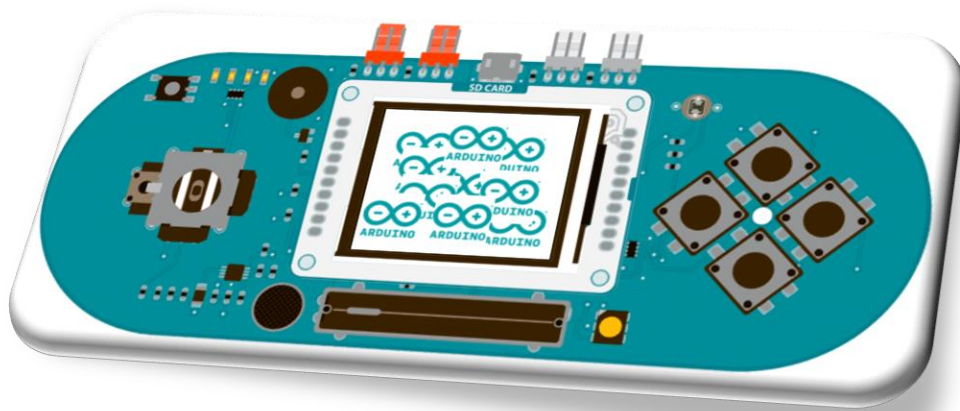


ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΑΣΥΡΜΑΤΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΣ ΣΤΑΘΜΟΣ ΜΕ
ARDUINO**



ΡΑΦΑΗΛ ΜΙΧΑΛΑΚΑΚΗΣ

A.M.:11211

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΒΑΣΙΛΕΙΟΣ ΡΑΠΤΗΣ

ΑΡΤΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2018



ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΑΣΥΡΜΑΤΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΣ ΣΤΑΘΜΟΣ ΜΕ
ARDUINO**

ΡΑΦΑΗΛ ΜΙΧΑΛΑΚΑΚΗΣ

A.M.:11211

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΒΑΣΙΛΕΙΟΣ ΡΑΠΤΗΣ

ΑΡΤΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2018

WIRELESS METEOROLOGICAL STATION WITH ARDUINO

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα 28/09/2018

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. ΚΑΘΗΓΗΤΗΣ

Ράπτης Βασίλειος Ακ. Υπότροφος

2. ΚΑΘΗΓΗΤΗΣ

Στεργίου Ελευθέριος Αν. Καθηγητής

3. ΚΑΘΗΓΗΤΗΣ

Χρονόπουλος Σπυρίδων Ακ. Υπότροφος

Ο Προϊστάμενος του τμήματος

Καθηγητής

Στύλιος Χρυσόστομος Καθηγητής

© Μιχαλακάκης, Ραφαήλ, 2018.

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 πτυχιακή εργασία είναι εξ 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.

ΡΑΦΑΗΛ ΜΙΧΑΛΑΚΑΚΗΣ

ΥΠΟΓΡΑΦΗ

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Βασίλειο Ράπτη, που μου έδωσε την δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα και για την υποστήριξη, την καθοδήγηση και τις πολύτιμες συμβουλές του καθ' όλη την διάρκεια διεκπεραίωσης της πτυχιακής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω όλους του καθηγητές του τμήματος μηχανικών πληροφορικής για τις πολύτιμες γνώσεις που μου έδωσαν όλα αυτά τα χρόνια.

Τέλος, ευχαριστώ τους γονείς μου, τον αδερφό μου, τους συναδέλφους μου και τους φίλους μου που με στήριξαν με κάθε τρόπο σε όλη την διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής αυτής είναι η δημιουργία μιας ασύρματης επικοινωνίας με χρήση μικροελεγκτών όπου θα μπορεί να γίνεται η ανάγνωση της θερμοκρασίας και υγρασίας στον χώρο που είμαστε μέσα από μία οθόνη. Οι μικροελεγκτές με τους οποίους θα υλοποιηθεί η ασύρματη επικοινωνία είναι οι Arduino. Η λήψη και η μετάδοση των δεδομένων θερμοκρασίας θα γίνει μέσω ασύρματης ζεύξης. Οι Arduino του πομπού και του δέκτη είναι προγραμματισμένοι για την αποστολή, λήψη και την ανίχνευση τυχών απώλειας της σύνδεσης. Η υλοποίηση θα γίνει στο Arduino IDE και η γλώσσα προγραμματισμού του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++. Επίσης χρησιμοποιήθηκε και το πρόγραμμα Fritzing μέσα στο οποίο έγιναν αναπαραστάσεις των υλικών που χρησιμοποιήθηκαν.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Μικροελεγκτές, Arduino, αποστολή, λήψη, ασύρματη επικοινωνία, αισθητήρας, θερμοκρασία.

