick values
  int angleV1 = map(X_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of X-axis as degrees of range 0-180
  int angleV2 = map(Y_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input of Y-axis as degrees of range 0-180
  delay(100); // waits 100ms for the servo to reach the corresponding angle

  // Send joystick information (status & values) to receiver
  radio.write(&angleV1, sizeof(angleV1));
  Serial.print("Joystick value for X-axis: ");
  Serial.print(angleV1);
  Serial.println("");
  delay(25);

  radio.write(&angleV2, sizeof(angleV2));
  Serial.print("Joystick value for Y-axis: ");
  Serial.print(angleV2);
  Serial.println("");
  Serial.print("\n");
  delay(500);
}
```

ΚΩΔΙΚΑΣ_16 Κώδικας λειτουργίας για το Κύκλωμα TS του Σ.Η.Α.Ε.

3.3.2 Ανάπτυξη Κώδικα για το Κύκλωμα RS

Η διαδικασία που ακολουθείται εδώ είναι όμοια με αυτήν της προηγούμενης υποενότητας και ο προγραμματιστικός κώδικας είναι ο κατωτέρω:

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>      // add the servo library

//create an RF24 object
RF24 radio(9, 10); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "00001";

// include 2 servos of 180 degree rotation
Servo myServo1;      // Define 'myServo1' as Servo1
Servo myServo2;      // Define 'myServo2' as Servo2

void setup() {
  Serial.begin(9600);
  radio.begin();
  //set the address
  radio.openReadingPipe(0, address);

  myServo1.attach(5);      // attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6);      // attaches Servo2 Signal Line to pin 6 (PWM)
}

void loop() {
  //Set module as receiver
  radio.startListening();

  //Read the data if available in buffer
  if (radio.available()) {
    while (radio.available()) {
      int angleV_1 = 0;      // Initialize the variable to store Joystick values of X-axis for Servo No1
      int angleV_2 = 0;      // Initialize the variable to store Joystick value of Y-axis for Servo No2

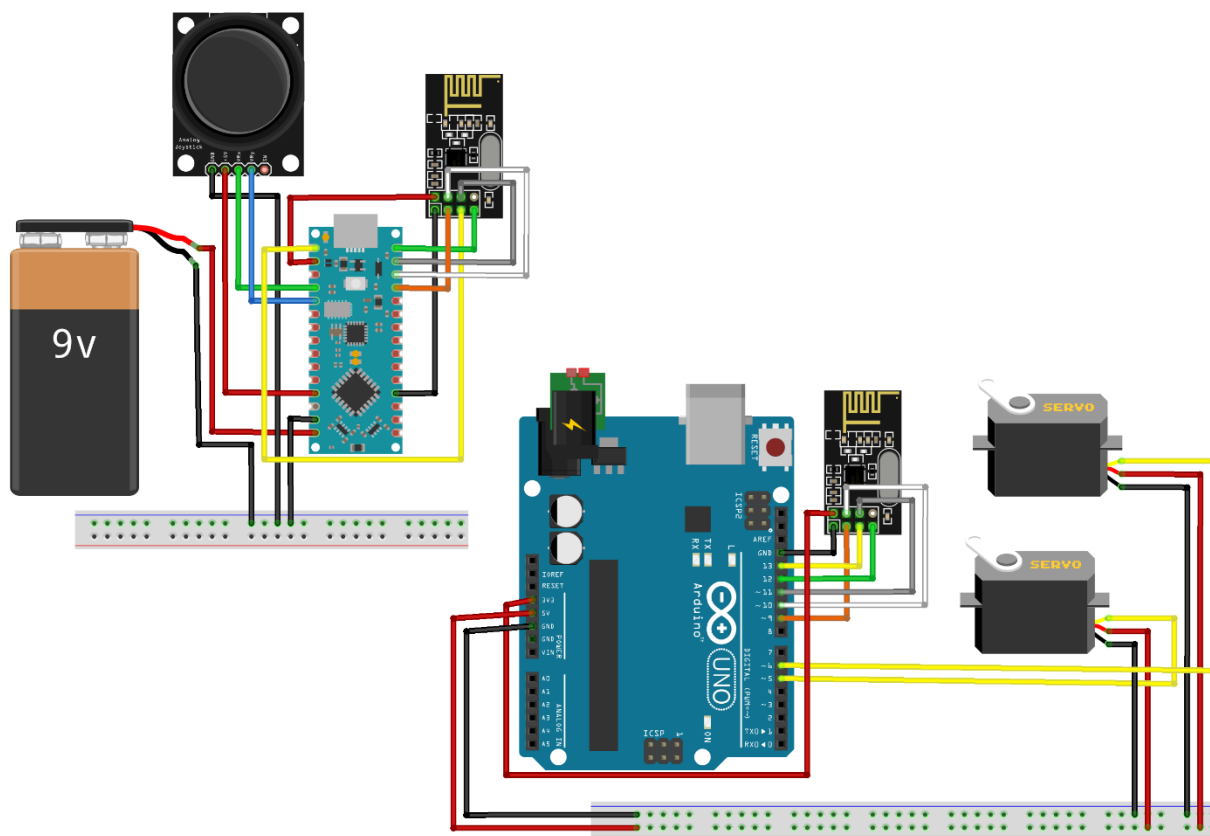
      // Receive joystick information to rotate the 2 Servo motors accordingly
      radio.read(&angleV_1, sizeof(angleV_1));
      Serial.print("Servo 1 position/angle is: ");      // read & store joystick X-axis information
      Serial.print(angleV_1);
      Serial.println("");
      myServo1.write(angleV_1);      // use the variable value to rotate Servo 1
      delay(100);

      radio.read(&angleV_2, sizeof(angleV_2));      // read & store joystick Y-axis information
      Serial.print("Servo 2 position/angle is: ");
      Serial.print(angleV_2);
      Serial.println("");
      myServo2.write(angleV_2);      // use the variable value to rotate Servo 2
      Serial.print("\n");
    }
  }
  delay(100);
}
```

ΚΩΔΙΚΑΣ_17 Κώδικας λειτουργίας για το Κύκλωμα RS του Σ.Η.Α.Ε.

3.3.3 Έλεγχος Απόκρισης I/O για λειτουργία του συστήματος ως Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.)

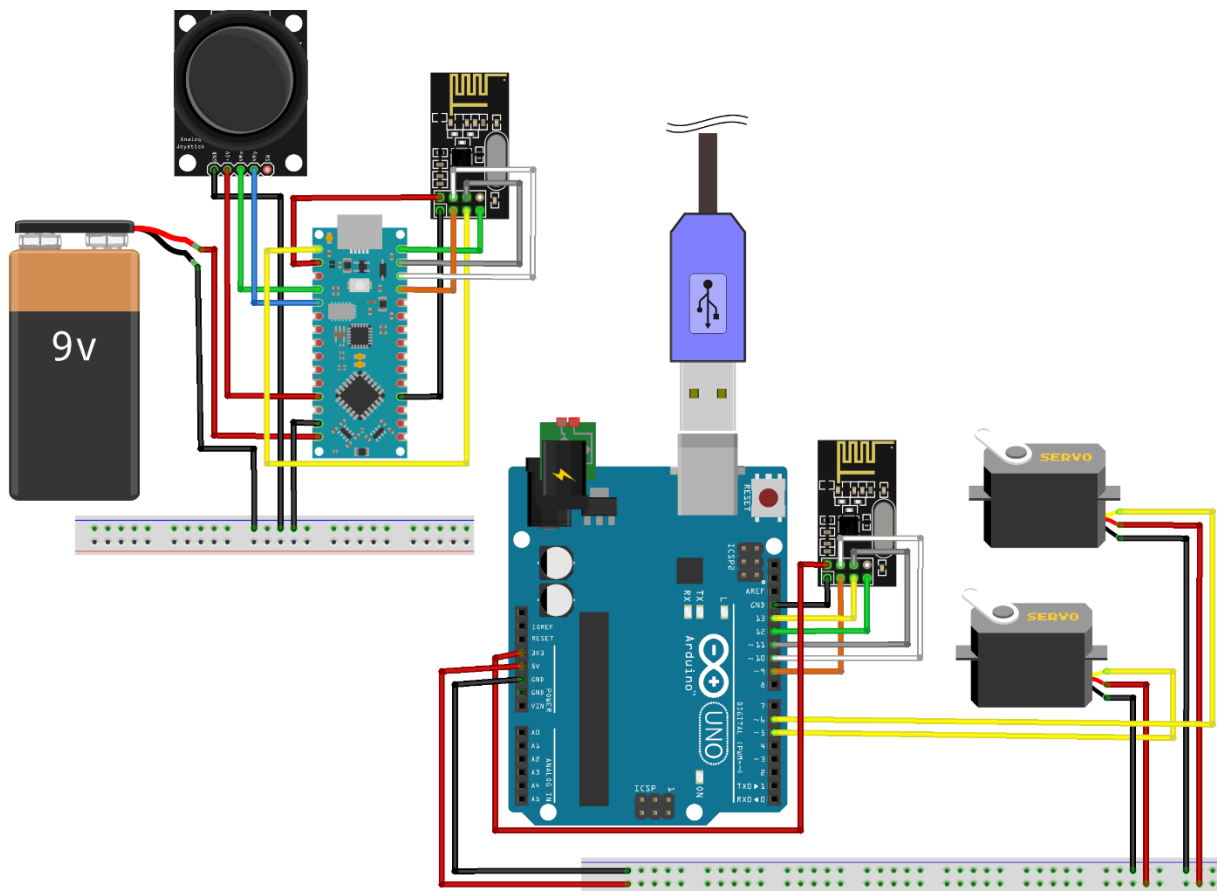
Με την ανάπτυξη κατάλληλου προγραμματιστικού κώδικα για ένα υποσύστημα και «ανέβασμά» τους σε αυτά, έχει πλέον ολοκληρωθεί τόσο ο σχεδιασμός και ο προγραμματισμός όσο και η υλοποίηση συστήματος ηλεκτρονικού ασύρματου ελεγκτή, με τις εικόνες και τα παραδείγματα που έπονται να παρουσιάζουν την ορθή και εύρυθμη λειτουργία του ως σύστημα απόκρισης εισόδου/εξόδου που δύναται να προσαρμόζεται καταλλήλως για χρήση σε ευρύ φάσμα Βιομηχανικών και μη Εφαρμογών.



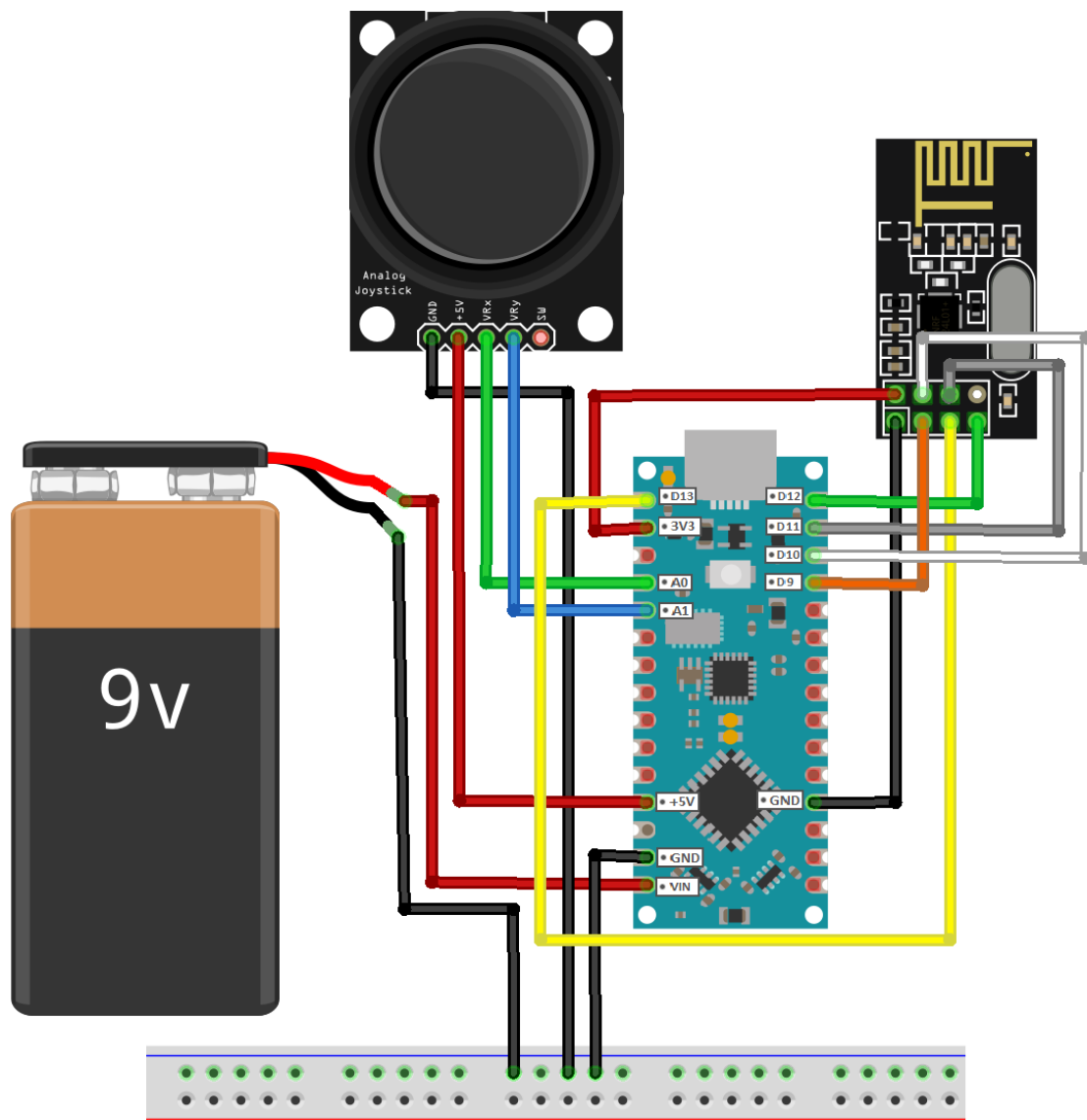
Εικόνα 3.72 Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές

Λόγω της αρκετά μεγάλης ευελιξίας συνδεσμολογίας που παρέχει ο μικροελεγκτής UNO λόγω κατασκευής του, το κύκλωμα RS δύναται να λειτουργεί με πληθώρα διαφορετικών μεθόδων τροφοδοσίας, είτε αυτές περιλαμβάνουν φορητές συσκευές όπως μπαταρίες, power banks και μικρές φορητές γεννήτριες είτε όχι όπως το δίκτυο της ΔΕΗ και οι γνωστές στην βιομηχανία μη φορητές γεννήτριες.

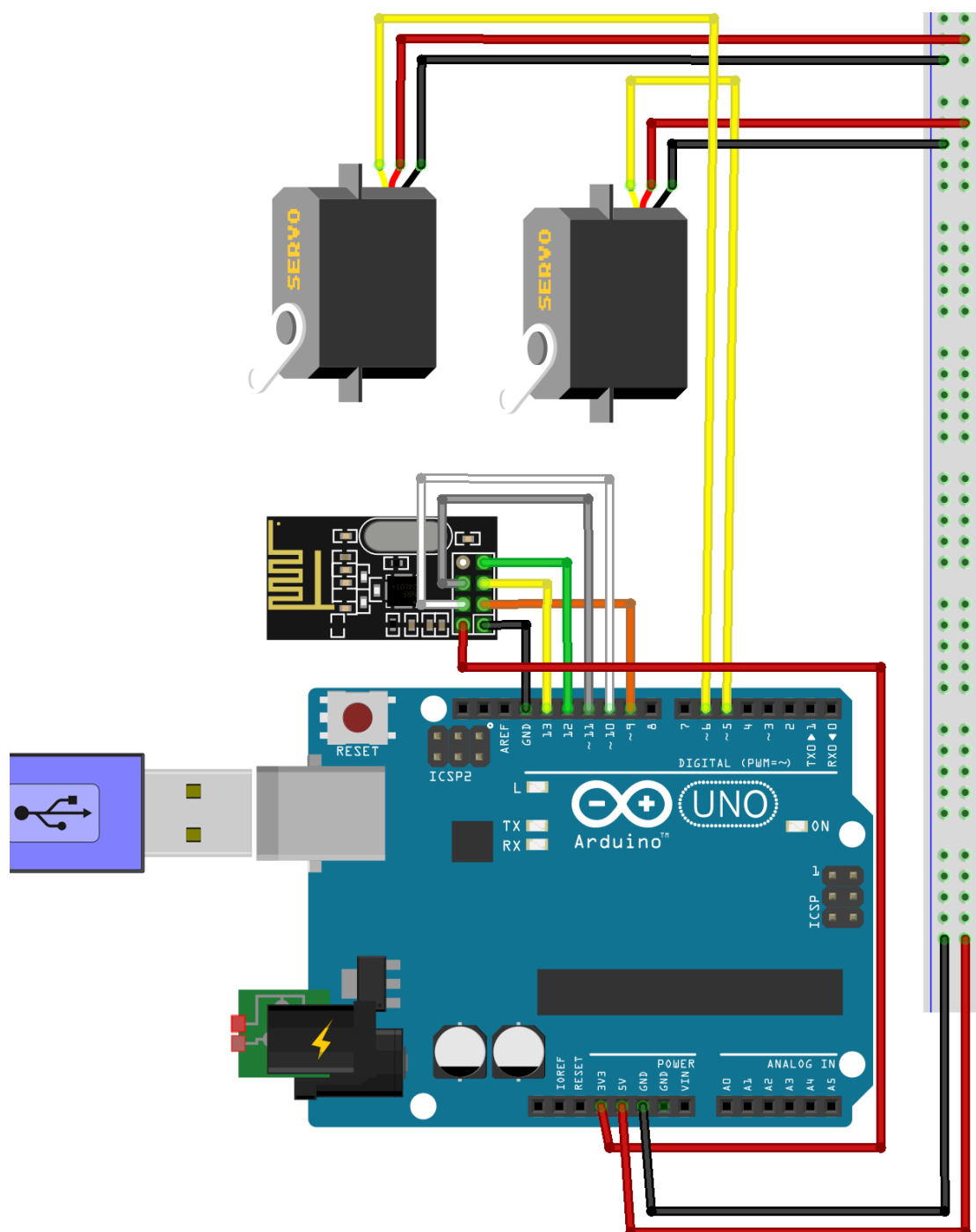
Με αφορμή καθαρά και μόνο την εύρυθμη λειτουργία του συστήματος αλλά και λαμβάνοντας υπόψη το γεγονός ότι το Arduino UNO διαθέτει θύρα συμβατή με καλώδιο της πλέον διαδεδομένης θύρας USB και η οποία συναντάται σχεδόν σε όλες τις φορητές και μη συσκευές παροχής τροφοδοσίας, για αυτήν την εργασία η τροφοδοσία παρέχεται κατά κόρων από καλώδιο USB για το Κύκλωμα RS, δεν παύει όμως να διατηρείται και ξεχωριστή αναμονή για επιθυμητή μελλοντική σύνδεση με μπαταρία.



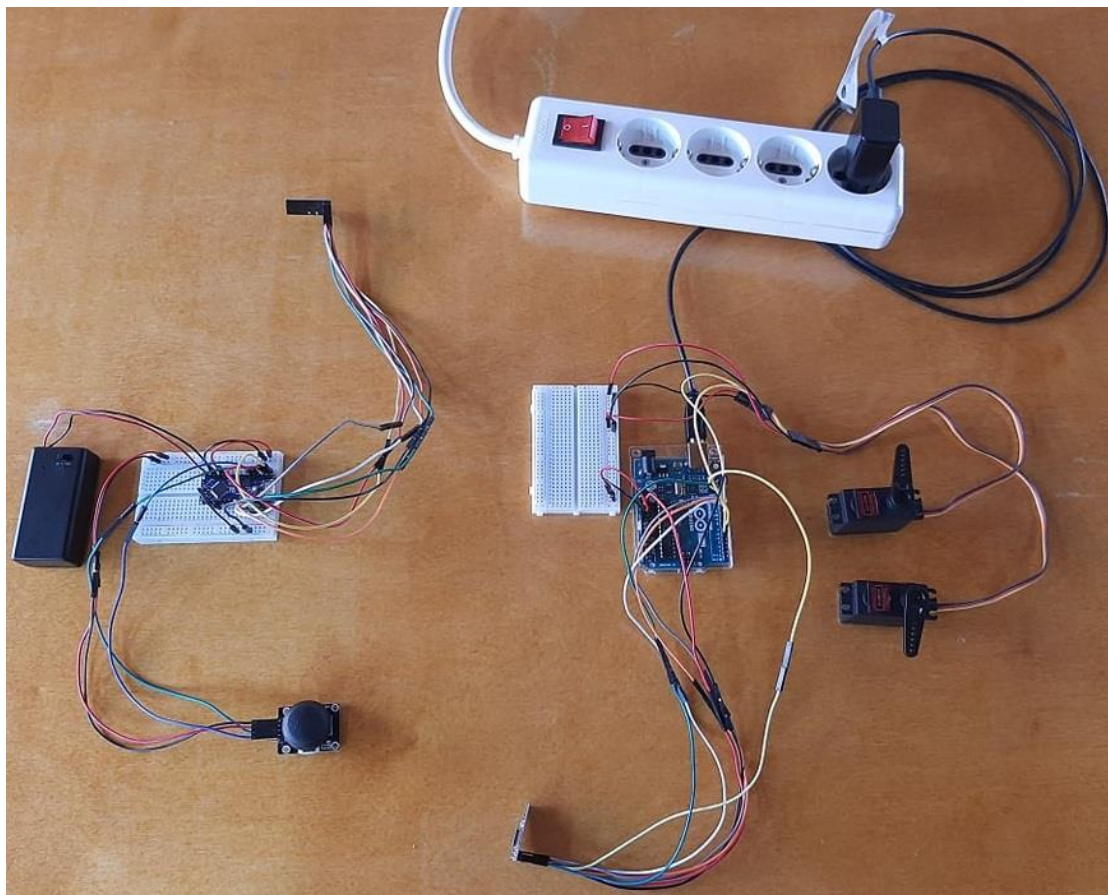
Εικόνα 3.73 Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές, εναλλακτικού τρόπου τροφοδοσίας, σε περιβάλλον προσομοίωσης



Εικόνα 3.74 Υποσύστημα Πομπού του συστήματος ηλεκτρονικού ασύρματου ελεγκτή ή αλλιώς Κύκλωμα TS



Εικόνα 3.75 Υποσύστημα Δέκτη του συστήματος ηλεκτρονικού ασύρματου ελεγκτή ή αλλιώς Κύκλωμα RS



Εικόνα 3.76 Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές, υλοποιημένο σε επαγγελματικό εργαστήριο

Με χρήση του Serial Monitor για το Κύκλωμα TS, εμφανίζονται στην εικόνα 3.77 οι τιμές των 2 ποτενσιομέτρων του αναλογικού μοχλού για τους X & Y άξονες αντίστοιχα, μετατρεπόμενες σε μοίρες για ευκολία έκφρασης και αντιστοίχισης των ολισθήσεων μεταξύ μοχλού και σερβοκινητήρων. Συγκεκριμένα, στην εν λόγω εικόνα εμφανίζονται οι τιμές των 2 ποτενσιομέτρων όταν ο μοχλός βρίσκεται σε αδράνεια.

```
Joystick value for X-axis: 93°  
Joystick value for Y-axis: 89°
```

Εικόνα 3.77 Τιμές των 2 ποτενσιομέτρων του αναλογικού μοχλού σε αδράνεια

Με χρήση του Serial Monitor για το Κύκλωμα RS, εμφανίζονται στην εικόνα 3.78 οι τιμές που λαμβάνει επιτυχώς το Κύκλωμα RS από την εκάστοτε θέση που βρίσκεται ο μοχλός του Κυκλώματος TS.

Στην εικόνα αυτή, το πρώτο ζεύγος ενδείξεων παρουσιάζει τις τιμές του μοχλού όταν αυτός βρίσκεται σε αδράνεια.

Όταν ο μοχλός ολισθαίνει στις κατευθύνσεις αριστερά & επάνω, δηλαδή κατεύθυνση στους άξονες X & Y ταυτόχρονα, τότε οι τιμές που λαμβάνει το Κύκλωμα RS και κατευθύνει αντίστοιχα τους σερβοκινητήρες είναι 0° και 0°.

Με ολίσθηση στις κατευθύνσεις δεξιά & κάτω οι τιμές που λαμβάνει το Κύκλωμα RS για μετατόπιση του άξονα των 2 σερβοκινητήρων είναι 180° και 180°.

Με επαναφορά του μοχλού στην αρχική θέση ηρεμίας οι τιμές που αποστέλλονται στο Κύκλωμα RS είναι 93° για τον σερβοκινητήρα 1 και 89° για τον σερβοκινητήρα 2.

```
Servo 1 position/angle is: 93°  
Servo 2 position/angle is: 89°  
  
Servo 1 position/angle is: 0°  
Servo 2 position/angle is: 0°  
  
Servo 1 position/angle is: 180°  
Servo 2 position/angle is: 180°  
  
Servo 1 position/angle is: 93°  
Servo 2 position/angle is: 89°
```

Εικόνα 3.78 Διάφορες τιμές των 2 σερβοκινητήρων με ολίσθηση του αναλογικού μοχλού



Εικόνα 3.79 Χρήση “power bank” ως εναλλακτική μέθοδος τροφοδοσίας του μικροελεγκτή Arduino UNO R3 ο οποίος με την σειρά του τροφοδοτεί ολόκληρο το Υποσύστημα Δέκτη ή αλλιώς Κύκλωμα RS του συστήματος ηλεκτρονικού ασύρματου ελεγκτή

3.4 Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) με Πρόσθετες Περιφερειακές Μονάδες

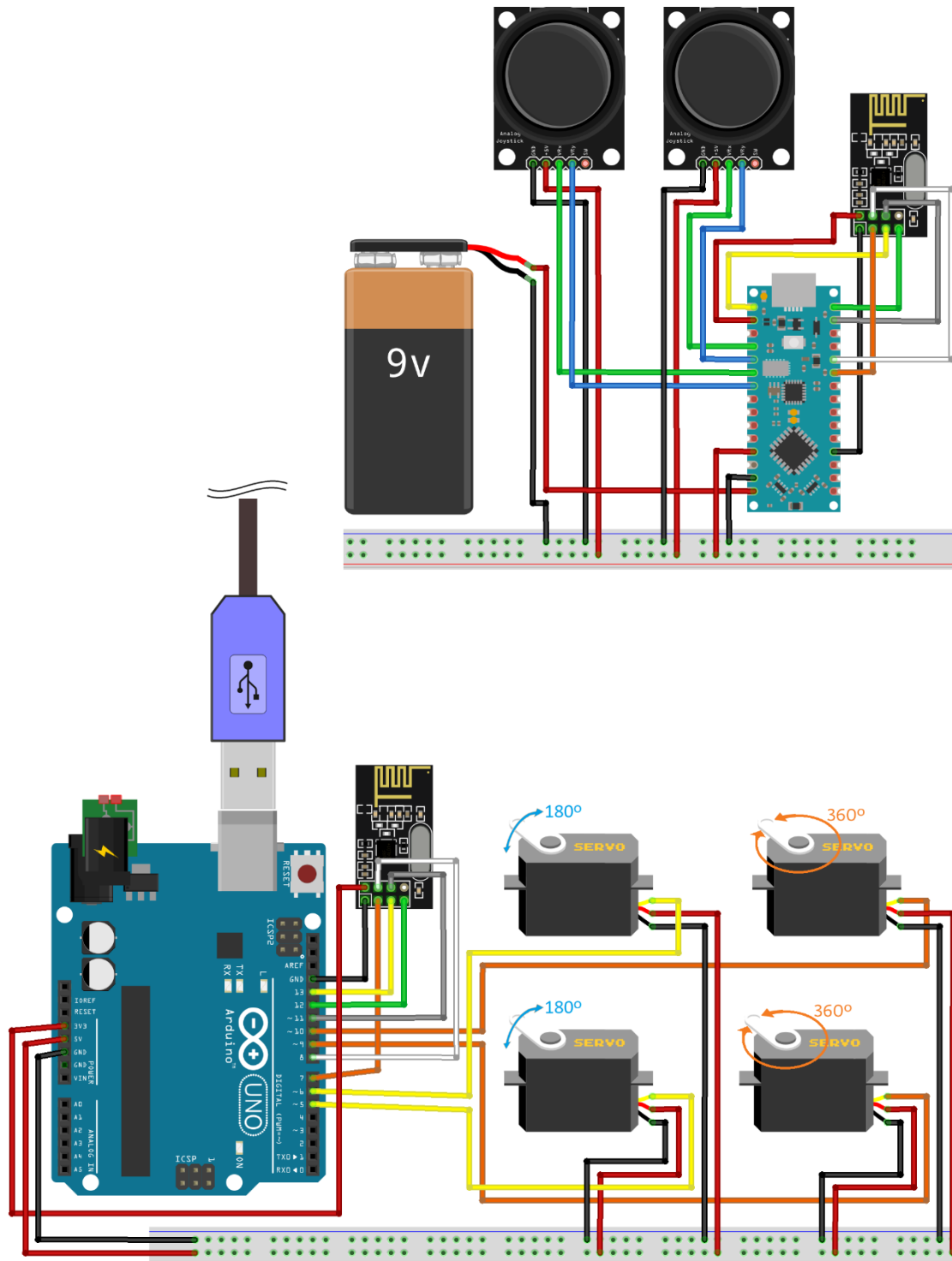
Αποσκοπώντας στην παρουσίαση της δυναμικής που διαθέτει το σύστημα ηλεκτρονικού ασύρματου ελεγκτή για χρήση του σε πληθώρα εφαρμογών βάσει του εκάστοτε σχεδιασμού και προγραμματισμού αλλά και ενδεχόμενης ανάγκης για τροποποίησή του, όντας υλοποιημένο με μικροελεγκτές της οικογένειας Arduino σε συνδυασμό με ένα αρκετά μεγάλο εύρος ηλεκτρονικών μονάδων συμβατών με Arduino, στην παρούσα ενότητα γίνεται συμπερίληψη 6 χαρακτηριστικών και ευρέως γνωστών περιφερειακών μονάδων για διεύρυνση των δυνατοτήτων του συστήματος σε σύγκριση με αυτές που ήταν καθορισμένο να διαθέτει στην προηγούμενη ενότητα 3.3.

Αρχικά γίνεται ένταξη και δοκιμή ορθής λειτουργίας ενός επιπλέον **αναλογικού μοχλού 2** αξόνων και ενός επιπλέον **ζεύγους σερβοκινητήρων** με δυνατότητα περιστροφής του άξονά τους κατά 360°.

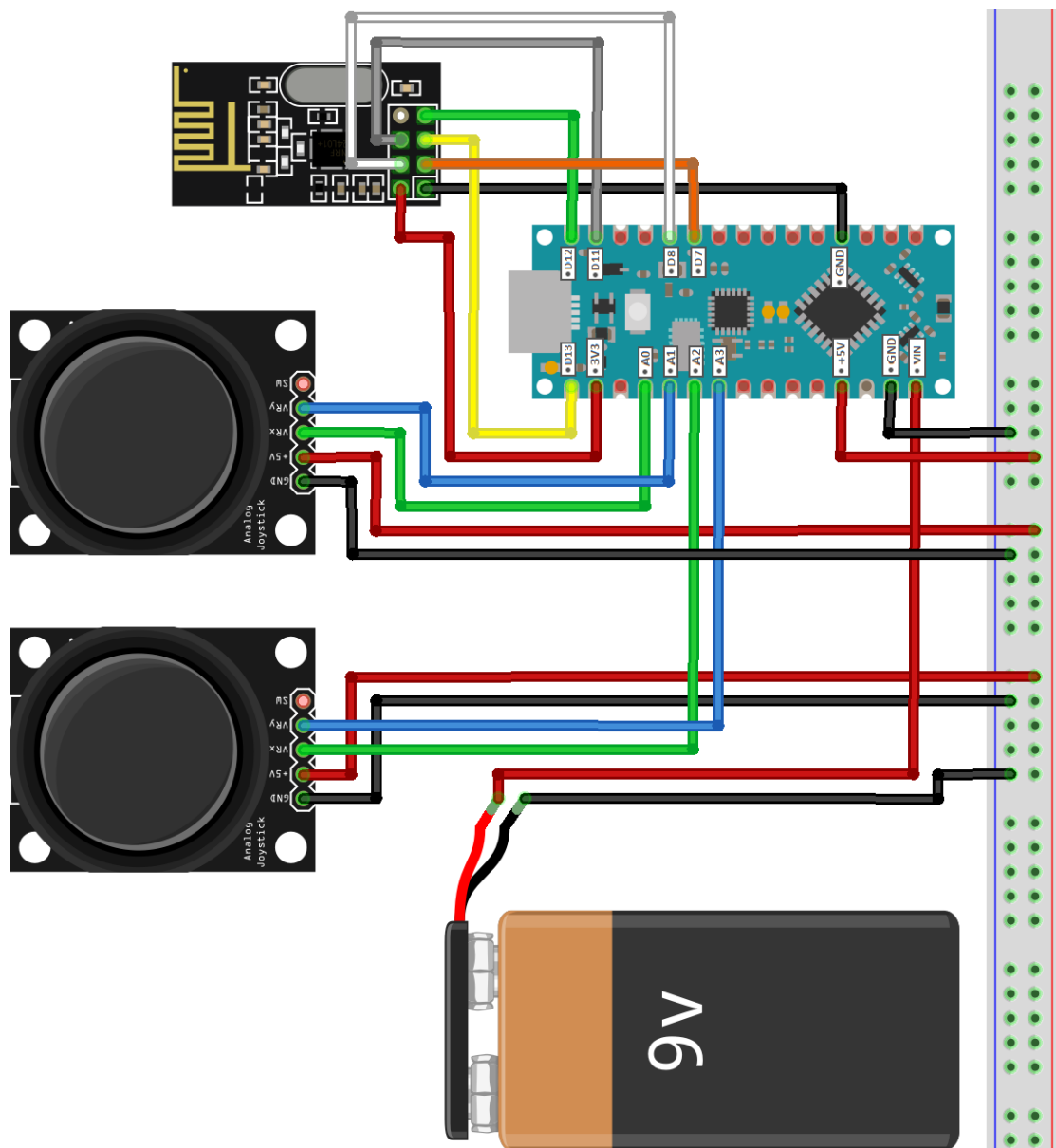
Στην συνέχεια γίνεται δοκιμή διασύνδεσης και ορθής λειτουργίας μιας μονάδας **αισθητήρα υπερήχων HC-SR04**, ενός πιεζοηλεκτρικού ηχείου **Buzzer** και ενός κόκκινου **LED** και τέλος μιας μονάδας **αισθητήρα MPU6050** επιταχυνσιομέτρου & γυροσκοπίου 3 αξόνων.

Με πραγματοποίηση του ορθού ελέγχου λειτουργίας των ανωτέρω μονάδων ξεχωριστά, γίνεται στην συνέχεια προσθήκη τους στο σύστημα για δημιουργία ενός νέου Σ.Η.Α.Ε. με πρόσθετες περιφερειακές μονάδες. Έτσι το σύστημα αυτό αποκτά νέες δυνατότητες και άρα χρήση σε περισσότερες εφαρμογές.

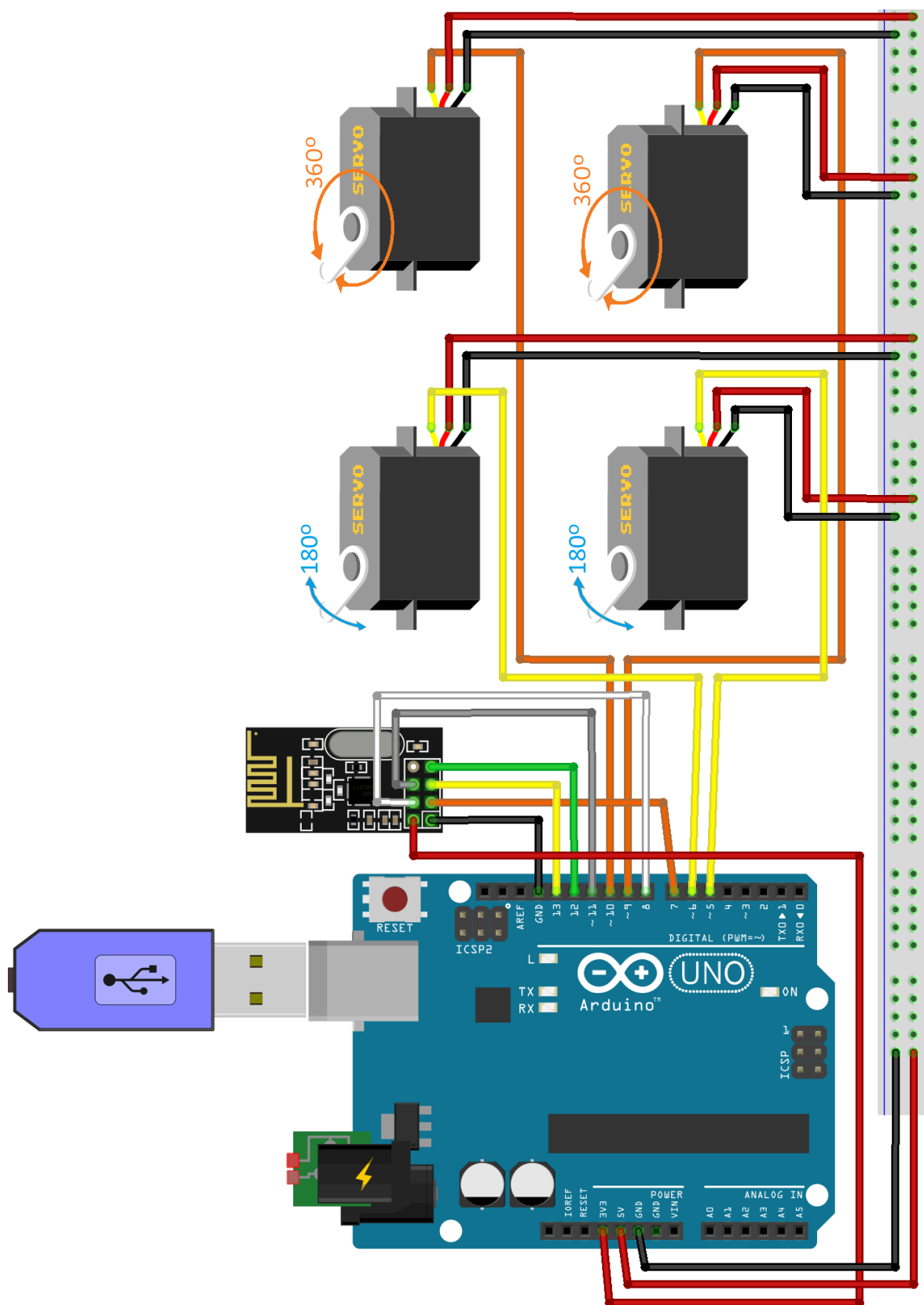
3.4.1 2^{ος} Αναλογικός Μοχλός Δύο Αξόνων & 2 Ψηφιακοί Σερβοκινητήρες Περιστροφής 360°



Εικόνα 3.80 Σύστημα ηλεκτρονικού ασύρματου ελεγκτή (Σ.Η.Α.Ε.) με 2 εισόδους και 4 εξόδους, υλοποιημένο σε περιβάλλον προσομοίωσης

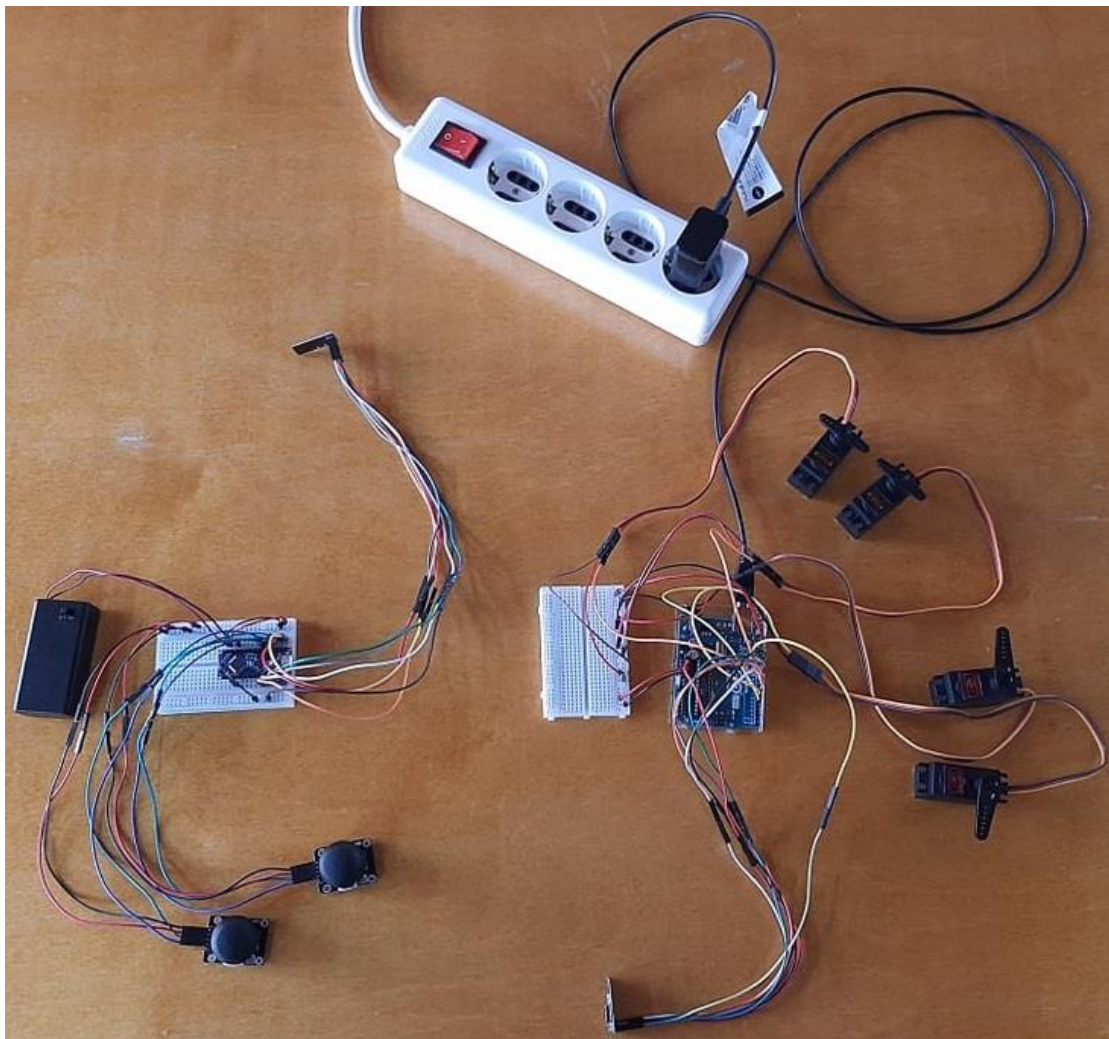


Εικόνα 3.81 Κόκλωμα TS του συστήματος ηλεκτρονικού ασύρματου ελεγκτή με 2 αναλογικούς μοχλούς για είσοδο



Εικόνα 3.82 Κύκλωμα RS του συστήματος ηλεκτρονικού ασύρματου ελεγκτή με 4 ψηφιακούς σερβοκινητήρες για έξοδο

Το παρόν σύστημα ηλεκτρονικού ασύρματου ελεγκτή (Σ.Η.Α.Ε.) μπορεί κάλλιστα να θεωρηθεί ως μια ασύρματη παραλλαγή του συστήματος ηλεκτρονικού ενσύρματου ελεγκτή (Σ.Η.Ε.Ε.) που υλοποιήθηκε και παρουσιάστηκε μέσω των εικόνων 3.58 και 3.59 καθώς το ενσύρματο σύστημα λειτουργεί κατά βάση με τις ίδιες περιφερειακές μονάδες αν εξαιρέσουμε τον ασύρματο τρόπο με τον οποίο ελέγχονται οι 4 σερβοκινητήρες από τους 2 αναλογικούς μοχλοί λόγω ύπαρξης των 2 πομποδεκτών NRF24L01.



***Εικόνα 3.83** Σύστημα ηλεκτρονικού ασύρματου ελεγκτή (Σ.Η.Α.Ε.) με 2 εισόδους και 4 εξόδους, υλοποιημένο σε επαγγελματικό εργαστήριο*

Ανάπτυξη Κώδικα για το Κύκλωμα TS

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(7, 8); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "00001";

// include an analog joystick of 2-axis
const int joystick_VRX = A0; // Connect the X-axis potentiometer to Arduino pin A0
const int joystick_VRY = A1; // Connect the Y-axis potentiometer to Arduino pin A1
const int joystick_VRX_2 = A2; // Connect the X_2-axis potentiometer to Arduino pin A2
const int joystick_VRY_2 = A3; // Connect the Y_2-axis potentiometer to Arduino pin A3

void setup() {
  Serial.begin(9600);
  radio.begin();
  //set the address
  radio.openWritingPipe(address);
}

void loop() {
  //Set module as transmitter
  radio.stopListening();

  // calculation process of joystick & No2 joystick values
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int X_2_ADC_Value = analogRead(joystick_VRX_2); // Read the analog/mechanical input of X_2-axis
  int Y_2_ADC_Value = analogRead(joystick_VRY_2); // Read the analog/mechanical input of Y_2-axis
  delay(100);

  // variables to store joystick values
  int angleV1 = map(X_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input signal of X-axis as values of range 0-180°
  int angleV2 = map(Y_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input signal of Y-axis as values of range 0-180°
  int angleV3 = map(X_2_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input signal of X_2-axis as values of range 0-180°
  int angleV4 = map(Y_2_ADC_Value, 0, 1023, 0, 180); // Set the analog/mechanical input signal of Y_2-axis as values of range 0-180°
  delay(100); // waits 100ms for the servos to reach the corresponding angle

  // Send joystick information (status & values) to receiver
  radio.write(&angleV1, sizeof(angleV1));
  Serial.print("Joystick value for X-axis: ");
  Serial.print(angleV1);
  Serial.print("\n");
  delay(25);

  radio.write(&angleV2, sizeof(angleV2));
  Serial.print("Joystick value for Y-axis: ");
  Serial.print(angleV2);
  Serial.print("\n");
  delay(25);

  radio.write(&angleV3, sizeof(angleV3));
  Serial.print("Joystick No2 value for X_2-axis: ");
  Serial.print(angleV3);
  Serial.print("\n");
  delay(25);

  radio.write(&angleV4, sizeof(angleV4));
  Serial.print("Joystick No2 value for Y_2-axis: ");
  Serial.print(angleV4);
  Serial.println("\n");
  delay(500);
}
```

ΚΩΔΙΚΑΣ_18 Κώδικας λειτουργίας για το Κύκλωμα TS
του Σ.Η.Α.Ε. με δύο αναλογικούς μοχλούς

Ανάπτυξη Κώδικα για το Κύκλωμα RS

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h> // add the servo library

//create an RF24 object
RF24 radio(7, 8); // CE, CSN
//address through which two modules communicate.
const byte address[6] = "00001";

// include 2 servos of 180 degree rotation
Servo myServo1; // Define 'myServo1' as Servo1 servo motor
Servo myServo2; // Define 'myServo2' as Servo2 servo motor
Servo myServo3; // Define 'myServo3' as Servo3 servo motor
Servo myServo4; // Define 'myServo4' as Servo4 servo motor

// variables to store joysticks' values sent by the transmitter (TS Circuit)
int angleV_1 = 0; // Initialize the variable to store Joystick values of X-axis for Servo 1
int angleV_2 = 0; // Initialize the variable to store Joystick value of Y-axis for Servo 2
int angleV_3 = 0; // Initialize the variable to store Joystick No2 values of X_2-axis for Servo3 rotational speed
int angleV_4 = 0; // Initialize the variable to store Joystick No2 values of Y_2-axis for Servo4 rotational speed

void setup() {
  Serial.begin(9600);
  radio.begin();

  //set the address
  radio.openReadingPipe(0, address);

  myServo1.attach(5); // Attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // Attaches Servo2 Signal Line to pin 6 (PWM)

  myServo3.write(90); // Initialize Servo1 rotation speed to zero (90 degrees)
  myServo4.write(90); // Initialize Servo2 rotation speed to zero (90 degrees)
  myServo3.attach(9); // Attaches Servo3 Signal Line to pin 9 (PWM)
  myServo4.attach(10); // Attaches Servo4 Signal Line to pin 10 (PWM)
}

void loop() {
  //Set module as receiver
  radio.startListening();

  //Read the data if available in buffer
  if (radio.available()) {
    while (radio.available()) {

      // Receive joystick information to rotate the 2 Servo motors accordingly
      radio.read(&angleV_1, sizeof(angleV_1)); // Read & store X-axis information of Joystick
      Serial.print("For X-axis value: ");
      Serial.print(angleV_1);
      Serial.print("\n");
      if (angleV_1 < 10 || angleV_1 > 170) {
        Serial.print(" ---->Servo position/angle is ");
        Serial.print(angleV_1);
        Serial.println("\n");
        myServo1.write(angleV_1); // Use the variable value to rotate Servo 1
      }
      else if (angleV_1 > 10 && angleV_1 < 170) {
        // print the standby position in degrees
        Serial.println(" ---->Servo on standby mode");
      }
    }
    delay(100);
  }
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

radio.read(&angleV_2, sizeof(angleV_2)); // Read & store Y-axis information of Joystick
Serial.print("For Y-axis value: ");
Serial.print(angleV_2);
Serial.print("");
if (angleV_2 < 10 || angleV_2 > 170) {
  Serial.print(" ---->Servo2 position/angle is ");
  Serial.print(angleV_2);
  Serial.println("");
  myServo2.write(angleV_2); // Use the variable value to rotate Servo 2
}
else if (angleV_2 > 10 && angleV_2 < 170) {
  // print the standby position in degrees
  Serial.println(" ---->Servo2 on standby mode");
}
delay(100);

radio.read(&angleV_3, sizeof(angleV_3));
Serial.print("For X_2-axis value: "); // Read & store X_2-axis information of Joystick No2
Serial.print(angleV_3);
Serial.print("");
if (angleV_3 < 85 || angleV_3 > 95) {
  Serial.print(" -->Servo3 rotational speed is ");
  Serial.print(angleV_3);
  Serial.println("");
  myServo3.write(angleV_3); // Use the variable value to rotate Servo 3
}
else if (angleV_3 > 85 && angleV_3 < 95) {
  Serial.println(" -->Servo3 in inertia");
  myServo3.write(90); // Use the variable value to stagnate Servo 3
}
delay(100);

radio.read(&angleV_4, sizeof(angleV_4)); // Read & store Y_2-axis information of Joystick No2
Serial.print("For Y_2-axis value: ");
Serial.print(angleV_4);
Serial.print("");
if (angleV_4 <= 85) {
  Serial.print(" -->Servo4 rotational speed is ");
  Serial.print(angleV_4);
  Serial.println("", " but fixed at 80°");
  myServo4.write(80); // Use the variable value to rotate Servo 4
}
else if (angleV_4 >= 95) {
  Serial.print(" -->Servo4 rotational speed is ");
  Serial.print(angleV_4);
  Serial.println("", " but fixed at 100°");
  myServo4.write(100); // Use the variable value to rotate Servo 4
}
else if (angleV_4 > 85 && angleV_4 < 95) {
  Serial.println(" -->Servo4 in inertia");
  myServo4.write(90); // Use the variable value to stagnate Servo 4
}
Serial.print("\n");
}
}
delay(100);
}

```

ΚΩΔΙΚΑΣ_19 Κώδικας λειτουργίας για το Κόκλωμα RS
του Σ.Η.Α.Ε. με 4 ψηφιακούς σερβοκινητήρες

Με χρήση του Serial Monitor για το Κύκλωμα TS, γίνεται παρακολούθηση των τιμών των αναλογικών μοχλών που αποστέλλονται στο Κύκλωμα RS μέσω των μονάδων πομποδεκτών NRF24L01:

```
Joystick value for X-axis: 93°  
Joystick value for Y-axis: 89°  
Joystick No2 value for X_2-axis: 91°  
Joystick No2 value for Y_2-axis: 89°  
  
Joystick value for X-axis: 93°  
Joystick value for Y-axis: 89°  
Joystick No2 value for X_2-axis: 91°  
Joystick No2 value for Y_2-axis: 89°  
  
Joystick value for X-axis: 93°  
Joystick value for Y-axis: 89°  
Joystick No2 value for X_2-axis: 91°  
Joystick No2 value for Y_2-axis: 89°
```

Εικόνα 3.84 Ανάγνωση του ζεύγους τιμών εκάστου αναλογικού μοχλού του Κυκλώματος TS στην θέση ηρεμίας (αδράνεια)

Κατ' αντιστοιχία, για την παρακολούθηση των τιμών των μοχλών που λαμβάνει το Κύκλωμα RS γίνεται ξανά χρήση του Serial Monitor:

```
For X-axis value: 93° ----->Servo1 on standby mode  
For Y-axis value: 89° ----->Servo2 on standby mode  
For X_2-axis value: 91° -->Servo3 in inertia  
For Y_2-axis value: 89° -->Servo4 in inertia  
  
For X-axis value: 93° ----->Servo1 on standby mode  
For Y-axis value: 89° ----->Servo2 on standby mode  
For X_2-axis value: 91° -->Servo3 in inertia  
For Y_2-axis value: 89° -->Servo4 in inertia  
  
For X-axis value: 93° ----->Servo1 on standby mode  
For Y-axis value: 89° ----->Servo2 on standby mode  
For X_2-axis value: 91° -->Servo3 in inertia  
For Y_2-axis value: 89° -->Servo4 in inertia  
  
For X-axis value: 93° ----->Servo1 on standby mode  
For Y-axis value: 89° ----->Servo2 on standby mode  
For X_2-axis value: 91° -->Servo3 in inertia  
For Y_2-axis value: 89° -->Servo4 in inertia
```

Εικόνα 3.85 Ανάγνωση του ζεύγους τιμών εκάστου μοχλού και αντιστοιχία των με τους σερβοκινητήρες του Κυκλώματος RS στην θέση ηρεμίας (αδράνεια)

Με ολίσθηση των δύο αναλογικών μοχλών μεταβάλλονται και οι τιμές των ποτενσιόμετρων που αντιστοιχούν στους X και Y άξονες.

Συγκεκριμένα, στην εικόνα 3.86 παρατηρείται αρχικά η μεταβολή των τιμών για τα 2 ποτενσιόμετρα του ενός μοχλού όταν ολισθαίνει κατά τον X και Y άξονα αντίστοιχα, ενώ ο 2^{ος} μοχλός που ελέγχει τους σερβοκινητήρες 3 & 4 παραμένει αδρανής.

```
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 0° ----->Servo1 position/angle is 0°
For Y-axis value: 145° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 50° ----->Servo1 on standby mode
For Y-axis value: 0° ----->Servo2 position/angle is 0°
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 180° ----->Servo1 position/angle is 180°
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 180° ----->Servo2 position/angle is 180°
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia
```

Εικόνα 3.86 Παρακολούθηση Σειριακής Οθόνης για το Κύκλωμα RS και ολίσθηση του 1^{ου} αναλογικού μοχλού

Στην εικόνα 3.87 απεικονίζεται η μεταβολή των τιμών των ποτενσιομέτρων του 2^{ου} μοχλού κατά την μετατόπισή του στους X και Y άξονες, με τον 1^ο μοχλό που ελέγχει τους σερβοκινητήρες 1 & 2 να βρίσκεται σε κατάσταση αναμονής (αδράνεια).

```
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 92° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 0° -->Servo3 rotational speed is 0°
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 0° -->Servo4 rotational speed is 0°, but fixed at 80°

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 180° -->Servo3 rotational speed is 180°
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 180° -->Servo4 rotational speed is 180°, but fixed at 100°
```

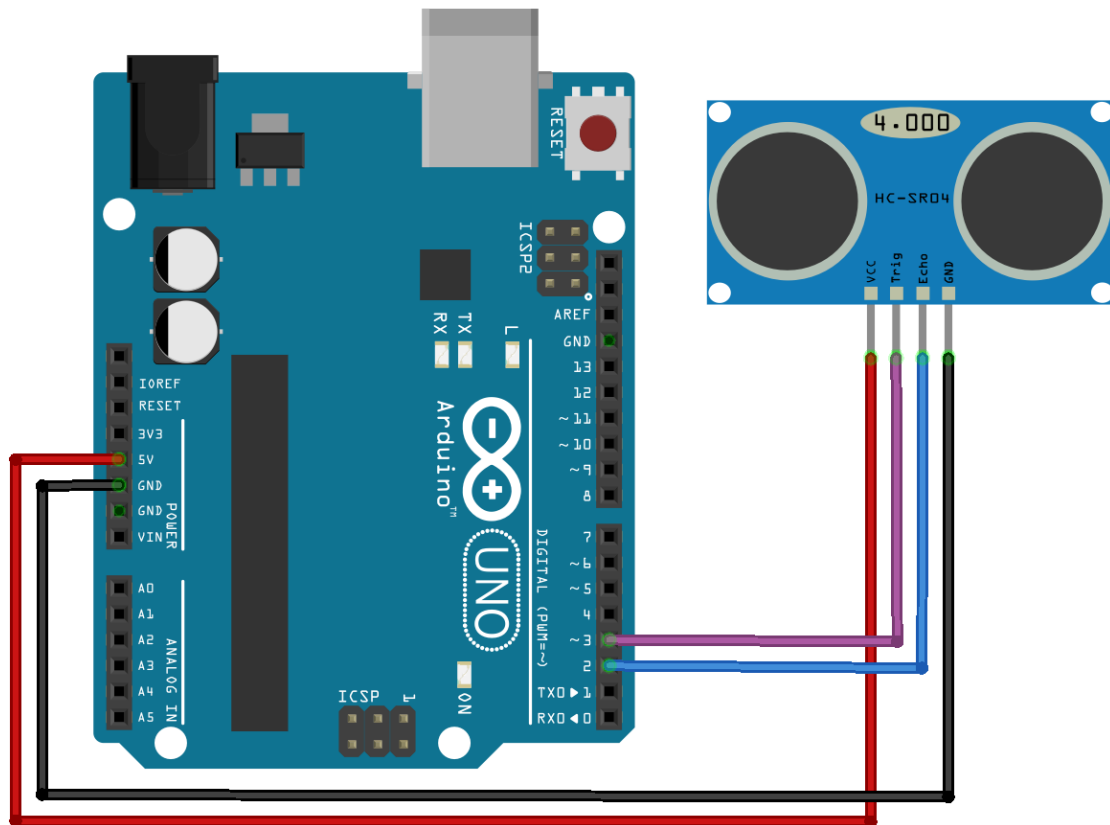
Εικόνα 3.87 Παρακολούθηση Σειριακής Οθόνης για το Κύκλωμα RS και ολίσθηση του 2^{ου} αναλογικού μοχλού

Ειδικά για τον 4^ο σερβοκινητήρα με το όνομα “*Servo4*” έχει γίνει κατάλληλη ανάπτυξη του προγραμματιστικού κώδικα ώστε να ελέγχεται η ταχύτητα με την οποία θα περιστραφεί και άρα να καθίσταται εμφανής η ευελιξία προγραμματισμού, προσαρμογής και τροποποίησης που επιδέχεται το Σ.Η.Α.Ε. για να καλύπτει πληθώρα εφαρμογών σε έναν χώρο, καθώς μέσω της εικόνας 3.87 γίνεται αντιληπτό ότι η ταχύτητα του εκάστοτε σερβοκινητήρα αλλά και γενικά η συμπεριφορά του μπορεί να ελέγχεται και να καθορίζεται βάσει των επιθυμιών του προγραμματιστή και του χρήστη και αναλόγως την χρησιμότητα για την εκάστοτε εφαρμογή για την οποία προορίζεται.

3.4.2 Δοκιμή διασύνδεσης Arduino UNO R3 & Αισθητήρα Υπερήχων HC-SR04

ΠΙΝΑΚΑΣ 3_6

| | | | | |
|---------------------------------------|-----|-----|------|------|
| Ακροδέκτες αισθητήρα υπερήχων HC-SR04 | Vcc | Gnd | Trig | Echo |
| Ακροδέκτες Arduino UNO R3 | 5V | GND | D3 | D2 |



Εικόνα 3.88 Συνδεσμολογία Arduino UNO & μονάδας αισθητήρα HC-SR04

Ανάπτυξη Κώδικα για τον αισθητήρα HC-SR04

```
// Define pins numbers
const int Trig_pin = 3; // Attach Trigger pin to Arduino pin D3
const int Echo_pin = 2; // Attach Echo pin to Arduino pin D2

// Define variables
float duration, distance;

void setup() {
  pinMode(Trig_pin, OUTPUT); // Set the Trig pin as an Output
  pinMode(Echo_pin, INPUT); // Set the Echo pin as an Input

  Serial.begin(9600); // Starts the serial communication
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // Print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}

void loop() {

  // Clear the Trig pin condition
  digitalWrite(Trig_pin, LOW);
  delayMicroseconds(2);

  // Set the Trig pin HIGH (ACTIVE) for 10µs
  digitalWrite(Trig_pin, HIGH); // Generate the sound wave
  delayMicroseconds(10);
  digitalWrite(Trig_pin, LOW); // Clear the Trig pin condition

  // Read the Echo pin and return the sound wave travel time in µs
  duration = pulseIn(Echo_pin, HIGH); // The reflected wave turns the Echo pin ON (HIGH) and the timing starts.
  // When the wave ends the Echo pin goes OFF (LOW) and the timing stops,
  // returning the length of the reflected sound wave in µs.

  // Calculate the distance
  distance = (duration * 0.034) / 2; // Speed of sound wave divided by 2 (transmission & reception time travel)

  // Display the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(100);
}
```

ΚΩΔΙΚΑΣ_20 Κώδικας λειτουργίας της μονάδας αισθητήρα υπερήχων HC-SR04 και εμφάνισης των παραμέτρων της μέσω του Serial Monitor

Με «ανέβασμα» του κώδικα Νο20 και εκκίνηση της λειτουργίας του Serial Monitor του Arduino IDE για παρακολούθηση της συμπεριφοράς του αισθητήρα υπερήχων, εμφανίζονται στις δύο κάτωθι εικόνες τα αποτελέσματα από συνεχή μέτρηση αποστάσεων σε έναν χώρο.

```
Ultrasonic Sensor HC-SR04 Test  
with Arduino UNO R3  
Distance: 335.48 cm  
Distance: 337.86 cm  
Distance: 337.84 cm  
Distance: 337.89 cm  
Distance: 337.96 cm  
Distance: 337.98 cm  
Distance: 337.98 cm  
Distance: 337.89 cm
```

Εικόνα 3.89 Αρχική κατάσταση του παραθύρου του Serial Monitor, με τον αισθητήρα υπερήχων να μετρά απόσταση ~337,9cm μεταξύ αυτού και ενός αντικειμένου σε έναν χώρο

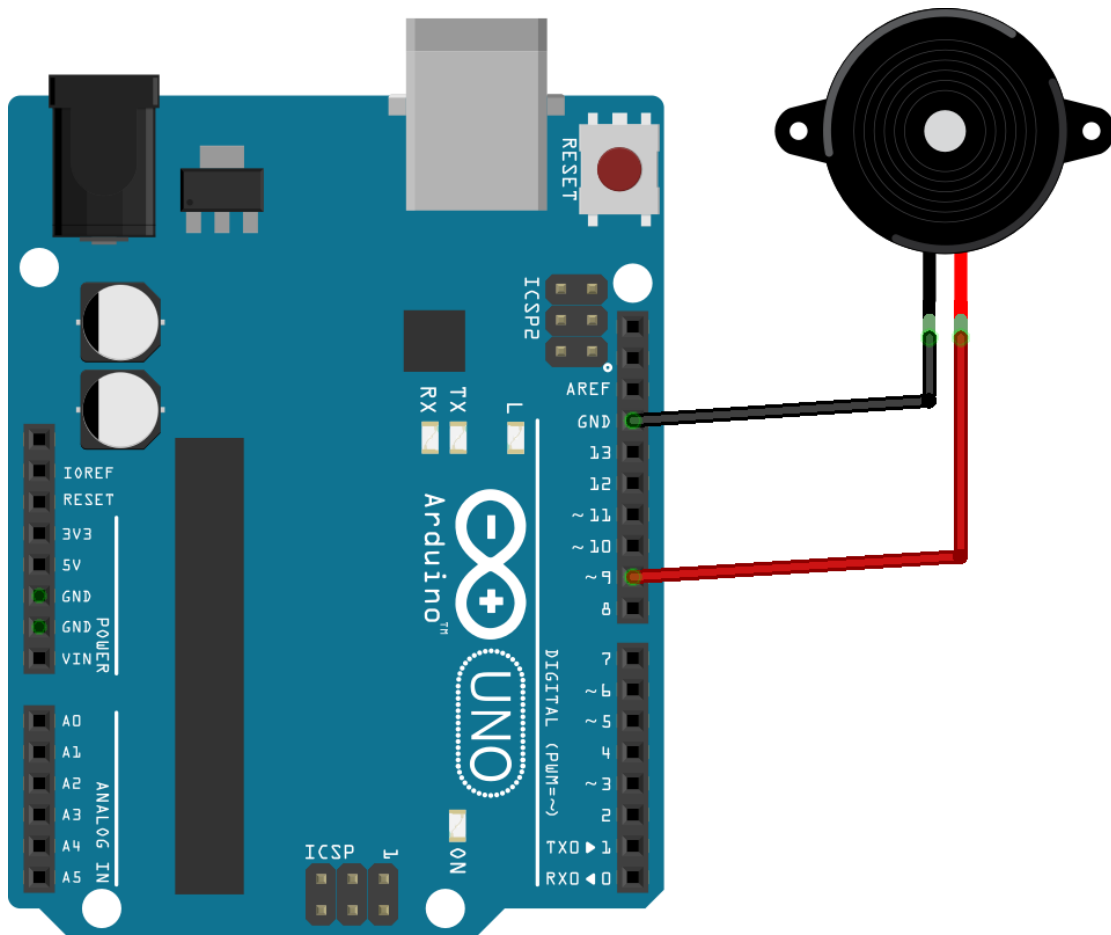
```
Distance: 337.82 cm  
Distance: 337.94 cm  
Distance: 55.30 cm  
Distance: 337.81 cm  
Distance: 28.51 cm  
Distance: 17.83 cm  
Distance: 10.18 cm  
Distance: 4.17 cm  
Distance: 9.89 cm  
Distance: 28.24 cm  
Distance: 48.28 cm  
Distance: 30.57 cm  
Distance: 18.79 cm  
Distance: 29.63 cm  
Distance: 3.86 cm  
Distance: 4.17 cm  
Distance: 14.01 cm  
Distance: 21.25 cm  
Distance: 33.66 cm  
Distance: 46.75 cm  
Distance: 59.09 cm  
Distance: 64.11 cm  
Distance: 337.79 cm  
Distance: 337.86 cm
```

Εικόνα 3.90 Με επιλογή ενός οποιουδήποτε αντικειμένου για αξιοποίησή του ως εμπόδιο για τον αισθητήρα, στην εμβέλεια που αυτός καλύπτει, φανερώνουν την εκάστοτε μεταβολή της απόστασης του αντικειμένου ως προς τον αισθητήρα

3.4.3 Δοκιμή διασύνδεσης Arduino UNO R3 & Buzzer

ΠΙΝΑΚΑΣ 3_7

| | | |
|--|-----|-----|
| Ακροδέκτες πιεζοηλεκτρικού ηχείου “Buzzer” | Vcc | Gnd |
| Ακροδέκτες Arduino UNO R3 | D9 | GND |



Εικόνα 3.91 Συνδεσμολογία Arduino UNO & πιεζο-ηχείου “Buzzer”

Ανάπτυξη Κώδικα για το Buzzer

```
const int buzzer = 9; // Connect Buzzer pin "Vcc" to Arduino pin "9"

void setup() {
  Serial.begin(9600);

  pinMode(buzzer, OUTPUT); // Set the connected to pin 9 buzzer as an output
}

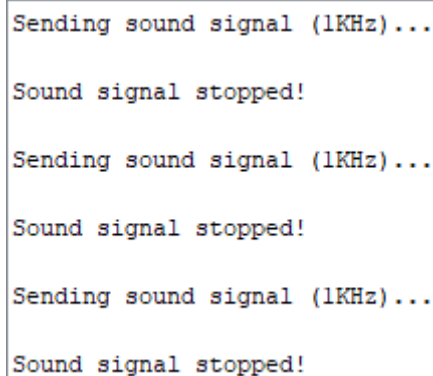
void loop() {

  Serial.print("Sending sound signal (1kHz)...");
  tone(buzzer, 1000); // Send 1kHz sound signal
  Serial.println("\n");
  delay(2000); // ...for 2000ms or 2sec

  Serial.print("Sound signal stopped!");
  noTone(buzzer); // Stop the sound signal
  Serial.println("\n");
  delay(2000); // ...for 2000ms or 2sec
}
```

ΚΩΔΙΚΑΣ_21 Κώδικας λειτουργίας (ενεργοποίησης/απενεργοποίησης) ενός πιεζοηλεκτρικού ηχείου (Buzzer) μέσω του μικροελεγκτή UNO R3

Ως γίνεται αντιληπτό τόσο από τα σχόλια του ανωτέρω κώδικα όσο και από την κατωτέρω εικόνα 3.92, με «ανέβασμα» του εν λόγω κώδικα και χρήση του Serial Monitor για μελέτη της λειτουργίας του πιεζοηλεκτρικού ηχείου “Buzzer”, το πιεζο-ηχείο παράγει και αποστέλλει ηχητικό σήμα του 1kHz διάρκειας 2 δευτερολέπτων (2000ms), έπειτα αδρανοποιείται για άλλα 2 δευτερόλεπτα και συνεχίζει από την αρχή ο ίδιος κύκλος εργασιών.

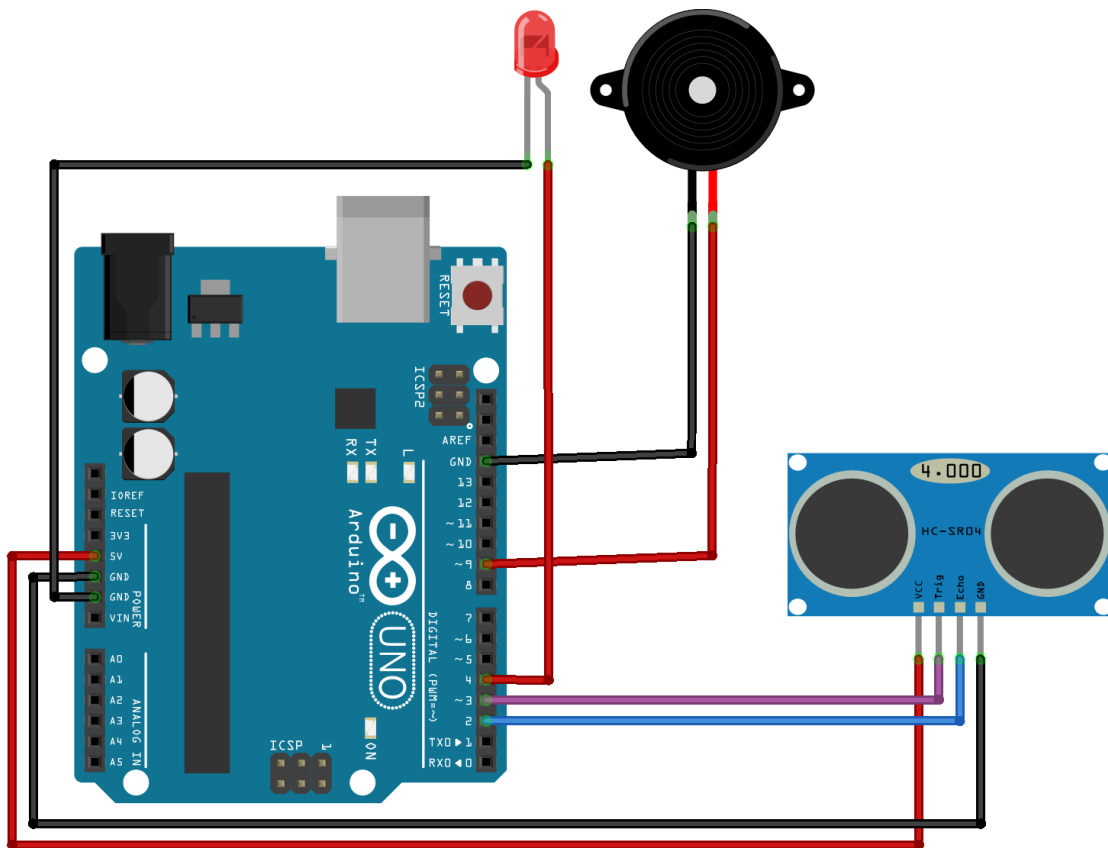


```
Sending sound signal (1kHz)...
Sound signal stopped!
Sending sound signal (1kHz)...
Sound signal stopped!
Sending sound signal (1kHz)...
Sound signal stopped!
```

Εικόνα 3.92 Εκκίνηση και παύση αποστολής ηχητικού σήματος 1kHz, με διάρκεια 2sec για την κάθε ενέργεια

3.4.3.1 Σύστημα Συναγερμού – Ειδοποίησης & Ελέγχου της Κίνησης

Με χρήση των δύο ανωτέρω πινάκων συνδεσμολογίας που αφορούν τις μονάδες HC-SR04, Buzzer και Arduino UNO R3 υλοποιείται ένα ενσύρματο σύστημα παρακολούθησης, μέτρησης και καταγραφής μιας συγκεκριμένης απόστασης σε έναν χώρο, ικανό για εφαρμογή του ως Σύστημα Συναγερμού σε περίπτωση παραβίασης του χώρου (π.χ. οικία, δημόσια περιουσία) ή ως σύστημα ειδοποίησης και ελέγχου της κίνησης στον χώρο (π.χ. είσοδος/έξοδος ενός καταστήματος).



Εικόνα 3.93 Συνδεσμολογία του Arduino UNO R3 με τις μονάδες HC-SR04, Buzzer και LED για χρήση ως Σύστημα Συναγερμού – Ειδοποίησης & Ελέγχου της Κίνησης, υλοποιημένο σε περιβάλλον προσομοίωσης

Ανάπτυξη Κώδικα για το Σύστημα Συναγερμού

```
// Define pins numbers
const int Trig_pin = 3;    // Attach Trigger pin to Arduino pin D3
const int Echo_pin = 2;    // Attach Echo pin to Arduino pin D2
const int buzzer = 9;     // Connect Buzzer pin "Vcc" to Arduino pin "9"
const int RED_LED = 4;    // Connect RED LED to Arduino pin "4"

// Define variables
float duration, distance;

void setup() {
  Serial.begin(9600); // Starts the serial communication
  pinMode(Trig_pin, OUTPUT); // Set the Trig pin as an Output
  pinMode(Echo_pin, INPUT); // Set the Echo pin as an Input
  pinMode(buzzer, OUTPUT); // Set the connected to pin 9 Buzzer as an output
  pinMode(RED_LED, OUTPUT); // Set the RED LED as an Output
}

void loop() {
  // Clear the Trig pin condition
  digitalWrite(Trig_pin, LOW);
  delayMicroseconds(2);

  // Set the Trig pin HIGH (ACTIVE) for 10µs
  digitalWrite(Trig_pin, HIGH); // Generate the sound wave
  delayMicroseconds(10);
  digitalWrite(Trig_pin, LOW); // Clear the Trig pin condition

  // Read the Echo pin and return the sound wave travel time in µs
  duration = pulseIn(Echo_pin, HIGH); // The reflected wave turns the Echo pin ON (HIGH) and the timing starts.
  // When the wave ends the Echo pin goes OFF (LOW) and the timing stops,
  // returning the length of the reflected sound wave in µs.

  // Calculate the distance
  distance = (duration * 0.034) / 2; // Speed of sound wave divided by 2 (transmission & reception time travel)

  // Display the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(100);

  if (distance <= 12) {
    Serial.print("Sending sound signal (1kHz)...");
    tone(buzzer, 1000); // Send 1kHz sound signal
    Serial.println("");
    Serial.print("RED LED is turned ON !");
    digitalWrite(RED_LED, HIGH);
    Serial.println("\n");
    delay(2000); // ...for 2000ms or 2sec
  }
  else {
    Serial.print("Sound signal stopped!");
    noTone(buzzer); // Stop the sound signal
    Serial.println("");
    digitalWrite(RED_LED, LOW);
    Serial.print("RED LED is turned OFF !");
    Serial.println("\n");
    delay(25); // ...for 25ms
  }
}
```

ΚΩΔΙΚΑΣ_22 Κώδικας μετατροπής του συστήματος (Arduino UNO, HC-SR04, Buzzer & LED) Σε Σύστημα Συναγερμού – Ειδοποίησης και Ελέγχου της κίνησης σε έναν χώρο

Με «ανέβασμα» του κώδικα No22 για παρακολούθηση της συμπεριφοράς των τεσσάρων ηλεκτρονικών μονάδων ως Σύστημα Συναγερμού – Ειδοποίησης & Ελέγχου της κίνησης σε έναν χώρο, παρατηρούμε στην εικόνα 3.94 ότι δεν αποστέλλεται ηχητικό σήμα και άρα οι μονάδες Buzzer και LED παραμένουν αδρανείς.

```
Distance: 336.87 cm  
Sound signal stopped!  
RED LED is turned OFF !  
  
Distance: 334.63 cm  
Sound signal stopped!  
RED LED is turned OFF !  
  
Distance: 334.34 cm  
Sound signal stopped!  
RED LED is turned OFF !  
  
Distance: 334.20 cm
```

Εικόνα 3.94 Συνδεσμολογία του Arduino UNO R3 με τις μονάδες HC-SR04, Buzzer και LED

Έχοντας ένα αντικείμενο να πλησιάζει σε απόσταση τον αισθητήρα υπερήχων HC-SR04, όταν το αντικείμενο αποκτήσει απόσταση μικρότερη των 12cm ως ορίστηκε στον κώδικα, το Buzzer στέλνει ηχητικό σήμα 1kHz και το LED κόκκινου χρώματος φωτοβολεί, ειδοποιώντας τόσο οπτικά όσο και ηχητικά έναν χρήστη για ύπαρξη κάποιας δραστηριότητας στον χώρο λόγω μεταβολής των αποστάσεων κατά την μέτρηση.

Το Σύστημα Συναγερμού είναι προγραμματισμένο έτσι ώστε να απενεργοποιείται (*Buzzer & LED → OFF*) όταν το αντικείμενο αποκτά απόσταση μικρότερη ή ίση των 12cm ως προς τον αισθητήρα.

```
Distance: 15.15 cm
Sound signal stopped!
RED LED is turned OFF !

Distance: 9.28 cm
Sending sound signal (1KHz)...
RED LED is turned ON !

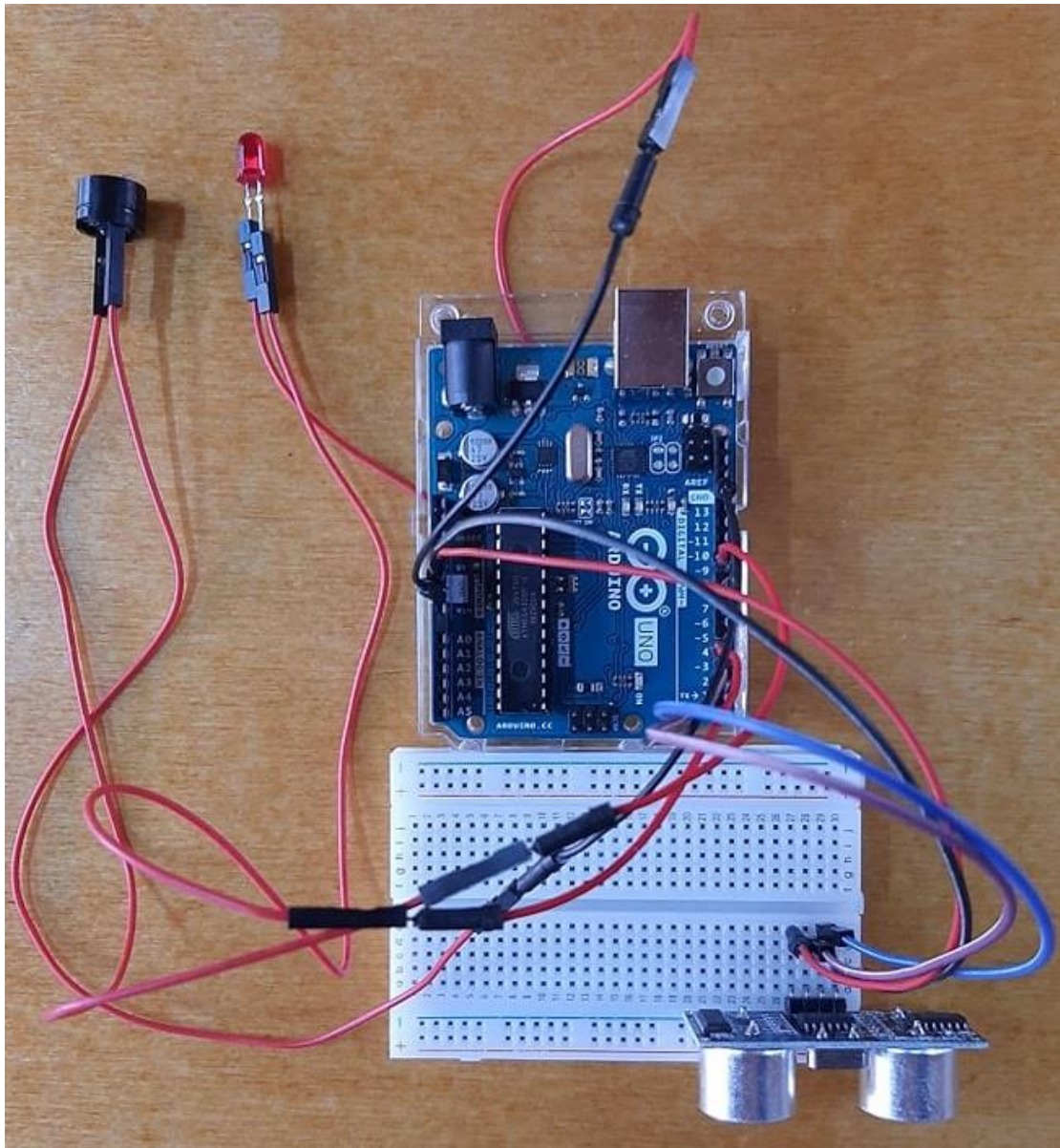
Distance: 3.16 cm
Sending sound signal (1KHz)...
RED LED is turned ON !

Distance: 2.99 cm
Sending sound signal (1KHz)...
RED LED is turned ON !

Distance: 2.89 cm
Sending sound signal (1KHz)...
RED LED is turned ON !

Distance: 332.16 cm
Sound signal stopped!
RED LED is turned OFF !
```

Εικόνα 3.95 Όσο η απόσταση μεταξύ ενός αντικειμένου και της μονάδας HC-SR04 διατηρείται μεγαλύτερη των 12cm, το Σύστημα Συναγερμού παραμένει αδρανές

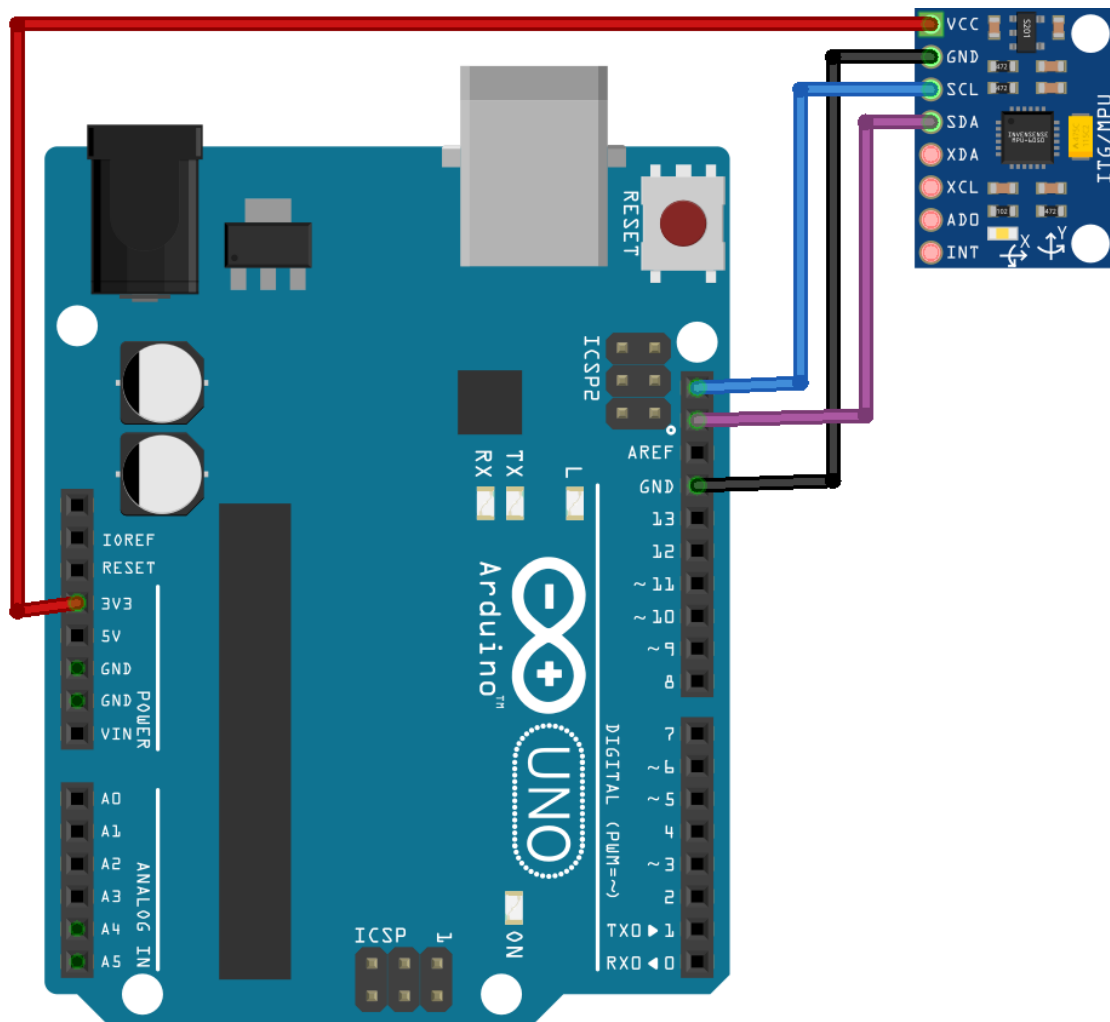


Εικόνα 3.96 Συνδεσμολογία του Arduino UNO με τις μονάδες HC-SR04, Buzzer και LED για χρήση ως Σύστημα Συναγερμού – Ειδοποίησης & Ελέγχου της Κίνησης, υλοποιημένο σε επαγγελματικό εργαστήριο

3.4.4 Δοκιμή διασύνδεσης των Arduino UNO & NANO με την Μονάδα Αισθητήρα Επιταχυνσιόμετρου-Γυροσκοπίου

ΠΙΝΑΚΑΣ 3_8

| | | | | |
|---------------------------------|------|-----|-----|-----|
| Ακροδέκτες αισθητήρα MPU6050 | VCC | GND | SCL | SDA |
| Ακροδέκτες Arduino UNO R3 | 3.3V | GND | SCL | SDA |



Εικόνα 3.97 Συνδεσμολογία Arduino UNO & μονάδας αισθητήρα MPU6050

Με επιτυχή διασύνδεση μεταξύ Arduino UNO R3 και μονάδας αισθητήρα MPU6050, πραγματοποιείται στην συνέχεια σύνδεση του μικροελεγκτή με Η/Υ και ανάπτυξη κατάλληλου κώδικα (*ΚΩΔΙΚΑΣ_23*) για προγραμματισμό του Arduino και παρακολούθηση της συμπεριφοράς του αισθητήρα μέσω του Serial Monitor.

```
#include <Wire.h>
#include <GY6050.h>
GY6050 gyro(0x68); // Create Gyro object with "0x68" as I2C address (the address of the MPU6050)

void setup() {

  gyro.initialisation();
  Serial.begin(9600);
  Serial.println("MPU6050 Test");
  delay(2500);

}

void loop() {

  Serial.print("AcX = ");
  int AcX = gyro.refresh('A', 'X');
  Serial.print(AcX); // Ask for the X axis of the Accelerometer and print it

  Serial.print(" | AcY = ");
  int AcY = gyro.refresh('A', 'Y');
  Serial.print(AcY); // Ask for the Y axis of the Accelerometer and print it

  Serial.print(" | AcZ = ");
  int AcZ = gyro.refresh('A', 'Z');
  Serial.print(AcZ); // Ask for the Z axis of the Accelerometer and print it

  Serial.print(" | Tmp = ");
  int Tmp_C = gyro.refresh('T', 'C');
  Serial.print(Tmp_C); // Ask for the Temperature in Celsius and print it
  Serial.print(" C or ");
  int Tmp_F = gyro.refresh('T', 'F');
  Serial.print(Tmp_F); // Ask for the Temperature in Farenheit and print it
  Serial.print(" F");

  Serial.print(" | GyX = ");
  int GyX = gyro.refresh('G', 'X');
  Serial.print(GyX); // Ask for the X axis of the Gyroscope and print it

  Serial.print(" | GyY = ");
  int GyY = gyro.refresh('G', 'Y');
  Serial.print(GyY); // Ask for the Y axis of the Gyroscope and print it

  Serial.print(" | GyZ = ");
  int GyZ = gyro.refresh('G', 'Z');
  Serial.println(GyZ); // Ask for the Z axis of the Gyroscope and print it

  delay(1000);
}
```

***ΚΩΔΙΚΑΣ_23** Κώδικας λειτουργίας της μονάδας
MPU6050 επιταχυνσιόμετρου & γυροσκοπίου 3 αξόνων*

Με χρήση του Serial Monitor για μελέτη της συμπεριφοράς της μονάδας MPU6050, παρατηρούμε ότι αυτή διαθέτει 7 χωρικές παραμέτρους οι οποίες προκύπτουν από τον τρόπο με τον οποίο το επιταχυνσιόμετρο, το γυροσκόπιο και ο αισθητήρας θερμοκρασίας είναι σχεδιασμένα να λειτουργούν:

- Οι 3 παράμετροι τις τιμές των οποίων φέρουν οι μεταβλητές **AcX**, **AcY** και **AcZ** αναφέρονται στο ενσωματωμένο επιταχυνσιόμετρο (*Accelerometer*) της μονάδας MPU6050 και εκφράζουν την (γραμμική) **επιτάχυνση** του αισθητήρα σε έναν χώρο κατά τους άξονες X, Y και Z.
 - Οι 3 παράμετροι που εκφράζονται με τις μεταβλητές **GyX**, **GyY** και **GyZ** αναφέρονται στο ενσωματωμένο γυροσκόπιο (*Gyroscope*) της μονάδας και εκφράζουν την **γωνιακή ταχύτητα και προσανατολισμό** του αισθητήρα στο χώρο κατά τους 3 άξονες.
 - Η παράμετρος που εκφράζεται με την μεταβλητή **Tmp** αναφέρεται στον ενσωματωμένο μικροαισθητήρα θερμοκρασίας (*Temperature Sensor*) της μονάδας και εκφράζει την **εσωτερική θερμοκρασία** της σε βαθμούς Κελσίου ($^{\circ}C$) ή Φαρενάιτ ($^{\circ}F$).
- Η παρακολούθηση της εσωτερικής θερμοκρασίας χρησιμεύει ώστε να βαθμονομείται σωστά η μονάδα για την μέτρηση των παραμέτρων επιταχυνσιομέτρου και γυροσκοπίου.

Στην εικόνα 3.98 εμφανίζονται οι 7 χωρικές παράμετροι του αισθητήρα MPU6050 σε κατάσταση ηρεμίας (αδράνεια) και άρα οι τιμές που παρουσιάζονται στην εικόνα διατηρούνται ως έχουν μέχρι η μονάδα να αλλάξει θέση στον τρισδιάστατο χώρο στον οποίο βρίσκεται.

Ενώ η μονάδα αισθητήρα βρίσκεται σε αδράνεια, οι τιμές που ορίζουν επιτάχυνση στους αντίστοιχους άξονες δεν είναι μηδέν.

```
MPU6050 Test
AcX = -6 | AcY = -4 | AcZ = 95 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 50 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 0
```

Εικόνα 3.98 Έλεγχος της συμπεριφοράς του αισθητήρα MPU6050 με μελέτη των παραμέτρων του μέσω του Serial Monitor

Αυτό συμβαίνει διότι το επιταχυνσιόμετρο σε ηρεμία στην επιφάνεια της Γης μετρά την ονομαζόμενη ως σωστή επιτάχυνση (*proper acceleration*) λόγω ύπαρξης της βαρύτητας της Γης.

Αντίθετα, αν η μονάδα MPU6050 εκτελέσει ελεύθερη πτώση (*free fall*) το επιταχυνσιόμετρο θα μετρήσει μηδενική επιτάχυνση.

Φυσικά η απόπειρα εκτέλεσης ελεύθερης πτώσης για δοκιμή μετρήσεων των παραμέτρων δεν είναι πρακτικά δύσκολη, είναι όμως αρκετά ριψοκίνδυνη για την διατήρηση της καλής κατάστασης και λειτουργίας της μονάδας αισθητήρα, για αυτό τον λόγο και στις μετρήσεις που ελήφθησαν και παρουσιάζονται στο πλαίσιο της παρούσας υποενότητας σπάνια απαντάται παράμετρος επιταχυνσιομέτρου με τιμή μηδέν ($AcX=0, AcY=0, AcZ=0$).