ABSTRACT

The purpose of this diploma thesis is to create a wireless communication system using microcontrollers where it is possible to read the temperature and humidity in the place through a screen. The microcontrollers use for this will be Arduino. We will have a transmitter and a receiver in place where we will be able to send and receive data. Arduino's transmitter and receiver are programmed to send, receive, detect and alert any errors. The implementation will be done on the Arduino IDE and the Arduino programming language is based on the Wiring language, a C / C ++ variant. Also used was the Fritzing program in which representations of the materials used were made.

KEYWORDS

Microcontrollers, Arduino, sending, receiving, wireless communication, sensor, temperature.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευχαριστίες.....	8
Περίληψη.....	9
Abstract.....	10
Πίνακας περιεχομένων.....	11
Κατάλογος πινάκων.....	13
Κατάλογος εικόνων.....	14
Κατάλογος Σχεδιαγραμμάτων.....	15
Εισαγωγή.....	17
 Κεφάλαιο 1 – Μικροελεγκτές.....	 17
1.1 Τι είναι μικροελεγκτές.....	17
1.2 Τι είναι Arduino.....	19
1.3 Ιστορικά στοιχεία Arduino.....	20
1.4 Μοντέλα πλακετών Arduino.....	21
1.5 Shields Arduino.....	24
1.6 Σύγκριση μεταξύ των μοντέλων Arduino.....	27
1.7 Εκδόσεις Arduino.....	28
1.8 Πλεονεκτήματα Arduino.....	28
 Κεφάλαιο 2 – Οι πλακέτες Arduino Mega και Nano.....	 29
2.1 Χαρακτηριστικά Arduino Mega.....	29
2.1.1 Τροφοδοσία Arduino Mega.....	30
2.1.2 Ακροδέκτες Arduino Mega.....	31
2.1.3 Μνήμες Arduino Mega.....	33
2.2 Χαρακτηριστικά Arduino Nano.....	33
2.2.1 Τροφοδοσία Arduino Nano.....	34
2.2.2 Ακροδέκτες Arduino Nano.....	34

2.2.3 Μνήμες Arduino Nano.....	35
Κεφάλαιο 3 – Περιβάλλον ανάπτυξης του Arduino.....	36
3.1 Λογισμικό Arduino IDE.....	36
3.2 Εγκατάσταση Arduino IDE.....	36
3.3 Έλεγχος προγράμματος / Compiler.....	41
3.4 Το GUI του Arduino IDE.....	42
3.5 Βασικές εντολές και δομή του προγράμματος.....	45
3.6 Συναρτήσεις ακροδεκτών.....	51
3.6.1 Συναρτήσεις αναλογικών ακροδεκτών.....	51
3.6.2 Συναρτήσεις ψηφιακών ακροδεκτών.....	53
Κεφάλαιο 4 – Υλοποίηση της εργασίας.....	55
4.1 Λειτουργία του συστήματος.....	55
4.2 Υλικά.....	56
4.2.1 Αισθητήρας DHT22.....	56
4.2.2 Ρολόι χρονισμού DS3231 RTC.....	58
4.2.3 NRF24L01+ Long Range Wireless.....	59
4.2.4 Οθόνη TFT_320QDT_9341.....	62
4.2.5 TFT LCD Shield για το Arduino Mega 2560.....	66
4.3 Συνδεσμολογία στο πρόγραμμα Fritzing.....	67
Κεφάλαιο 5 – Κώδικας υλοποίησης της εργασίας.....	70
5.1 Κώδικας Arduino Mega 2560.....	70
5.2 Κώδικας Arduino Nano.....	75
Κεφάλαιο 6 – Συμπεράσματα.....	77
Βιβλιογραφία.....	78

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1.1 – Μοντέλα Arduino.	21
1.2 – Σύγκριση διαφορετικών τύπων Arduino A'.	27
1.3 – Σύγκριση διαφορετικών τύπων Arduino B'.	27
4.1 – Τεχνικά χαρακτηριστικά αισθητήρα DHT22.	57
4.2 – Τεχνικά χαρακτηριστικά NRF24L01+.	60

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

1.1 – Λογότυπο Arduino.	19
1.2 – Arduino WiFi Shield.	24
1.3 – Arduino Ethernet Shield.	24
1.4 – Arduino Wireless SD Shield.	25
1.5 – Arduino Wireless Proto Shield.	25
1.6 – Arduino Motor Shields.	26
1.7 – Arduino GSM Shield 2.	26
1.8 – Arduino MKR CAN Shield.	26
2.1 – Arduino Mega 2560 Rev3.	29
2.2 – Arduino Nano.	33
3.1 Το GUI του Arduino IDE.	42
4.1 Αισθητήρας DHT22.	55
4.2 Ρολόι χρονισμού RTC DS-3241.	56
4.3 NRF24L01+ Long Range Wireless.	58
4.4 TFT_320QDT_9341.	59
4.5 TFT LCD Shield για το Arduino Mega 2560.	62
5.1 Το αποτέλεσμα της λειτουργίας του συστήματος.	66