Οι τιμές των παραμέτρων της ανωτέρω εικόνας ελήφθησαν με τον αισθητήρα αδρανής σε ελαφρώς επικλινή επιφάνεια ($GyX=2, GyY=0, GyZ=0$).

```
AcX = -6 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = 7 | AcY = 2 | AcZ = 94 | Tmp = 49 C or 121 F | GyX = -6 | GyY = -16 | GyZ = -8
AcX = 80 | AcY = 0 | AcZ = 42 | Tmp = 49 C or 120 F | GyX = -8 | GyY = -19 | GyZ = -4
AcX = 87 | AcY = -6 | AcZ = 1 | Tmp = 49 C or 120 F | GyX = 1 | GyY = -6 | GyZ = -4
AcX = 87 | AcY = -6 | AcZ = -3 | Tmp = 49 C or 120 F | GyX = 1 | GyY = 1 | GyZ = 1
AcX = 85 | AcY = -10 | AcZ = 15 | Tmp = 49 C or 121 F | GyX = 29 | GyY = 10 | GyZ = -6
AcX = 54 | AcY = 7 | AcZ = 79 | Tmp = 49 C or 120 F | GyX = 0 | GyY = 22 | GyZ = -7
AcX = -29 | AcY = 4 | AcZ = 88 | Tmp = 49 C or 121 F | GyX = -4 | GyY = 31 | GyZ = -4
AcX = -89 | AcY = -11 | AcZ = -14 | Tmp = 49 C or 121 F | GyX = -15 | GyY = 41 | GyZ = -6
AcX = -72 | AcY = -7 | AcZ = -47 | Tmp = 49 C or 120 F | GyX = 3 | GyY = 0 | GyZ = 1
AcX = -88 | AcY = -26 | AcZ = -25 | Tmp = 49 C or 121 F | GyX = 12 | GyY = -39 | GyZ = 4
AcX = -21 | AcY = 9 | AcZ = 97 | Tmp = 49 C or 121 F | GyX = 19 | GyY = -20 | GyZ = 4
AcX = 10 | AcY = 84 | AcZ = -14 | Tmp = 49 C or 121 F | GyX = 12 | GyY = 0 | GyZ = -1
AcX = 12 | AcY = 86 | AcZ = -15 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 1
AcX = 13 | AcY = 85 | AcZ = -18 | Tmp = 49 C or 120 F | GyX = 3 | GyY = 0 | GyZ = 2
AcX = 19 | AcY = 56 | AcZ = 84 | Tmp = 49 C or 121 F | GyX = -69 | GyY = 9 | GyZ = 3
AcX = 11 | AcY = -73 | AcZ = 53 | Tmp = 49 C or 121 F | GyX = -2 | GyY = -1 | GyZ = 1
AcX = 9 | AcY = -76 | AcZ = 53 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 1
AcX = 2 | AcY = -5 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 1
```

Εικόνα 3.99 Εμφάνιση διάφορων τιμών των παραμέτρων του αισθητήρα MPU6050 με χρήση του Serial Monitor, όταν αυτός ολισθαίνει γραμμικά και περιστροφικά κατά τους άξονες X, Y και Z

```

AcX = 1 | AcY = -5 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = 1 | AcY = -6 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = 1 | AcY = -5 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = 1 | AcY = -5 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = 1 | AcY = -4 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = -1
AcX = 20 | AcY = -6 | AcZ = 131 | Tmp = 49 C or 120 F | GyX = 36 | GyY = 15 | GyZ = -11
AcX = -3 | AcY = 9 | AcZ = 175 | Tmp = 49 C or 120 F | GyX = 73 | GyY = -32 | GyZ = 34
AcX = 147 | AcY = 117 | AcZ = 40 | Tmp = 49 C or 120 F | GyX = 110 | GyY = -109 | GyZ = 110
AcX = 16 | AcY = -8 | AcZ = 89 | Tmp = 49 C or 120 F | GyX = 25 | GyY = 21 | GyZ = -21
AcX = 23 | AcY = 0 | AcZ = 94 | Tmp = 49 C or 120 F | GyX = -1 | GyY = -2 | GyZ = -4
AcX = 3 | AcY = -13 | AcZ = 94 | Tmp = 49 C or 120 F | GyX = -29 | GyY = 4 | GyZ = 24
AcX = 54 | AcY = -15 | AcZ = 88 | Tmp = 49 C or 120 F | GyX = -108 | GyY = 109 | GyZ = -63
AcX = 6 | AcY = -12 | AcZ = 99 | Tmp = 49 C or 121 F | GyX = 110 | GyY = -29 | GyZ = -25
AcX = -60 | AcY = -69 | AcZ = 140 | Tmp = 49 C or 120 F | GyX = -108 | GyY = 90 | GyZ = -39
AcX = -2 | AcY = -4 | AcZ = 97 | Tmp = 49 C or 121 F | GyX = 3 | GyY = -1 | GyZ = 1
AcX = 0 | AcY = -3 | AcZ = 96 | Tmp = 49 C or 121 F | GyX = 2 | GyY = 0 | GyZ = 1
AcX = 0 | AcY = -3 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0

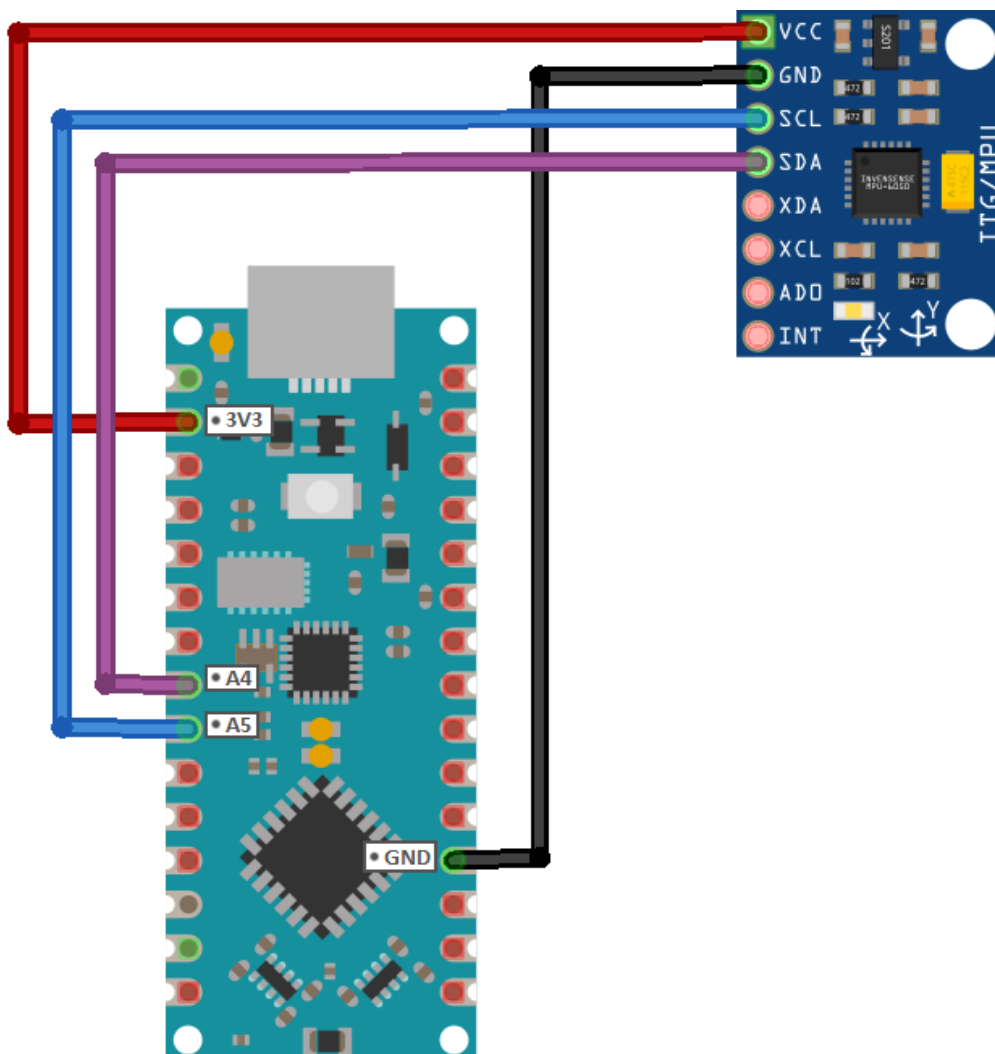
```

Εικόνα 3.100 Εμφάνιση διαφορών τιμών των παραμέτρων του αισθητήρα MPU6050 με χρήση του *Serial Monitor*, όταν αυτός ολισθαίνει γραμμικά και περιστροφικά κατά τους άξονες X, Y και Z

Στον πίνακα 3_9 και στην εικόνα 3.101 παρουσιάζεται η συνδεσμολογία μεταξύ του Arduino NANO EVERY και της μονάδας αισθητήρα MPU6050, με τον προγραμματιστικό κώδικα για την λειτουργία του κυκλώματος να είναι ο ίδιος με τον κώδικα Νο23.

ΠΙΝΑΚΑΣ 3_9

| | | | | |
|--|-------------|------------|------------|------------|
| Ακροδέκτες αισθητήρα MPU6050 | VCC | GND | SCL | SDA |
| Ακροδέκτες Arduino NANO EVERY | 3.3V | GND | A5 | A4 |



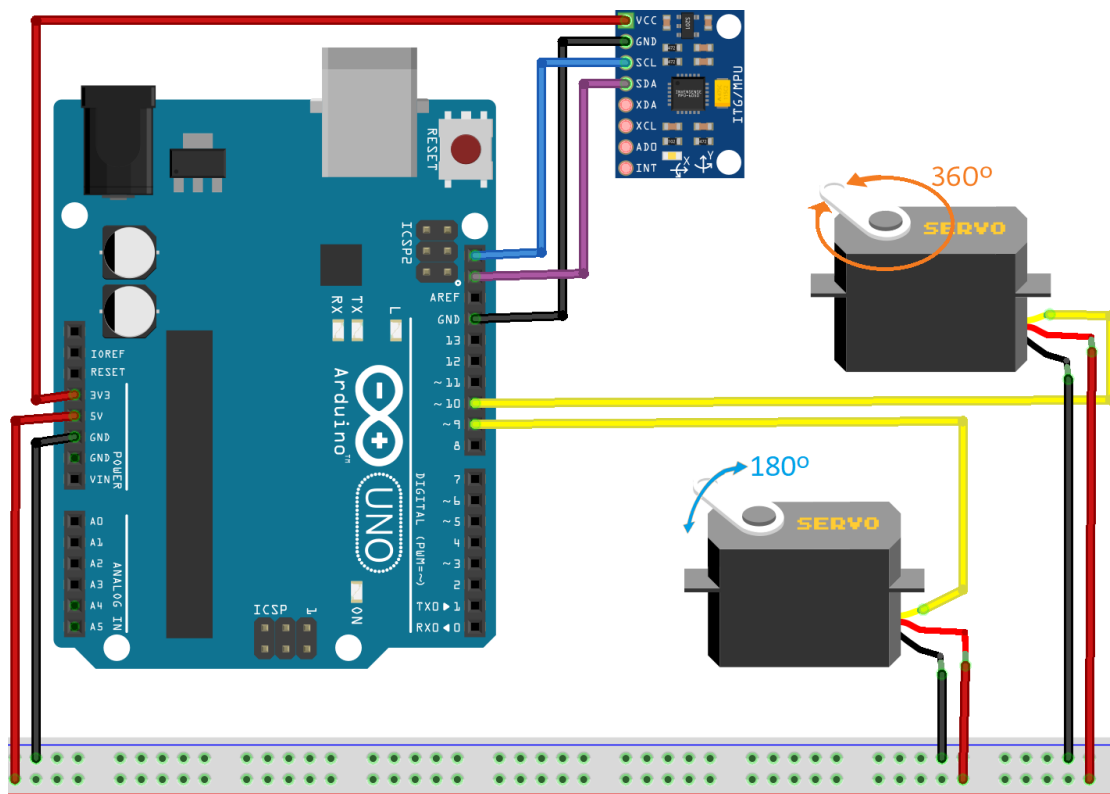
Εικόνα 3.101 Συνδεσμολογία μεταξύ Arduino NANO EVEY & μονάδας αισθητήρα MPU6050

Μονάδα MPU6050 ως ενσύρματο χειριστήριο 2 αξόνων για έλεγχο 2 σερβοκινητήρων

Για αυτή την περίπτωση, ο προγραμματιστικός κώδικας No23 τροποποιείται με τέτοιο τρόπο ώστε η μονάδα MPU6050 να λειτουργεί ως ενσύρματη μονάδα εισόδου εντολών για το κύκλωμα της εικόνας 3.102, ελέγχοντας 2 ψηφιακούς σερβοκινητήρες.

Όπως ακριβώς με έναν αναλογικό μοχλό 2 αξόνων, ο αισθητήρας MPU6050 ρυθμίζεται για να ολισθαίνει στους 2 άξονες X & Y, έτσι, ώστε με «πλοήγησή» του στον χώρο κατά τον X άξονα να ολισθαίνει τον σερβοκινητήρα MG995R δυνατότητας περιστροφής 180° και κατά τον Y άξονα τον σερβοκινητήρα MG995 δυνατότητας περιστροφής 360°.

Ο νέος κώδικας, υπεύθυνος για την λειτουργία αυτή, παρουσιάζεται στην επόμενη σελίδα.



Εικόνα 3.102(α) Μονάδα MPU6050 ως ενσύρματο χειριστήριο 2 αξόνων, σε περιβάλλον προσομοίωσης

```

/*
 * ΑΞΙΟΠΟΙΗΣΗ ΤΗΣ ΜΟΝΑΔΑΣ ΑΙΣΘΗΤΗΡΑ MPU6050 ΩΣ ΕΝΣΥΡΜΑΤΟ ΧΕΙΡΙΣΤΗΡΙΟ 2 ΑΞΟΝΩΝ
 * ΓΙΑ ΕΛΕΓΧΟ 2 ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ (1 MG995R 180 ΜΟΙΡΩΝ & 1 MG995 360 ΜΟΙΡΩΝ)
 */

#include <GY6050.h> // Library for GYRO
#include <Wire.h>
#include <Servo.h>
Servo myservo; // Create servo object to control a servo
Servo myservo2; // Create servo object to control a servo
int X = 0;
int Y = 0;

// Create Gyro object with "0x68" as I2C address (the address of the MPU6050) to save GYRO data
GY6050 gyro(0x68);

void setup() {

  Serial.begin(9600);
  Wire.begin(); // Initialize GYRO
  gyro.initialisation();
  delay(100);

  myservo.attach(9); // Set the 180-degree servo to Arduino pin 9
  myservo2.attach(10); // Set the 360-degree servo to Arduino pin 10

}

void loop() {

  // Mapping the gyro data according to angle limitation of servo motor
  X = map(gyro.refresh('A', 'X'), -90, 90, 0, 180);
  Serial.print("X Axis: ");
  Serial.print(X);
  Serial.print("\t");
  myservo.write(X); // Rotate servo with X-axis movement

  // Mapping the gyro data according to angle design of servo motor 2
  Y = map(gyro.refresh('A', 'Y'), -90, 90, 0, 180);
  Serial.print("Y Axis: ");
  Serial.println(Y);
  myservo2.write(Y); // Rotate servo 2 with Y-axis movement

  delay(200);
}

```

ΚΩΔΙΚΑΣ_24 Κώδικας λειτουργίας της μονάδας MPU6050 ως χειριστήριο 2 αξόνων για τον έλεγχο 2 ψηφιακών σερβοκινητήρων

| | |
|------------|------------|
| X Axis: 77 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 91 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 77 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 77 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 91 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 91 |

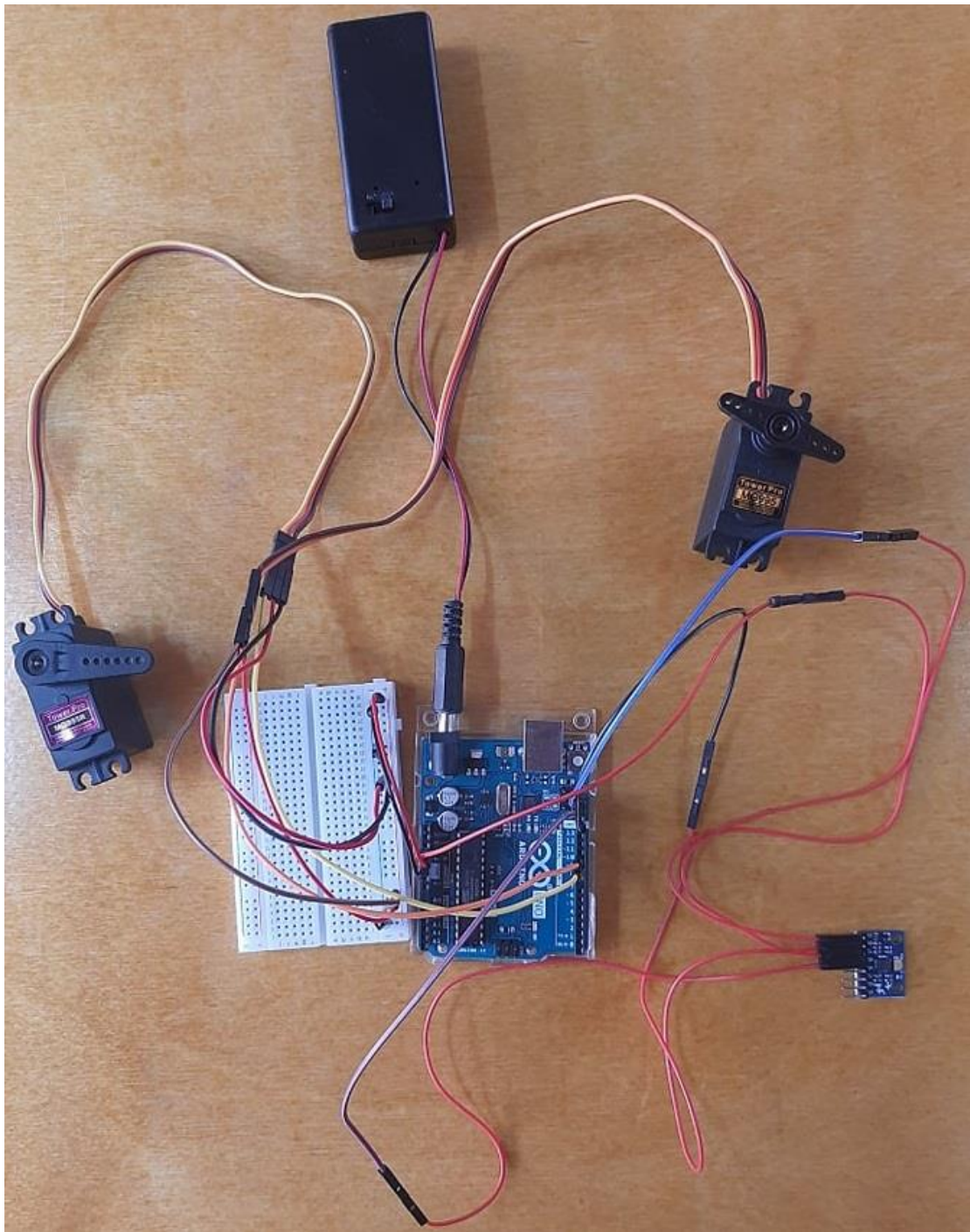
Εικόνα 3.103 Τιμές του οριζόντιου & κατακόρυφου άξονα X & Y αντίστοιχα, με διατήρηση του αισθητήρα MPU6050 σε αδράνεια. Παρότι ο αισθητήρας παραμένει στη θέση ηρεμίας, οι συγκεκριμένες τιμές αποτυπώθηκαν με αυτόν να βρίσκεται τοποθετημένος σε επιφάνεια με ελαφριά κλίση και όχι σε απόλυτα οριζόντια επιφάνεια.

| | |
|-------------|------------|
| X Axis: 77 | Y Axis: 92 |
| X Axis: 78 | Y Axis: 92 |
| X Axis: 76 | Y Axis: 94 |
| X Axis: 91 | Y Axis: 91 |
| X Axis: 117 | Y Axis: 85 |
| X Axis: 148 | Y Axis: 81 |
| X Axis: 154 | Y Axis: 79 |
| X Axis: 170 | Y Axis: 77 |
| X Axis: 167 | Y Axis: 75 |
| X Axis: 175 | Y Axis: 80 |
| X Axis: 174 | Y Axis: 80 |
| X Axis: 176 | Y Axis: 84 |
| X Axis: 173 | Y Axis: 81 |
| X Axis: 164 | Y Axis: 79 |
| X Axis: 161 | Y Axis: 76 |
| X Axis: 145 | Y Axis: 86 |
| X Axis: 111 | Y Axis: 80 |
| X Axis: 103 | Y Axis: 86 |
| X Axis: 92 | Y Axis: 87 |
| X Axis: 63 | Y Axis: 93 |
| X Axis: 41 | Y Axis: 88 |
| X Axis: 28 | Y Axis: 85 |
| X Axis: 10 | Y Axis: 79 |
| X Axis: 4 | Y Axis: 81 |
| X Axis: 1 | Y Axis: 84 |
| X Axis: 1 | Y Axis: 82 |
| X Axis: 5 | Y Axis: 77 |
| X Axis: 14 | Y Axis: 77 |
| X Axis: 13 | Y Axis: 80 |
| X Axis: 39 | Y Axis: 75 |
| X Axis: 61 | Y Axis: 78 |
| X Axis: 82 | Y Axis: 82 |
| X Axis: 84 | Y Axis: 89 |
| X Axis: 75 | Y Axis: 91 |
| X Axis: 75 | Y Axis: 92 |
| X Axis: 75 | Y Axis: 91 |
| X Axis: 76 | Y Axis: 91 |
| X Axis: 76 | Y Axis: 91 |
| X Axis: 75 | Y Axis: 92 |
| X Axis: 76 | Y Axis: 92 |
| X Axis: 77 | Y Axis: 91 |

Εικόνα 3.104 Διάφορες τιμές της μονάδας αισθητήρα MPU6050 για τους άξονες X & Y με σχετική ολίσθησή της στον περιβάλλοντα χώρο

| | |
|-------------|-------------|
| X Axis: 79 | Y Axis: 91 |
| X Axis: 81 | Y Axis: 93 |
| X Axis: 84 | Y Axis: 69 |
| X Axis: 117 | Y Axis: 41 |
| X Axis: 118 | Y Axis: 31 |
| X Axis: 116 | Y Axis: 35 |
| X Axis: 77 | Y Axis: 29 |
| X Axis: 61 | Y Axis: 96 |
| X Axis: 27 | Y Axis: 142 |
| X Axis: 38 | Y Axis: 126 |
| X Axis: 35 | Y Axis: 132 |
| X Axis: 60 | Y Axis: 120 |
| X Axis: 93 | Y Axis: 82 |
| X Axis: 79 | Y Axis: 89 |
| X Axis: 87 | Y Axis: 102 |
| X Axis: 111 | Y Axis: 139 |
| X Axis: 125 | Y Axis: 158 |
| X Axis: 116 | Y Axis: 154 |
| X Axis: 107 | Y Axis: 133 |
| X Axis: 57 | Y Axis: 39 |
| X Axis: 47 | Y Axis: 26 |
| X Axis: 37 | Y Axis: 24 |
| X Axis: 35 | Y Axis: 22 |
| X Axis: 63 | Y Axis: 53 |
| X Axis: 72 | Y Axis: 71 |
| X Axis: 75 | Y Axis: 82 |
| X Axis: 74 | Y Axis: 93 |
| X Axis: 75 | Y Axis: 92 |
| X Axis: 75 | Y Axis: 91 |
| X Axis: 75 | Y Axis: 91 |
| X Axis: 76 | Y Axis: 91 |
| X Axis: 75 | Y Axis: 91 |
| X Axis: 76 | Y Axis: 92 |
| X Axis: 76 | Y Axis: 91 |

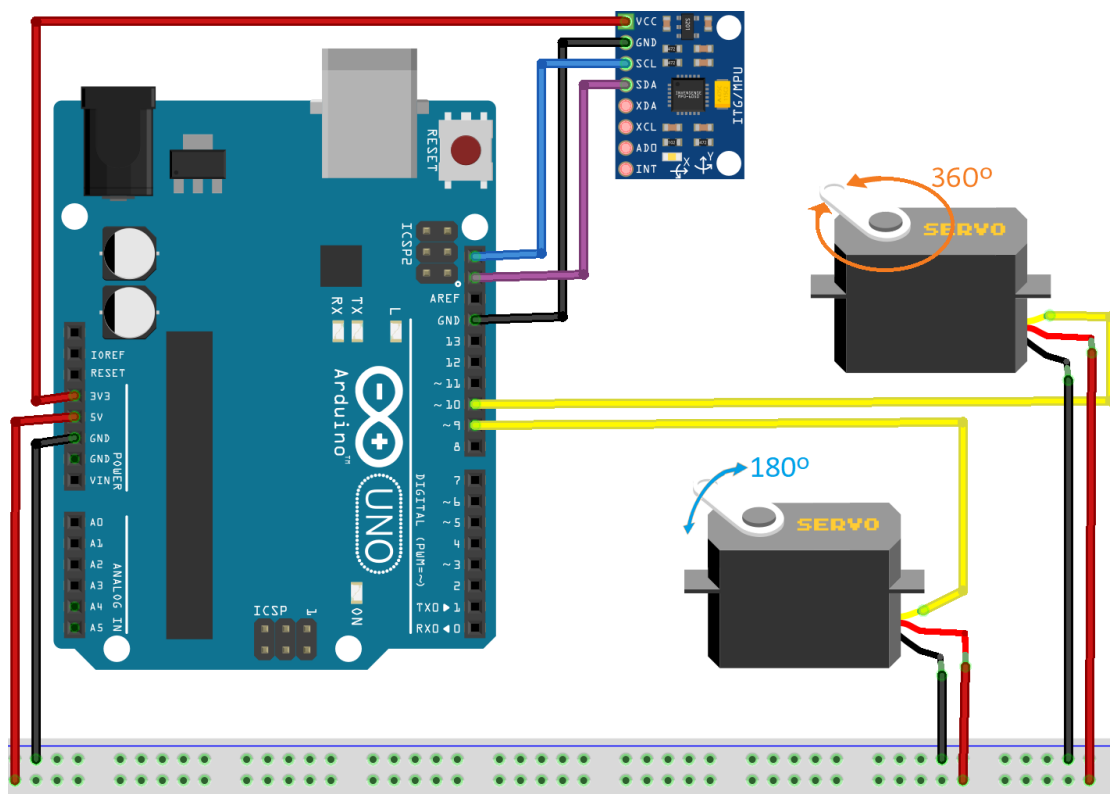
Εικόνα 3.105 Διάφορες τιμές της μονάδας MPU6050 για τους 2 άξονες X & Y με ολίσθησή της στον περιβάλλοντα χώρο και έπειτα τοποθέτησή της σε επικλινή επιφάνεια με αποτύπωση των τιμών $X = 76$ και $Y = 91$ για τον αισθητήρα σε αδράνεια.



*Εικόνα 3.106 Μονάδα MPU6050 ως ενσύρματο χειριστήριο
2 αξόνων υλοποιημένο σε επαγγελματικό εργαστήριο*

Μονάδα MPU6050 ως βελτιωμένο ενσύρματο χειριστήριο 3 αξόνων για έλεγχο 2 σερβοκινητήρων

Για το ίδιο κύκλωμα (εικ. 3.102), με τροποποίηση του κώδικα No24 για «πλοήγηση» της μονάδας αισθητήρα MPU6050 στον χώρο κατά τους 3 άξονες X, Y, και Z, γίνεται αντιληπτό ότι η παρακολούθηση της μετατόπισης του νέου βελτιωμένου «ενσύρματου χειριστηρίου» στον χώρο καθίσταται περισσότερο διακριτή και η παροχή εντολών εισόδου στο κύκλωμα εκτενέστερη.



Εικόνα 3.102(β) Μονάδα MPU6050 ως ενσύρματο χειριστήριο 3 αξόνων υλοποιημένο σε περιβάλλον προσομοίωσης

```

#include <Wire.h>
#include <GY6050.h>
// Create Gyro object with "0x68" as I2C address (most likely the address of your MPU6050)
GY6050 gyro(0x68);
void setup() {
  gyro.initialisation();
  Serial.begin(9600);
  Serial.println("MPU6050 Test of parameters measurement");
  Serial.println("----- The joystick will be temporarily disabled in case some of the parameters exceed certain limits -----");
  delay(2500);
}
void loop() {
  Serial.print("AcX = ");
  int AcX = gyro.refresh('A', 'X');
  Serial.print(AcX); // Ask for the X axis of the Accelerometer and print it

  Serial.print(" | AcY = ");
  int AcY = gyro.refresh('A', 'Y');
  Serial.print(AcY); // Ask for the Y axis of the Accelerometer and print it

  Serial.print(" | AcZ = ");
  int AcZ = gyro.refresh('A', 'Z');
  Serial.print(AcZ); // Ask for the Z axis of the Accelerometer and print it

  Serial.print(" | Tmp = ");
  int Tmp_C = gyro.refresh('T', 'C');
  Serial.print(Tmp_C); // Ask for the Temperature in Celsius and print it
  Serial.print(" C or ");
  int Tmp_F = gyro.refresh('T', 'F');
  Serial.print(Tmp_F); // Ask for the Temperature in Farenheit and print it
  Serial.print(" F");

  Serial.print(" | GyX = ");
  int GyX = gyro.refresh('G', 'X');
  Serial.print(GyX); // Ask for the X axis of the Gyroscope and print it

  Serial.print(" | GyY = ");
  int GyY = gyro.refresh('G', 'Y');
  Serial.print(GyY); // Ask for the Y axis of the Gyroscope and print it

  Serial.print(" | GyZ = ");
  int GyZ = gyro.refresh('G', 'Z');
  Serial.println(GyZ); // Ask for the Z axis of the Gyroscope and print it
  delay(1000);

  if (AcX > 80 || AcX < -80) {
    Serial.print("\n");
    Serial.print("AcX parameter off limits... joystick temporarily disabled !!!");
    Serial.print("\n");
  }
  if (AcY > 80 || AcY < -80) {
    Serial.print("\n");
    Serial.print("AcY parameter off limits... joystick temporarily disabled !!!");
    Serial.print("\n");
  }
  if (GyX > 80 || GyX < -80) {
    Serial.print("\n");
    Serial.print("GyX parameter off limits... joystick temporarily disabled !!!");
    Serial.print("\n");
  }
  if (GyY > 80 || GyY < -80) {
    Serial.print("\n");
    Serial.print("GyY parameter off limits... joystick temporarily disabled !!!");
    Serial.print("\n");
  }
  if (GyZ > 80 || GyZ < -80) {
    Serial.print("\n");
    Serial.print("GyZ parameter off limits... joystick temporarily disabled !!!");
    Serial.print("\n");
  }
  Serial.print("\n");
}

```

ΚΩΔΙΚΑΣ_25 Κώδικας λειτουργίας της μονάδας MPU6050 ως χειριστήριο 3 αξόνων για τον έλεγχο 2 ψηφιακών σερβοκινητήρων

Στον παρόν κώδικα, εκτός της μέτρησης των 7 χωρικών παραμέτρων, αναπτύσσεται τμήμα κώδικα (συνθήκες *if*) που επισημαίνει τις περιπτώσεις όπου το «χειριστήριο» ξεπερνάει καθορισμένα από τον προγραμματιστή χωρικά όρια, ενέργεια που θα αποβεί εξαιρετικά χρήσιμη σε μετέπειτα υλοποιήσεις.

Στις εικόνες 3.107 και 3.108 που ακολουθούν εμφανίζονται οι 7 χωρικές παράμετροι του αισθητήρα MPU6050 σε κατάσταση ηρεμίας (αδράνεια), με τοποθέτησή του σε μία ελαφρώς επικλινή επιφάνεια.

```

MPU6050 Test of parameters measurement
----- The joystick will be temporarily disabled in case some of the parameters exceed certain limits -----
AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
    
```

Εικόνα 3.107 Παρακολούθηση των τιμών των παραμέτρων της μονάδας MPU6050 σε κατάσταση ηρεμίας επί ελαφρώς επικλινούς επιφάνειας. Παρότι αδρανής, η μονάδα μετρά την σωστή επιτάχυνση (proper acceleration) λόγω της βαρύτητας (g) της Γης

```

MPU6050 Test of parameters measurement
----- The joystick will be temporarily disabled in case some of the parameters exceed certain limits -----
AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -11 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
    
```

Εικόνα 3.108 Μεγέθυνση του παραθύρου του Serial Monitor για παρακολούθηση των τιμών των παραμέτρων της μονάδας MPU6050 σε κατάσταση ηρεμίας. Εάν η μονάδα εκτελούσε ελεύθερη πτώση (free fall) τότε η μετρούμενη επιτάχυνση θα ήταν μηδέν

Στις εικόνες 3.109, 3.110 και 3.111 γίνεται μέσω του Serial Monitor αντιληπτή η παρακολούθηση και μέτρηση των 7 χωρικών παραμέτρων της μονάδας MPU6050 με ολίσθηση ή αλλιώς πλοήγησή της κατά τους 3 άξονες, καθώς και η εμφάνιση σχετικών μηνυμάτων όταν αυτή ξεπερνάει καθορισμένα χωρικά όρια που έχει θέσει ο προγραμματιστής.

```
AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -11 | AcY = 2 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -75 | AcY = 12 | AcZ = 57 | Tmp = 46 C or 115 F | GyX = 9 | GyY = 43 | GyZ = 5
AcX = -92 | AcY = -2 | AcZ = -7 | Tmp = 46 C or 115 F | GyX = 1 | GyY = 2 | GyZ = -1
AcX parameter off limits... joystick temporarily disabled !!!
AcX = -89 | AcY = 2 | AcZ = 15 | Tmp = 46 C or 115 F | GyX = -2 | GyY = -28 | GyZ = 8
AcX parameter off limits... joystick temporarily disabled !!!
AcX = -16 | AcY = 10 | AcZ = 103 | Tmp = 46 C or 115 F | GyX = -12 | GyY = -2 | GyZ = 0
AcX = -16 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -4 | AcY = -73 | AcZ = 53 | Tmp = 46 C or 115 F | GyX = -50 | GyY = -11 | GyZ = 9
AcX = -3 | AcY = -90 | AcZ = 0 | Tmp = 46 C or 115 F | GyX = -2 | GyY = 2 | GyZ = 0
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -9 | AcY = -84 | AcZ = 1 | Tmp = 46 C or 115 F | GyX = 28 | GyY = -4 | GyZ = 7
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -12 | AcY = 36 | AcZ = 82 | Tmp = 46 C or 115 F | GyX = 35 | GyY = 0 | GyZ = 5
AcX = -16 | AcY = 86 | AcZ = 15 | Tmp = 46 C or 115 F | GyX = 20 | GyY = 0 | GyZ = -2
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -22 | AcY = 88 | AcZ = 2 | Tmp = 46 C or 116 F | GyX = 3 | GyY = 0 | GyZ = 0
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -13 | AcY = 37 | AcZ = 92 | Tmp = 46 C or 116 F | GyX = -39 | GyY = 7 | GyZ = 3
AcX = -14 | AcY = 11 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 1 | GyY = 0 | GyZ = -1
AcX = -15 | AcY = 7 | AcZ = 95 | Tmp = 46 C or 116 F | GyX = 1 | GyY = 0 | GyZ = 0
```

Εικόνα 3.109 Παρακολούθηση των τιμών των παραμέτρων της μονάδας MPU6050, με χρήση και ολίσθησή της ως χειριστήριο 3 αξόνων στον περιβάλλοντα χώρο

```

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -11 | AcY = 2 | AcZ = 96 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -75 | AcY = 12 | AcZ = 57 | Tmp = 46 C or 115 F | GyX = 9 | GyY = 43 | GyZ = 5
AcX = -92 | AcY = -2 | AcZ = -7 | Tmp = 46 C or 115 F | GyX = 1 | GyY = 2 | GyZ = -1
AcX parameter off limits... joystick temporarily disabled !!!
AcX = -89 | AcY = 2 | AcZ = 15 | Tmp = 46 C or 115 F | GyX = -2 | GyY = -28 | GyZ = 8
AcX parameter off limits... joystick temporarily disabled !!!
AcX = -16 | AcY = 10 | AcZ = 103 | Tmp = 46 C or 115 F | GyX = -12 | GyY = -2 | GyZ = 0
AcX = -16 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -4 | AcY = -73 | AcZ = 53 | Tmp = 46 C or 115 F | GyX = -50 | GyY = -11 | GyZ = 9
AcX = -3 | AcY = -90 | AcZ = 0 | Tmp = 46 C or 115 F | GyX = -2 | GyY = 2 | GyZ = 0
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -9 | AcY = -84 | AcZ = 1 | Tmp = 46 C or 115 F | GyX = 28 | GyY = -4 | GyZ = 7
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -12 | AcY = 36 | AcZ = 82 | Tmp = 46 C or 115 F | GyX = 35 | GyY = 0 | GyZ = 5
AcX = -16 | AcY = 86 | AcZ = 15 | Tmp = 46 C or 115 F | GyX = 20 | GyY = 0 | GyZ = -2
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -22 | AcY = 88 | AcZ = 2 | Tmp = 46 C or 116 F | GyX = 3 | GyY = 0 | GyZ = 0
AcY parameter off limits... joystick temporarily disabled !!!
AcX = -13 | AcY = 37 | AcZ = 92 | Tmp = 46 C or 116 F | GyX = -39 | GyY = 7 | GyZ = 3
AcX = -14 | AcY = 11 | AcZ = 95 | Tmp = 46 C or 115 F | GyX = 1 | GyY = 0 | GyZ = -1
AcX = -15 | AcY = 7 | AcZ = 95 | Tmp = 46 C or 116 F | GyX = 1 | GyY = 0 | GyZ = 0

```

Εικόνα 3.110 Παρακολούθηση των τιμών των παραμέτρων της μονάδας MPU6050, με χρήση και ολίσησή της ως χειριστήριο 3 αξόνων στον περιβάλλοντα χώρο

```

AcX = -12 | AcY = 1 | AcZ = 96 | Tmp = 47 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -11 | AcY = 2 | AcZ = 95 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -38 | AcY = -63 | AcZ = 81 | Tmp = 46 C or 116 F | GyX = -108 | GyY = 109 | GyZ = -109
GyX parameter off limits... joystick temporarily disabled !!!
GyY parameter off limits... joystick temporarily disabled !!!
GyZ parameter off limits... joystick temporarily disabled !!!

AcX = -5 | AcY = -110 | AcZ = 94 | Tmp = 47 C or 116 F | GyX = -108 | GyY = -30 | GyZ = 58
AcY parameter off limits... joystick temporarily disabled !!!
GyX parameter off limits... joystick temporarily disabled !!!

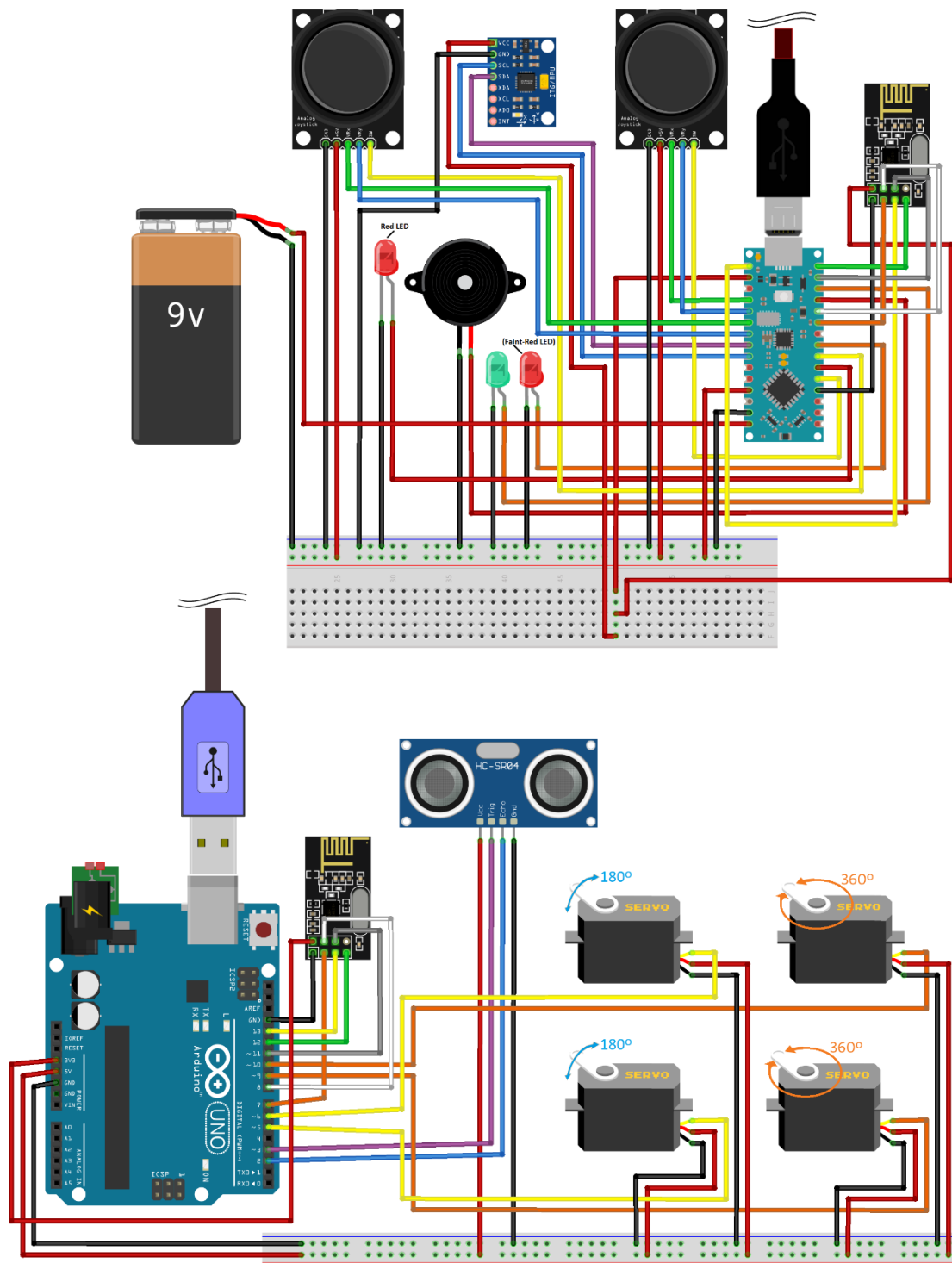
AcX = 25 | AcY = 3 | AcZ = 90 | Tmp = 46 C or 116 F | GyX = 8 | GyY = 11 | GyZ = -4
AcX = -8 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0
AcX = -7 | AcY = 1 | AcZ = 96 | Tmp = 46 C or 116 F | GyX = 2 | GyY = 0 | GyZ = 0

```

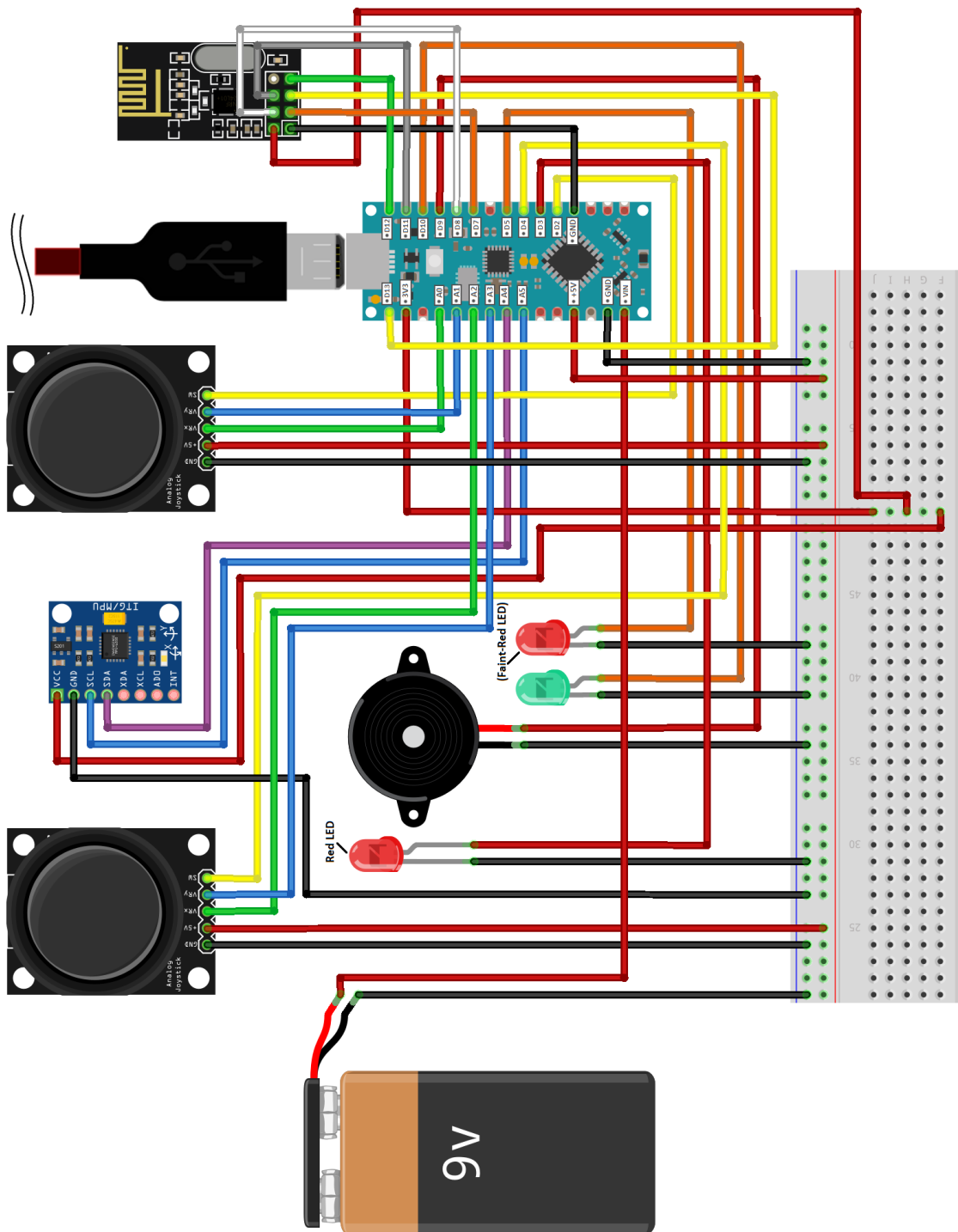
Εικόνα 3.111 Παρακολούθηση των τιμών των παραμέτρων της μονάδας MPU6050, με χρήση και ολίσθησή της ως χειριστήριο 3 αξόνων στον περιβάλλοντα χώρο

Ολοκληρώνοντας στο σημείο αυτό όλες τις ανωτέρω δοκιμές με επιτυχή διασύνδεση του μικροελεγκτή Arduino με εκάστη εκ των προαναφερθέντων 6 περιφερειακών μονάδων και εκτέλεση ορθού ελέγχου λειτουργίας των ως αυτόνομα ηλεκτρονικά κυκλώματα, υλοποιείται στην υποενότητα 3.4.5 που ακολουθεί ένα σύστημα ηλεκτρονικού ασύρματου ελεγκτή με προσθήκη και των 6 ανωτέρω περιφερειακών.

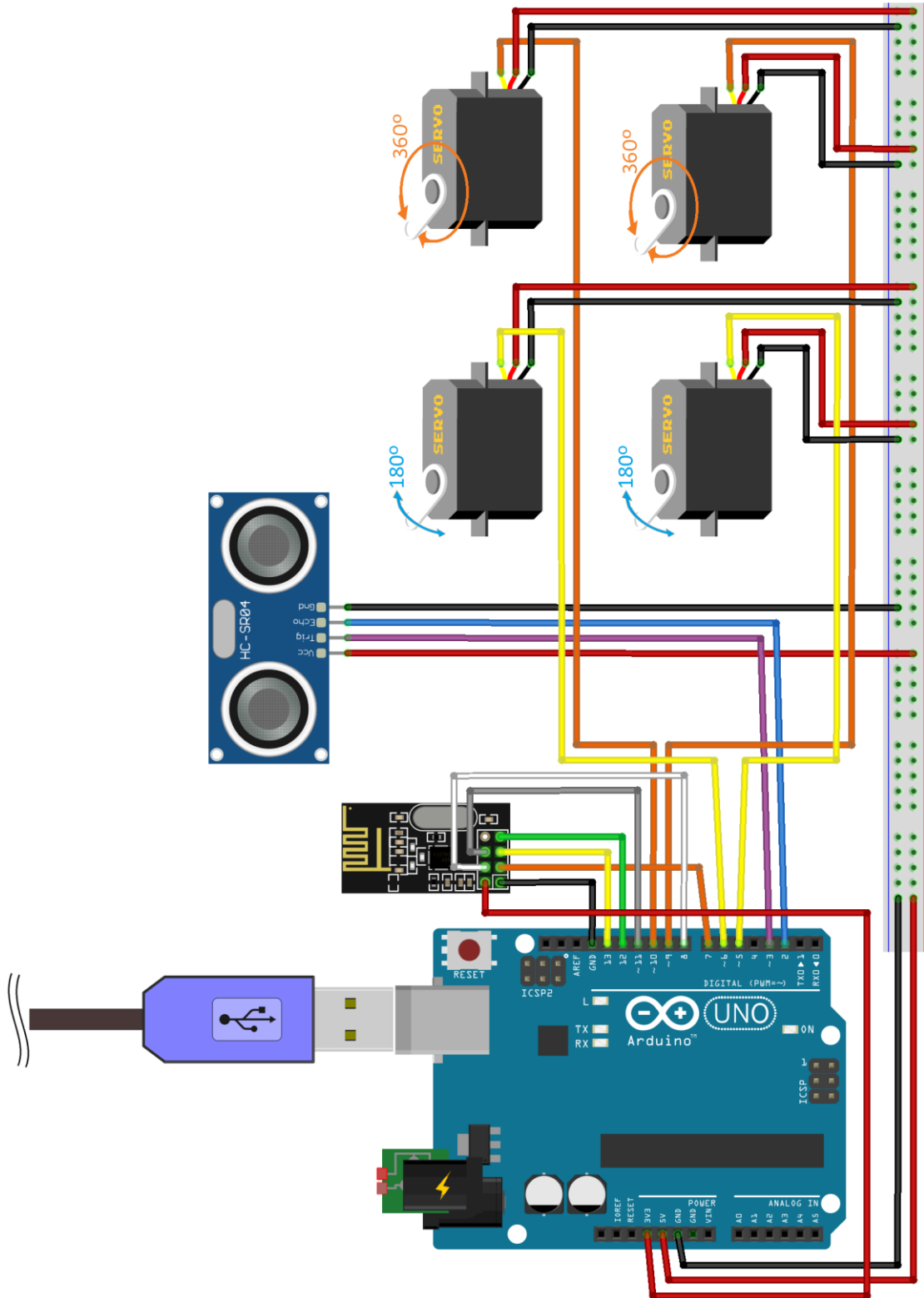
3.4.5 Έλεγχος Απόκρισης του Σ.Η.Α.Ε. με Πρόσθετες Περιφερειακές Μονάδες



Εικόνα 3.112 Σύστημα ηλεκτρονικού ασύρματου ελεγκτή (Σ.Η.Α.Ε.) με πρόσθετες περιφερειακές μονάδες, υλοποιημένο σε περιβάλλον προσομοίωσης



Εικόνα 3.113 Υποσύστημα Πομπού (Κύκλωμα TS) του συστήματος ηλεκτρονικού ασύρματου ελεγκτή



Εικόνα 3.114 Υποσύστημα Δέκτη (Κύκλωμα RS) του συστήματος ηλεκτρονικού ασύρματου ελεγκτή

Η βασική λειτουργία του **Σ.Η.Α.Ε.** με πρόσθετες περιφερειακές μονάδες είναι η **ασύρματη αποστολή και λήψη πληροφοριών από και προς τα δύο Υποσυστήματά του.**

Αρχικά το Υποσύστημα Πομπού, *Κύκλωμα TS* ή αλλιώς *Ασύρματο Χειριστήριο*, αποστέλλει την μηχανική κίνηση των 2 αναλογικών μοχλών του –εκφρασμένη σε μοίρες– στο Υποσύστημα Δέκτη, *Κύκλωμα RS* ή απλώς *Δέκτη* και ως αποτέλεσμα οι 4 ψηφιακοί σερβοκινητήρες περιστρέφονται κατά τις μοίρες αυτές.

Ως δευτερεύουσα λειτουργία, το Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή έχει σχεδιαστεί ώστε να λειτουργεί ως **Σύστημα Συναγερμού**, καθώς η περιφερειακή μονάδα αισθητήρα υπερήχων που είναι ενταγμένη στο Υποσύστημα Δέκτη μπορεί να ειδοποιήσει το Υποσύστημα Πομπού για εμφάνιση κίνησης σε έναν χώρο με κατάλληλη τοποθέτηση του αισθητήρα σε κάποιο σημείο του χώρου (*π.χ. είσοδος/έξοδος ενός εργασιακού χώρου ή χώρου στάθμευσης οχημάτων*).

Το Σ.Η.Α.Ε. περιλαμβάνει επίσης ένα **Σύστημα Ασφαλείας Χειρισμού** για το ασύρματο χειριστήριο, καθώς ενέργειες από τον χρήστη που περιλαμβάνουν αργή ή γρήγορη περιστροφική κίνηση μεγάλης κλίσης, έντονη ταλάντευση, απότομη περιστροφή ή πτώση του ασύρματου χειριστηρίου προκαλούν την ενεργοποίηση του κατάλληλα προγραμματισμένου κυκλώματός του με αποτέλεσμα οι 2 αναλογικοί μοχλοί του χειριστηρίου να βγαίνουν εκτός λειτουργίας, με μόνη ενέργεια επαναφοράς τους για έλεγχο των σερβοκινητήρων να είναι το πάτημα ενός εκ των 2 αναλογικών μοχλών για μερικά δευτερόλεπτα.

Προγραμματιστικός Κώδικας για το Κύκλωμα TS

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Wire.h>
#include <GY6050.h>
//create an RF24 object
RF24 radio(7, 8); // CE, CSN
//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};
// include an analog joystick of 2-axis
const int joystick_VRX = A0; // Connect the X-axis potentiometer to Arduino pin A0
const int joystick_VRY = A1; // Connect the Y-axis potentiometer to Arduino pin A1
const int joystick_VRX_2 = A2; // Connect the X_2-axis potentiometer to Arduino pin A2
const int joystick_VRY_2 = A3; // Connect the Y_2-axis potentiometer to Arduino pin A3
//include a Buzzer (piezo-speaker)
const int BUZZER = 9; // Connect Buzzer pin "Vcc" to Arduino D9
int Sensor_State = 0; //Initialize variable for the Buzzer & the Red LED
//include MPU6050 sensor unit (accelerometer & gyroscope)
GY6050 gyro(0x68); // Create Gyro object with "0x68" as I2C address (address of the MPU6050)
int SW_State, SW_2_State = 0; // Initialize 2 variables for the 2 joystick switches/push buttons,
// respectively

void setup() {
  Serial.begin(9600);
  Serial.print("Wireless Controller activating...\n*****\n");
  delay(200);
  pinMode(2, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 2'
  pinMode(4, INPUT_PULLUP); // Set pin connections as 'joystick_2_SW_2 = 4'
  pinMode(BUZZER, OUTPUT); // Set the Buzzer as output
  //include a LED of 'red' colour
  pinMode(3, OUTPUT); // Set a Red LED as output
  //Include a LED of 'faint red' colour
  pinMode(5, OUTPUT); // Set a Faint-Red LED as output
  //Include a LED of 'green' colour
  pinMode(10, OUTPUT); // Set a Green LED as output

  radio.begin();
  //set the address
  radio.openWritingPipe(addresses[1]); // 00002
  radio.openReadingPipe(1, addresses[0]); // 00001

  gyro.initialisation();
  Serial.println("MPU6050 Sensor Unit on the Controller is activated");
  Serial.print("----- The Controller will be temporarily disabled in case some ");
  Serial.println("of the spatial parameters exceed certain limits -----\n");
  delay(2500);
}

void loop() {
  // Set the Green LED ON
  digitalWrite(10, HIGH); //Green LED ON --> Wireless Controller works properly
  // Set module as transmitter
  radio.stopListening();
  // Calculation process of joystick & No2 joystick values
  int X_ADC_Value = analogRead(joystick_VRX); // Read the analog/mechanical input of X-axis
  int Y_ADC_Value = analogRead(joystick_VRY); // Read the analog/mechanical input of Y-axis
  int X_2_ADC_Value = analogRead(joystick_VRX_2); // Read the analog/mechanical input of X-axis
  int Y_2_ADC_Value = analogRead(joystick_VRY_2); // Read the analog/mechanical input of Y-axis
  delay(100);
  // Variables to store joystick values of range 0-180°
  int angleV1 = map(X_ADC_Value, 0, 1023, 0, 180); // Set the analog input signal of X-axis as values
  int angleV2 = map(Y_ADC_Value, 0, 1023, 0, 180); // Set the analog input signal of Y-axis as values
  int angleV3 = map(X_2_ADC_Value, 0, 1023, 0, 180); // Set the analog input signal of X_2-axis as values
  int angleV4 = map(Y_2_ADC_Value, 0, 1023, 0, 180); // Set the analog input signal of Y_2-axis as values
  delay(100); // waits 100ms for the servos to reach the corresponding angle
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

// Send joystick information (status & values) to receiver
Serial.print("Joystick value for X-axis: ");
Serial.print(angleV1);
radio.write(&angleV1, sizeof(angleV1));
Serial.print("\n");
delay(25);
Serial.print("Joystick value for Y-axis: ");
Serial.print(angleV2);
radio.write(&angleV2, sizeof(angleV2));
Serial.print("\n");
delay(25);
Serial.print("Joystick No2 value for X_2-axis: ");
Serial.print(angleV3);
radio.write(&angleV3, sizeof(angleV3));
Serial.print("\n");
delay(25);
Serial.print("Joystick No2 value for Y_2-axis: ");
Serial.print(angleV4);
radio.write(&angleV4, sizeof(angleV4));
Serial.println("\n");
delay(500);

// Set module as receiver
radio.startListening();
radio.read(&Sensor_State, sizeof(Sensor_State));
if (Sensor_State == 1) {
  do {
    Serial.print("The Red LED and the Buzzer for the ALARM are ON!");
    digitalWrite(3, HIGH);
    Serial.println("\n");
    tone(BUZZER, 1000);    // Send 1KHz sound signal
    delay(2000);
    radio.read(&Sensor_State, sizeof(Sensor_State));
  } while (Sensor_State == 1);
}
else if (Sensor_State == 0) {
  Serial.print("The Red LED and the Buzzer for the ALARM are OFF!");
  digitalWrite(3, LOW);
  Serial.println("\n");
  noTone(BUZZER);        // Stop the sound signal
}
delay(500);

// MPU6050 sensor unit parameters measurement
Serial.print("AcX = ");
int AcX = gyro.refresh('A', 'X');
Serial.print(AcX);      // Ask for the X axis of the Accelerometer and print it

Serial.print(" | AcY = ");
int AcY = gyro.refresh('A', 'Y');
Serial.print(AcY);      // Ask for the Y axis of the Accelerometer and print it

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΙΟΜΕΝΗ ΣΕΛΙΔΑ)

```

Serial.print(" | AcZ = ");
int AcZ = gyro.refresh('A', 'Z');
Serial.print(AcZ);      // Ask for the Z axis of the Accelerometer and print it

Serial.print(" | Tmp = ");
int Tmp_C = gyro.refresh('T', 'C');
Serial.print(Tmp_C);    // Ask for the Temperature in Celsius and print it
Serial.print(" C or ");
int Tmp_F = gyro.refresh('T', 'F');
Serial.print(Tmp_F);    // Ask for the Temperature in Farenheit and print it
Serial.print(" F");

Serial.print(" | GyX = ");
int GyX = gyro.refresh('G', 'X');
Serial.print(GyX);      // Ask for the X axis of the Gyroscope and print it

Serial.print(" | GyY = ");
int GyY = gyro.refresh('G', 'Y');
Serial.print(GyY);      // Ask for the Y axis of the Gyroscope and print it

Serial.print(" | GyZ = ");
int GyZ = gyro.refresh('G', 'Z');
Serial.println(GyZ);    // Ask for the Z axis of the Gyroscope and print it
delay(1000);

if (AcX > 70 || AcX < -70) {
  Serial.print("\n");
  delay(100);
  do {
    Serial.print("AcX parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);          // Turn the Green LED OFF
    digitalWrite(5, HIGH);         // Turn the Faint-Red LED ON
    //delay(25);
    SW_State = !digitalRead(2);     // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);   // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);        // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);     // Turn the Green LED ON
      break;
    }
  } while (1);
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΙΟΜΕΝΗ ΣΕΛΙΔΑ)

```

if (AcY > 70 || AcY < -70) {
  Serial.print("\n");
  do {
    Serial.print("AcY parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);           // Turn the Green LED OFF
    digitalWrite(5, HIGH);          // Turn the Faint-Red LED ON
    //delay(25);
    SW_State = !digitalRead(2);      // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);    // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);          // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);        // Turn the Green LED ON
      break;
    }
  } while (1);
}

if (GyX > 80 || GyX < -80) {
  Serial.print("\n");
  do {
    Serial.print("GyX parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);           // Turn the Green LED OFF
    digitalWrite(5, HIGH);          // Turn the Faint-Red LED ON
    //delay(25);
    SW_State = !digitalRead(2);      // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);    // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);          // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);        // Turn the Green LED ON
      break;
    }
  } while (1);
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)


```

if (GyY > 80 || GyY < -80) {
  Serial.print("\n");
  do {
    Serial.print("GyY parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);           // Turn the Green LED OFF
    digitalWrite(5, HIGH);          // Turn the Faint-Red LED ON
    //delay(25);
    SW_State = !digitalRead(2);      // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);    // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);          // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);        // Turn the Green LED ON
      break;
    }
  } while (1);
}

if (GyZ > 80 || GyZ < -80) {
  Serial.print("\n");
  do {
    Serial.print("GyZ parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);           // Turn the Green LED OFF
    digitalWrite(5, HIGH);          // Turn the Faint-Red LED ON
    //delay(25);
    SW_State = !digitalRead(2);      // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);    // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);          // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);        // Turn the Green LED ON
      break;
    }
  } while (1);
}
Serial.print("\n");
}

```

ΚΩΔΙΚΑΣ_26 Κώδικας λειτουργίας του Κυκλώματος TS (Ασύρματο Χειριστήριο)

Προγραμματιστικός Κώδικας για το Κύκλωμα RS

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>      // Add the servo library

// Create an RF24 object
RF24 radio(7, 8); // CE, CSN
//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};

// Include 2 servos of 180 degree rotation
Servo myServo1; // Define 'myServo1' as Servo1 servo motor
Servo myServo2; // Define 'myServo2' as Servo2 servo motor
Servo myServo3; // Define 'myServo3' as Servo3 servo motor
Servo myServo4; // Define 'myServo4' as Servo4 servo motor

// variables to store joysticks' values sent by the transmitter (TS Circuit)
int angleV_1 = 0; // Initiallize variable to store Joystick values of X-axis for Servo1
int angleV_2 = 0; // Initiallize variable to store Joystick value of Y-axis for Servo2
int angleV_3 = 0; // Initiallize variable to store Joystick No2 values of X_2-axis for Servo3
int angleV_4 = 0; // Initiallize variable to store Joystick No2 values of Y_2-axis for Servo4

// Include 1 HC-RS04 ultrasonic sensor
const int Trig_pin = 3;
const int Echo_pin = 2;
// define variables
float duration, distance;
int Sensor_State = 0;

void setup() {
  pinMode(Trig_pin, OUTPUT); // Set the Trig pin as an Output
  pinMode(Echo_pin, INPUT); // Set the Echo pin as an Input
  //pinMode(buzzer, OUTPUT); // Set the connected to pin 9 buzzer as an output
  Serial.begin(9600);

  radio.begin();
  //set the address
  radio.openWritingPipe(addresses[0]); // 00001
  radio.openReadingPipe(1, addresses[1]); // 00002

  myServo1.attach(5); // attaches Servo1 Signal Line to pin 5 (PWM)
  myServo2.attach(6); // attaches Servo2 Signal Line to pin 6 (PWM)

  myServo3.write(90); // Initialize Servo1 rotation speed to zero (90 degrees)
  myServo4.write(90); // Initialize Servo2 rotation speed to zero (90 degrees)
  myServo3.attach(9); // attaches Servo3 Signal Line to pin 9 (PWM)
  myServo4.attach(10); // attaches Servo4 Signal Line to pin 10 (PWM)
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

void loop() {
  //Set module as receiver
  radio.startListening();

  //Read the data if available in buffer
  if (radio.available()) {
    while (radio.available()) {

      // Receive joystick information to rotate the 2 Servo motors accordingly
      radio.read(&angleV_1, sizeof(angleV_1)); // Read & store X-axis information of Joystick
      Serial.print("For X-axis value: ");
      Serial.print(angleV_1);
      Serial.print("\n");
      if (angleV_1 < 10 || angleV_1 > 170) {
        Serial.print(" ---->Servol position/angle is ");
        Serial.print(angleV_1);
        Serial.println("\n");
        myServo1.write(angleV_1); // Use the variable value to rotate Servo 1
      }
      else if (angleV_1 > 10 && angleV_1 < 170) {
        // print the standby position in degrees
        Serial.println(" ---->Servol on standby mode");
      }
    }
    delay(100);

    radio.read(&angleV_2, sizeof(angleV_2)); // Read & store Y-axis information of Joystick
    Serial.print("For Y-axis value: ");
    Serial.print(angleV_2);
    Serial.print("\n");
    if (angleV_2 < 10 || angleV_2 > 170) {
      Serial.print(" ---->Servo2 position/angle is ");
      Serial.print(angleV_2);
      Serial.println("\n");
      myServo2.write(angleV_2); // Use the variable value to rotate Servo 2
    }
    else if (angleV_2 > 10 && angleV_2 < 170) {
      // print the standby position in degrees
      Serial.println(" ---->Servo2 on standby mode");
    }
  }
  delay(100);

  radio.read(&angleV_3, sizeof(angleV_3));
  Serial.print("For X_2-axis value: "); // Read & store X_2-axis information of Joystick No2
  Serial.print(angleV_3);
  Serial.print("\n");
  if (angleV_3 < 85 || angleV_3 > 95) {
    Serial.print(" -->Servo3 rotational speed is ");
    Serial.print(angleV_3);
    Serial.println("\n");
    myServo3.write(angleV_3); // Use the variable value to rotate Servo 3
  }
  else if (angleV_3 > 85 && angleV_3 < 95) {
    Serial.println(" -->Servo3 in inertia");
    myServo3.write(90); // Use the variable value to stagnate Servo 3
  }
}
delay(100);

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΙΟΜΕΝΗ ΣΕΛΙΔΑ)

```

// Read & store Y_2-axis information of Joystick No2
radio.read(&angleV_4, sizeof(angleV_4));
Serial.print("For Y_2-axis value: ");
Serial.print(angleV_4);
Serial.print("°");
if (angleV_4 <= 85) {
  Serial.print(" -->Servo4 rotational speed is ");
  Serial.print(angleV_4);
  Serial.println("°, but fixed at 80°");
  myServo4.write(80);          // Use the variable value to rotate Servo 4
}
else if (angleV_4 >= 95) {
  Serial.print(" -->Servo4 rotational speed is ");
  Serial.print(angleV_4);
  Serial.println("°, but fixed at 100°");
  myServo4.write(100);       // Use the variable value to rotate Servo 4
}
else if (angleV_4 > 85 && angleV_4 < 95) {
  Serial.println(" -->Servo4 in inertia");
  myServo4.write(90);       // Use the variable value to stagnate Servo 4
}
Serial.print("\n");
delay(25);

// Clear the Trig pin condition
digitalWrite(Trig_pin, LOW);
delayMicroseconds(2);

// Set the Trig pin HIGH (ACTIVE) for 10µs
digitalWrite(Trig_pin, HIGH);    // Generate the sound wave
delayMicroseconds(10);
digitalWrite(Trig_pin, LOW);    // Clear the Trig pin condition

// Read the Echo pin and return the sound wave travel time in µs
duration = pulseIn(Echo_pin, HIGH);
// The reflected wave turns the Echo pin ON (HIGH) and the timing starts.
// When the wave ends the Echo pin goes OFF (LOW) and the timing stops,
// returning the length of the reflected sound wave in µs.

// Calculate the distance
distance = (duration * 0.034) / 2;
// Speed of sound wave divided by 2 (transmission & reception time travel)

// Display the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
delay(100);

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΙΟΜΕΝΗ ΣΕΛΙΔΑ)

```

if (distance <= 20) {
  //Set module as transmitter
  radio.stopListening();
  do {
    myServo3.write(90);
    myServo4.write(90);

    Serial.print("Sending sound signal (1KHz)...");
    //tone(buzzer, 1000);          // Send 1KHz sound signal
    Serial.print("Obstacle at: ");
    Serial.print(distance);
    Serial.println("cm!\n");
    Sensor_State = 1;
    radio.write(&Sensor_State, sizeof(Sensor_State));
    delay(50);          // ...for 50ms or 2sec

    // Clear the Trig pin condition
    digitalWrite(Trig_pin, LOW);
    delayMicroseconds(2);

    // Set the Trig pin HIGH (ACTIVE) for 10µs
    digitalWrite(Trig_pin, HIGH);    // Generate the sound wave
    delayMicroseconds(10);
    digitalWrite(Trig_pin, LOW);    // Clear the Trig pin condition

    // Read the Echo pin and return the sound wave travel time in µs
    duration = pulseIn(Echo_pin, HIGH);
    // The reflected wave turns the Echo pin ON (HIGH) and the timing starts.
    // When the wave ends the Echo pin goes OFF (LOW) and the timing stops,
    // returning the length of the reflected sound wave in µs.

    // Calculate the distance
    distance = (duration * 0.034) / 2;
    // Speed of sound wave divided by 2 (transmission & reception time travel)
    delay(25);
  } while (distance <= 20);

  delay(200);
}
}
delay(200);
}
}

```

ΚΩΔΙΚΑΣ_27 Κώδικας λειτουργίας του Κυκλώματος RS (Δέκτης)

Το Κύκλωμα RS λαμβάνει τις τιμές των 4 ποτενσιομέτρων των 2 αναλογικών μοχλών του ασύρματου χειριστηρίου (Κύκλωμα TS) σε μοίρες ώστε να αντιστοιχίζεται πλήρως η ολίσθηση των 2 μοχλών του χειριστηρίου με την περιστροφική κίνηση του άξονα των 4 σερβοκινητήρων.

Με χρήση του Serial Monitor για το Κύκλωμα TS, στην εικόνα 3.115 παρουσιάζονται τα σχετικά μηνύματα που υποδεικνύουν την ενεργοποίηση του ασύρματου χειριστηρίου και αμέσως μετά ακολουθούν οι ενδείξεις των τιμών των 4 ποτενσιομέτρων κατά την ενεργοποίησή του, εκφρασμένες σε μοίρες. Όταν το χειριστήριο βρίσκεται σε αδράνεια, οι εν λόγω τιμές διατηρούνται όπως παρουσιάζονται στην εικόνα και αποστέλλονται διαρκώς με την βοήθεια των πομποδεκτών NRF24L01 από το Κύκλωμα TS στο Κύκλωμα RS.

```

Wireless Controller activating...
*****
MPU6050 Sensor Unit on the Controller is activated
----- The Controller will be temporarily disabled in case some of the spatial parameters exceed certain limits -----

Joystick value for X-axis: 93°
Joystick value for Y-axis: 89°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -8 | AcY = -1 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

```

***Εικόνα 3.115** Εμφάνιση των τιμών των παραμέτρων του ασύρματου χειριστηρίου (Κύκλωμα TS) σε αδράνεια*

Η εικόνα 3.116 αποτυπώνει την συνεχή λήψη τιμών αδρανείας του χειριστηρίου, όπως αυτές λαμβάνονται από το Κύκλωμα RS μέσω του πομποδέκτη NRF και εμφανίζονται με χρήση του Serial Monitor:

```

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 234.06 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 232.68 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 232.07 cm

```

***Εικόνα 3.116** Εμφάνιση των τιμών των παραμέτρων του Υποσυστήματος Δέκτη (Κύκλωμα RS), με το χειριστήριο σε αδράνεια*

Στην εικόνα 3.117 γίνεται χρήση του Serial Monitor για το Κύκλωμα TS, με εμφανή την μεταβολή των τιμών των ποτενσιομέτρων για διάφορες περιπτώσεις ολίσθησης των μοχλών του χειριστηρίου κατά τους X και Y άξονες. Η εικόνα αυτή αποτελεί συνέχεια της εικόνας 3.115, με το Κύκλωμα TS να στέλνει αδιάκοπα στο Κύκλωμα RS την κατάσταση των αναλογικών μοχλών, σε μοίρες.

```

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°
}
ΑΝΑΛΟΓΙΚΟΙ ΜΟΧΛΟΙ 1 & 2
ΣΕ ΑΔΡΑΝΕΙΑ

The Red LED and the Buzzer for the ALARM are OFF! ← ΣΥΣΤΗΜΑ ΣΥΝΑΓΕΡΜΟΥ ΑΝΕΝΕΡΓΟ

AcX = -9 | AcY = 0 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 0°
Joystick value for Y-axis: 0°
}
ΑΡΙΣΤΕΡΗ & ΑΝΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 1 (X=0 & Y=0)
Joystick No2 value for X_2-axis: 180°
Joystick No2 value for Y_2-axis: 180°
}
ΔΕΞΙΑ & ΚΑΤΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 2 (X_2=180 & Y_2=180)

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -8 | AcY = 0 | AcZ = 97 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°
}
ΜΟΧΛΟΙ 1 & 2
ΣΕ ΑΔΡΑΝΕΙΑ

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -9 | AcY = 0 | AcZ = 96 | Tmp = 48 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 180°
Joystick value for Y-axis: 179°
}
ΔΕΞΙΑ & ΚΑΤΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 1 (X=180 & Y=180)
Joystick No2 value for X_2-axis: 0°
Joystick No2 value for Y_2-axis: 0°
}
ΑΡΙΣΤΕΡΗ & ΑΝΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 2 (X_2=0 & Y_2=0)

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -9 | AcY = 0 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°
}
ΜΟΧΛΟΙ 1 & 2
ΣΕ ΑΔΡΑΝΕΙΑ

The Red LED and the Buzzer for the ALARM are OFF!

```

Εικόνα 3.117 Διάφορες καταστάσεις ολίσθησης και αντίστοιχες τιμές των μοχλών του ασύρματου χειριστηρίου, με το Σύστημα Συναγερμού (ALARM) σε κατάσταση OFF

Για την εικόνα 3.117, εφόσον οι ενδείξεις “Red LED” και “Buzzer” είναι απενεργοποιημένες (OFF) σημαίνει ότι το Σύστημα Συναγερμού (ALARM) είναι ανενεργό, δηλαδή βάσει του κώδικα Νο27 ο αισθητήρας υπερήχων του Κυκλώματος RS δεν ενεργοποιείται κι έτσι δεν αποστέλλει σήμα για ενεργοποίηση του Συστήματος Συναγερμού (ΚΩΔΙΚΑΣ_26) που διαθέτει το Κύκλωμα TS.

Στην εικόνα 3.118 της επόμενης σελίδας γίνεται και πάλι χρήση του Serial Monitor για το Κύκλωμα RS, με τις εκάστοτε ληφθείσες τιμές να εξαρτώνται από τους αναλογικούς μοχλούς του Κυκλώματος TS.

Επίσης, με κατάλληλη τροποποίηση του κώδικα για το Κύκλωμα RS, ο σερβοκινητήρας «Servo4» περιστρέφεται είτε με σταθερή ταχύτητα 80° για ολίσθηση του 2^{ου} μοχλού οποιασδήποτε κλίσης εύρους 0-90° είτε με σταθερή ταχύτητα 100° για ολίσθηση του μοχλού στο εύρος των 90-180°, με την ενέργεια αυτή να αποτελεί απλά ένα δείγμα της δυναμικής που φέρει η εκάστοτε τροποποίηση του προγραμματιστικού κώδικα και επιφέρει αλλαγές τέτοιες που η απόκριση του Σ.Η.Α.Ε. δύναται να διαφοροποιείται για την κάλυψη των αναγκών της εκάστοτε εφαρμογής.


```

For Y_2-axis value: 89 -->Servo4 in inertia

Distance: 233.34 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 233.07 cm
For X-axis value: 0° ----->Servo1 position/angle is 0°
For Y-axis value: 180° ----->Servo2 position/angle is 180°
For X_2-axis value: 0° -->Servo3 rotational speed is 0°
For Y_2-axis value: 114° -->Servo4 rotational speed is 114°, but fixed at 100°

Distance: 232.76 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 232.07 cm
For X-axis value: 139° ----->Servo1 on standby mode
For Y-axis value: 0° ----->Servo2 position/angle is 0°
For X_2-axis value: 180° -->Servo3 rotational speed is 180°
For Y_2-axis value: 47° -->Servo4 rotational speed is 47°, but fixed at 80°

Distance: 106.20 cm
For X-axis value: 139° ----->Servo1 on standby mode
For Y-axis value: 0° ----->Servo2 position/angle is 0°
For X_2-axis value: 180° -->Servo3 rotational speed is 180°
For Y_2-axis value: 47° -->Servo4 rotational speed is 47°, but fixed at 80°

Distance: 231.64 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 232.93 cm

```

***Εικόνα 3.118** Διάφορες τιμές των 4 σερβοκινητήρων, παρεχόμενες σε μοίρες από το Υποσύστημα Πομπού. Η ένδειξη «Distance» μετρά την απόσταση μεταξύ αισθητήρα υπερήχων και κάποιου αντικειμένου, σε πραγματικό χρόνο*

Οι εικόνες 3.117 & 3.118 δεν έχουν χρονική συνέχεια μεταξύ τους, καθώς αποτελούν 2 διαφορετικά στιγμιότυπα εικόνων που ελήφθησαν σε διαφορετικό χρόνο για το κάθε κύκλωμα.

```

For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 233.56 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 231.66 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 31.35 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 25.23 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 20.35 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 22.42 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

```

Εικόνα 3.119 Τιμές χειριστηρίου σε αδράνεια και αντίστοιχη αδρανοποίηση σερβοκινητήρων, με την ένδειξη “Distance” να φανερώσει την μεταβολή της θέσης ενός αντικειμένου ως προς τον αισθητήρα υπερήχων με κάθε νέα μέτρηση

Όταν το χειριστήριο βρίσκεται σε αδράνεια, το Κύκλωμα RS λαμβάνει συνεχώς τις τιμές που απεικονίζονται στην εικόνα 3.120 και ως αποτέλεσμα οι σερβοκινητήρες θέτονται εξίσου σε αδράνεια.

Λόγω της αδράνειας αυτής και του κατάλληλου προγραμματιστικού κώδικα που αναπτύχθηκε για το Σ.Η.Α.Ε., το ζεύγος MG995R περιστροφής 180° βρίσκεται σε κατάσταση αναμονής (*standby mode*), διατηρώντας την θέση (*κλίση*) στην οποία είχε μετατοπισθεί για τελευταία φορά μεταξύ των δύο θέσεων εναλλαγής 0° ή 180°.

Το ζεύγος MG995 περιστροφής 360° παραμένει σε αδράνεια (*inertia*) λόγω των τιμών που λαμβάνει (οι τιμές 91° και 89° αντίστοιχα παρέχουν στους *Servo3* & *Servo4* μηδενική γωνιακή ταχύτητα).

Επίσης, βάσει του κώδικα προγραμματισμού, η μονάδα αισθητήρα υπερήχων του Δέκτη είναι ρυθμισμένη να στέλνει πληροφορίες στο Ασύρματο Χειριστήριο για τις περιπτώσεις που κάποιο αντικείμενο πλησιάσει την μονάδα σε απόσταση των 20cm και κάτω.

```
Distance: 233.65 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 22.78 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 19.63 cm
Sending sound signal (1KHz)...Obstacle at: 19.63cm!

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 232.88 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2
```

Εικόνα 3.120 Σερβοκινητήρες σε αδράνεια λόγω αδράνειας χειριστηρίου και μια περίπτωση αποστολής ηχητικού σήματος 1kHz με εμφάνιση εντός αντικειμένου στα 19,63cm

```

Distance: 233.05 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 18.94 cm
Sending sound signal (1kHz)...Obstacle at: 18.94cm!

Sending sound signal (1kHz)...Obstacle at: 19.13cm!

Sending sound signal (1kHz)...Obstacle at: 19.70cm!

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 16.85 cm
Sending sound signal (1kHz)...Obstacle at: 16.85cm!

Sending sound signal (1kHz)...Obstacle at: 16.76cm!

Sending sound signal (1kHz)...Obstacle at: 16.81cm!

Sending sound signal (1kHz)...Obstacle at: 17.12cm!

Sending sound signal (1kHz)...Obstacle at: 17.05cm!

Sending sound signal (1kHz)...Obstacle at: 17.05cm!

Sending sound signal (1kHz)...Obstacle at: 17.83cm!

Sending sound signal (1kHz)...Obstacle at: 19.89cm!

For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 90° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia
For Y_2-axis value: 89° -->Servo4 in inertia

Distance: 233.89 cm
For X-axis value: 93° ----->Servo1 on standby mode
For Y-axis value: 89° ----->Servo2 on standby mode
For X_2-axis value: 91° -->Servo3 in inertia

```

***Εικόνα 3.121** Σερβοκινητήρες σε αδράνεια λόγω αδράνειας χειριστηρίου και περιπτώσεις αποστολής ηχητικού σήματος 1kHz με εμφάνιση εντός αντικειμένου για διάφορες αποστάσεις άνω και κάτω των 20cm ως προς τον αισθητήρα*

Στην εικόνα 3.122 παρουσιάζεται με μηνύματα κόκκινου υπογραμμισμένου χρώματος η κατάσταση του ασύρματου χειριστηρίου όταν οι αναλογικοί του μοχλοί είναι σε αδράνεια και η μονάδα αισθητήρα υπερήχων του Δέκτη έχει ενημερώσει το χειριστήριο για την ύπαρξη αντικειμένου σε απόσταση κάτω των 20cm με αποτέλεσμα την ενεργοποίηση του Συστήματος Συναγερμού του χειριστηρίου, το οποίο αποτελείται από ένα κόκκινο LED που φωτοβολεί και ένα Buzzer που παράγει ηχητικό σήμα 1kHz για την οπτική και ηχητική ειδοποίηση του χρήστη.

```
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -11 | AcY = 2 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are ON!

The Red LED and the Buzzer for the ALARM are ON!

The Red LED and the Buzzer for the ALARM are ON!

AcX = -11 | AcY = 1 | AcZ = 96 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -11 | AcY = 2 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°
```

Εικόνα 3.122 Καταστάσεις ενεργοποίησης & απενεργοποίησης του Συστήματος Συναγερμού (ALARM) του Ασύρματου Χειριστηρίου

Το Σύστημα Ασφαλείας Χειρισμού που διαθέτει το Ασύρματο Χειριστήριο αποτρέπει την χρήση των αναλογικών μοχλών όταν το ίδιο το χειριστήριο περιστρέφεται εκτός καθορισμένων ορίων στον τρισδιάστατο χώρο.

Βάσει κατάλληλου προγραμματισμού του Κυκλώματος TS και των δύο LEDs πράσινου και αχνοκόκκινου (*Faint-Red*) χρώματος αντίστοιχα, όταν το χειριστήριο διαγράφει συγκεκριμένες κινήσεις στον χώρο που θεωρούνται εκτός των καθορισμένων ορίων (τιμές επιταχυνσιομέτρου $>70^\circ$ και γυροσκοπίου $>80^\circ$ προς μία ή και περισσότερες κατευθύνσεις κατά τους άξονες X, Y και Z), αυτόματα οι 2 αναλογικοί μοχλοί του χειριστηρίου παύουν να λειτουργούν και το πράσινο LED που υποδεικνύει την ορθή λειτουργία του χειριστηρίου σβήνει και ανάβει το αχνοκόκκινο LED στην θέση του. Στην κατάσταση αυτή οι μοχλοί θεωρούνται απενεργοποιημένοι και η χρήση τους δεν επιφέρει κανέναν επηρεασμό στην κατάσταση των σερβοκινητήρων.

Για επαναφορά των 2 μοχλών στην ενεργή τους κατάσταση αρκεί ένα μόνο παρατεταμένο πάτημα του ενός εκ των 2 για μερικά δευτερόλεπτα, με το αυτόματο σβήσιμο του αχνοκόκκινου και άναμμα του πράσινου LED να ακολουθεί αυτής της ενέργειας.

```
Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = 35 | AcY = 4 | AcZ = 87 | Tmp = 48 C or 119 F | GyX = -2 | GyY = -46 | GyZ = 3

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = 84 | AcY = -23 | AcZ = 10 | Tmp = 48 C or 120 F | GyX = 2 | GyY = 1 | GyZ = 0

AcX parameter off limits... joystick temporarily disabled !
AcX parameter off limits... joystick temporarily disabled !
AcX parameter off limits... joystick temporarily disabled !
```

Εικόνα 3.123 Ενεργοποίηση του Συστήματος Ασφαλείας Χειρισμού & απενεργοποίηση των μοχλών του ασύρματου χειριστηρίου με ολίσησή του εκτός των καθορισμένων ορίων του (κόκκινο πλαίσιο) στον τρισδιάστατο χώρο

```

The Red LED and the Buzzer for the ALARM are OFF!

AcX = 84 | AcY = -23 | AcZ = 10 | Tmp = 48 C or 120 F | GyX = 2 | GyY = 1 | GyZ = 0

AcX parameter off limits... joystick temporarily disabled !
AcX parameter off limits... joystick temporarily disabled !
AcX parameter off limits... joystick temporarily disabled !
AcX parameter off limits... joystick temporarily disabled !

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 92°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -6 | AcY = -67 | AcZ = 65 | Tmp = 48 C or 118 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

The Red LED and the Buzzer for the ALARM are OFF!

AcX = -5 | AcY = -70 | AcZ = 62 | Tmp = 48 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 93°
Joystick value for Y-axis: 90°
Joystick No2 value for X_2-axis: 91°
Joystick No2 value for Y_2-axis: 89°

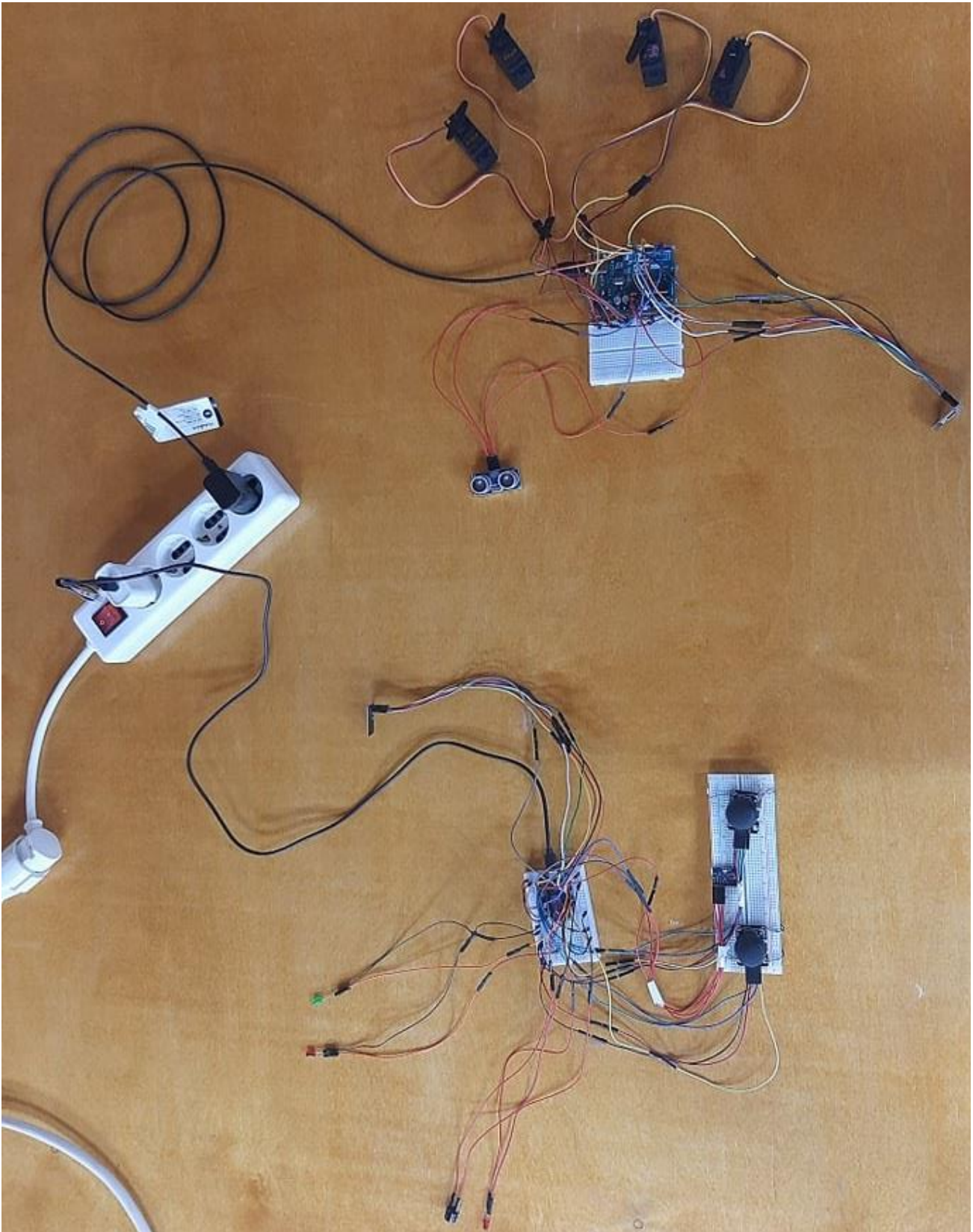
The Red LED and the Buzzer for the ALARM are OFF!

AcX = -5 | AcY = -71 | AcZ = 60 | Tmp = 48 C or 118 F | GyX = -2 | GyY = 0 | GyZ = 0

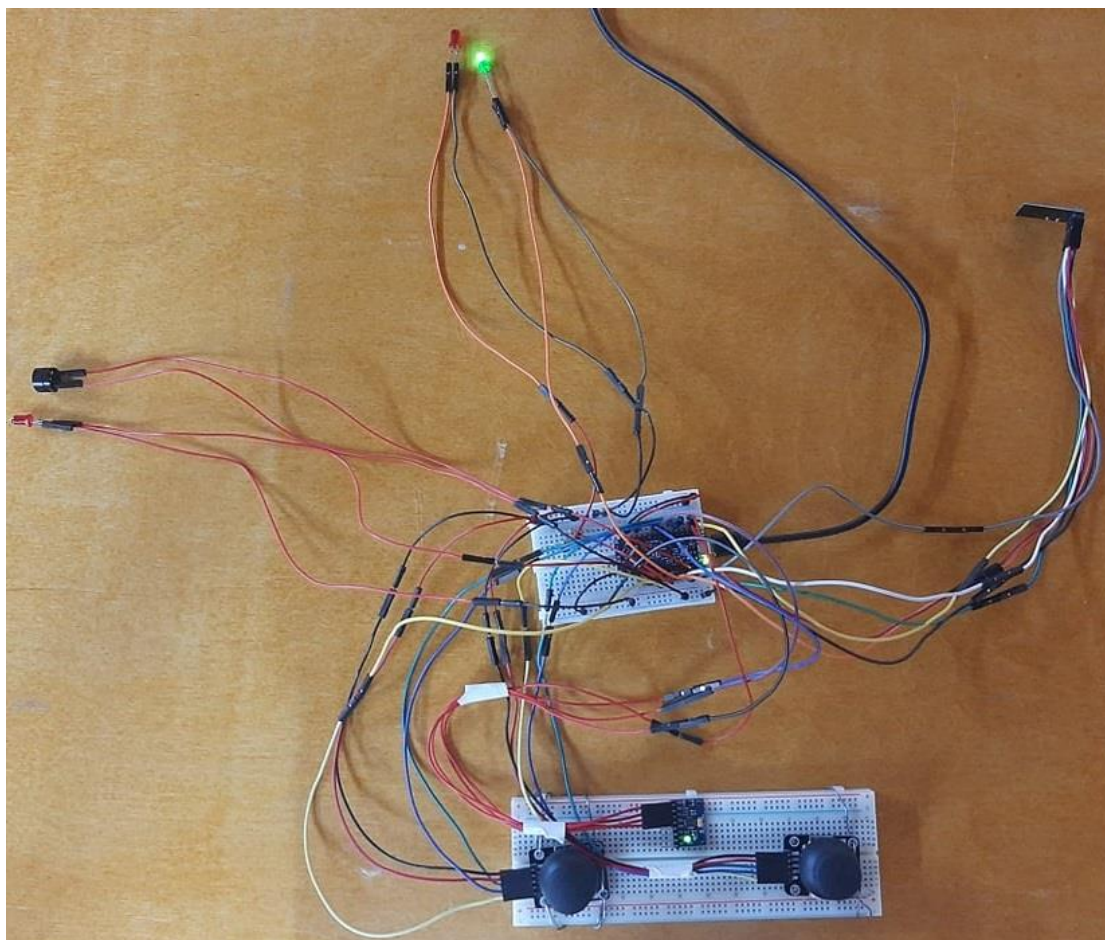
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !
AcY parameter off limits... joystick temporarily disabled !

```

***Εικόνα 3.124** Ενεργοποίηση του Συστήματος Ασφαλείας Χειρισμού & απενεργοποίηση των μοχλών του ασύρματου χειριστηρίου με ολίσησή του εκτός των καθορισμένων ορίων του (κόκκινο πλαίσιο) στον τρισδιάστατο χώρο*



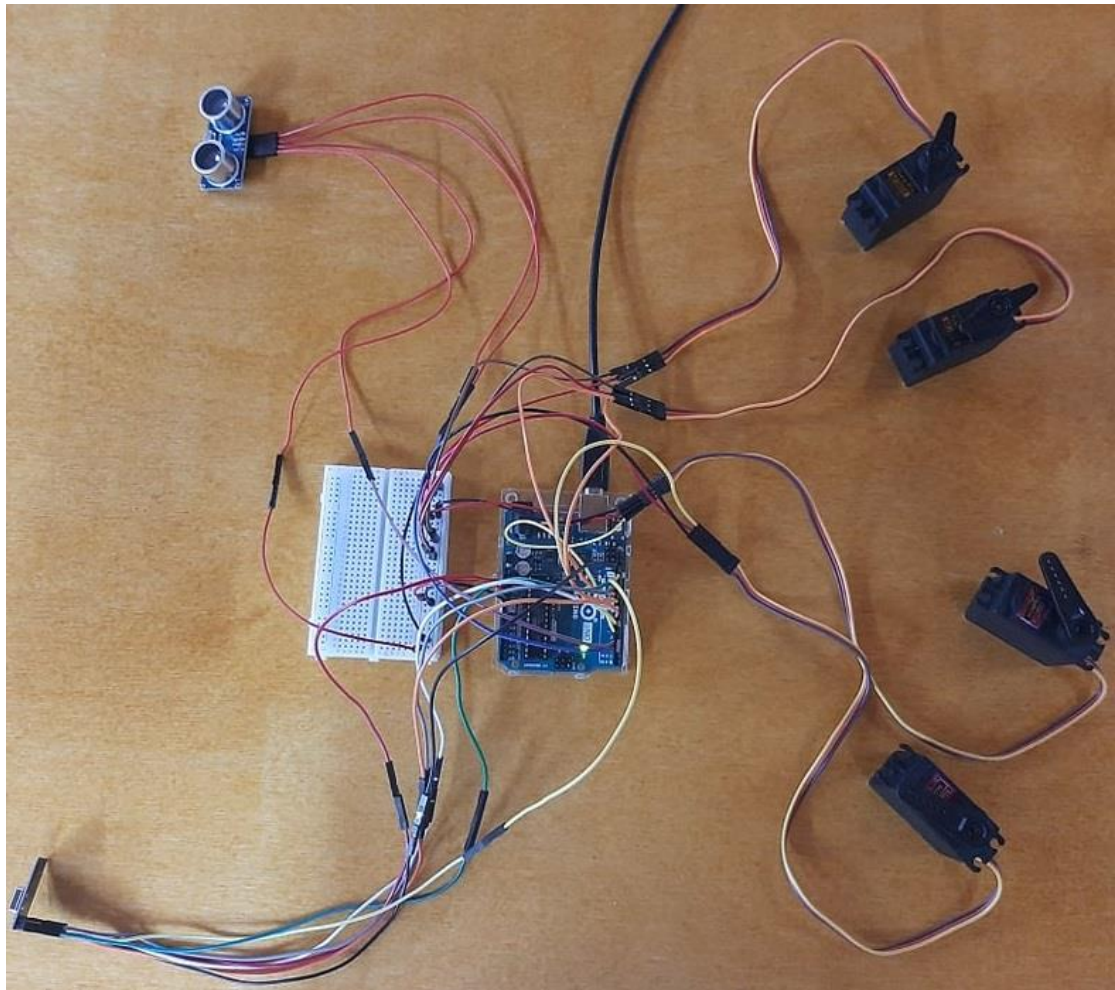
Εικόνα 3.125 Σύστημα ηλεκτρονικού ασύρματου ελεγκτή (Σ.Η.Α.Ε.) με πρόσθετες περιφερειακές μονάδες, υλοποιημένο σε επαγγελματικό εργαστήριο



*Εικόνα 3.126 Υποσύστημα Πομπού (Κύκλωμα TS)
του συστήματος ηλεκτρονικού ασύρματου ελεγκτή*

ΕΠΙΣΗΜΑΝΣΗ – ΠΑΡΑΤΗΡΗΣΗ:

Σε αντίθεση με το Κύκλωμα TS που υλοποιήθηκε σε περιβάλλον προσομοίωσης, σε πραγματικές συνθήκες υλοποίησης το κόκκινο LED δίπλα από το πράσινο συνδέθηκε σε σειρά με 2 αντιστάσεις των 10kΩ εκάστη ώστε να μειωθεί η ένταση της φωτεινότητάς του και να αποκτηθεί ένα φαινομενικά αχνοκόκκινο χρώμα για τον ευδιάκριτο οπτικό του διαχωρισμό από το κόκκινο LED που βρίσκεται δίπλα στο Buzzer και χρησιμοποιείται για το Σύστημα Συναγερμού (ALARM).



*Εικόνα 3.127 Υποσύστημα Δέκτη (Κύκλωμα RS)
του συστήματος ηλεκτρονικού ασύρματου ελεγκτή*

3.5 Ενδεικτική Ολοκληρωμένη Κατασκευή ενός Σ.Η.Α.Ε. για Βιομηχανικές Εφαρμογές

Στην παρούσα ενότητα πραγματοποιείται μια ενδεικτική ολοκληρωμένη κατασκευή ενός Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές.

Στην εν λόγω κατασκευή διατηρούνται όλες οι περιφερειακές μονάδες που συμπεριλήφθηκαν στην προηγούμενη ενότητα, πλην 2 εκ των 4 σερβοκινητήρων που αφαιρούνται εντελώς, αντικατάσταση του κόκκινου LED και των 2 αντιστάσεων συνδεδεμένων σε σειρά με ένα μόνο LED κίτρινου χρώματος για πιο ευδιάκριτα οπτικά αποτελέσματα, αντικατάσταση των μικροελεγκτών UNO R3 και NANO EVERY με δύο Arduino NANO V3 καθαρά από επιλογή και για πειραματική εναλλαγή μεταξύ των πλακετών Arduino, προσθήκη δύο μονάδων βάσης NRF24L01 (*NRF24L01 base modules*) για παροχή σταθερότερης τροφοδοσίας στους πομποδέκτες και δοκιμή εναλλαγής των πομποδεκτών NRF24L01 ενσωματωμένης κεραίας με μια παραλλαγή τους.

Για τις ηλεκτρονικές πλακέτες Arduino NANO V3 αξίζει να σημειωθεί ότι διαθέτουν την ίδια κατασκευή και άρα ίδιο αριθμό ακροδεκτών και λειτουργία αυτών, εκτός του μικροελεγκτή τους που είναι της κατηγορίας ATmega328P που διαθέτει και το Arduino UNO R3.

Επίσης, έχουν γίνει μερικές τροποποιήσεις στον κώδικα προγραμματισμού ώστε να διασφαλίζεται η λειτουργικότητα του συστήματος ως ασύρματο σύστημα απόκρισης I/O.

Στην εικόνα της επόμενης σελίδας απεικονίζεται το σχηματικό διάγραμμα της ολοκληρωμένης κατασκευής Σ.Η.Α.Ε., σχεδιασμένο σε περιβάλλον προσομοίωσης.

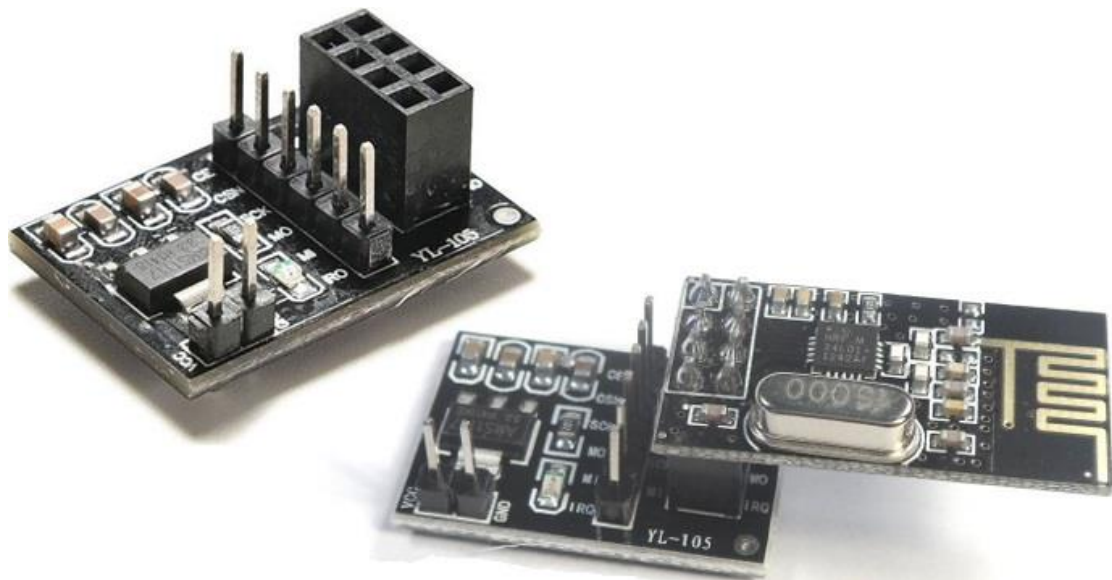
Ένταξη δύο μονάδων βάσης NRF24L01 στο Σ.Η.Α.Ε.

Στο Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή γίνεται ένταξη δύο μονάδων βάσης για τους πομποδέκτες NRF24L01.

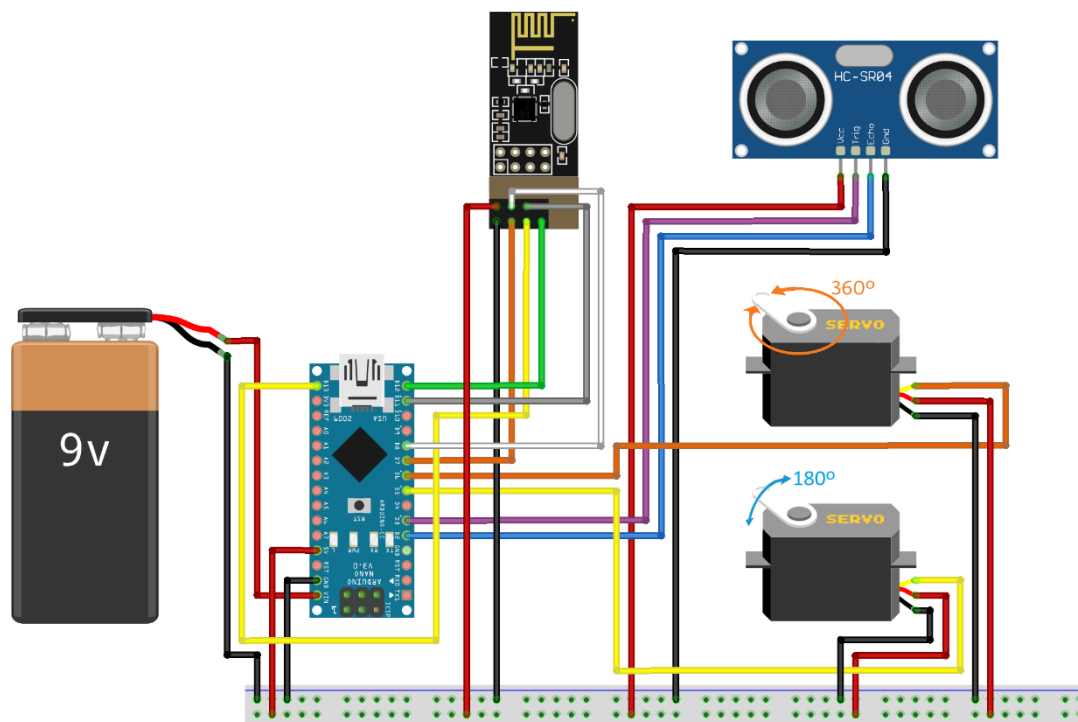
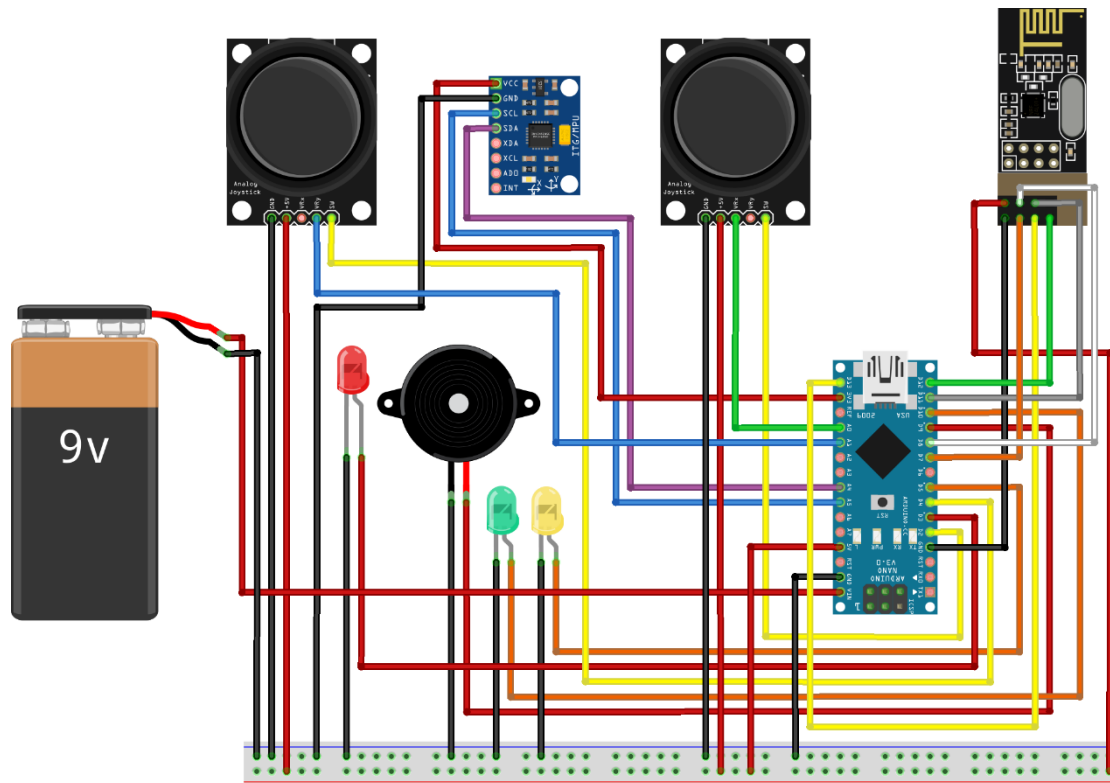
Αυτό γίνεται για την βέλτιστη λειτουργία του συστήματος καθώς η κάθε βάση τροφοδοτείται με 5V και άρα η παροχή ενέργειας των ίδιων των πομποδεκτών από αυτές διατηρούνται πάντα σταθερές στα 3,3V χωρίς να υπάρχουν διακυμάνσεις λόγω προσθήκης επιπλέον περιφερειακών στο σύστημα που στις περισσότερες περιπτώσεις καταλήγει σε υπολειτουργία των NRF λόγω μη επαρκούς τροφοδοσίας από απευθείας διασύνδεση με τον ακροδέκτη 3,3V που διαθέτουν τα Arduino.

Επίσης, για την ενσωμάτωση των πομποδεκτών στις ηλεκτρονικές πλακέτες για την επίτευξη της ολοκληρωμένης κατασκευής απαιτείται μόνιμη τοποθέτηση αυτών επί των πλακετών με μεθόδους συγκόλλησης, πράξη που σε περίπτωση καταστροφής ενός ή και των δύο πομποδεκτών προκαλείται μεγάλο πρόβλημα από πλευράς αντικατάστασής των.

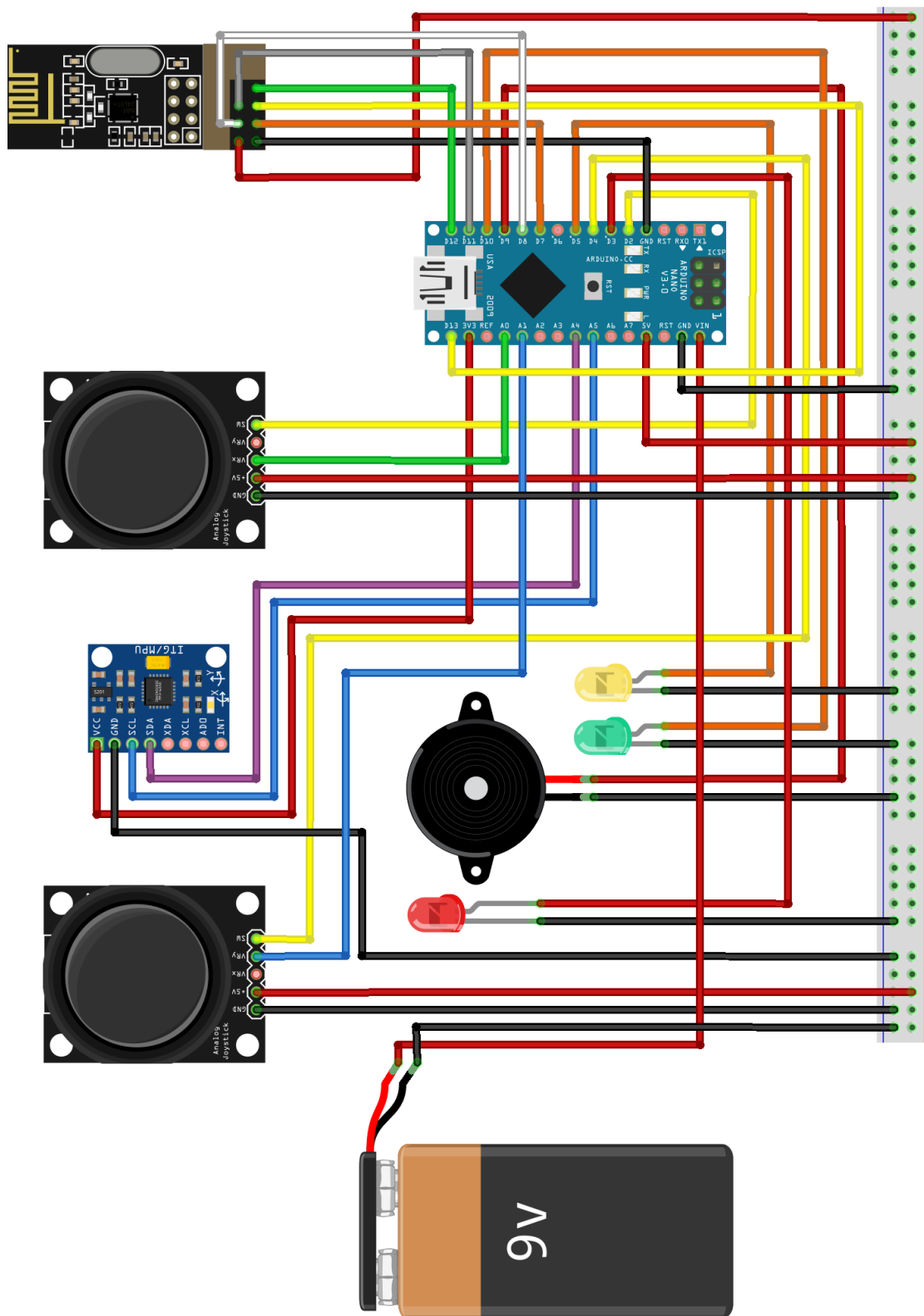
Με την ένταξη των μονάδων βάσης όχι μόνο αποτρέπεται το πρόβλημα αυτό αλλά τυχόν επιθυμητή διαδικασία αλλαγής των πομποδεκτών με νεότερα μοντέλα της οικογένειας NRF24L01 καθίσταται απλοϊκή, εύκολη και άμεση.



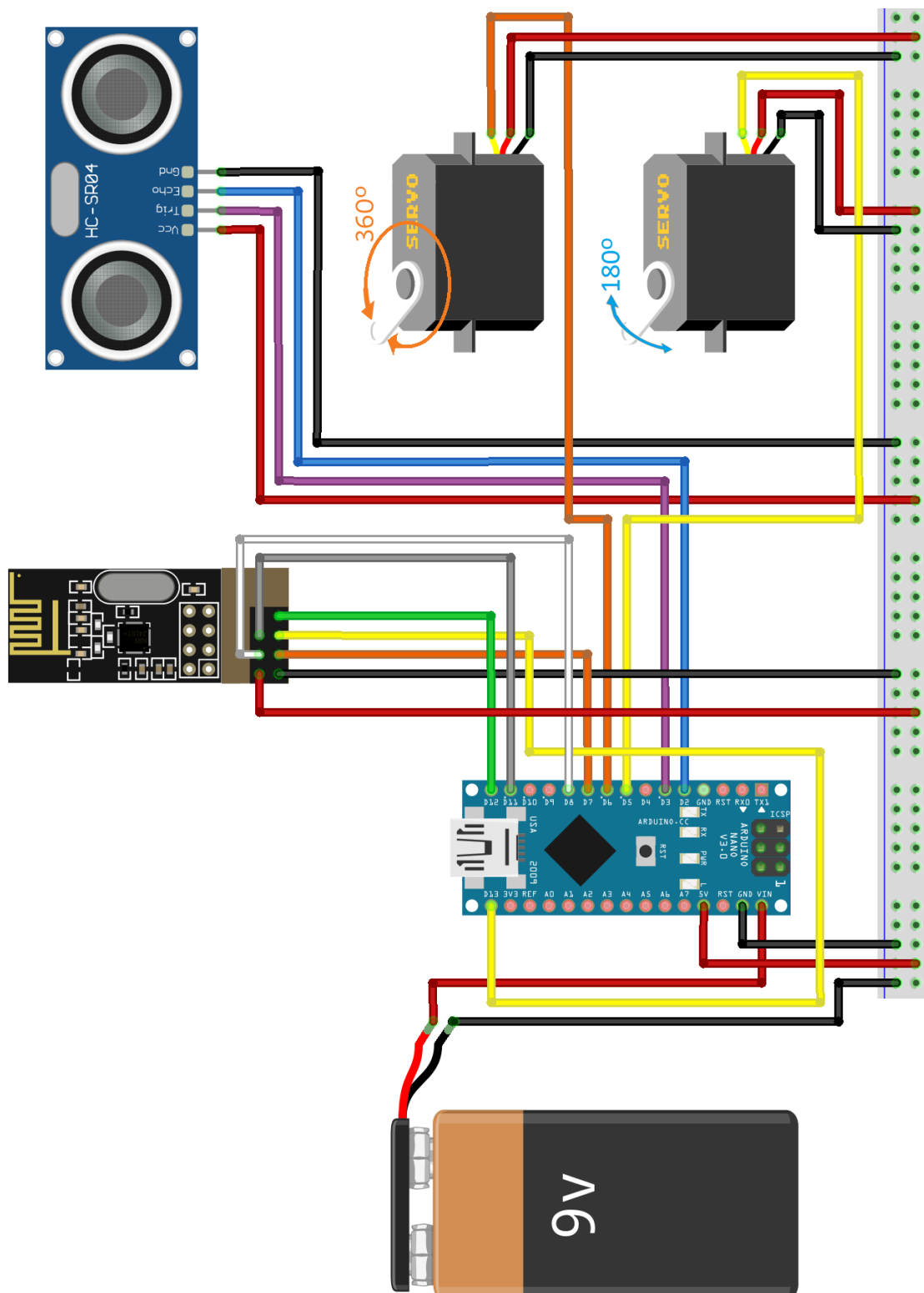
Εικόνα 3.128 Μονάδα βάσης για τον πομποδέκτη NRF24L01 (άνω αριστερά) και μονάδα βάσης με συνδεδεμένο τον πομποδέκτη (κάτω δεξιά)



Εικόνα 3.129 Σχηματικό διάγραμμα της Ολοκληρωμένης Κατασκευής Σ.Η.Α.Ε.



Εικόνα 3.130 Σχηματικό διάγραμμα του Ασύρματου Χειριστηρίου (Κύκλωμα TS) της Ολοκληρωμένης Κατασκευής Σ.Η.Α.Ε



Εικόνα 3.131 Σχηματικό διάγραμμα του Δέκτη (Κύκλωμα RS) της Ολοκληρωμένης Κατασκευής Σ.Η.Α.Ε

Η βασική λειτουργία που διαθέτει η **Ολοκληρωμένη Κατασκευή Σ.Η.Α.Ε.** είναι η **ασύρματη αποστολή και λήψη πληροφοριών από και προς τα δύο του Υποσυστήματα.**

Αρχικά το Υποσύστημα Πομπού, *Κύκλωμα TS* ή αλλιώς *Ασύρματο Χειριστήριο*, αποστέλλει την μηχανική κίνηση των 2 αναλογικών μοχλών του –εκφρασμένη σε μοίρες– στο Υποσύστημα Δέκτη, *Κύκλωμα RS* ή απλώς *Δέκτη* και ως αποτέλεσμα οι 2 ψηφιακοί σερβοκινητήρες περιστρέφονται κατά τις μοίρες αυτές.

Ως δευτερεύουσα λειτουργία, το Σ.Η.Α.Ε. έχει σχεδιαστεί ώστε να λειτουργεί ως **Σύστημα Συναγερμού**, καθώς η περιφερειακή μονάδα αισθητήρα υπερήχων που είναι ενταγμένη στο Υποσύστημα Δέκτη μπορεί να ειδοποιήσει το Υποσύστημα Πομπού για εμφάνιση κίνησης σε έναν χώρο με κατάλληλη τοποθέτηση του αισθητήρα σε κάποιο σημείο του χώρου (π.χ. είσοδος/έξοδος ενός εργασιακού χώρου ή χώρου στάθμευσης οχημάτων).

Το Σ.Η.Α.Ε. περιλαμβάνει επίσης ένα **Σύστημα Ασφαλείας Χειρισμού** για το ασύρματο χειριστήριο, καθώς ενέργειες από τον χρήστη που περιλαμβάνουν αργή ή γρήγορη περιστροφική κίνηση μεγάλης κλίσης, έντονη ταλάντευση, απότομη περιστροφή ή πτώση του ασύρματου χειριστηρίου προκαλούν την ενεργοποίηση του κατάλληλα προγραμματισμένου κυκλώματός του με αποτέλεσμα οι 2 αναλογικοί μοχλοί του χειριστηρίου να βγαίνουν εκτός λειτουργίας και άρα το σύστημα να προφυλάσσεται από ενδεχόμενη μη επιθυμητή παροχή εντολών εισόδου.

Επιπρόσθετα σε αυτό, λόγω του τροποποιημένου κώδικα του Πομπού, στην περίπτωση που ο χρήστης μεταχειρίζεται τους μοχλούς και το Σύστημα Ασφαλείας Χειρισμού ενεργοποιηθεί, τότε μαζί με τους μοχλούς απενεργοποιούνται αυτόματα και οι σερβοκινητήρες σταματώντας οποιαδήποτε περιστροφική τους κίνηση, αποφεύγοντας έτσι τυχόν ανεπιθύμητη απόκριση του συστήματος.

Με την επιπλέον δυνατότητα ρύθμισης των 2 αναλογικών μοχλών ως διακόπτες μπουτόν, με παρατεταμένο πάτημα ενός εκ των 2 για μερικά δευτερόλεπτα το Σύστημα Ασφαλείας Χειρισμού απενεργοποιείται και επαναφέρεται επιτυχώς η κατάσταση ενεργής λειτουργίας των αναλογικών μοχλών για επανέλεγχο των σερβοκινητήρων.

Προγραμματιστικός Κώδικας Ασύρματου Χειριστηρίου (Πομπό)

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Wire.h>
#include <GY6050.h>

//create an RF24 object
RF24 radio(7, 8); // CE, CSN

//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};

// include an analog joystick of 2-axis
const int joystick_VRX = A0; // connect the X-axis potentiometer to Arduino pin A0
const int joystick_VRY_2 = A1; // connect the Y-axis potentiometer to Arduino pin A1

//include a Buzzer (piezo-speaker)
const int BUZZER = 9; // Connect Buzzer pin "Vcc" to Arduino D9
int Sensor_State = 0; //Initialize variable for the Buzzer & the Red LED

//include MPU6050 sensor unit (accelerometer & gyroscope)
GY6050 gyro(0x68); // Create Gyro object with "0x68" as I2C address (MPU6050 address)

// Initialize 2 variables for the 2 joystick switches/push buttons, respectively
int SW_State, SW_2_State = 0;

void setup() {
  Serial.begin(9600);
  Serial.print("Wireless Controller activating...\n*****\n");
  delay(200);

  pinMode(2, INPUT_PULLUP); // Set pin connections as 'joystick_SW = 2'
  pinMode(4, INPUT_PULLUP); // Set pin connections as 'joystick_2 SW_2 = 4'
  pinMode(BUZZER, OUTPUT); // Set the Buzzer as output

  //include a LED of 'red' colour
  pinMode(3, OUTPUT); // Set a Red LED as output

  //Include a LED of 'yellow' colour
  pinMode(5, OUTPUT); // Set a Yellow LED as output

  //Include a LED of 'green' colour
  pinMode(10, OUTPUT); // Set a Green LED as output

  radio.begin();
  //set the address
  radio.openWritingPipe(addresses[1]); // 00002
  radio.openReadingPipe(1, addresses[0]); // 00001

  gyro.initialisation();
  Serial.println("MPU6050 Sensor Unit on the Controller is activated");
  Serial.print("----- The Controller will be temporarily disabled in case some ");
  Serial.println("of the spatial parameters exceed certain limits -----\n");
  delay(2500);
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

void loop() {
  // Set the Green LED ON
  digitalWrite(10, HIGH);    //Green LED ON --> Wireless Controller works properly

  //Set module as transmitter
  radio.stopListening();

  // calculation process of joystick & No2 joystick values
  int X_ADC_Value = analogRead(joystick_VRX);    // Read the analog/mechanical input of X-axis
  int Y_2_ADC_Value = analogRead(joystick_VRY_2); // Read the analog/mechanical input of Y-axis
  delay(100);

  // variables to store joystick values
  int angleV1 = map(X_ADC_Value, 1, 1023, 1, 180); // Set the analog/mechanical input signal
                                                    // of X-axis as values of range 1-180°
  int angleV2 = map(Y_2_ADC_Value, 2, 1023, 2, 180); // Set the analog/mechanical input signal
                                                    // of Y-axis as values of range 2-180°
  delay(100);    // waits 100ms for the servos to reach the corresponding angle

  // Send joystick information (status & values) to receiver
  Serial.print("Joystick value for X-axis: ");
  Serial.print(angleV1);
  radio.write(&angleV1, sizeof(angleV1));
  Serial.print("\n");
  delay(25);
  Serial.print("Joystick value for Y-axis: ");
  Serial.print(angleV2);
  radio.write(&angleV2, sizeof(angleV2));
  Serial.println("\n");
  delay(25);

  //Set module as receiver
  radio.startListening();

  radio.read(&Sensor_State, sizeof(Sensor_State));
  if (Sensor_State == 1) {
    do {
      Serial.print("|----Red LED & Buzzer are ON --> ALARM SYSTEM ACTIVATED!");
      digitalWrite(3, HIGH);
      Serial.println("\n");
      tone(BUZZER, 1000);    // Send 1KHz sound signal
      delay(2000);
      radio.read(&Sensor_State, sizeof(Sensor_State));
    } while (Sensor_State == 1);
  }
  else if (Sensor_State == 0) {
    Serial.print("----Red LED & Buzzer are OFF!");
    digitalWrite(3, LOW);
    Serial.println("\n");
    noTone(BUZZER);    // Stop the sound signal
  }
  delay(500);

  // MPU6050 sensor unit parameters measurement
  Serial.print("/---Spatial Parameters of MPU6050---\\n");
  Serial.print("AcX = ");
  int AcX = gyro.refresh('A', 'X');
  Serial.print(AcX);    // Ask for the X axis of the Accelerometer and print it

  Serial.print(" | AcY = ");
  int AcY = gyro.refresh('A', 'Y');
  Serial.print(AcY);    // Ask for the Y axis of the Accelerometer and print it

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

Serial.print(" | AcZ = ");
int AcZ = gyro.refresh('A', 'Z');
Serial.print(AcZ); // Ask for the Z axis of the Accelerometer and print it

Serial.print(" | Temp = ");
int Temp_C = gyro.refresh('T', 'C');
Serial.print(Temp_C); // Ask for the Temperature in Celsius and print it
Serial.print(" C or ");
int Temp_F = gyro.refresh('T', 'F');
Serial.print(Temp_F); // Ask for the Temperature in Farenheit and print it
Serial.print(" F");

Serial.print(" | GyX = ");
int GyX = gyro.refresh('G', 'X');
Serial.print(GyX); // Ask for the X axis of the Gyroscope and print it

Serial.print(" | GyY = ");
int GyY = gyro.refresh('G', 'Y');
Serial.print(GyY); // Ask for the Y axis of the Gyroscope and print it

Serial.print(" | GyZ = ");
int GyZ = gyro.refresh('G', 'Z');
Serial.println(GyZ); // Ask for the Z axis of the Gyroscope and print it
delay(1000);

if (AcX > 70 || AcX < -70) {
  Serial.print("\n");
  delay(100);
  do {
    Serial.print("AcX parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW); // Turn the Green LED OFF
    digitalWrite(5, HIGH); // Turn the Faint-Red LED ON
    //Set module as transmitter
    radio.stopListening();
    // Send joystick deactivated status to receiver
    Serial.print("Joystick value for X-axis: ");
    Serial.print(angleV1);
    radio.write(&angleV1, sizeof(angleV1));
    Serial.print("\n");
    delay(25);
    int angleV2_deact = 90;
    Serial.print("Joystick deactivated value for Y-axis inertia: ");
    Serial.print(angleV2_deact);
    radio.write(&angleV2_deact, sizeof(angleV2_deact));
    Serial.println("\n");
    delay(500);
    SW_State = !digitalRead(2); // Read the digital input of the Switch
    SW_2_State = !digitalRead(4); // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      digitalWrite(5, LOW); // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH); // Turn the Green LED ON
      break;
    }
  } while (1);
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

//Set module as receiver
radio.startListening();
}
if (AcY > 70 || AcY < -70) {
  Serial.print("\n");
  do {
    Serial.print("AcY parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);          // Turn the Green LED OFF
    digitalWrite(5, HIGH);         // Turn the Faint-Red LED ON
    //Set module as transmitter
    radio.stopListening();
    // Send joystick deactivated status to receiver
    Serial.print("Joystick value for X-axis: ");
    Serial.print(angleV1);
    radio.write(&angleV1, sizeof(angleV1));
    Serial.print("\n");
    delay(25);
    int angleV2_deact = 90;
    Serial.print("Joystick deactivated value for Y-axis inertia: ");
    Serial.print(angleV2_deact);
    radio.write(&angleV2_deact, sizeof(angleV2_deact));
    Serial.println("\n");
    delay(500);
    SW_State = !digitalRead(2);     // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);   // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      //delay(25);
      digitalWrite(5, LOW);        // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);     // Turn the Green LED ON
      break;
    }
  } while (1);
//Set module as receiver
radio.startListening();
}
if (GyX > 80 || GyX < -80) {
  Serial.print("\n");
  do {
    Serial.print("GyX parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);          // Turn the Green LED OFF
    digitalWrite(5, HIGH);         // Turn the Faint-Red LED ON
    //Set module as transmitter
    radio.stopListening();
    // Send joystick deactivated status to receiver
    Serial.print("Joystick value for X-axis: ");
    Serial.print(angleV1);
    radio.write(&angleV1, sizeof(angleV1));
    Serial.print("\n");
    delay(25);
    int angleV2_deact = 90;
    Serial.print("Joystick deactivated value for Y-axis inertia: ");
    Serial.print(angleV2_deact);
    radio.write(&angleV2_deact, sizeof(angleV2_deact));
    Serial.println("\n");
    delay(500);
    SW_State = !digitalRead(2);     // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);   // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      digitalWrite(5, LOW);        // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);     // Turn the Green LED ON
      break;
    }
  }
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

} while (1);
//Set module as receiver
radio.startListening();
}

if (GyY > 80 || GyY <-80) {
Serial.print("\n");
do {
Serial.print("GyY parameter off limits... joystick temporarily disabled !");
Serial.print("\n");
digitalWrite(10, LOW);          // Turn the Green LED OFF
digitalWrite(5, HIGH);          // Turn the Faint-Red LED ON
//Set module as transmitter
radio.stopListening();
// Send joystick deactivated status to receiver
Serial.print("Joystick value for X-axis: ");
Serial.print(angleV1);
radio.write(&angleV1, sizeof(angleV1));
Serial.print("\n");
delay(25);
int angleV2_deact = 90;
Serial.print("Joystick deactivated value for Y-axis inertia: ");
Serial.print(angleV2_deact);
radio.write(&angleV2_deact, sizeof(angleV2_deact));
Serial.println("\n");
delay(500);
SW_State = !digitalRead(2);      // Read the digital input of the Switch
SW_2_State = !digitalRead(4);    // Read the digital input of the Switch
if (SW_State == 1 || SW_2_State == 1) {
digitalWrite(5, LOW);           // Turn the Faint-Red LED OFF
digitalWrite(10, HIGH);         // Turn the Green LED ON
break;
}
} while (1);
//Set module as receiver
radio.startListening();
}

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

if (GyZ > 80 || GyZ < -80) {
  Serial.print("\n");
  do {
    Serial.print("GyZ parameter off limits... joystick temporarily disabled !");
    Serial.print("\n");
    digitalWrite(10, LOW);          // Turn the Green LED OFF
    digitalWrite(5, HIGH);         // Turn the Faint-Red LED ON
    //Set module as transmitter
    radio.stopListening();
    // Send joystick deactivated status to receiver
    Serial.print("Joystick value for X-axis: ");
    Serial.print(angleV1);
    radio.write(&angleV1, sizeof(angleV1));
    Serial.print("\n");
    delay(25);
    int angleV2_deact = 90;
    Serial.print("Joystick deactivated value for Y-axis inertia: ");
    Serial.print(angleV2_deact);
    radio.write(&angleV2_deact, sizeof(angleV2_deact));
    Serial.println("\n");
    delay(500);
    SW_State = !digitalRead(2);     // Read the digital input of the Switch
    SW_2_State = !digitalRead(4);   // Read the digital input of the Switch
    if (SW_State == 1 || SW_2_State == 1) {
      digitalWrite(5, LOW);        // Turn the Faint-Red LED OFF
      digitalWrite(10, HIGH);      // Turn the Green LED ON
      break;
    }
  } while (1);
  //Set module as receiver
  radio.startListening();
}
Serial.print("\n\n");

```

***ΚΩΔΙΚΑΣ_28** Κώδικας λειτουργίας του Ασύρματου Χειριστηρίου της ολοκληρωμένης κατασκευής Σ.Η.Α.Ε.*

Προγραμματιστικός Κώδικας Υποσυστήματος Δέκτη (Δέκτη)

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>    // Add the servo library

// Create an RF24 object
RF24 radio(7, 8); // CE, CSN
//address through which two modules communicate.
const byte addresses[][6] = {"00001", "00002"};

// include 2 servos of 180 degree rotation
Servo myServo1;    // Define 'myServo1' as Servo1 servo motor
Servo myServo2;    // Define 'myServo2' as Servo2 servo motor

// variables to store joystick values sent by the transmitter (TS Circuit)
int angleV_1 = 0;    // Initiallize variable to store Joystick values
                    // of X-axis for Servol (position or angle = 0)
int angleV_2 = 0; // Initiallize variable to store Joystick No2 values
                    // of X_2-axis for Servo3 rotational speed

// Include 1 HC-RS04 ultrasoni sensor
const int Trig_pin = 3;
const int Echo_pin = 2;
// define variables
float duration, distance;
int Sensor_State = 0;

void setup() {
  Serial.begin(9600);

  pinMode(Trig_pin, OUTPUT);    // Set the Trig pin as an Output
  pinMode(Echo_pin, INPUT);    // Set the Echo pin as an Input

  radio.begin();
  //set the address
  //radio.openReadingPipe(0, address);
  radio.openWritingPipe(addresses[0]); // 00001
  radio.openReadingPipe(1, addresses[1]); // 00002

  myServo1.attach(5);    // attaches Servol Signal Line to pin 5 (PWM)
  myServo2.attach(6);    // attaches Servo2 Signal Line to pin 6 (PWM)
}
```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

```

void loop() {
  //Set module as receiver
  radio.startListening();

  //Read the data if available in buffer
  if (radio.available()) {
    while (radio.available()) {

      // Receive joystick information to rotate the 2 Servo motors accordingly
      radio.read(&angleV_1, sizeof(angleV_1)); // Read & store X-axis information of Joystick
      Serial.print("For X-axis value: ");
      Serial.print(angleV_1);
      Serial.print("");
      if (angleV_1 == 1 || angleV_1 >= 178) {
        Serial.print(" ---->Servo1 position/angle is ");
        Serial.print(angleV_1);
        Serial.println("");
        myServo1.write(angleV_1); // Use the variable value to rotate Servo1
      }
      else {
        // print the standby position in degrees
        Serial.println(" ---->Servo1 on standby mode");
      }
      delay(100);

      radio.read(&angleV_2, sizeof(angleV_2)); // Read & store Y_2-axis information of Joy No2
      Serial.print("For Y-axis value: ");
      Serial.print(angleV_2);
      Serial.print("");
      if (angleV_2 == 2 || angleV_2 >= 178) {
        Serial.print(" ---->Servo2 position/angle is ");
        Serial.print(angleV_2);
        Serial.println("");
        myServo2.write(angleV_2); // Use the variable value to rotate Servo2
      }
      else {
        Serial.println(" -->Servo2 in inertia");
        myServo2.write(90); // Use the variable value to stagnate Servo2
      }
      delay(100);
      Serial.print("\n");
      delay(25);

      // Clear the Trig pin condition
      digitalWrite(Trig_pin, LOW);
      delayMicroseconds(2);

      // Set the Trig pin HIGH (ACTIVE) for 10µs
      digitalWrite(Trig_pin, HIGH); // Generate the sound wave
      delayMicroseconds(10);
      digitalWrite(Trig_pin, LOW); // Clear the Trig pin condition

      // Read the Echo pin and return the sound wave travel time in µs
      duration = pulseIn(Echo_pin, HIGH);
      // The reflected wave turns the Echo pin ON (HIGH) and the timing starts.
      // When the wave ends the Echo pin goes OFF (LOW) and the timing stops,
      // returning the length of the reflected sound wave in µs.

      // Calculate the distance
      distance = (duration * 0.034) / 2;
      // Speed of sound wave divided by 2 (transmission & reception time travel)

```

(ΣΥΝΕΧΕΙΑ ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ)

Με χρήση του Serial Monitor για το Ασύρματο Χειριστήριο (Κύκλωμα TS), παρουσιάζονται στην εικόνα 3.132 τα εξής χαρακτηριστικά μηνύματα που υποδεικνύουν την ενεργοποίηση του χειριστηρίου:

Wireless Controller activating . . .

MPU6050 Sensor Unit on the Controller is activated

----- The Controller will be temporarily disabled in case some of the spatial parameters exceed certain limits -----

και αμέσως μετά ακολουθούν οι ενδείξεις των τιμών των 4 ποτενσιομέτρων κατά την ενεργοποίησή του, σε μοίρες:

Joystick value for X-axis: 92°

Joystick value for Y-axis: 90°

Όταν το χειριστήριο βρίσκεται σε αδράνεια, οι εν λόγω τιμές διατηρούνται όπως παρουσιάζονται στην εν λόγω εικόνα και αποστέλλονται διαρκώς μέσω των πομποδεκτών NRF24L01 από το Χειριστήριο στον Δέκτη.

```
Wireless Controller activating...
*****
MPU6050 Sensor Unit on the Controller is activated
----- The Controller will be temporarily disabled in case some of the spatial parameters exceed certain limits -----

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 0 | AcZ = 96 | Temp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0
```

Εικόνα 3.132 Εμφάνιση των τιμών των παραμέτρων του Ασύρματου Χειριστηρίου (Κύκλωμα TS) σε αδράνεια

Η εικόνα 3.133 παρουσιάζει την συνεχή λήψη τιμών αδρανείας που λαμβάνει ο Δέκτης από το αδρανές Χειριστήριο μέσω του πομποδέκτη NRF, όπως αυτές εμφανίζονται με χρήση του Serial Monitor:

```
For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 313.99cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 313.48cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 313.97cm (Threshold is 20cm)
```

Εικόνα 3.133 Εμφάνιση των τιμών των παραμέτρων του Υποσυστήματος Δέκτη (Κύκλωμα RS), με το χειριστήριο σε αδράνεια

Στην εικόνα 3.134 γίνεται χρήση του Serial Monitor για το Χειριστήριο, με εμφανή την μεταβολή των τιμών των ποτενσιομέτρων για διάφορες περιπτώσεις ολίσθησης των μοχλών κατά τους X και Y άξονες, με το Κύκλωμα TS να στέλνει αδιάκοπα στο Κύκλωμα RS την κατάσταση των αναλογικών μοχλών, σε μοίρες.

Για την εικόνα 3.134, εφόσον οι ενδείξεις “Red LED” και “Buzzer” είναι “OFF” σημαίνει ότι το Σύστημα Συναγερμού (ALARM SYSTEM) είναι ανενεργό, δηλαδή βάσει του κώδικα No29 ο αισθητήρας υπερήχων του Δέκτη δεν ενεργοποιείται κι έτσι δεν αποστέλλει σήμα για ενεργοποίηση του Συστήματος Συναγερμού (ΚΩΔΙΚΑΣ_28) που διαθέτει το Ασύρματο Χειριστήριο.

```

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°
----Red LED & Buzzer are OFF!
/---Spatial Parameters of MPU6050---\
AcX = -3 | AcY = -2 | AcZ = 100 | Tmp = 49 C or 119 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 180°
Joystick value for Y-axis: 2°
----Red LED & Buzzer are OFF!
/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 3 | AcZ = 103 | Tmp = 48 C or 120 F | GyX = 2 | GyY = -1 | GyZ = 2

Joystick value for X-axis: 1°
Joystick value for Y-axis: 180°
----Red LED & Buzzer are OFF!
/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 0 | AcZ = 96 | Tmp = 49 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°
----Red LED & Buzzer are OFF!
/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 0 | AcZ = 96 | Tmp = 48 C or 120 F | GyX = 2 | GyY = 0 | GyZ = 0

```

➤ **ΑΝΑΛΟΓΙΚΟΙ ΜΟΧΛΟΙ 1 & 2 ΣΕ ΑΔΡΑΝΕΙΑ**

⬅ **ΣΥΣΤΗΜΑ ΣΥΝΑΓΕΡΜΟΥ ΑΝΕΝΕΡΓΟ**

➤ **ΔΕΞΙΑ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 1 (X=180)**

➤ **ΑΝΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 2 (Y=2)**

➤ **ΑΡΙΣΤΕΡΗ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 1 (X=1)**

➤ **ΚΑΤΩ ΟΛΙΣΘΗΣΗ ΜΟΧΛΟΥ 2 (Y=180)**

➤ **ΑΝΑΛΟΓΙΚΟΙ ΜΟΧΛΟΙ 1 & 2 ΣΕ ΑΔΡΑΝΕΙΑ**

Εικόνα 3.134 Διάφορες καταστάσεις ολίσθησης και αντίστοιχες τιμές των μοχλών του ασύρματου χειριστηρίου, με το Σύστημα Συναγερμού σε κατάσταση OFF

Στην εικόνα 3.135 γίνεται και πάλι χρήση του Serial Monitor για το Υποσύστημα Δέκτη, με τις εκάστοτε ληφθείσες τιμές να εξαρτώνται από τους αναλογικούς μοχλούς του Χειριστηρίου.

```
For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 243.36cm (Threshold is 20cm)

For X-axis value: 180° ----->Servo1 position/angle is 180°
For Y-axis value: 2° ----->Servo2 position/angle is 2°

\----Distance: 246.08cm (Threshold is 20cm)

For X-axis value: 1° ----->Servo1 position/angle is 1°
For Y-axis value: 2° ----->Servo2 position/angle is 2°

\----Distance: 243.15cm (Threshold is 20cm)

For X-axis value: 1° ----->Servo1 position/angle is 1°
For Y-axis value: 180° ----->Servo2 position/angle is 180°

\----Distance: 243.19cm (Threshold is 20cm)

For X-axis value: 180° ----->Servo1 position/angle is 180°
For Y-axis value: 180° ----->Servo2 position/angle is 180°

\----Distance: 245.09cm (Threshold is 20cm)
```

Εικόνα 3.135 Διάφορες τιμές των 4 σερβοκινητήρων, παρεχόμενες σε μοίρες από το Υποσύστημα Πομπού. Η ένδειξη «Distance» μετρά την απόσταση μεταξύ αισθητήρα υπερήχων και κάποιου αντικειμένου, σε πραγματικό χρόνο

Οι ανωτέρω εικόνες 3.134 & 3.135 δεν έχουν χρονική συνέχεια μεταξύ τους, καθώς αποτελούν 2 διαφορετικά στιγμιότυπα εικόνων που ελήφθησαν σε διαφορετικό χρόνο για το κάθε Υποσύστημα.

```
For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 312.78cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 312.36cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 30.45cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 27.03cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 22.36cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 244.85cm (Threshold is 20cm)
```

Εικόνα 3.136 Τιμές χειριστηρίου σε αδράνεια και αντίστοιχη αδρανοποίηση σερβοκινητήρων, με την ένδειξη “Distance” να φανερώνει την μεταβολή της θέσης ενός αντικειμένου ως προς τον αισθητήρα υπερήχων με κάθε νέα μέτρηση

Όταν το Χειριστήριο είναι αδρανές, ο Δέκτης λαμβάνει συνεχώς τις τιμές που απεικονίζονται στην εικόνα 3.137 με αποτέλεσμα οι σερβοκινητήρες να αδρανοποιούνται. Λόγω της αδράνειας αυτής και του κατάλληλου προγραμματιστικού κώδικα που αναπτύχθηκε για το Σ.Η.Α.Ε., ο σερβοκινητήρας “Servo1” περιστροφής 180° βρίσκεται σε κατάσταση ηρεμίας ή αναμονής (*standby mode*), διατηρώντας την θέση (*κλίση*) στην οποία είχε μετατοπισθεί για τελευταία φορά μεταξύ των δύο θέσεων εναλλαγής 0° ή 180°. Ο σερβοκινητήρας “Servo2” περιστροφής 360° παραμένει σε αδράνεια (*inertia*) λόγω της τιμής που λαμβάνει (*συνεχής τιμή 90° που παρέχει μηδενική γωνιακή ταχύτητα στον MG995*). Επίσης, βάσει του κώδικα προγραμματισμού, η μονάδα αισθητήρα υπερήχων του Υποσυστήματος Δέκτη είναι ρυθμισμένη να στέλνει πληροφορίες στο Ασύρματο Χειριστήριο για περιπτώσεις που κάποιο αντικείμενο πλησιάσει την μονάδα σε απόσταση των 20cm και κάτω.

```

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 20.59cm (Threshold is 20cm)

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 19.75cm (Threshold is 20cm)

Obstacle detected at: 19.75cm!
          |---->Sending Sensor State Report to TS Circuit...

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 23.24cm (Threshold is 20cm)

```

Εικόνα 3.137 Σερβοκινητήρες σε αδράνεια λόγω αδράνειας χειριστηρίου και μια περίπτωση αναγνώρισης αντικειμένου στα 19,75cm

```

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 40.56cm (Threshold is 20cm)

Obstacle detected at: 16.97cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 16.71cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 17.36cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 17.43cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 15.30cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 15.32cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 14.43cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 15.01cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 15.84cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 18.48cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 15.49cm!
      |--->Sending Sensor State Report to TS Circuit...

Obstacle detected at: 17.78cm!
      |--->Sending Sensor State Report to TS Circuit...

For X-axis value: 92° ----->Servo1 on standby mode
For Y-axis value: 90° -->Servo2 in inertia

\----Distance: 31.30cm (Threshold is 20cm)

```

***Εικόνα 3.138** Σερβοκινητήρες σε αδράνεια λόγω αδράνειας χειριστηρίου και περιπτώσεις αποστολής αναφοράς κατάστασης από τον Δέκτη στο Χειριστήριο, με εμφάνιση εντός αντικειμένων για διάφορες αποστάσεις άνω και κάτω των 20cm ως προς τον αισθητήρα.*

Στην εικόνα 3.139 παρουσιάζεται με μηνύματα κόκκινου υπογραμμισμένου χρώματος η κατάσταση του ασύρματου χειριστηρίου όταν οι αναλογικοί του μοχλοί είναι σε αδράνεια και η μονάδα αισθητήρα υπερήχων του Δέκτη έχει ενημερώσει το χειριστήριο για την ύπαρξη αντικειμένου σε απόσταση κάτω των 20cm με αποτέλεσμα την ενεργοποίηση του Συστήματος Συναγερμού του χειριστηρίου, το οποίο αποτελείται από ένα κόκκινο LED που φωτοβολεί και ένα Buzzer που παράγει ηχητικό σήμα 1kHz για την οπτική και ηχητική ειδοποίηση του χρήστη.

```

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

```

```

/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 1 | AcZ = 96 | Tmp = 53 C or 127 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

```

```

/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 1 | AcZ = 96 | Tmp = 53 C or 128 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

|----Red LED & Buzzer are ON --> ALARM SYSTEM ACTIVATED!

```

```

|----Red LED & Buzzer are ON --> ALARM SYSTEM ACTIVATED!

```

```

|----Red LED & Buzzer are ON --> ALARM SYSTEM ACTIVATED!

```

```

/---Spatial Parameters of MPU6050---\
AcX = -3 | AcY = 0 | AcZ = 96 | Tmp = 53 C or 128 F | GyX = 2 | GyY = 0 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

```

```

/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = 0 | AcZ = 96 | Tmp = 53 C or 127 F | GyX = 2 | GyY = 0 | GyZ = 0

```

Εικόνα 3.139 Καταστάσεις ενεργοποίησης & απενεργοποίησης του Συστήματος Συναγερμού (ALARM SYSTEM) του Ασύρματου Χειριστηρίου

Το Σύστημα Ασφαλείας Χειρισμού που διαθέτει το Ασύρματο Χειριστήριο αποτρέπει την χρήση των αναλογικών μοχλών όταν το ίδιο το χειριστήριο περιστρέφεται εκτός των καθορισμένων ορίων του στον τρισδιάστατο χώρο.

Βάσει κατάλληλου προγραμματισμού του Κυκλώματος TS και των δύο LEDs πράσινου και κίτρινου χρώματος αντίστοιχα, όταν το χειριστήριο διαγράφει συγκεκριμένες κινήσεις στον περιβάλλοντα χώρο που θεωρούνται εκτός των καθορισμένων ορίων (τιμές επιταχυνσιομέτρου $>70^\circ$ και γυροσκοπίου $>80^\circ$ προς μία ή και περισσότερες κατευθύνσεις κατά τους άξονες X, Y και Z), αυτόματα οι 2 αναλογικοί μοχλοί του χειριστηρίου παύουν να λειτουργούν και το πράσινο LED που υποδεικνύει την ορθή λειτουργία του χειριστηρίου σβήνει και ανάβει το κίτρινο LED στην θέση του. Στην κατάσταση αυτή οι μοχλοί θεωρούνται απενεργοποιημένοι και η χρήση τους δεν επιφέρει κανέναν επηρεασμό στην κατάσταση των σερβοκινητήρων.

Για επαναφορά των 2 μοχλών στην ενεργή τους κατάσταση αρκεί ένα μόνο παρατεταμένο πάτημα του ενός εκ των 2 μοχλών για μερικά δευτερόλεπτα, με το αυτόματο σβήσιμο του κίτρινου και άναμμα του πράσινου LED να ακολουθεί αυτής της ενέργειας.

```
Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = 48 | AcY = 8 | AcZ = 83 | Tmp = 52 C or 127 F | GyX = 3 | GyY = -2 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = 85 | AcY = -1 | AcZ = -10 | Tmp = 52 C or 127 F | GyX = 2 | GyY = 1 | GyZ = 0
AcX parameter off limits... joystick temporarily disabled !
Joystick value for X-axis: 92°
Joystick deactivated value for Y-axis inertia: 90°
```

Εικόνα 3.140 Ενεργοποίηση του Συστήματος Ασφαλείας Χειρισμού και ταυτόχρονη απενεργοποίηση των μοχλών του ασύρματου χειριστηρίου με ολίσθησή του εκτός των καθορισμένων ορίων του (κόκκινο πλαίσιο) στον τρισδιάστατο χώρο

```

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = -3 | AcY = 4 | AcZ = 98 | Tmp = 52 C or 127 F | GyX = 0 | GyY = 1 | GyZ = 1

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = -4 | AcY = -43 | AcZ = 84 | Tmp = 53 C or 127 F | GyX = -5 | GyY = 1 | GyZ = 0

Joystick value for X-axis: 92°
Joystick value for Y-axis: 90°

----Red LED & Buzzer are OFF!

/---Spatial Parameters of MPU6050---\
AcX = -2 | AcY = -79 | AcZ = 45 | Tmp = 52 C or 126 F | GyX = 0 | GyY = -1 | GyZ = 1

```

```

AcY parameter off limits... joystick temporarily disabled !
Joystick value for X-axis: 92°
Joystick deactivated value for Y-axis inertia: 90°

AcY parameter off limits... joystick temporarily disabled !
Joystick value for X-axis: 92°
Joystick deactivated value for Y-axis inertia: 90°

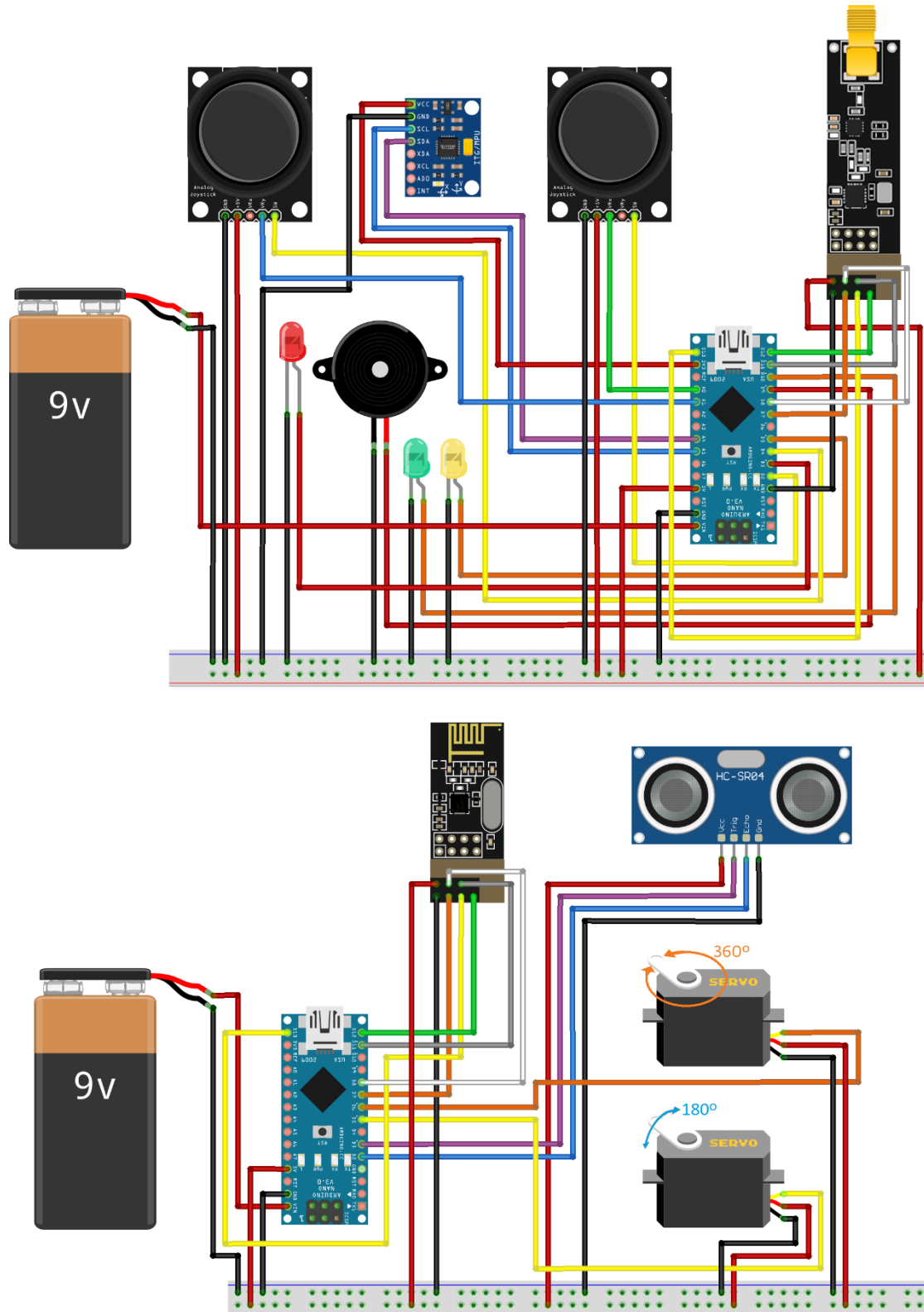
AcY parameter off limits... joystick temporarily disabled !
Joystick value for X-axis: 92°
Joystick deactivated value for Y-axis inertia: 90°

AcY parameter off limits... joystick temporarily disabled !
Joystick value for X-axis: 92°
Joystick deactivated value for Y-axis inertia: 90°

```

***Εικόνα 3.141** Ενεργοποίηση του Συστήματος Ασφαλείας Χειρισμού και ταυτόχρονη απενεργοποίηση των μοχλών του ασύρματου χειριστηρίου με ολίσθησή του εκτός των καθορισμένων ορίων του (κόκκινο πλαίσιο) στον τρισδιάστατο χώρο*

Δοκιμή λειτουργίας του Σ.Η.Α.Ε. με το μοντέλο πομποδέκτη NRF24L01 υποδοχής SMA για εξωτερική κεραία και τσιπ RFX2401C



Εικόνα 3.142 Σχηματικό διάγραμμα της Ολοκληρωμένης Κατασκευής Σ.Η.Α.Ε. με διαφορετικά μοντέλα πομποδεκτών NRF24L01

Με τον σχεδιασμό όλων των μοντέλων NRF24L01 να διαθέτουν την ίδια σειρά ακροδεκτών και άρα τον ίδιο τρόπο συνδεσμολογίας σε μια ηλεκτρονική διάταξη, με εναλλαγή του μοντέλου πομποδέκτη NRF24L01 ενσωματωμένης κεραίας με την παραλλαγή πομποδέκτη NRF24L01 που διαθέτει υποδοχή SMA για σύνδεση εξωτερικής κεραίας και ένα τσιπ RFX2401C ενισχυτή σήματος (PA) και ενισχυτή χαμηλού θορύβου (LNA), επιτυγχάνεται εξίσου αποτελεσματική ασύρματη επικοινωνία.

Interface



| Pin # | Name | Description |
|-------|------|--|
| 1 | GND | Ground (0V) |
| 2 | VCC | Supply (+1.9V - +3.6V DC) |
| 3 | CE | Chip Enable Activates RX or TX mode (Never >3.6V) |
| 4 | CSN | SPI Chip Select |
| 5 | SCK | SPI Clock |
| 6 | MOSI | SPI Slave Data Input |
| 7 | MISO | SPI Slave Data Output, with tri-state option |
| 8 | IRQ | Maskable interrupt pin. Active low |

Εικόνα 3.143 Διάγραμμα διάταξης ακροδεκτών & λειτουργίες τους

Επειδή όμως στην πράξη είναι κάπως χρονοβόρες οι καταστάσεις εναλλαγής –σύνδεσης, αποσύνδεσης και επανασύνδεσης– μεταξύ διαφορετικών μονάδων πομποδεκτών και ιδίως όταν αυτοί φέρουν πολλούς ακροδέκτες, στην συγκεκριμένα υλοποίηση έγινε επίσης προσθήκη 2 μονάδων βάσης NRF24L01 (*NRF24L01 base modules*) για αρκετά ευκολότερη εναλλαγή μεταξύ των παραλλαγμένων μοντέλων τους.

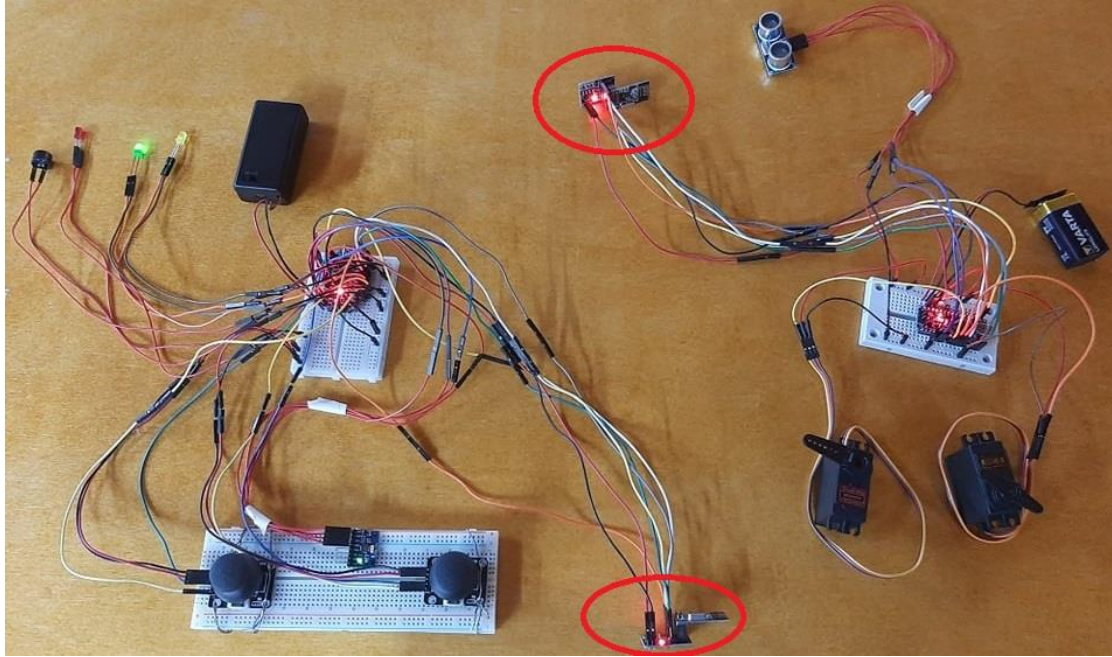
Επίσης, καθώς στην παρούσα περίπτωση της ολοκληρωμένης κατασκευής του Σ.Η.Α.Ε. η μονάδα βάσης διατηρείται σταθερή και ενσωματωμένη στην ηλεκτρονική πλακέτα εκάστου Υποσυστήματος, σε περίπτωση αλλαγής του μοντέλου πομποδέκτη απλά και μόνο από επιλογή του χρήστη ή σε περιπτώσεις αντικατάστασης λόγω βλάβης, η εναλλαγή μεταξύ των μοντέλων πομποδεκτών καθίσταται πολύ πιο εύκολα πραγματοποιήσιμη.

Λαμβάνοντας υπόψη τα ανωτέρω, γίνεται δοκιμή υλοποίησης συνδεσμολογίας μεταξύ δύο διαφορετικών παραλλαγών μοντέλου NRF24L01 και παρατηρείται ότι τα δύο Υποσυστήματα επικοινωνούν το ίδιο αποτελεσματικά με όλες τις προηγούμενες διαδικασίες που εκτελέστηκαν έως τώρα με χρήση των εν λόγω πομποδεκτών.

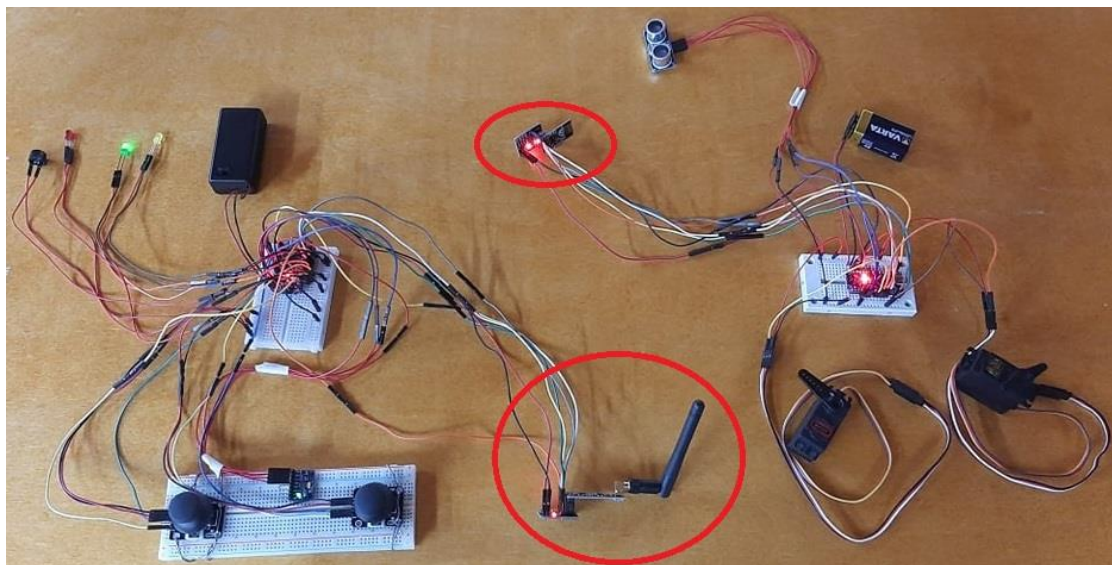
Αποκλειστικά για αυτήν την περίπτωση διαφορετικών παραλλαγών πομποδεκτών σε ένα Σ.Η.Α.Ε., επειδή συγκεκριμένα το μοντέλο NRF24L01 με ενσωματωμένη κεραία καλύπτει ένα εύρος 100m ενώ το μοντέλο NRF24L01 με υποδοχή SMA για εξωτερική κεραία καλύπτει εύρος 1000m, καθώς σε απόσταση άνω των 100m θα μπορεί μόνο το Υποσύστημα που υποστηρίζει ασύρματη επικοινωνία 1000m να αποστέλλει πληροφορίες στο άλλο Υποσύστημα, τέτοια δυνατότητα δύναται να αξιοποιηθεί για ιδιαίτερες καταστάσεις εφαρμογών όπου μετά από ένα ορισμένο μήκος είναι επιθυμητή η διατήρηση επικοινωνίας από το ένα Υποσύστημα του Σ.Η.Α.Ε.

Στις δύο εικόνες που ακολουθούν παρατηρείται η επιτυχής λειτουργία του Σ.Η.Α.Ε.:

- με 2 πομποδέκτες NRF24L01 ενσωματωμένης κεραίας (εικ. 3.144)
- με αντικατάσταση του μοντέλου NRF24L01 του Υποσυστήματος Πομπού με τον NRF24L01 υποδοχής SMA για κεραία (εικ. 3.145)

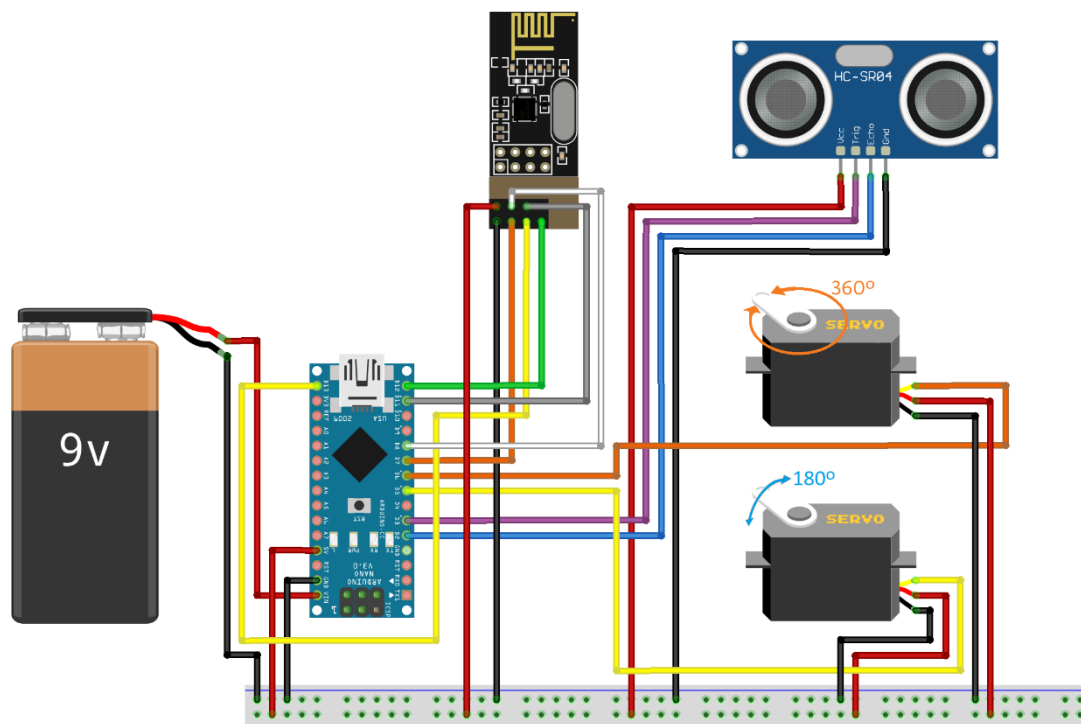
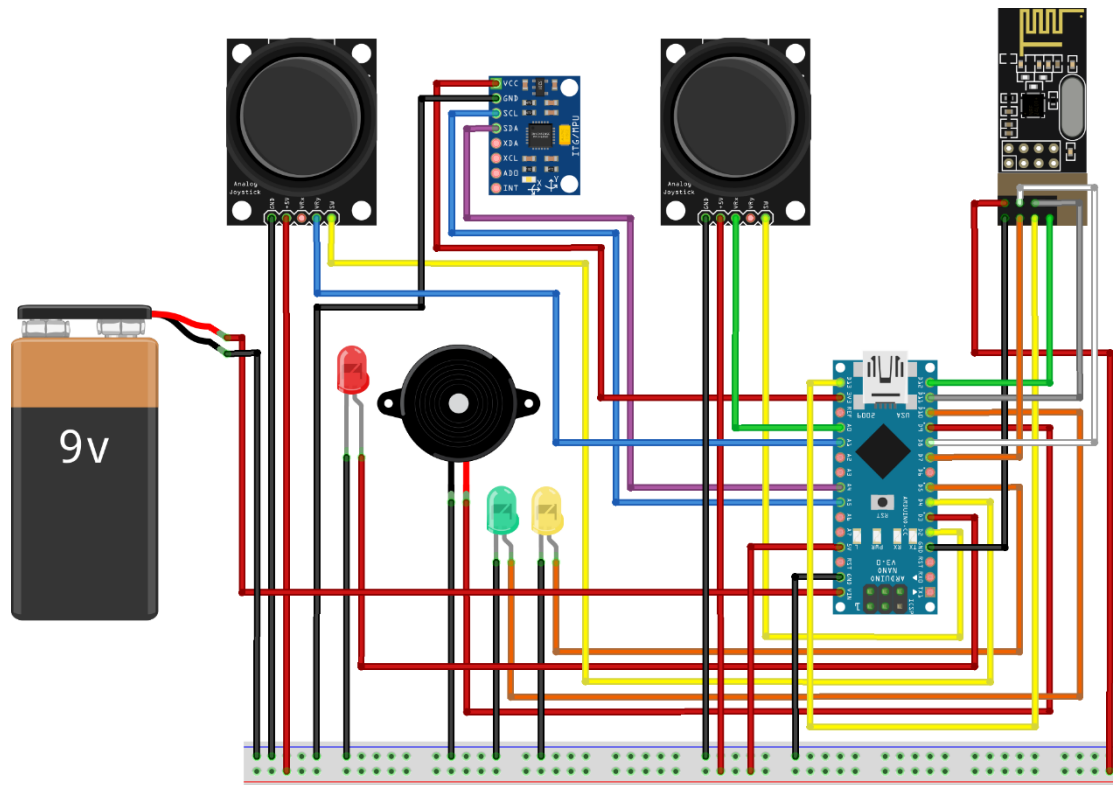


Εικόνα 3.144 Συνδεσμολογία Σ.Η.Α.Ε. με πομποδέκτη NRF24L01 ενσωματωμένης κεραίας, υλοποιημένη σε επαγγελματικό εργαστήριο



Εικόνα 3.145 Συνδεσμολογία Σ.Η.Α.Ε. με διαφορετικό μοντέλο πομποδέκτη NRF24L01 ανά Υποσύστημα, υλοποιημένη σε επαγγελματικό εργαστήριο

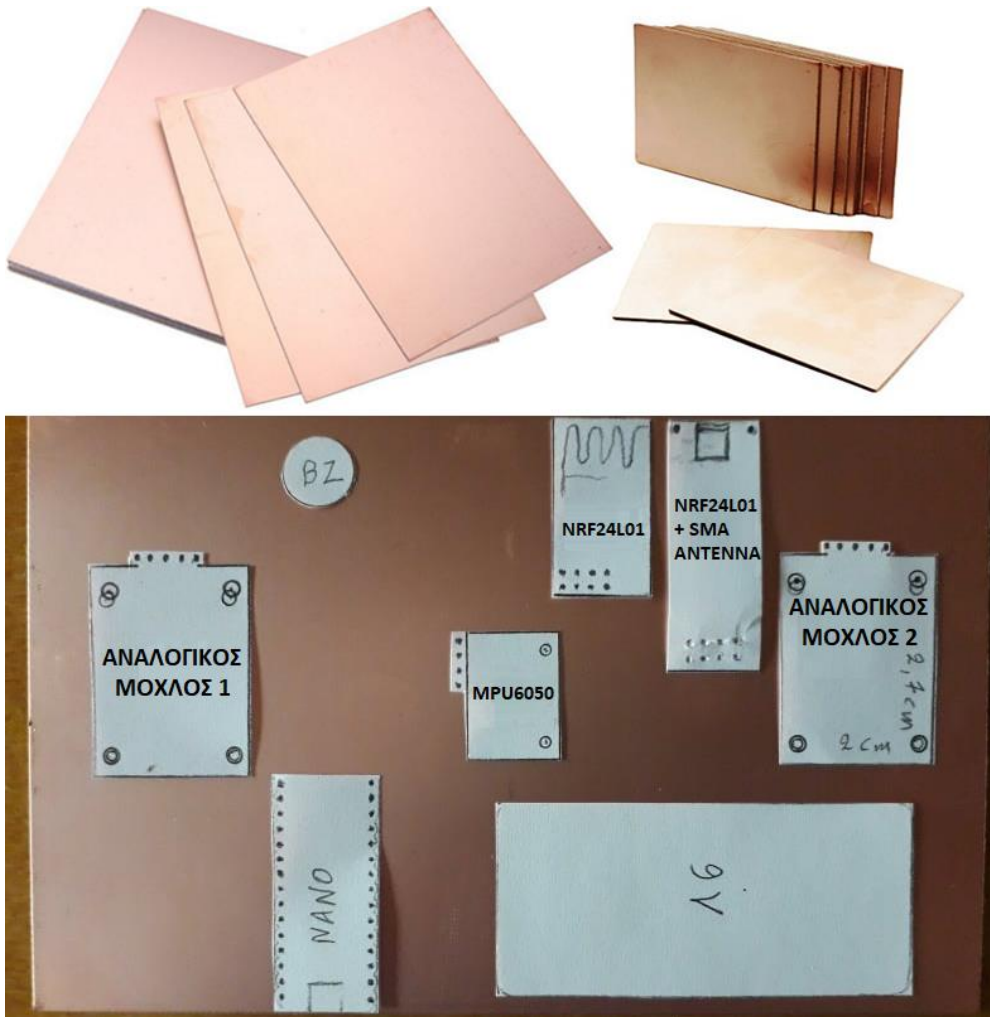
3.5.1 Διαδικασία Υλοποίησης της Ενδεικτικής Ολοκληρωμένης Κατασκευής του Σ.Η.Α.Ε.



Εικόνα 3.146 Σχηματικό διάγραμμα της Ενδεικτικής Ολοκληρωμένης Κατασκευής του Σ.Η.Α.Ε.

Η μέθοδος που επιλέγεται για την δημιουργία των τυπωμένων κυκλωμάτων της ενδεικτικής ολοκληρωμένης κατασκευής του Σ.Η.Α.Ε. είναι αυτή της αποχάλκωσης.

Με χρήση επιχάλκωμένων πλακετών δύο όψεων (*double sided copper plated plastic boards*), αρχικά γίνεται η οριοθέτηση της εκάστοτε ηλεκτρονικής μονάδας που απαρτίζει Σ.Η.Α.Ε. ώστε να γίνει η ορθή κατανομή τους επί των πλακετών χαλκού.



Εικόνα 3.147 Επιχάλκωμένες πλακέτες (άνω) και οριοθέτηση ηλεκτρονικών μονάδων επί πλακέτας χαλκού σε επαγγελματικό εργαστήριο για το Ασύρματο Χειριστήριο (κάτω)

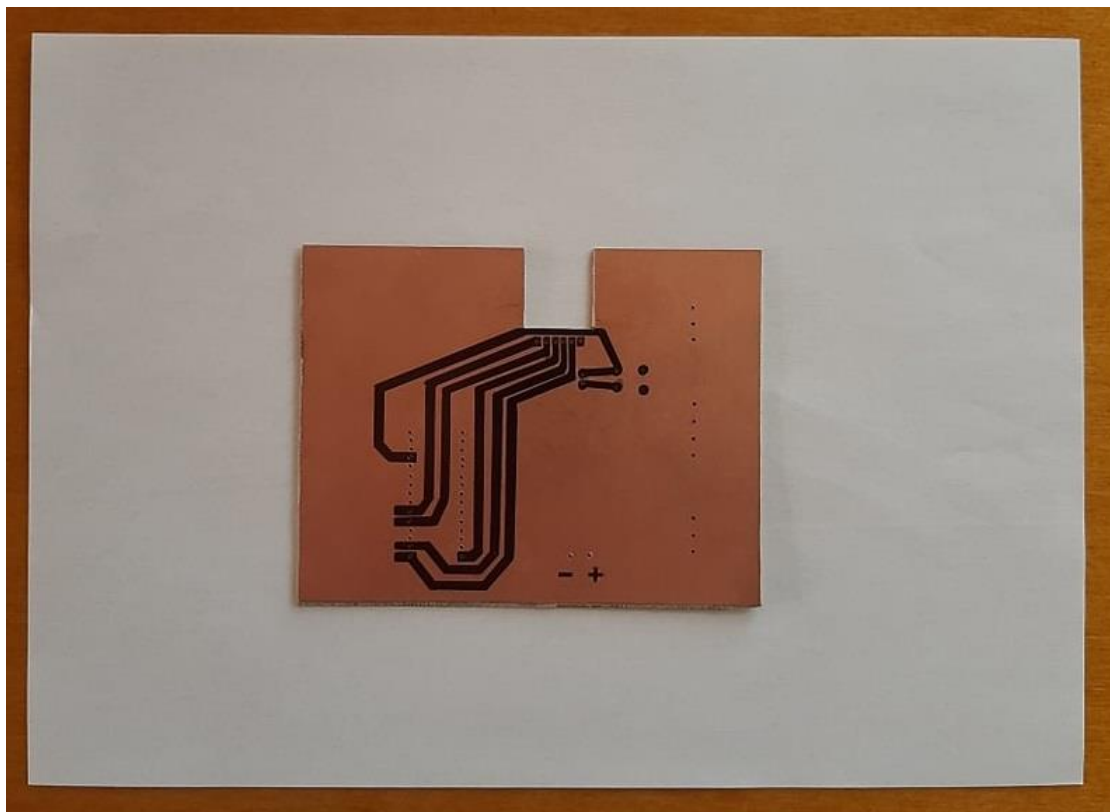
ΣΗΜΕΙΩΣΗ – ΕΠΙΣΗΜΑΝΣΗ: Ο αρχικός σχεδιασμός προέβλεπε την ένταξη και των δύο μοντέλων πομποδέκτη NRF24L01 που αναφέρονται στην παρούσα εργασία, με έναν διακόπτη εναλλαγής για λειτουργία του ενός εκ των δύο, αναλόγως της επιθυμίας του χρήστη.

Η ιδέα αυτή απορρίφθηκε με την απόφαση ένταξης της μονάδας βάσης NRF24L01 στο σύστημα.

Σύμφωνα με την εικόνα 3.136, εφόσον έχει επιλεγεί η επιθυμητή κατανομή των μονάδων του Κυκλώματος TS στην πλακέτα χαλκού, γίνεται χρήση ενός ανεξίτηλου μαρκαδόρου μαύρου χρώματος και διαστάσεων 1mm για τον σχεδιασμό των περιγραμμάτων των μονάδων επί της επιχλωμένης πλακέτας δύο όψεων.

Αντίστοιχη διαδικασία εκτελείται και για το Υποσύστημα Δέκτη.

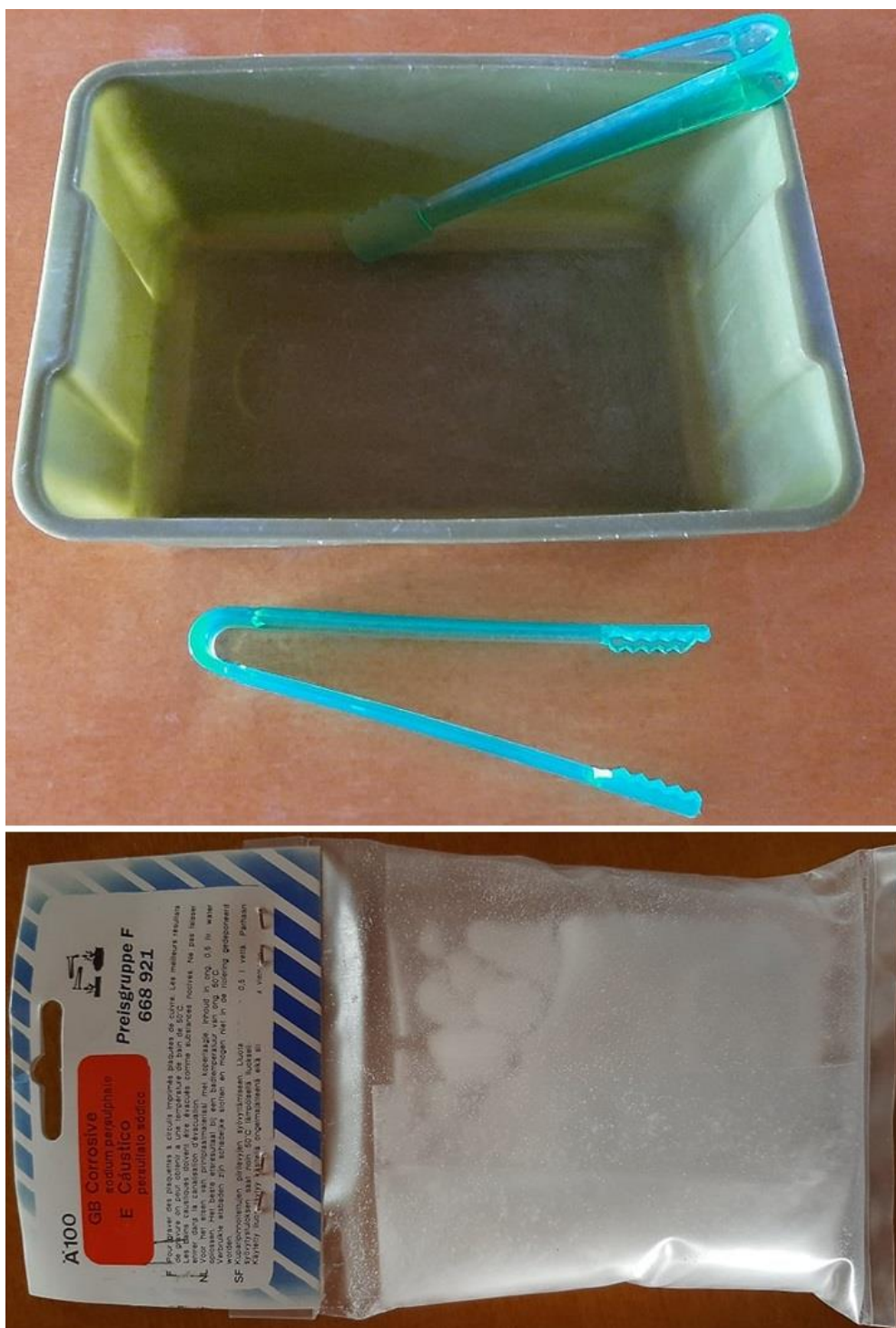
Με την ολοκλήρωση του σχεδιασμού των πλακετών για κάθε Υποσύστημα, γίνεται στην συνέχεια χρήση ενός δοχείου για την αποχάλκωση μερών της επιφάνειας των πλακετών που δεν χρειαζόμαστε και συγκεκριμένα των ελεύθερων επιφανειών χαλκού, αυτών δηλαδή που δεν έχουν γραμμοσκιαστεί με ανεξίτηλο μαύρο μελάνι.



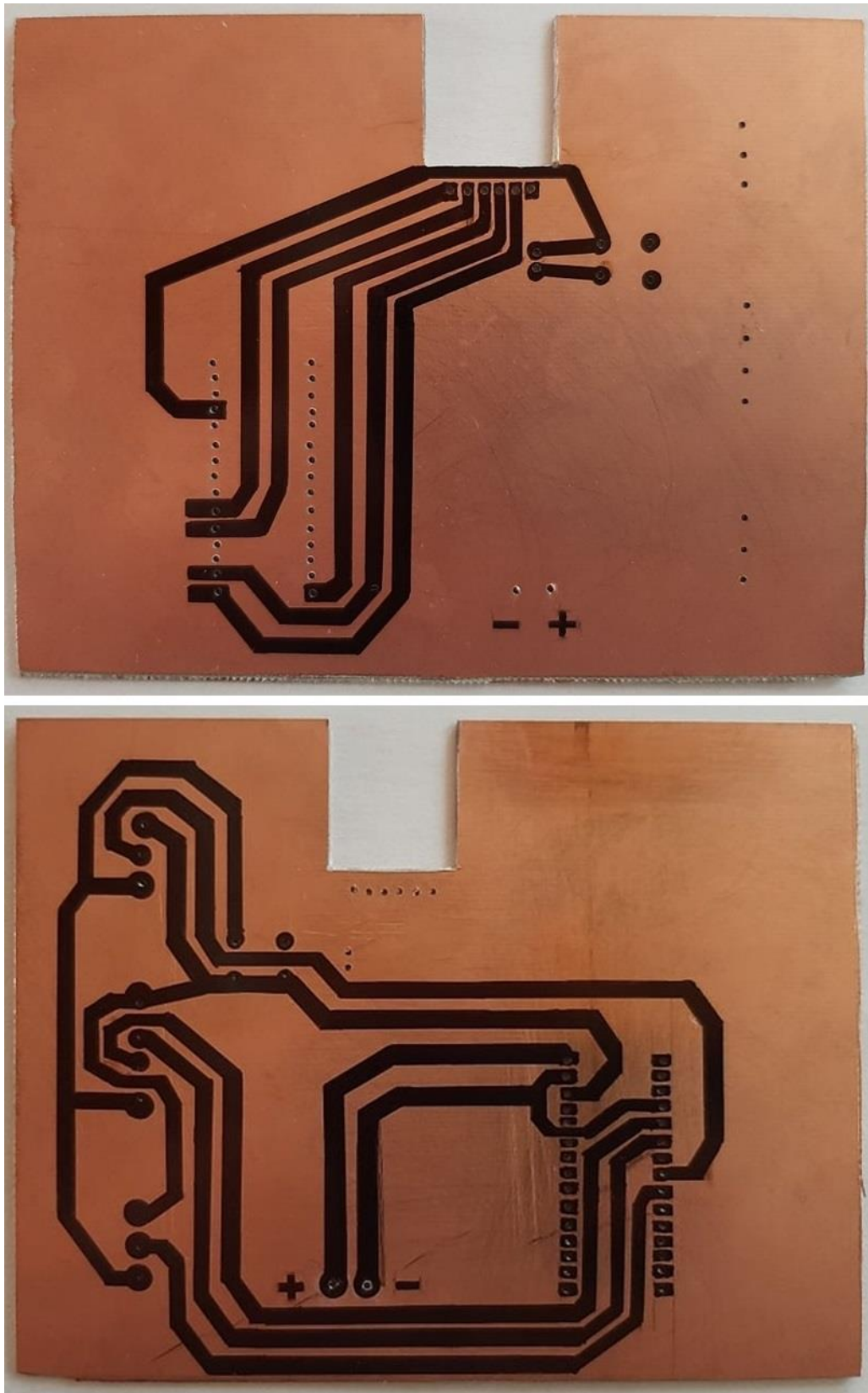
Εικόνα 3.148 Άνω όψη της πλακέτας του Κυκλώματος RS, τοποθετημένο σε χαρτί A4, με σχεδιασμένες τις διαδρομές για τις απαραίτητες συνδέσεις μεταξύ των μονάδων

Για τον σχεδιασμό επιλέχθηκε να γίνει κατασκευή τυπωμένων κυκλωμάτων διπλής όψης, δηλαδή ο σχεδιασμός κάθε πλακέτας θα περιέχει διαδρομές σχεδιασμένες με μαρκαδόρο και από τις δύο πλευρές, καθώς η διαδικασία αυτή απελευθερώνει αρκετό χώρο στις πλακέτες με τον διαμοιρασμό των συνδεσμολογιών σε δύο επιφάνειες.

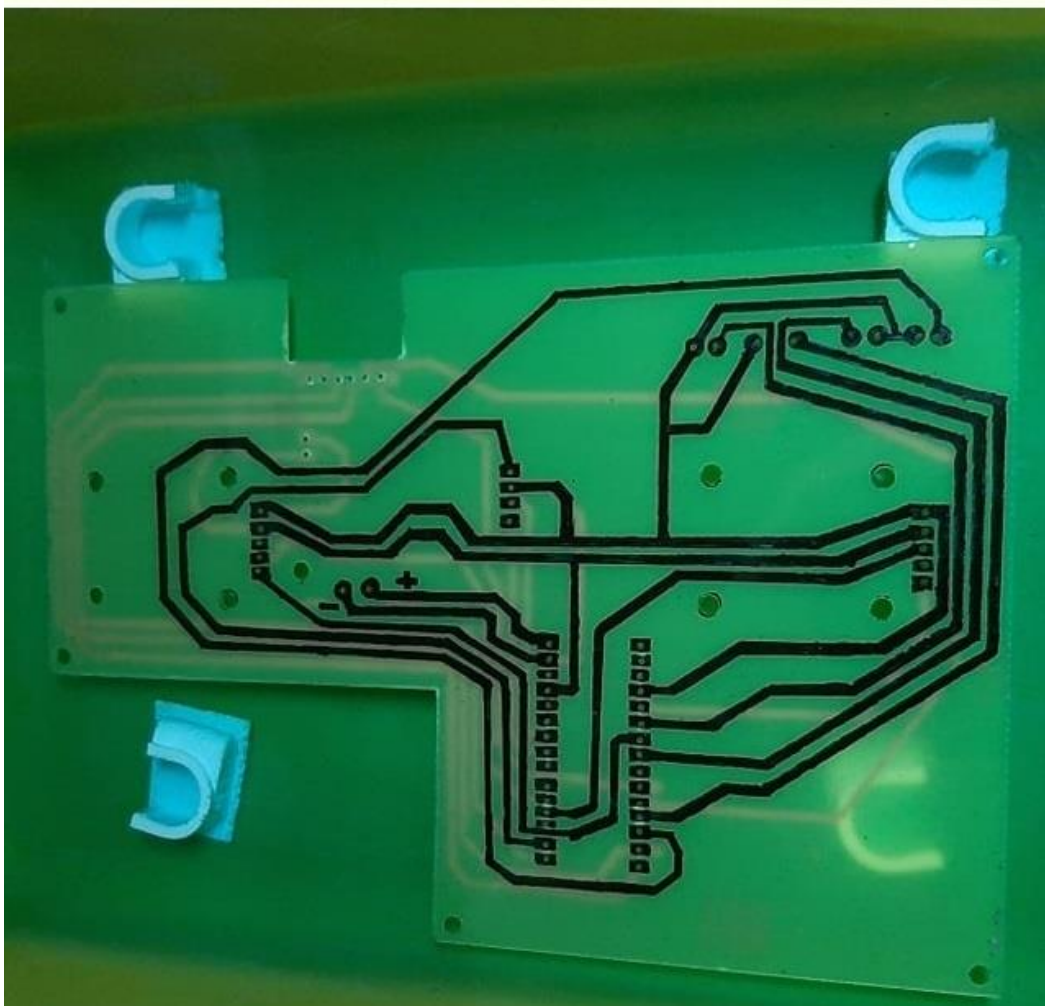
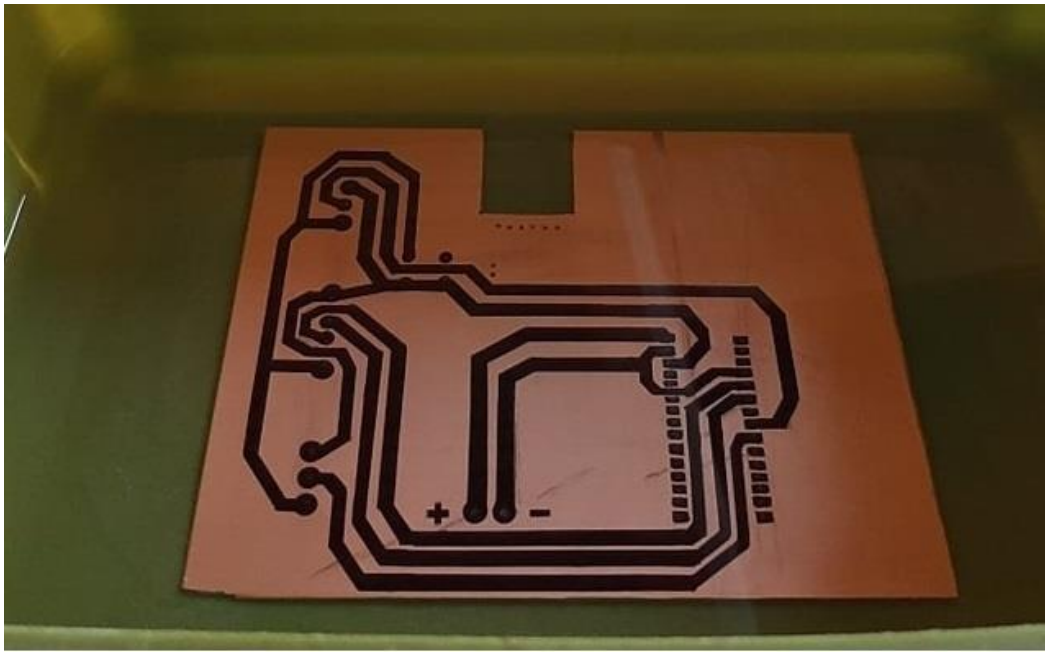
Για την μέθοδο αποχάλκωσης της πλακέτας Πομπού και Δέκτη γίνεται χρήση ενός πλαστικού δοχείου και μιας συσκευασίας 100gr καυστικής σόδας ($FeCl_3$) που όταν ανακατευτεί και διαλυθεί σε νερό (0,5lit στους 50°C) δημιουργεί αποχάλκωση στις βυθιζόμενες στο δοχείο πλακέτες.



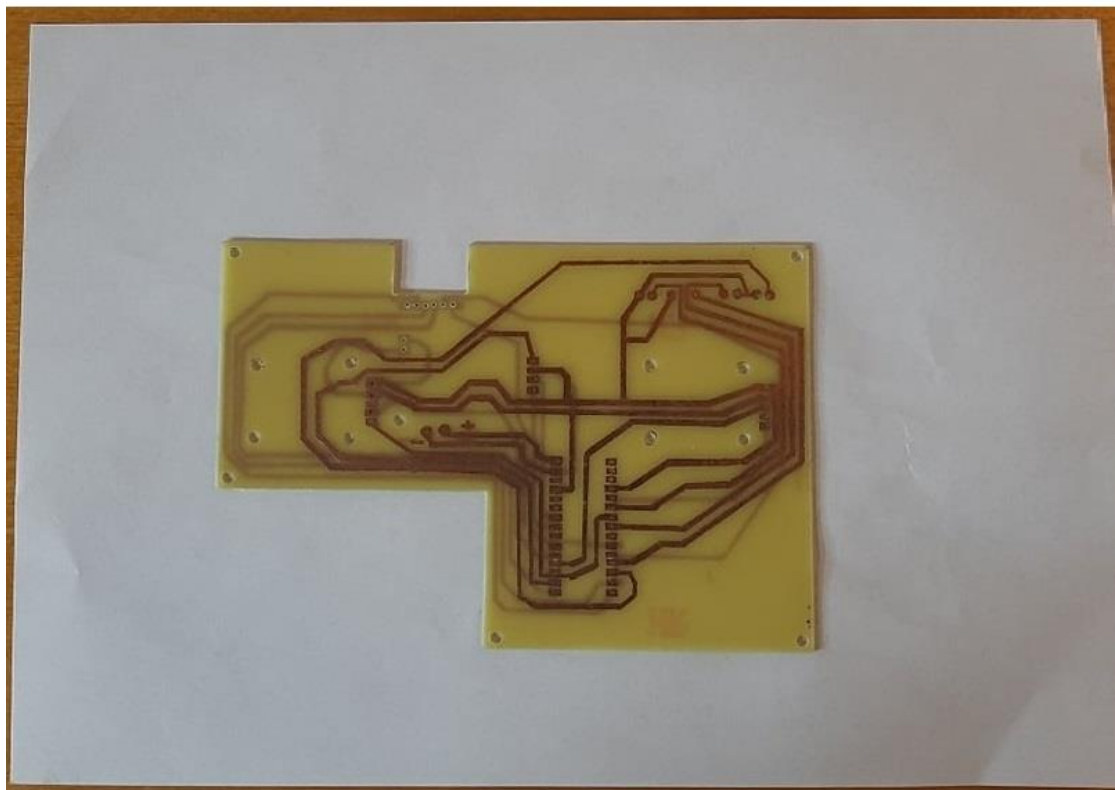
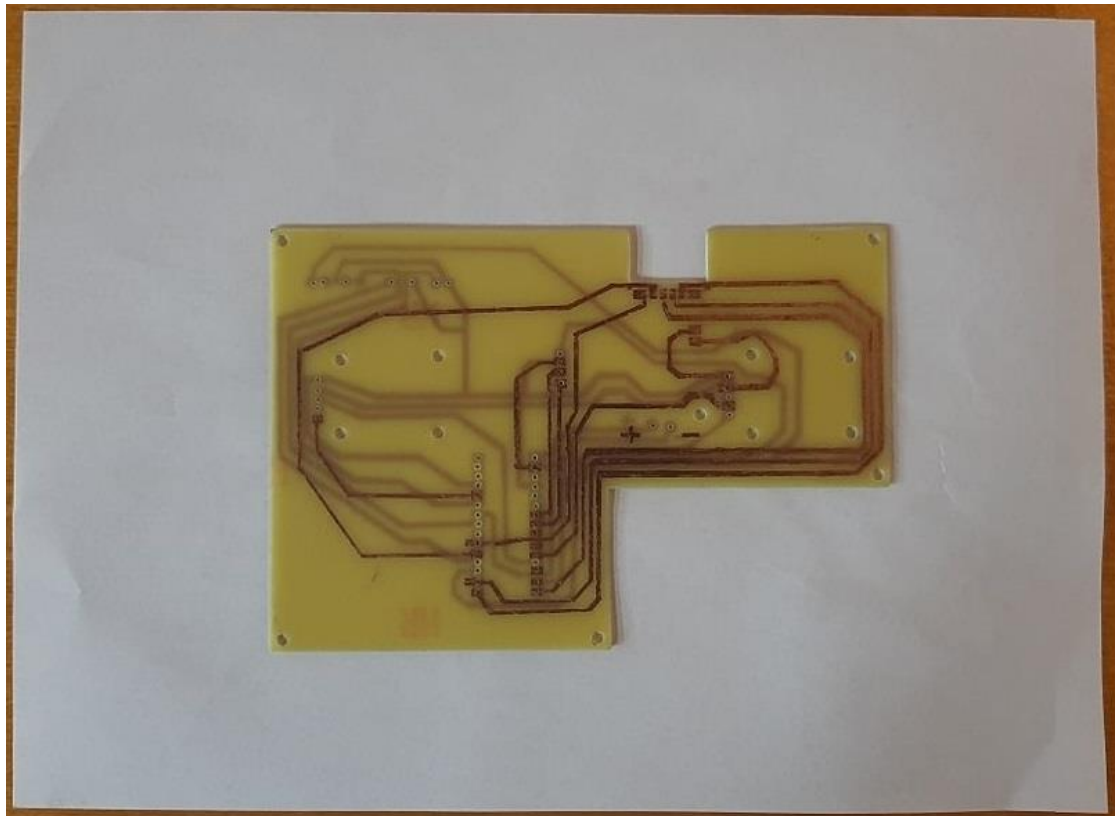
Εικόνα 3.149 Πλαστικό δοχείο (άνω) και συσκευασία καυστικής σόδας (κάτω)



Εικόνα 3.150 Άνω και κάτω όψη της πλακέτας του Υποσυστήματος Δέκτη, πριν την εφαρμογή της μεθόδου αποχάλκωσης



Εικόνα 3.151 Στιγμιότυπο από διαδικασία αποχάλκωσης πλακέτας Δέκτη αμέσως μετά την βύθιση (άνω) και πλακέτας Ασύρματου Χειριστηρίου μετά από 90' βύθισης (κάτω)



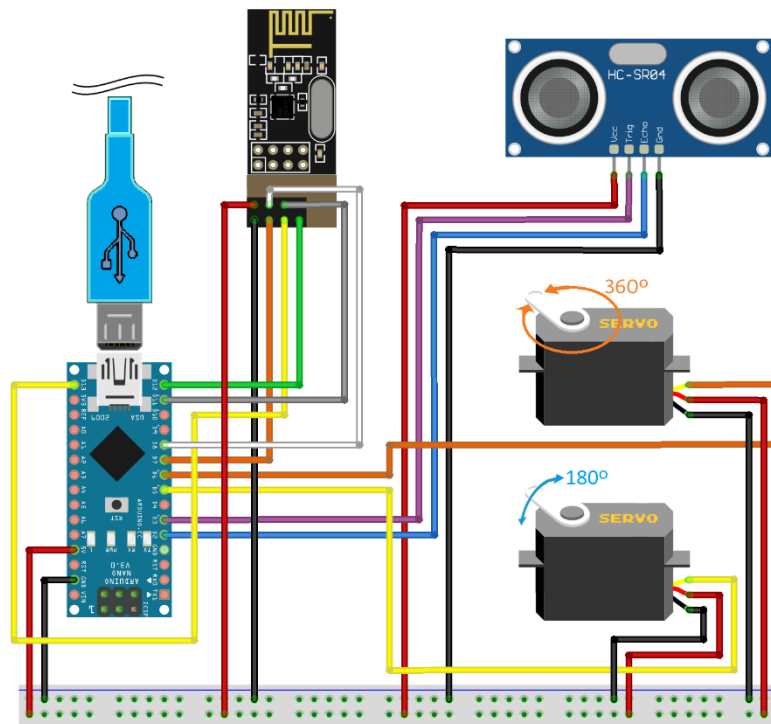
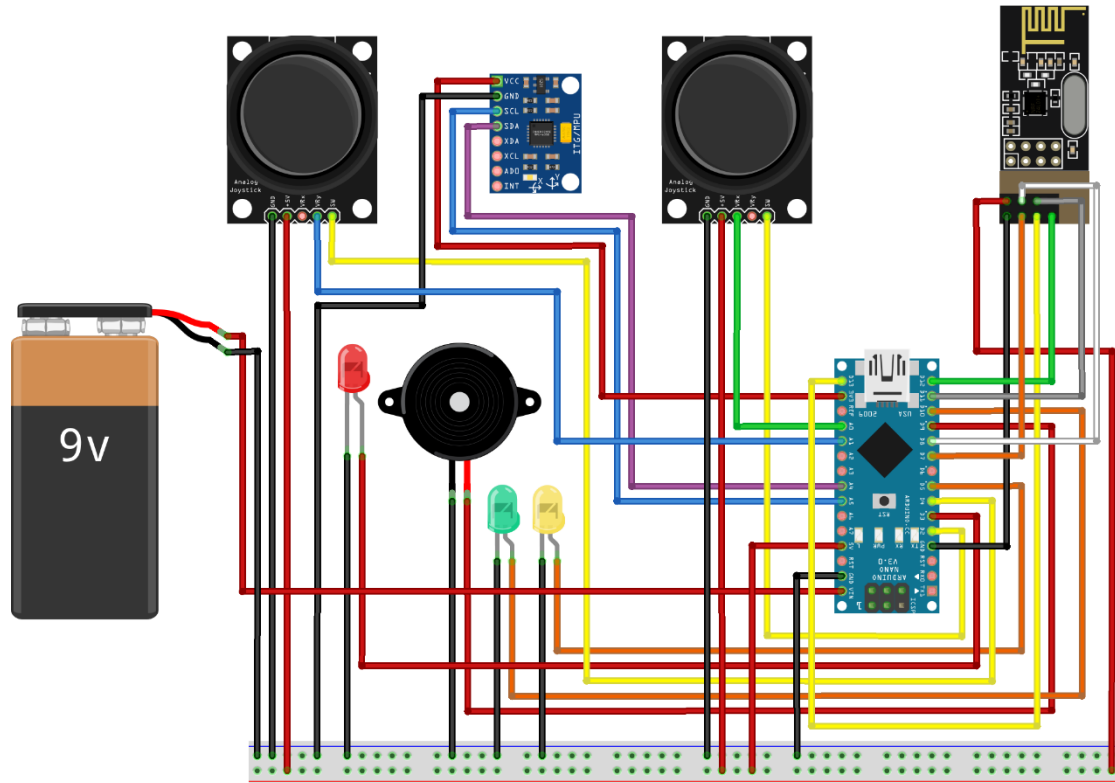
Εικόνα 3.152 Άνω και κάτω όψη της πλακέτας του Ασύρματου Χειριστηρίου, τοποθετημένο σε χαρτί A4, μετά την εφαρμογή της μεθόδου αποχάλκωσης

Με τα τυπωμένα κυκλώματα των δύο Υποσυστημάτων του Σ.Η.Α.Ε. να έχουν υλοποιηθεί και με δημιουργία οπών σε αυτά κατάλληλων για τοποθέτηση και συγκόλληση των ηλεκτρονικών μονάδων μέσω των ακροδεκτών τους, η τελευταία ενέργεια που εκτελέστηκε για την ολοκλήρωση της κατασκευής ήταν η κατάλληλη επεξεργασία 4 διάφανων φύλλων τύπου πλέξιγκλας για την δημιουργία δοχείου προστασίας για έκαστο Υποσύστημα.

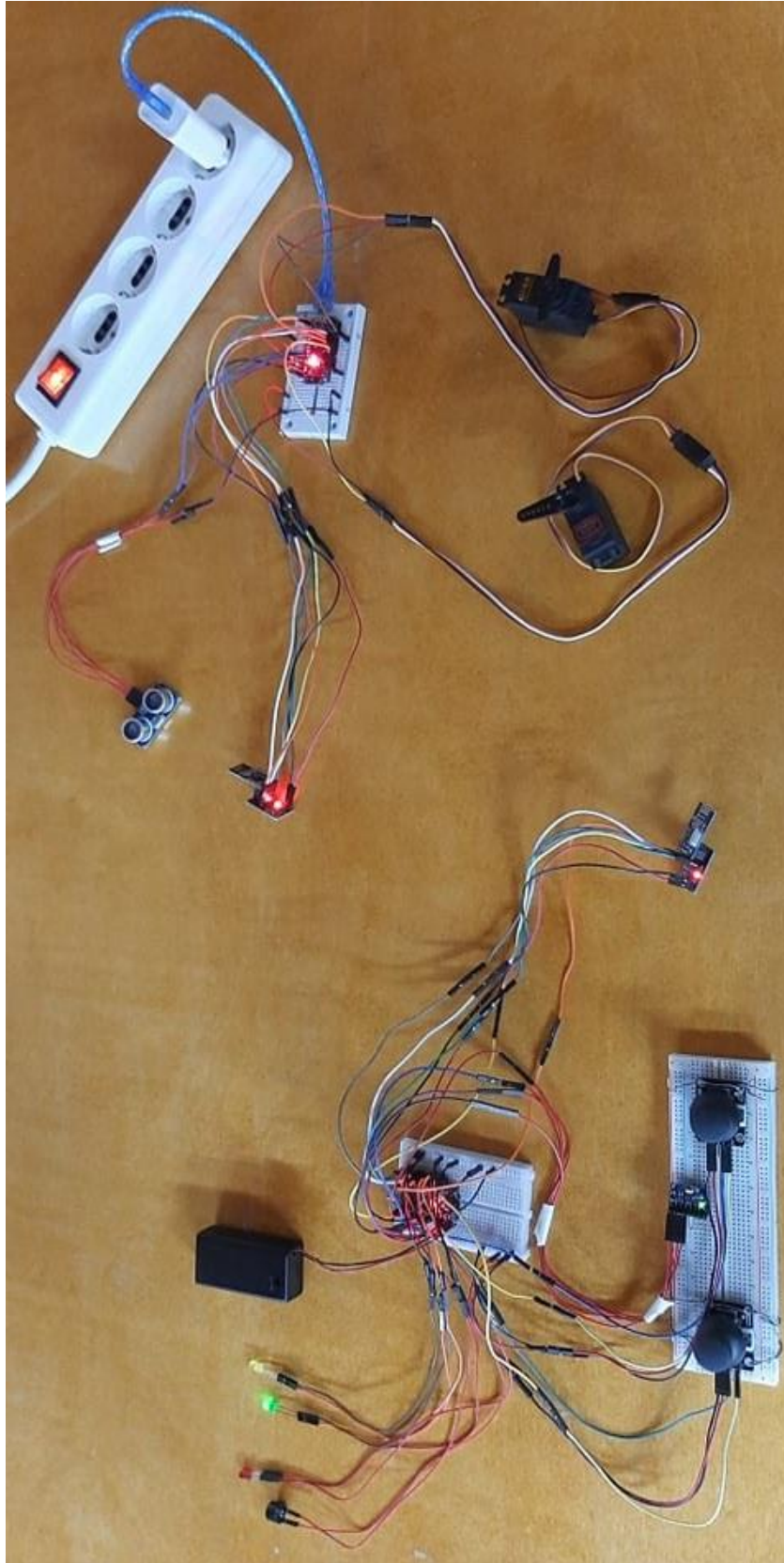
Με την ολοκλήρωση όλων των ανωτέρω διεργασιών, έχει πλέον ολοκληρωθεί η ενδεικτική κατασκευή του Σ.Η.Α.Ε. για Βιομηχανικές Εφαρμογές.

Στις εικόνες 3.153, 3.154 και 3.155 που ακολουθούν παρουσιάζονται με την σειρά το σχηματικό διάγραμμα, η σχετική συνδεσμολογία και η αντίστοιχη ολοκληρωμένη κατασκευή του Σ.Η.Α.Ε.

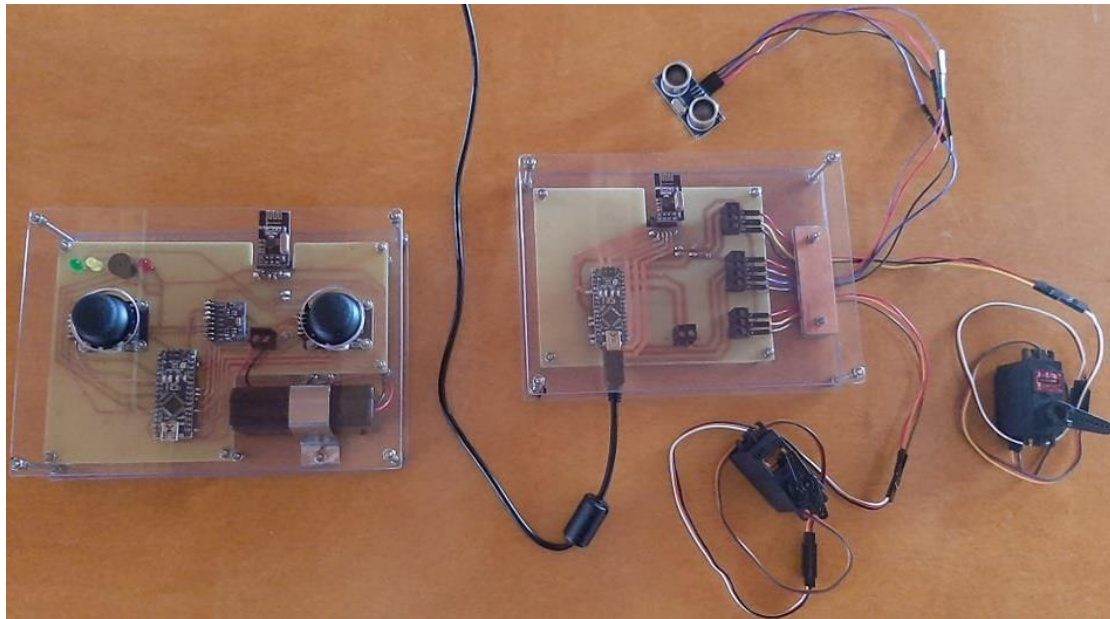
Για το Υποσύστημα Δέκτη επελέγη η τροφοδοσία να παρέχεται μέσω καλωδίου USB από το δίκτυο της ΔΕΗ για χρήση του ως μη φορητό υποσύστημα, έχει όμως σχεδιαστεί και κατασκευαστεί και για διασύνδεση με μπαταρία στην περίπτωση που η φορητή ιδιότητα και των δύο Υποσυστημάτων ζητηθεί βάσει κάποιας απαιτητικής βιομηχανικής εφαρμογής.



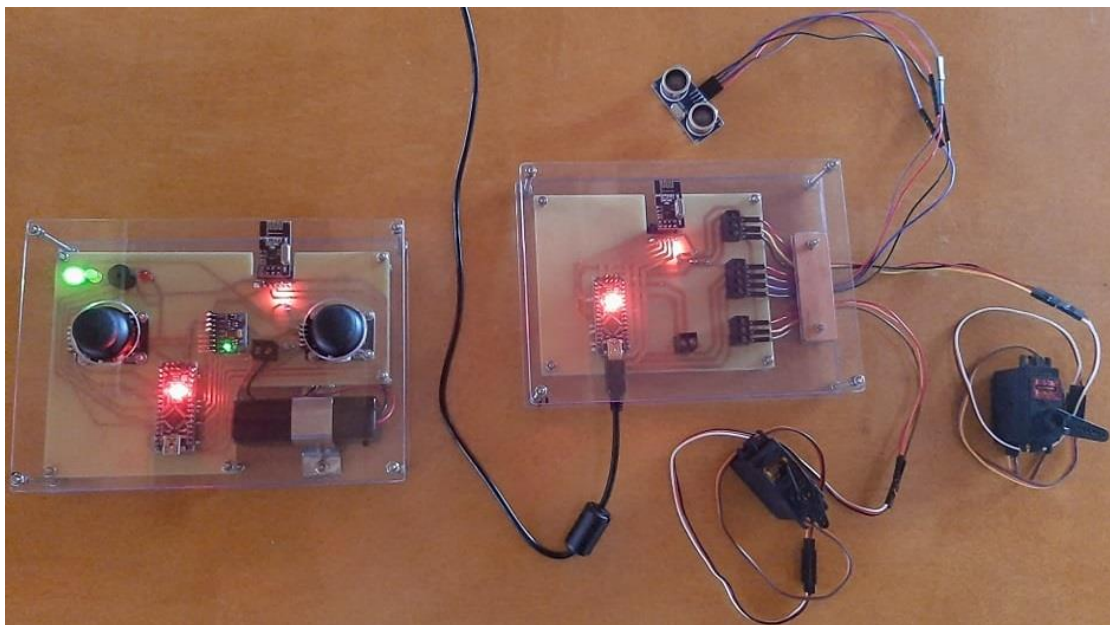
Εικόνα 3.153 Σχηματικό διάγραμμα της ενδεικτικής ολοκληρωμένης κατασκευής του Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή για Βιομηχανικές Εφαρμογές, σχεδιασμένο σε περιβάλλον προσομοίωσης



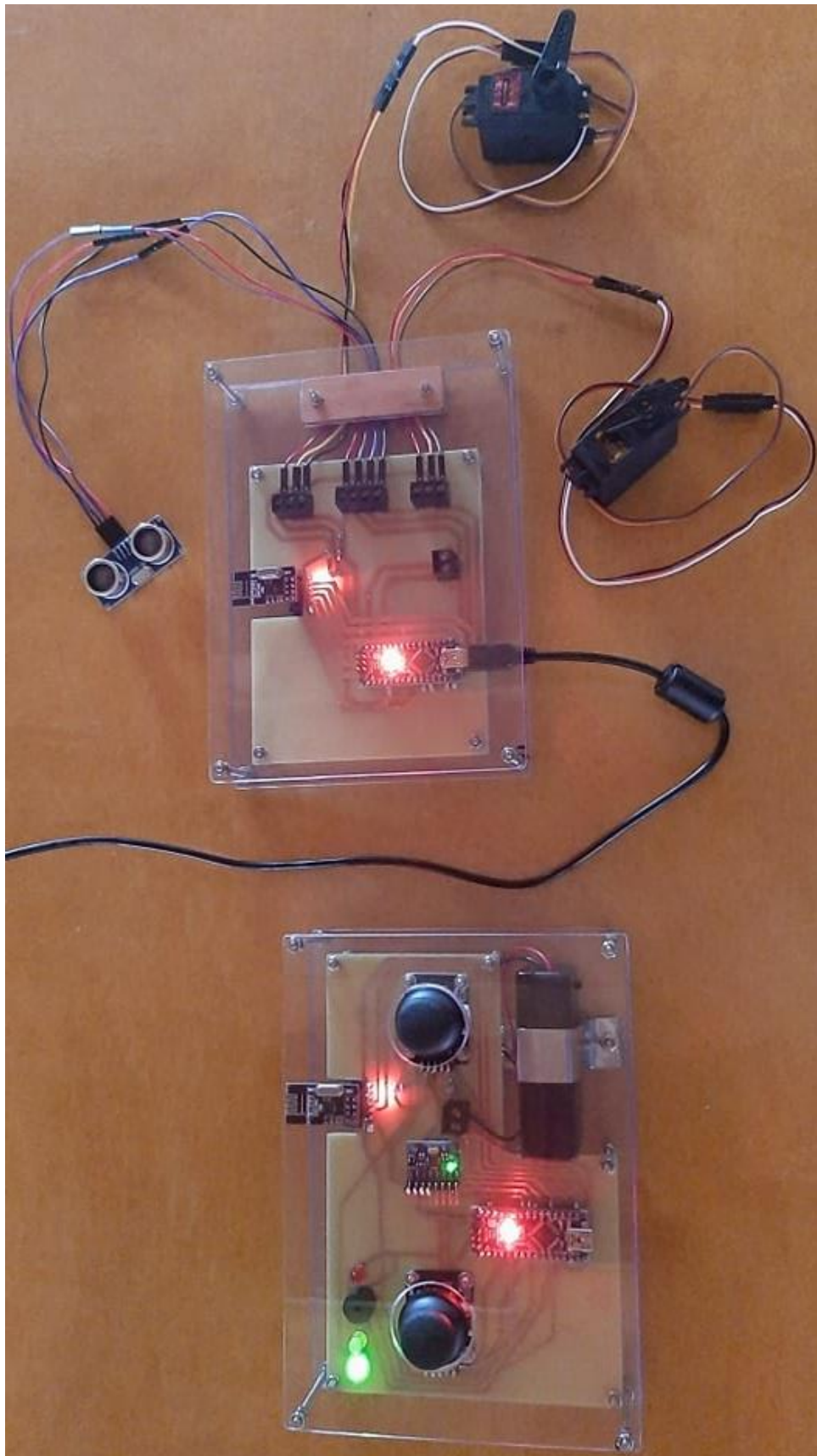
Εικόνα 3.154 Συνδεσμολογία της ενδεικτικής ολοκληρωμένης κατασκευής του Σ.Η.Α.Ε. για Βιομηχανικές Εφαρμογές, υλοποιημένη σε επαγγελματικό εργαστήριο



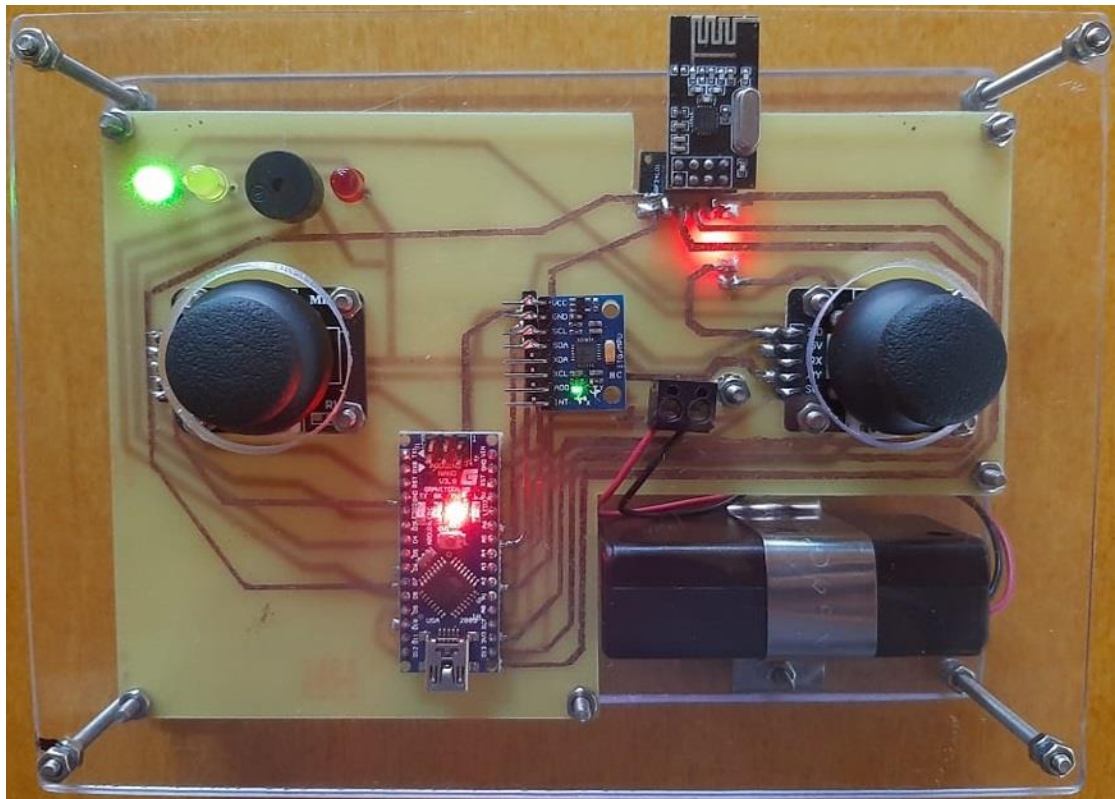
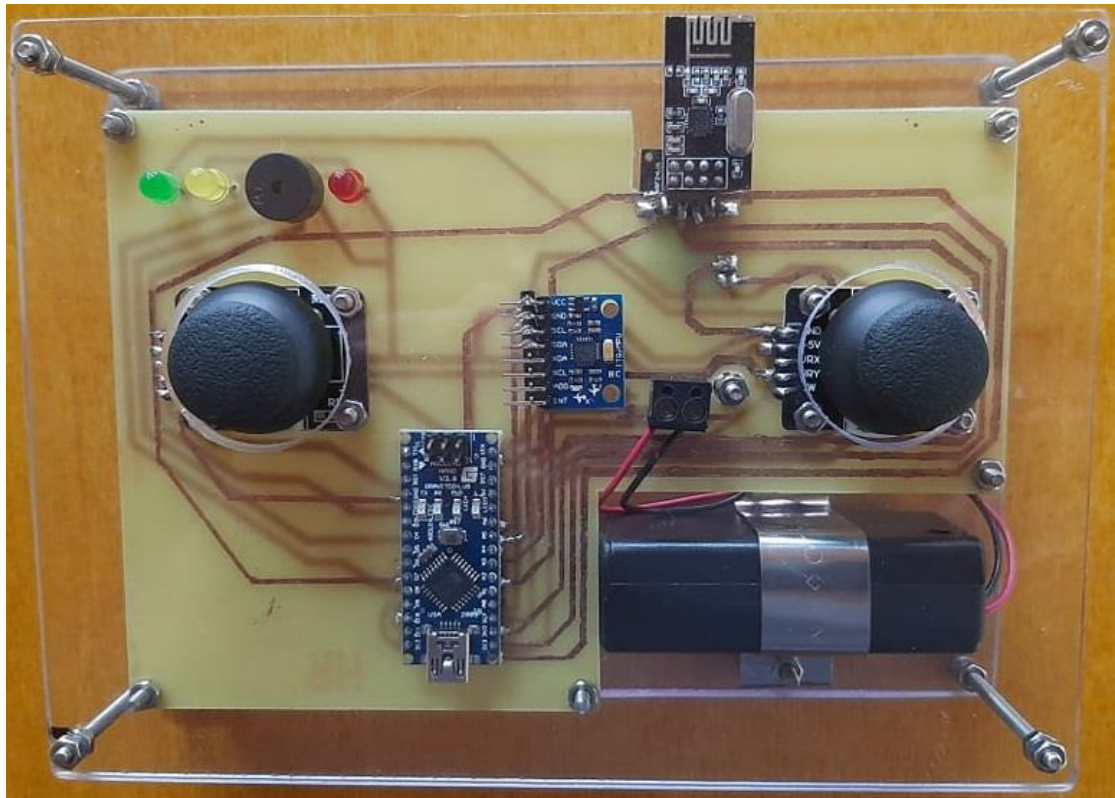
Εικόνα 3.155 Ολοκληρωμένη Κατασκευή ενός Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) για Βιομηχανικές Εφαρμογές, υλοποιημένη σε επαγγελματικό εργαστήριο



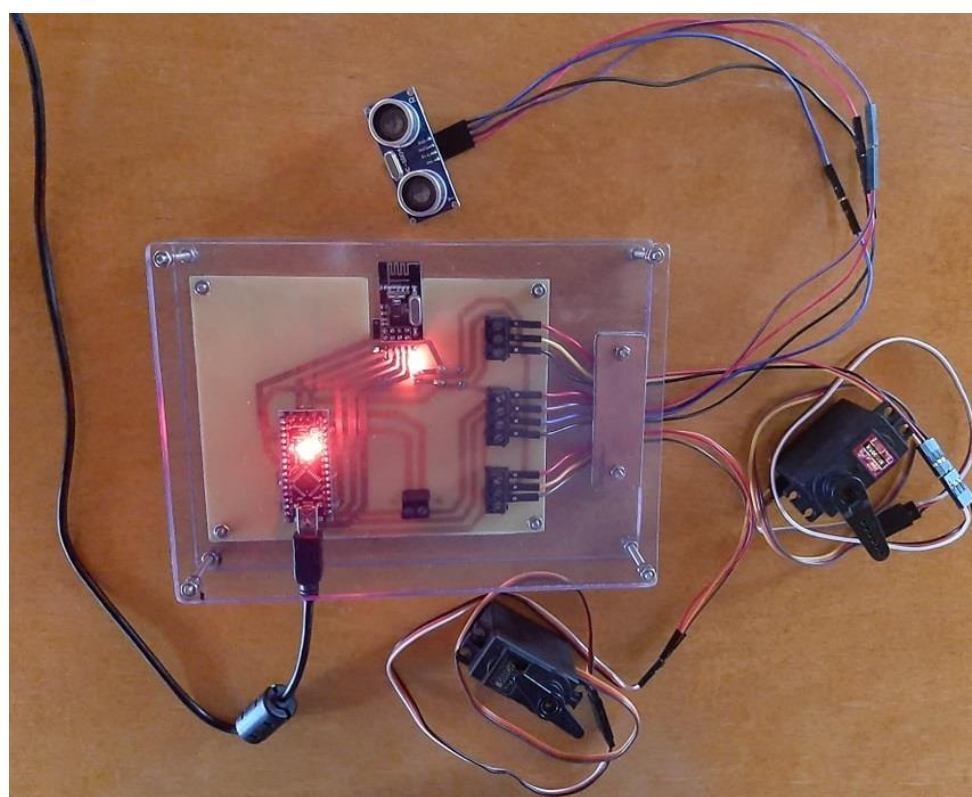
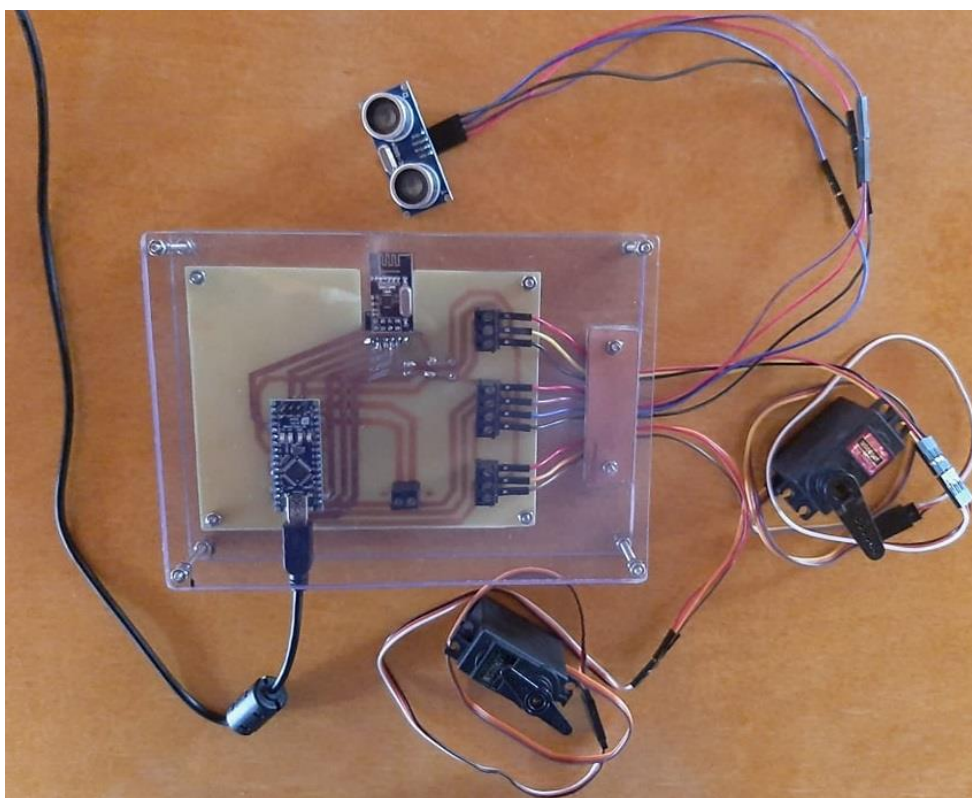
Εικόνα 3.156 Ολοκληρωμένη Κατασκευή ενός Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.), σε κατάσταση ενεργής λειτουργίας



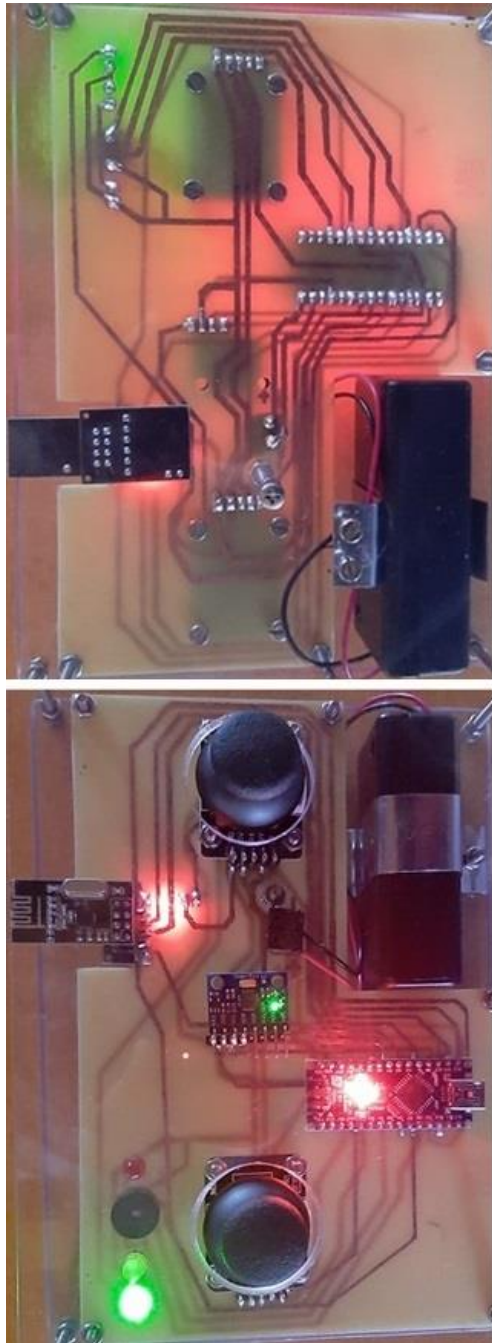
Εικόνα 3.156 – Zoom in Ολοκληρωμένη Κατασκευή ενός Συστήματος Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.), σε λειτουργία



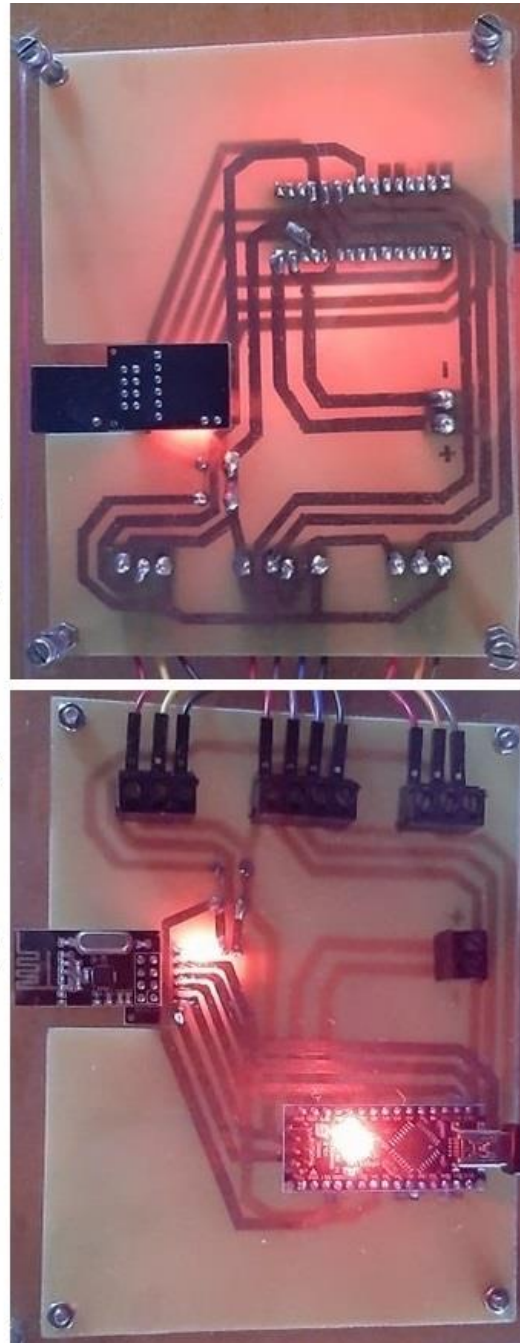
Εικόνα 3.157 Ασύρματο Χειριστήριο του Σ.Η.Α.Ε.,
ανενεργό (άνω) και σε κατάσταση λειτουργίας (κάτω)



Εικόνα 3.158 Υποσύστημα Δέκτη του Σ.Η.Α.Ε.,
ανενεργό (άνω) και σε κατάσταση λειτουργίας (κάτω)



Άνω και κάτω όψη του Ασύρματου Χειριστηρίου σε λειτουργία



Άνω και κάτω όψη του Υποσυστήματος Δέκτη σε λειτουργία

Εικόνα 3.159 Άνω και κάτω όψη της ολοκληρωμένης κατασκευής του Σ.Η.Α.Ε., με εμφανή τις επιφάνειες διπλής όψης των 2 τυπωμένων κυκλωμάτων του

3.6 Συμπεράσματα

Κατά το παρελθόν, η Βιομηχανία αλλά και γενικότερα όλοι οι υπόλοιποι τομείς της τεχνολογίας ήταν αρκετά περιορισμένοι ως προς την ενσωμάτωση μορφών ασύρματης επικοινωνίας για εκτέλεση εφαρμογών, σε σχέση με την σημερινή εποχή.

Με πλεονεκτήματα όπως φορητότητα, παραγωγικότητα, ασφάλεια, ευχρηστία, χαμηλό κόστος και δυνατότητα κλιμάκωσης, τα συστήματα ασύρματης επικοινωνίας άρχισαν με σταθερό ρυθμό να αντικαθιστούν τις αντίστοιχες ενσύρματες επικοινωνίες σε διάφορες τεχνολογικές εφαρμογές και κυρίως στον βιομηχανικό τομέα. Επιπλέον, εκτός των πλεονεκτημάτων που προσδίδουν από πλευράς εφαρμογών, λόγω της ευελιξίας παροχής ελέγχου σε έναν χώρο, όλο και περισσότερες είναι οι εφαρμογές στις οποίες περιλαμβάνεται ασύρματη επικοινωνία.

Υπάρχουν βεβαίως και μειονεκτήματα από την χρήση ασύρματης επικοινωνίας, όπως η ενδεχόμενη απώλεια διαδρομής, η εμφάνιση παρεμβολών λόγω πολλαπλών διαδρομών ή ραδιοσημάτων, η απαίτηση για διαχείριση της ενέργειας λόγω των παρεμβολών αυτών, η ασυμβατότητα μεταξύ συστημάτων και η ασφάλεια ως προς την ορθή και ομαλή λειτουργία του εκάστοτε ασύρματου συστήματος. Τα μειονεκτήματα αυτά είναι ένας σημαντικός λόγος που δεν έχουν αντικατασταθεί όλες οι ενσύρματες τεχνολογίες με ασύρματες, μα καθώς η επιστημονική μελέτη και έρευνα γίνεται εκτενέστερη, συνέχεια θα συναντάται πρόοδος στον τομέα των βιομηχανικών εφαρμογών που ενσωματώνουν ασύρματη επικοινωνία.

Με την ολοκλήρωση της Διπλωματικής εργασίας, προκύπτει ένα Σύστημα Ηλεκτρονικού Ασύρματου Ελεγκτή (Σ.Η.Α.Ε.) που παρέχει δυνατότητα εκτέλεσης και ελέγχου εφαρμογών από απόσταση μέσω ασύρματης επικοινωνίας. Με αξιοποίησή του, ένας χρήστης δύναται να ελέγχει εφαρμογές τις οποίες σε διαφορετική περίπτωση θα ήταν πολύ δύσκολο και χρονοβόρο να εκτελέσει με επιτυχία, για πληθώρα λόγων και αφορμών όπως για παράδειγμα η δυσμενής πρόσβαση σε έναν περιβάλλοντα χώρο που όμως το Υποσύστημα Δέκτη του Σ.Η.Α.Ε. δεν θα διέτρεχε κανένα κίνδυνο να εκτελέσει την κατά τα άλλα επικίνδυνη για τον άνθρωπο διεργασία.

Επιπροσθέτως αυτής της δυναμικής, με περαιτέρω επεκτάσεις ή τροποποιήσεις σε περιφερειακές μονάδες και προγραμματιστικό κώδικα, το Σ.Η.Α.Ε. καθίσταται ικανό να καλύπτει πρόσθετες βιομηχανικές εφαρμογές πέραν των ήδη υπαρχόντων και άρα μπορεί να ανταποκρίνεται συνεχώς στις μεταβαλλόμενες ανάγκες της καθημερινότητας και τις νέες τάσεις της τεχνολογίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. "*Electronics*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Electronics/>

<https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE>

2. "*Electrical network*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Electrical_network/

https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%B9%CE%BA%CF%8C_%CE%BA%CF%8D%CE%BA%CE%BB%CF%89%CE%BC%CE%B1

3. "*Electronic component*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Electronic_component

4. "*Input device*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Input_device/

https://el.wikipedia.org/wiki/%CE%A3%CF%85%CF%83%CE%BA%CE%B5%CF%85%CE%AE_%CE%B5%CE%B9%CF%83%CF%8C%CE%B4%CE%BF%CF%85

5. "*Output device*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Output_device/

https://el.wikipedia.org/wiki/%CE%A3%CF%85%CF%83%CE%BA%CE%B5%CF%85%CE%AE_%CE%B5%CE%BE%CF%8C%CE%B4%CE%BF%CF%85

6. "*Input/output*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Input/output/>

https://el.wikipedia.org/wiki/%CE%A3%CF%85%CF%83%CE%BA%CE%B5%CF%85%CE%AE_%CE%B5%CE%B9%CF%83%CF%8C%CE%B4%CE%BF%CF%85-%CE%B5%CE%BE%CF%8C%CE%B4%CE%BF%CF%85

7. "*Peripheral*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Peripheral> /

https://el.wikipedia.org/wiki/%CE%A0%CE%B5%CF%81%CE%B9%CF%86%CE%B5%CF%81%CE%B5%CE%B9%CE%B1%CE%BA%CE%AE_%CF%83%CF%85%CF%83%CE%BA%CE%B5%CF%85%CE%AE

8. "*Integrated circuit*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Integrated_circuit /

https://el.wikipedia.org/wiki/%CE%9F%CE%BB%CE%BF%CE%BA%CE%BB%CE%B7%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF_%CE%BA%CF%8D%CE%BA%CE%BB%CF%89%CE%BC%CE%B1

9. "*Controller (computing)*", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Controller_\(computing\)](https://en.wikipedia.org/wiki/Controller_(computing))

10. "*ISM band*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/ISM_band

11. "*Baud*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Baud>

12. "*Modem*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Modem> /

<https://el.wikipedia.org/wiki/%CE%9C%CF%8C%CE%BD%CF%84%CE%B5%CE%BC>

13. "*Switch*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Switch> /

<https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%BA%CF%8C%CF%80%CF%84%CE%B7%CF%82>

14. "*Arduino.cc*", The Arduino.cc Official Site,

<https://www.arduino.cc/>

15. "*Official boards*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Arduino#Official_boards
16. "*Arduino UNO*", EPro Labs Home / ePro Labs WiKi,
https://wiki.eprolabs.com/index.php?title=Arduino_UNO
17. "*Arduino UNO*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Arduino>
18. "*Arduino Uno*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Arduino_Uno
19. "Programming language", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Programming_language/
https://el.wikipedia.org/wiki/%CE%93%CE%BB%CF%8E%CF%83%CF%83%CE%B1_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D
20. "Open source", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Open_source
21. "Open-source hardware", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Open-source_hardware/
https://el.wikipedia.org/wiki/%CE%A5%CE%BB%CE%B9%CF%83%CE%BC%CE%B9%CE%BA%CF%8C_%CE%B1%CE%BD%CE%BF%CE%B9%CE%BA%CF%84%CE%AE%CF%82_%CF%80%CE%B7%CE%B3%CE%AE%CF%82
22. "*The Serial Monitor*", adafruit.com,
<https://learn.adafruit.com/adafruit-arduino-lesson-5-the-serial-monitor/the-serial-monitor>
23. "*Arduino Lesson 5. The Serial Monitor*", adafruit.com,
<https://learn.adafruit.com/adafruit-arduino-lesson-5-the-serial-monitor>

24. "*HOW-TO Use the ARDUINO SERIAL MONITOR*", instructables.com,
<https://www.instructables.com/HOW-TO-use-the-ARDUINO-SERIAL-MONITOR/>
25. "How to use the Serial Monitor with the Arduino IDE 2.0", arduino.cc,
<https://www.arduino.cc/en/Tutorial/getting-started-with-ide-v2/ide-v2-serial-monitor>
26. "*Serial Monitor: Arduino Basics*", arduino.cc,
<https://create.arduino.cc/projecthub/glowascii/serial-monitor-arduino-basics-399eb6>
27. "*Arduino - Serial Monitor*", ARDUINO GET STARTED,
<https://arduinogetstarted.com/tutorials/arduino-serial-monitor>
28. "*Arduino Serial Monitor Tutorial: Basics and Alternatives to try*", seeedstudio.com,
<https://www.seeedstudio.com/blog/2019/12/16/how-to-use-arduino-serial-monitor-and-alternatives-to-try/>
29. "*Επικοινωνία με το Serial Monitor*", learn electronics.gr,
<https://learnelectronics.gr/επικοινωνία-με-το-serial-monitor/>
30. "*USE SERIAL.PRINT() TO DISPLAY ARDUINO OUTPUT ON YOUR COMPUTER MONITOR: PART 1*", PROGRAMMING ELECTRONICS ACADEMY,
<https://www.programmingelectronics.com/using-the-print-function-with-arduino-part-1/>
31. "*Chapter 4. Serial Communications*", O'REILLY,
<https://www.oreilly.com/library/view/arduino-cookbook/9781449399368/ch04.html>
32. "*AVR microcontrollers*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/AVR_microcontrollers
33. "*Atmel AVR instruction set*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Atmel_AVR_instruction_set

34. "*In-system programming*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/In-system_programming
35. "*Booting*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Booting#BOOT-LOADER>
36. "*Linear regulator*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Linear_regulator
37. "*Crystal oscillator*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Crystal_oscillator
38. "*Building an Arduino on a Breadboard*", The Arduino.cc Official Site,
<https://www.arduino.cc/en/main/standalone>
39. "*Downloads / Arduino IDE 1.8.13*", The Arduino.cc Official Site,
<https://www.arduino.cc/en/software>
40. "*Arduino Uno Tutorial [Pinout]*", diyIoT, Microcontroller Tutorials and Smart Home Projects, <https://diyIoT.com/arduino-uno-tutorial/>
41. "*Real Arduino UNO vs. Clone UNO R3 - Is the Genuine a Genu-Win??"*", YouTube^{GR},
<https://www.youtube.com/watch?v=874COrIDXcM>
42. "*Getting Started with Arduino Web Editor on Various Platforms*", Arduino Project Hub / Arduino SSO / create.arduino.cc / Arduino.cc,
https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a?f=1
43. "*How to Test Arduino Circuit and Code Online*", indestructibles circuits,
<https://www.instructables.com/How-to-Test-Arduino-Circuit-and-Code-Online/>

44. "*Arduino Products*", Arduino.cc,
<https://www.arduino.cc/en/Main/Products>
45. "*GRECO*" - Arduino Object Avoiding Robot for beginners, arduino.cc,
<https://www.ardumotive.com/greco-roboten.html>
46. "*An Overview of Arduino Nano Board*", EL-PRO-CUS / ELECTRONICS-PROJECTS-FOCUS,
<https://www.elprocus.com/an-overview-of-arduino-nano-board/>
47. "*Arduino Nano*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Arduino_Nano
48. "*Arduino Pro Mini*", EPro Labs Home / ePro Labs Wiki,
https://wiki.eprolabs.com/index.php?title=Arduino_Pro_Mini
49. "*Logic Level Converter*", EPro Labs Home / ePro Labs Wiki,
https://wiki.eprolabs.com/index.php?title=Logic_Level_Converter
50. "*Arduino Uno Datasheet*", Farnell – Electronic Component Distributors,
<http://www.farnell.com/datasheets/1682209.pdf>
51. "*Arduino barrel jack connector with batteries*", Google / Pictures,
https://www.google.com/search?q=Arduino+barrel+jack+connector+with+batteries&tbm=isch&ved=2ahUKEwi59f60msPtAhVNP-wKHcUFCRwQ2-cCegQIABAA&oq=Arduino+barrel+jack+connector+with+batteries&gs_lcp=CgNpbWcQA1CcQli_W2CaXGgAcAB4AIABjgGIAd4NkgEEMC4xNJgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=ifjRX_mMHc3-sAfFi6TgAQ&bih=1297&biw=2560
52. "*How Long Can An Arduino Run On Batteries*", YouTube,
<https://www.youtube.com/watch?v=5cYN5-Spnos>

53. “Αντάπτορας Arduino - Barrel Jack 5.5x2.1mm”, Google,
https://www.google.com/search?q=%CE%91%CE%BD%CF%84%CE%AC%CF%80%CF%84%CE%BF%CF%81%CE%B1%CF%82+%CE%B3%CE%B9%CE%B1+Arduino+-+9V+%CE%9C%CF%80%CE%B1%CF%84%CE%B1%CF%81%CE%AF%CE%B1+%CF%83%CE%B5+Jack+5.5x2.1mm&sxsrf=ALeKk02Hb52dy3XV2oPC1zLfvW93pwWKuA:1612529848721&source=lnms&tbn=isch&sa=X&ved=2ahUKEwjEgMrm5dLuAhUYNuWKHZjmDkoQ_AUoAXoECBAQAw&biw=1312&bih=1253

54. “Arduino Uno DIP Edition”, Polulu, Robotics & Electronics,
<https://www.pololu.com/product/1616>

55. “Dual in-line package”, Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Dual_in-line_package

56. “Through-hole technology”, Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Through-hole_technology

57. “Surface-mount technology, Wikipedia, the free encyclopedia”,
https://en.wikipedia.org/wiki/Surface-mount_technology

58. “What is a Surface Mount Device or SMD?”, EuroCircuits,
<https://www.eurocircuits.com/smd-surface-mount-device/>

59. “Chip carrier”, Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Chip_carrier

60. “Printed circuit board”, Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Printed_circuit_board/
https://el.wikipedia.org/wiki/%CE%A0%CE%BB%CE%B1%CE%BA%CE%AD%CF%84%CE%B1_%CF%84%CF%85%CF%80%CF%89%CE%BC%CE%AD%CE%BD%CE%BF%CF%85_%CE%BA%CF%85%CE%BA%CE%BB%CF%8E%CE%BC%CE%B1%CF%84%CE%BF%CF%82

61. "*Semiconductor package*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Semiconductor_package
62. "What is the difference between Arduino UNO SMD and the standard UNO", Arduino.cc,
<https://support.arduino.cc/hc/en-us/articles/360018161639-What-is-the-difference-between-Arduino-UNO-SMD-and-the-standard-UNO>
63. "What is the difference between an Arduino Uno and an Arduino Uno SMD?", Quora,
<https://www.quora.com/What-is-the-difference-between-an-Arduino-Uno-and-an-Arduino-Uno-SMD>
64. "*Telecommunication*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Telecommunication> /
<https://el.wikipedia.org/wiki/%CE%A4%CE%B7%CE%BB%CE%B5%CF%80%CE%B9%CE%BA%CE%BF%CE%B9%CE%BD%CF%89%CE%BD%CE%AF%CE%B5%CF%82>
65. "*Information*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Information> /
<https://el.wikipedia.org/wiki/%CE%A0%CE%BB%CE%B7%CF%81%CE%BF%CF%86%CE%BF%CF%81%CE%AF%CE%B1>
66. "*Wire*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Wire>
67. "*Radio*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Radio> /
<https://el.wikipedia.org/wiki/%CE%A1%CE%B1%CE%B4%CE%B9%CF%8C%CF%86%CF%89%CE%BD%CE%BF>
68. "*Radio wave*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio_wave /
<https://el.wikipedia.org/wiki/%CE%A1%CE%B1%CE%B4%CE%B9%CE%BF%CE%BA%CF%8D%CE%BC%CE%B1%CF%84%CE%B1>

69. "*Communication*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Communication/>

<https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B9%CE%BA%CE%BF%CE%B9%CE%BD%CF%89%CE%BD%CE%AF%CE%B1>

70. "*Radio frequency*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Radio_frequency/

<https://el.wikipedia.org/wiki/%CE%A1%CE%B1%CE%B4%CE%B9%CE%BF%CF%83%CF%85%CF%87%CE%BD%CF%8C%CF%84%CE%B7%CF%84%CE%B1>

71. "*Frequency modulation*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Frequency_modulation/

https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%BC%CF%8C%CF%81%CF%86%CF%89%CF%83%CE%B7_%CF%83%CF%85%CF%87%CE%BD%CF%8C%CF%84%CE%B7%CF%84%CE%B1%CF%82

72. "*Radio-frequency engineering*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Radio-frequency_engineering

73. "*Electric current*", Electric current / Radio frequency From Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Radio_frequency#Current

74. "*Radio resource management*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Radio_resource_management

75. "*ISM band*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/ISM_band

76. "*RF module*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/RF_module

77. "*Applications*", Applications / ISM band From Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/ISM_band#Applications

78. "*Frequency allocation*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Frequency_allocation

79. "*Fixed wireless*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Fixed_wireless

80. "*2.4 GHz radio use*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/2.4_GHz_radio_use

81. "*Bluetooth Low Energy*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Bluetooth_Low_Energy

82. "*Wi-Fi*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Wi-Fi> / <https://el.wikipedia.org/wiki/Wi-Fi>

83. "*Antenna (radio)*", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Antenna_\(radio\)](https://en.wikipedia.org/wiki/Antenna_(radio)) /

<https://el.wikipedia.org/wiki/%CE%9A%CE%B5%CF%81%CE%B1%CE%AF%CE%B1>

84. "*Tuner (radio)*", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Tuner_\(radio\)](https://en.wikipedia.org/wiki/Tuner_(radio))

85. "*Radar*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Radar> /

<https://el.wikipedia.org/wiki/%CE%A1%CE%B1%CE%BD%CF%84%CE%AC%CF%81>

86. "*Transmission line*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Transmission_line/

https://el.wikipedia.org/wiki/%CE%95%CE%BD%CF%83%CF%8D%CF%81%CE%BC%CE%B1%CF%84%CE%BF_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF

87. "*Amplifier*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Amplifier>

88. "*Modulation*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Modulation/>

https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%BC%CF%8C%CF%81%CF%86%CF%89%CF%83%CE%B7_%CF%83%CE%AE%CE%BC%CE%B1%CF%84%CE%BF%CF%82

89. "*RF power amplifier*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/RF_power_amplifier

90. "*Electromagnetic shielding*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Electromagnetic_shielding

91. "*RF and microwave filter*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/RF_and_microwave_filter

92. "*Broadcast engineering*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Broadcast_engineering

93. "*Serial port*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Serial_port

94. "*Transmitter*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Transmitter/>

<https://el.wikipedia.org/wiki/%CE%A0%CE%BF%CE%BC%CF%80%CF%8C%CF%82>

95. "Radio receiver", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio_receiver
96. "Wireless Product Portfolio", Nordic Semiconductor,
<https://www.nordicsemi.com/Products> / <https://www.nordicsemi.com/>
97. "NRF24L01 Wireless Transceiver Module", SunFounder,
http://wiki.sunfounder.cc/index.php?title=NRF24L01_Wireless_Transceiver_Module
98. "Nordic Semiconductor", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Nordic_Semiconductor
99. "SparkFun Electronics", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/SparkFun_Electronics
100. "NRF24L01 module", Google,
https://www.google.com/search?q=NRF24L01%2B+wikipedia&sxsrf=ALeKk02sqPqLYLr3aO7IK79Y7H82L6_cyg:1607542179773&source=lnms&tbn=isch&sa=X&ved=2ahUKEwiVs dik0cHtAhUKKBoKHcrXCgQQ_AUoAXoECAcQAw&biw=2560&bih=1240
101. "Nordic Semiconductor nRF24L01 Transceiver module",
<https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus Preliminary Product Specification v1.0.pdf>
102. "nRF24L01 datasheet/tutorial", DIYembedded,
http://www.diyembedded.com/tutorials/nrf24l01_0/nrf24l01_tutorial_0.pdf
103. "RFM73-D 2.4 GHz transceiver module", VOTI webshop catalogus,
<https://www.voti.nl/shop/catalog.html?HF-RFM73-D>
104. "nRF24L01+ Single Chip 2.4GHz Transceiver", Digi-Key,
<https://www.digikey.com/catalog/en/partgroup/nrf24l01/44008>

105. “*arduino/Arduino*”, GitHub,
<https://github.com/arduino/Arduino>
106. "nRF24L01 2.4GHz Wireless Transceiver Module", Grobotronics,
<https://grobotronics.com/nrf24l01-2.4ghz-wireless-transceiver-module.html>
107. “*nRF24L01 2.4GHz Wireless Transceiver Module with Antenna*”, Grobotronics,
<https://grobotronics.com/nrf24l01-2.4ghz-wireless-transceiver-module-with-antenna.html?sl=en>
108. “*How nRF24L01+ Wireless Module Works & Interface with Arduino*”,
Last Minute Engineers,
<https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>
109. "*Frequency-shift keying*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Frequency-shift_keying ,
https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%BC%CF%8C%CF%81%CF%86%CF%89%CF%83%CE%B7_%CE%BC%CE%B5%CF%84%CE%B1%CF%84%CF%8C%CF%80%CE%B9%CF%83%CE%B7%CF%82_%CF%83%CF%85%CF%87%CE%BD%CF%8C%CF%84%CE%B7%CF%84%CE%B1%CF%82
110. "*Gaussian frequency-shift keying*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Frequency-shift_keying#Gaussian_frequency-shift_keying
111. "*Gaussian filter*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Gaussian_filter
112. "*GFSK vs FSK*", Home of RF and Wireless Vendors and Resources –
One Stop For Your RF and Wireless Need,
<https://www.rfwireless-world.com/Terminology/GFSK-vs-FSK.html>
113. "*Level GFSK*", Fisher Telecommunication,
<https://www.fishercom.xyz/access-points/level-gfsk-1.html>

114. "Gaussian filter impulse response with BT of 0.5 and 0.3 respectively", ResearchGate, https://www.researchgate.net/figure/Gaussian-filter-impulse-response-with-BT-of-05-and-03-respectively_fig2_44139168

115. "Impulse response of a Gaussian filter, $d e f c g$ ", ResearchGate, https://www.researchgate.net/figure/Impulse-response-of-a-Gaussian-filter-d-e-f-c-g_fig5_37536492

116. "Impulse response and Frequency response of Gaussian filter $h(t)$, and Gaussian filtered frequency pulse $g(t)$ with $B = 0.5, 0.3, 0.25$ ", ResearchGate, https://www.researchgate.net/figure/Impulse-response-and-Frequency-response-of-Gaussian-filter-h-t-and-Gaussian-filtered_fig1_225356894

117. "Noise figure", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Noise_figure/
https://www.google.com/search?q=Noise+figure&sxsrf=ALeKk00-g8U3NqJU64fyWaR3ToGleufMkw%3A1619350493623&source=hp&ei=3VOFYJfVI42-UpS4pcgI&iflsig=AINFCbYAAAAAYIVh7cYxm7eNo3mRKntUILT6ZH1aOaoO&oq=Noise+figure&gs_lcp=Cgdnd3Mtd2l6EAMyBAgjECcyAgguMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCABoqNYogNgsgZoAHAAeACAAXqIAXqSAQMwLjGYAQCgAQkAQGqAQdnd3Mtd2l6&scient=gws-wiz&ved=0ahUKEwjXt7_XppnwAhUNnxQKHRRcCYkQ4dUDCAc&uact=5

118. "Voice over IP", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Voice_over_IP/
<https://el.wikipedia.org/wiki/VoIP>

119. "Static random-access memory", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Static_random-access_memory

120. "Wire Library", arduino.cc, <https://www.arduino.cc/en/Reference/Wire>

121. "Clock signal", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Clock_signal
122. "Pull-up resistor", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Pull-up_resistor
123. "Using the I2C Bus", ROBOT ELECTRONICS,
<https://www.robot-electronics.co.uk/i2c-tutorial>
124. "Potentiometer", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Potentiometer/>
<https://el.wikipedia.org/wiki/%CE%A0%CE%BF%CF%84%CE%B5%CE%BD%CF%83%CE%B9%CF%8C%CE%BC%CE%B5%CF%84%CF%81%CE%BF>
125. "Potentiometers - Technical Definitions", novotechnik,
<https://www.novotechnik.de/en/products/sensor-technologies/potentiometers/>
126. "Potentiometers LINEARITY TECHNICAL NOTE",
https://www.bourns.com/docs/technical-documents/technical-library/sensors-controls/technical-notes/Bourns_pot_linearity_technote.pdf
127. "Temperature coefficient", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Temperature_coefficient
128. "Analog 2-axis Thumb Joystick with Select Button + Breakout Board", adafruit.com,
<https://www.adafruit.com/product/512>
129. "Mini Analog Joystick - 10K Potentiometers", adafruit.com,
<https://www.adafruit.com/product/3102>
130. "JH-D202X-R2/R4 Joystick 5K", Proto Supplies,
<https://protosupplies.com/product/jh-d202x-r2-r4-joystick-5k/>

131. "*How 2-Axis Joystick Works & Interface with Arduino + Processing*",
Last Minute Engineers,
<https://lastminuteengineers.com/joystick-interfacing-arduino-processing/>

132. "*Soldering*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Soldering/>
<https://el.wikipedia.org/wiki/%CE%9A%CF%8C%CE%BB%CE%BB%CE%B7%CF%83%CE%B7>

133. "*Servomotor*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Servomotor>

134. "*Electric motor*," Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Electric_motor/
https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%B9%CE%BA%CF%8C%CF%82_%CE%BA%CE%B9%CE%BD%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82

135. "*Difference Between DC, Servo & Stepper Motors*", The PiHut –
The Ultimate Raspberry Pi & Maker Store,
<https://thepihut.com/blogs/raspberry-pi-roundup/whats-the-difference-between-dc-servo-amp-stepper-motors>

136. "*DC motor*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/DC_motor

137. "*Electric machine*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Electric_machine

138. "*Armature Controlled DC Motor*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Armature_Controlled_DC_Motor

139. "Armature (electrical)", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Armature_\(electrical\)](https://en.wikipedia.org/wiki/Armature_(electrical))

140. "Torque and speed of a DC motor", Torque and speed of a DC motor /

Brushed DC electric motor From Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Brushed_DC_electric_motor#Torque_and_speed_of_a_DC_motor

141. "Actuator", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Actuator>

142. "What is a Servo Motor – How a Servo Motor Works (Control)", Ettron,

<https://ettron.com/what-is-a-servo-motor-how-a-servo-motor-works-control/>

143. "Different Types of Servo Motor and Its Applications",

EDGEFX.US – KITS & SOLUTIONS,

<https://www.efxkits.us/different-types-servo-motor-applications/>

144. "Steering Gear Car Model System Kit with servomotor", Pinterest,

<https://id.pinterest.com/pin/725924033666248028/>

145. "Rudder", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Rudder>

146. "Aluminum Servo Arm", ESPRITtech,

<https://www.espritmodel.com/aluminum-servo-arm-half-v2-futaba.aspx>

147. "Servo Horn Arm Set", ANYTHINGHELI.COM,

<https://www.anythingheli.com/sab-helidivision-servo-horn-arm-set-bk-futaba-7pcs-goblin-380-420/>

148. "DOUBLE ARM SERVO HORN", XMODELIX,
<https://modelixracing.com/en/7969-SERVO-HORN-GAS-FRENO-ALUMINIO-FUTABA-POWER-HD-25T-1U-.html>

149. "Torque", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Torque/>
<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CF%80%CE%AE>

150. "International System of Units", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/International_System_of_Units/
<https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B5%CE%B8%CE%BD%CE%AD%CF%82%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1%CE%BC%CE%BF%CE%BD%CE%AC%CE%B4%CF%89%CE%BD>

151. "Newton-metre", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Newton-metre>

152. "Kilogram-force", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Kilogram-force>

153. "Pound (force)", Wikipedia, the free encyclopedia,
[https://en.wikipedia.org/wiki/Pound_\(force\)](https://en.wikipedia.org/wiki/Pound_(force))

154. "Torque tester", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Torque_tester

155. "SI derived unit", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/SI_derived_unit/
<https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CF%81%CE%AC%CE%B3%CF%89%CE%B3%CE%B5%CF%82%CE%BC%CE%BF%CE%BD%CE%AC%CE%B4%CE%B5%CF%82%CF%84%CE%BF%CF%85%SI>

156. "*Radio control*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio_control
157. "*Precision-guided munition*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Precision-guided_munition
158. "*Radio-controlled aircraft*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio-controlled_aircraft
159. "*Radio-controlled boat*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio-controlled_boat
160. "*Radio-controlled car*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio-controlled_car
161. "*Radio-controlled helicopter*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Radio-controlled_helicopter
162. "*Remote control*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Remote_control
163. "*Remote control vehicle*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Remote_control_vehicle
164. "*Telecommand*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Telecommand>
165. "*Teletank*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Teletank>
166. "*Telerobotics*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Telerobotics>

167. "*Teleoperation*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Teleoperation>

168. "*Remote manipulator*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Remote_manipulator

169. "*Remote administration*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Remote_administration

170. "*Remotely operated underwater vehicle*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Remotely_operated_underwater_vehicle

171. "*Industrial robot*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Industrial_robot/

https://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%BF%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AC_%CF%81%CE%BF%CE%BC%CF%80%CF%8C%CF%84

172. "*Unmanned aerial vehicle*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle/

https://el.wikipedia.org/wiki/%CE%9C%CE%B7_%CE%B5%CF%80%CE%B1%CE%BD%CE%B4%CF%81%CF%89%CE%BC%CE%AD%CE%BD%CE%BF_%CE%B1%CE%B5%CF%81%CE%BF%CF%83%CE%BA%CE%AC%CF%86%CE%BF%CF%82

173. "*Remote surgery*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Remote_surgery/

<https://el.wikipedia.org/wiki/%CE%A4%CE%B7%CE%BB%CE%B5%CF%87%CE%B5%CE%B9%CF%81%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CE%AE>

174. "*Goliath tracked mine*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Goliath_tracked_mine

- 175.** "*RC AIRPLANE*", The RC Servo,
<https://www.rc-airplane-world.com/rc-servo.html>
- 176.** "*RC plane servo setup*", The RC Servo,
<https://www.rc-airplane-world.com/rc-plane-servo-setup.html>
- 177.** "*Radio control gear explained*", The RC Servo,
<https://www.rc-airplane-world.com/radio-control-gear.html>
- 178.** "*Radio control functions*", The RC Servo,
<https://www.rc-airplane-world.com/radio-control-functions.html>
- 179.** "*Understanding RC Servo Torque Ratings*", rccarstars.com – The RC Service Pros,
<https://www.rccarstars.com/rc-maintenance-blog/understanding-rc-servo-torque-ratings>
- 180.** "*Using 16 Servos -- Question About Servo Torque*", adafruit,
<https://forums.adafruit.com/viewtopic.php?f=25&p=311947>
- 181.** "*RC Servos - The Muscles of Our Hobby*", RC HELICOPTER FUN,
<https://www.rchelicoptertfun.com/rc-servos.html>
- 182.** "*Servo Motor SG-90*", componenets101,
<https://components101.com/servo-motor-basics-pinout-datasheet>
- 183.** "*SG90 Analog*", TowerPro Online Shop,
<http://www.towerpro.com.tw/product/sg90-analog/>
- 184.** "*Servo Motor Introduction*", Electronic Wings,
<https://www.electronicwings.com/sensors-modules/servo-motor>

- 185.** “*Servo Motor Opening Window*”, Google,
https://www.google.com/search?q=servo+motor+opening+window&sxsrf=ALeKk00H0fwjHYdiAVRISGPU4yT0Qy78Kw:1610378420311&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjn7siNl5TuAhVI6aQKHVhfASwQ_AUoAXoECBQQAw
- 186.** "*Arduino Controlled Window for Cat*", instructables circuits,
<https://www.instructables.com/Arduino-controlled-window-for-cat/>
- 187.** "*Different servo motors (from 35g to 3kg.m)*", YouTube,
<https://www.youtube.com/watch?v=1wbi-L4epMY>
- 188.** "*Arduino - Servo.attach()*", ARDUINO GET STARTED,
<https://arduinogetstarted.com/reference/library/servo-attach>
- 189.** "*Controlling two Servo with same signal*", arduino.cc forum,
<https://forum.arduino.cc/t/controlling-two-servo-with-same-signal/396496>
- 190.** "*Multiple servos attached to one pin*", arduino.cc forum,
<https://forum.arduino.cc/t/multiple-servos-attached-to-one-pin/172499>
- 191.** "*Continuous Rotation Servo*", arduino.cc forum,
<https://forum.arduino.cc/t/continuous-rotation-servo/114459>
- 192.** "*Continuous rotation servo*", Creative Technologies in the Classroom (CTC),
<http://ctc-dev.verkstad.cc/en/course-literature/continuous-rotation-servo/>
- 193.** "*Two continuous servo with one axis of joystick and vary speed*", arduino.cc forum,
<https://forum.arduino.cc/t/two-continuous-servo-with-one-axis-of-joystick-and-vary-speed/330783>
- 194.** "*Powering Two Continuous Servos*", arduino.cc forum,
<https://forum.arduino.cc/t/powering-two-continuous-servos/292236>

- 195.** "*How to control continuous rotation servo motor (360°)?*", arduino.cc forum,
<https://forum.arduino.cc/t/how-to-control-continuous-rotation-servo-motor-360/610983>
- 196.** "*Arduino Servo Bot*", instructables circuits,
<https://www.instructables.com/Arduino-Servo-Bot/>
- 197.** "*Continuous servo speed control with Arduino Uno controller and servo shield*",
Stack Exchange Network,
<https://arduino.stackexchange.com/questions/70968/continuous-servo-speed-control-with-arduino-uno-controller-and-servo-shield>
- 198.** "*Continuous Rotation Servo Motor*", Informatics Diversity - Enhanced Workforce
(iDEW),
<https://docs.idew.org/internet-of-things-project/references-for-wiring-and-coding/continuous-rotation-servo-motor>
- 199.** "*map() function*", arduino.cc,
<https://www.arduino.cc/reference/en/language/functions/math/map/>
- 200.** "*Light-emitting diode*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Light-emitting_diode/
https://el.wikipedia.org/wiki/%CE%94%CE%AF%CE%BF%CE%B4%CE%BF%CF%82_%CE%B5%CE%BA%CF%80%CE%BF%CE%BC%CF%80%CE%AE%CF%82_%CF%86%CF%89%CF%84%CF%8C%CF%82
- 201.** "*Logic level*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Logic_level

- 202.** "*Καλώδιο με θύρα usb arduino*", Google,
https://www.google.com/search?q=%CE%BA%CE%B1%CE%BB%CF%89%CE%B4%CE%B9%CE%BF+%CE%BC%CE%B5+%CE%B8%CF%85%CF%81%CE%B1+usb+arduino&sxsrf=ALeKk01JF-xUIM1PSJYxCtIAcqwmu6rw_g:1608826883454&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjksZKXg-ftAhXD0eAKHZ3_CWEQ_AUoAXoECBIQAw&biw=1322&bih=1260
- 203.** "*USB Jack & Interface*", adafruit,
<https://learn.adafruit.com/ladyadas-learn-arduino-lesson-number-0/usb-jack-and-interface>
- 204.** "*Jump wire*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Jump_wire
- 205.** "*Corpco 4 x AA Battery Holder with Standard snap Connector 6V Output Type BH341 2 Pack*", amazon,
<https://www.amazon.com/Corpco-Battery-Holder-Standard-Connector/dp/B07R3C2PL5>
- 206.** "*Electrical wiring*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Electrical_wiring
- 207.** "*Crocodile clip*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Crocodile_clip
- 208.** "*Banana connector*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Banana_connector
- 209.** "*Registered jack*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Registered_jack
- 210.** "*RCA connector*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/RCA_connector

211. "*RF connector*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/RF_connector

212. "*arduino nano v3 pin header*", Google,

https://www.google.com/search?sxsrf=ALeKk03tK8fxittgaeA4z0J5sEHeK5qM-Q%3A1609782942781&ei=nlbzX5KRL8mbkwWZtKqADg&q=arduino+nano+v3+pin+header&oq=arduino+nano+v3+pin+header&gs_lcp=CgZwc3ktYWlQAzIICCEQFhAdEB46BAgAEEc6BwgjEOoCECc6CQgjEOoCECcQEzoECCMQJzoECAAQQzoCCAA6BggjECcQEzoHCAAQsQMQQzoFCAAQsQM6BQgAEMsBUMYJWMMoYPIpaAFwBHgAgAGKAYgBrAySAQQyLjEymAEAoAEB0AECqgEHZ3dzLXdperABCsgBAsABAQ&scient=psy-ab&ved=0ahUKEwiS2ODj7ILuAhXJzaQKHRmaCuAQ4dUDCA0&uact=5

213. "*Καλώδιο με θύρα micro usb arduino nano*", Google,

https://www.google.com/search?q=%CE%BA%CE%B1%CE%BB%CF%89%CE%B4%CE%B9%CE%BF+%CE%BC%CE%B5+%CE%B8%CF%85%CF%81%CE%B1+micro+usb+arduino+nano&tbm=isch&ved=2ahUKEwjP5YaYg-ftAhXPwYUKHfTWA_IQ2-cCegQIABAA&oq=%CE%BA%CE%B1%CE%BB%CF%89%CE%B4%CE%B9%CE%BF+%CE%BC%CE%B5+%CE%B8%CF%85%CF%81%CE%B1+micro+usb+arduino+nano&gs_lcp=CgNpbWcQA1DS2wJYrvECYN_3AmgAcAB4AIABbIgb3wiSAQMzLjiYAQCgAQGqAQotnd3Mtd2l6LWltZ8ABAQ&scient=img&ei=BcDkX4_PFc-DlwT04Y-QDw&bih=1260&biw=1322

214. "*Pin header*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Pin_header

215. "*DC connector*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/DC_connector

216. "*Insulation-displacement connector*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Insulation-displacement_connector

217. "*Breadboard*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Breadboard>

218. "What is a Servo", SERVOCITY,

<https://www.servocity.com/servo-faqs/>

219. "ΕΞΑΡΤΗΜΑΤΑ ΜΗΧΑΝΙΣΜΟΥ καθετων περσιδων", Google,

https://www.google.com/search?q=%CE%95%CE%9E%CE%91%CE%A1%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91+%CE%9C%CE%97%CE%A7%CE%91%CE%9D%CE%99%CE%A3%CE%9C%CE%9F%CE%A5+%CE%BA%CE%B1%CE%B8%CE%B5%CF%84%CF%89%CE%BD+%CF%80%CE%B5%CF%81%CF%83%CE%B9%CE%B4%CF%89%CE%BD&sxsrf=ALeKk01eWNOEXDFTQb9XK4Vn7Fe9BWuFgw:1611321620488&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjp9brm0K_uAhUK-aQKHWD6AygQ_AUoAXoECAgQAw&biw=1312&bih=1253#imgrc=uYaf73O7YjU25M

220. "Ultrasonic Module", SunFounder,

http://wiki.sunfounder.cc/index.php?title=Ultrasonic_Module

221. "Ultrasonic Sensor (HC-SR04) + Arduino", Washington University in St. Louis / ESE205 Wiki,

[https://classes.engineering.wustl.edu/ese205/core/index.php?title=Ultrasonic_Sensor_\(HC-SR04\)_%2B_Arduino](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Ultrasonic_Sensor_(HC-SR04)_%2B_Arduino)

222. "Ultrasonic Ranging Module HC-SR04", itead.cc,

https://www.itead.cc/wiki/Ultrasonic_Ranging_Module_HC-SR04

223. "Buzzer", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Buzzer>

224. "Piezoelectric speaker", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Piezoelectric_speaker

225. "Piezoelectricity", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Piezoelectricity>

226. "*Piezoelectricity*", explainthatstuff.com,

<https://www.explainthatstuff.com/piezoelectricity.html>

227. "*BUZZER MODULE*", arduino.cc,

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45?fbclid=IwAR0WuvtXs4B30r67-sOWZHxbdKwcFnoTETMgb5WySqzzlcAPKwJARSwNz7E>

228. "*USE TONE() WITH ARDUINO FOR AN EASY WAY TO MAKE NOISE*",

programmingelectronics,

https://www.programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/?fbclid=IwAR0OA1Uo1FnpX6D_8_cpq0Etvo90f7tdVbDLjuQ8Ax57YUYFFokRQp8wszI

229. "*Play a Melody using the tone() function*", arduino.cc,

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody>

230. "*Arduino: Piezo speakers (buzzers)*", shallowsky,

https://shallowsky.com/arduino/class/buzzer.html?fbclid=IwAR22fA4ZR-hpxtB1-uUCXiKzt-ZP4Cw6GZGvcM9kGdmiUKFEd7E_jkqmem8

231. "*Alarm device*" Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Alarm_device

232. "*Alarm clock*" Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Alarm_clock/

<https://el.wikipedia.org/wiki/%CE%9E%CF%85%CF%80%CE%BD%CE%B7%CF%84%CE%AE%CF%81%CE%B9>

233. "*Klaxon*", Vehicle horn / Klaxon, Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Vehicle_horn#Klaxon

234. "*Security alarm*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Security_alarm

235. "*Distributed control system*" Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Distributed_control_system / <https://el.wikipedia.org/wiki/DCS>

236. "*Civil defense siren*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Civil_defense_siren

237. "*Warning system*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Warning_system

238. "*Sensor*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Sensor> /

<https://el.wikipedia.org/wiki/%CE%91%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82>

239. "*List of sensors*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/List_of_sensors

240. "*Instrumentation*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Instrumentation>

241. "*tone() function*", Arduino.cc,

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

242. "*PWM*", Arduino.cc,

<https://www.arduino.cc/en/Tutorial/Foundations/PWM>

243. "*LED circuit*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/LED_circuit /

https://el.wikipedia.org/wiki/%CE%9A%CF%8D%CE%BA%CE%BB%CF%89%CE%BC%CE%B1_LED

244. "LED Current Limiting Resistors",
[sparkfun, https://www.sparkfun.com/tutorials/219](https://www.sparkfun.com/tutorials/219)
245. "Gyroscope", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Gyroscope/>
<https://el.wikipedia.org/wiki/%CE%93%CF%85%CF%81%CE%BF%CF%83%CE%BA%CF%8C%CF%80%CE%B9%CE%BF>
246. "Gyroscope", Britannica,
<https://www.britannica.com/technology/gyroscope>
247. "Gyroscope Technology and Applications: A Review in the Industrial Perspective",
NCBI, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5677445/>
248. "Gyro sensors - How they work and what's ahead", EPSON,
https://www5.epsondevice.com/en/information/technical_info/gyro/
249. "Advances in optical gyroscopes", SPIE.DIFITAL LIBRARY,
<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11199/111990T/Advances-in-optical-gyroscopes/10.1117/12.2542713.short?SSO=1>
250. "Accelerometer", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Accelerometer>
251. "How to Measure Acceleration?", OMEGA Engineering,
<https://www.omega.com/en-us/resources/accelerometers>
252. "Proper acceleration", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Proper_acceleration
253. "Free fall", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Free_fall

- 254.** *"Everything You Ever Wanted to Know About Gyroscopes & Accelerometers"*, INTERESTING ENGINEERING,
<https://interestingengineering.com/what-gyroscopes-are-how-they-work-and-their-importance>
- 255.** *"Accelerometer vs. Gyroscope: What's the Difference?"*, LIVESCIENCE,
<https://www.livescience.com/40103-accelerometer-vs-gyroscope.html>
- 256.** *"MPU-6050 3 Axis Gyroscope + Accelerometer Module for Arduino GY-521 (MPU6050)"*, CABLEWORKS,
<https://www.cableworks.gr/elektronika/arduino-and-microcontrollers/mcu-and-components/tilt/mpu-6050-3-axis-gyroscope-accelerometer-module-for-arduino-gy-521/>
- 257.** *"Activity Recognition with Smartphone Sensors"*, ResearchGate,
https://www.researchgate.net/publication/263285563_Activity_Recognition_with_Smartphone_Sensors
- 258.** *"Activity recognition with smartphone sensors"*, IEEE Xplore,
<https://ieeexplore.ieee.org/document/6838194>
- 259.** *"Gyroscopic Instruments - Gyroscopes Theory"*, DUTCHOPS.COM,
www.dutchops.com/portfolio_marcel/articles/instruments/gyroscopic_instruments/theory_gyroscopes.htm
- 260.** *"gyroscopic systems and instruments"*, PILOTFRIEND,
www.pilotfriend.com/training/flight_training/fxd_wing/gyro.htm
- 261.** *"Apply Sensor Fusion to Accelerometers and Gyroscopes"*, Digi-Key ELECTRONICS,
<https://www.digikey.com/en/articles/apply-sensor-fusion-to-accelerometers-and-gyroscopes>
- 262.** *"MPU6050 - Accelerometer and Gyroscope Module"*, COMPONENTS 101,
<https://components101.com/sensors/mpu6050-module>

- 263.** "ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ – ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ ΚΑΙ ΣΕ ΤΙ ΧΡΗΣΙΜΕΥΕΙ;",
TME ELECTRONIC COMPONENTS,
<https://www.tme.eu/gr/news/library-articles/page/22568/epitakhunsiometro-pos-leitourgei-kai-se-ti-khresimeuei/#Co-to-jest-akcelerometr>
- 264.** "MPU6050 (Gyroscope + Accelerometer + Temperature) Sensor Module",
ElectronicWings,
<https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>
- 265.** "MPU-6000 and MPU-6050 Product Specification Revision 3.4", INVENSENSE,
<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- 266.** "How to Interface Arduino and the MPU 6050 Sensor", MAKER.PRO,
<https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor>
- 267.** "MPU6050 3 Axis Gyroscope Accelerometer Sensor Module (3V-5V Compatible)",
ICs & Robot Gadgets,
www.icstation.com/mpu6050-axis-gyroscope-accelerometer-sensor-module-compatible-p-1502.html
- 268.** "MEMS Accelerometer Gyroscope Magnetometer & Arduino",
HOW TO MECHATRONICS,
<https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/>
- 269.** "MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2",
InvenSense,
<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>
- 270.** "Tilt Sensing Using a Three-Axis Accelerometer", NXP,
https://www.nxp.com/files-static/sensors/doc/app_note/AN3461.pdf

271. "*Gravitational acceleration*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Gravitational_acceleration/

https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B9%CF%84%CE%AC%CF%87%CF%85%CE%BD%CF%83%CE%B7_%CF%84%CE%B7%CF%82_%CE%B2%CE%B1%CF%81%CF%8D%CF%84%CE%B7%CF%84%CE%B1%CF%82

272. "*Acceleration*", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Acceleration/>

<https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B9%CF%84%CE%AC%CF%87%CF%85%CE%BD%CF%83%CE%B7>

273. "*Centrifugal force*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Centrifugal_force/

https://el.wikipedia.org/wiki/%CE%A6%CF%85%CE%B3%CF%8C%CE%BA%CE%B5%CE%BD%CF%84%CF%81%CE%BF%CF%82_%CE%B4%CF%8D%CE%BD%CE%B1%CE%BC%CE%B7

274. "*Standard gravity*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Standard_gravity/

https://el.wikipedia.org/wiki/%CE%A4%CF%85%CF%80%CE%B9%CE%BA%CE%AE_%CE%B2%CE%B1%CF%81%CF%8D%CF%84%CE%B7%CF%84%CE%B1

275. "*Centrifugal force*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Centrifugal_force

276. "*Spin (physics)*", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Spin_\(physics\)/](https://en.wikipedia.org/wiki/Spin_(physics)/)

<https://el.wikipedia.org/wiki/%CE%A3%CF%80%CE%B9%CE%BD>

277. "*Yaw (rotation)*", Wikipedia, the free encyclopedia,

[https://en.wikipedia.org/wiki/Yaw_\(rotation\)](https://en.wikipedia.org/wiki/Yaw_(rotation))

- 278.** "*Yaw-rate sensor*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Yaw-rate_sensor
- 279.** "*Center of gravity*", Center of mass / Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Center_of_mass#Center_of_gravity
- 280.** "*Directional stability*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Directional_stability
- 281.** "*Coriolis force*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Coriolis_force/
<https://el.wikipedia.org/wiki/%CE%94%CF%8D%CE%BD%CE%B1%CE%BC%CE%B7%CE%9A%CE%BF%CF%81%CE%B9%CF%8C%CE%BB%CE%B9%CF%82>
- 282.** "*Six degrees of freedom*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Six_degrees_of_freedom
- 283.** "*Adverse yaw*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Adverse_yaw
- 284.** "*Aircraft principal axes*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Aircraft_principal_axes
- 285.** "*Pitch, yaw, and roll*", Simple English Wikipedia, the free encyclopedia,
https://simple.wikipedia.org/wiki/Pitch,_yaw,_and_roll
- 286.** "*Yaw, pitch, and roll rotations*", PLANNING ALGORITHMS,
<http://planning.cs.uiuc.edu/node102.html>
- 287.** "*Roll, Pitch, and Yaw*", HOW THINGS FLY,
<https://howthingsfly.si.edu/flight-dynamics/roll-pitch-and-yaw>

288. "*Aircraft Rotations - Body Axes*",
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION,
<https://www.grc.nasa.gov/www/k-12/airplane/rotations.html>

289. "*Aircraft Yaw Motion*",
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION,
<https://www.grc.nasa.gov/www/k-12/airplane/yaw.html>

290. "*Rotation*", Wikipedia, the free encyclopedia,
<https://en.wikipedia.org/wiki/Rotation/>
<https://el.wikipedia.org/wiki/%CE%A0%CE%B5%CF%81%CE%B9%CF%83%CF%84%CF%81%CE%BF%CF%86%CE%AE>

291. "*Plane of rotation*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Plane_of_rotation

292. "*Rotation (mathematics)*", Wikipedia, the free encyclopedia,
[https://en.wikipedia.org/wiki/Rotation_\(mathematics\)](https://en.wikipedia.org/wiki/Rotation_(mathematics))

293. "*Three-dimensional space*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Three-dimensional_space/
https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%B9%CF%83%CE%B4%CE%B9%CE%AC%CF%83%CF%84%CE%B1%CF%84%CE%BF%CF%82_%CF%87%CF%8E%CF%81%CE%BF%CF%82

294. "*Rotation around a fixed axis*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Rotation_around_a_fixed_axis

295. "*Instant centre of rotation*", Wikipedia, the free encyclopedia,
https://en.wikipedia.org/wiki/Instant_centre_of_rotation

296. "Velocity", Wikipedia, the free encyclopedia,

<https://en.wikipedia.org/wiki/Velocity/>

<https://el.wikipedia.org/wiki/%CE%A4%CE%B1%CF%87%CF%8D%CF%84%CE%B7%CF%84%CE%B1>

297. "Angular velocity", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Angular_velocity/

https://el.wikipedia.org/wiki/%CE%93%CF%89%CE%BD%CE%B9%CE%B1%CE%BA%CE%AE_%CF%84%CE%B1%CF%87%CF%8D%CF%84%CE%B7%CF%84%CE%B1

298. "Circular motion", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Circular_motion/

https://el.wikipedia.org/wiki/%CE%9A%CF%85%CE%BA%CE%BB%CE%B9%CE%BA%CE%AE_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7

299. "Two's complement", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Two%27s_complement/

https://el.wikipedia.org/wiki/%CE%A3%CF%85%CE%BC%CF%80%CE%BB%CE%AE%CF%81%CF%89%CE%BC%CE%B1_%CF%89%CF%82_%CF%80%CF%81%CE%BF%CF%82_2

300. "Understanding and Converting 2's Complement into Decimal for ITG3205 Gyroscope",
arduino.cc,

<https://forum.arduino.cc/t/understanding-and-converting-2s-complement-into-decimal-for-itg3205-gyroscope/252408>

301. "Two's Complement", tutorials point,

<https://www.tutorialspoint.com/two-s-complement>

302. "Two's Complement", CS.CORNELL.EDU,

<https://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html>

303. "*MPU 6050 Gyro, Accelerometer Communication With Arduino*",
instructables circuits,

<https://www.instructables.com/Accelerometer-MPU-6050-Communication-With-AVR-MCU/>

304. "*Raw data*", Wikipedia, the free encyclopedia,

https://en.wikipedia.org/wiki/Raw_data