ΚΑΤΑΛΟΓΟΣ ΣΧΕΔΙΑΓΡΑΜΜΑΤΩΝ

3.1 – Αναλογικοί ακροδέκτες Arduino Mega 2560.	52
3.2 – Ψηφιακοί ακροδέκτες Arduino Mega 2560.	53
4.1 – Συνδεσμολογία αισθητήρα DHT22 με την πλακέτα Nano.	58
4.2 – Συνδεσμολογία με την πλακέτα.	59
4.3 – Συνδεσμολογία με την πλακέτα Arduino Nano.	61
4.4 – Συνδεσμολογία με την πλακέτα Arduino Mega.	61
4.5 – Συνδεσμολογία οθόνης – shield – Arduino Mega.	67
4.6 – Συνδεσμολογία Arduino Mega 2560	68
4.7 – Συνδεσμολογία Arduino Nano.	69

Εισαγωγή

Απαραίτητοι για την αποστολή, την λήψη, την μετάδοση και την εμφάνιση των δεδομένων στην οθόνη που θα χρησιμοποιηθεί για την εμφάνιση της θερμοκρασίας από απομακρυσμένο σημείο είναι οι μικροελεγκτές. Σε αυτό το κεφάλαιο θα συζητηθεί με αναφορά στις υπάρχουσες τεχνολογίες οι διαθέσιμοι στην αγορά μικροελεγκτές που είναι ενσωματωμένοι σε πλακέτες για την πλήρη λειτουργία τους ως ανεξάρτητα συστήματα.

Κεφάλαιο 1

1.1 Τι είναι μικροελεγκτές

Ένας μικροελεγκτής είναι ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα το οποίο διαθέτει επεξεργαστή, μνήμη, διάφορα περιφερειακά κυκλώματα καθώς επίσης και θύρες εισόδου/εξόδου για επικοινωνία με εξωτερικές συσκευές. Μερικές από τις εφαρμογές των μικροελεγκτών στον βιομηχανικό κυρίως τομέα είναι σε αυτόματο έλεγχο συσκευών, όπως συστήματα ελέγχου κινητήρα αυτοκινήτων, τηλεχειρισμούς και άλλα. Η μείωση του κόστους αλλά και του μεγέθους τους, καθιστά εφικτό τον οικονομικότερο ψηφιακό έλεγχο πολλών συσκευών και εφαρμογών. Ιδιαίτερα εύχρηστοι έχουν γίνει οι μικροελεγκτές μικτού σήματος, οι οποίοι έχουν ενσωματωμένα και αναλογικά εξαρτήματα επάνω τους για τη σύνδεση και σε μη ψηφιακές συσκευές. Για εξοικονόμηση ενέργειας και χαμηλής κατανάλωσης, ορισμένοι μικροελεγκτές χρησιμοποιούν τεσσάρων bits (4 bits) λέξεις και λειτουργούν σε χαμηλές συχνότητες, όπως τα 4 kHz. Με αυτόν τον τρόπο, είναι δυνατή η χρήση τους σε περιπτώσεις όπου είναι απαραίτητο να βρίσκονται σε κατάσταση αναμονής και να ενεργοποιηθούν μετά από κάποιο συμβάν, όπως ένας διακόπτης. Έτσι, η κατανάλωση ενέργειας περιορίζεται στην τάξη των nanowatts, με αποτέλεσμα να επεκτείνεται έτσι σημαντικά η διάρκεια ζωής τους όταν η τροφοδοσία τους γίνεται μέσω μπαταρίας. Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους χρησιμοποιείται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (π.χ. οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου Von Neumann.

Κάποιες από τις γνωστότερες εταιρίες που κατασκευάζουν μικροελεγκτές είναι οι εξής:

- ❖ ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα)

- ❖ Atmel (εξαγοράστηκε από την Microchip)
- ❖ Epson
- ❖ Freescale Semiconductor (πρώην Motorola)
- ❖ Hitachi
- ❖ Maxim (μετά την εξαγορά της Dallas)
- ❖ Microchip
- ❖ NEC
- ❖ Toshiba
- ❖ Texas Instruments

Τα πλεονεκτήματα των μικροελεγκτών είναι:

- Αυτονομία.
- Η ενσωμάτωση περιφερειακών: Γίνεται μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία: Λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές: Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος): Λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Ενώ κάποια από τα μειονεκτήματα είναι η δυσκολία του προγραμματισμού του γιατί δεν γίνεται η αλλαγή του προγράμματος διότι αυτό είναι γραμμένο στην ROM.[2][17][4]

1.2 Τι είναι Arduino

Το Arduino είναι μία πλακέτα ανοικτού κώδικα με την οποία κάποιος χωρίς ιδιαίτερες γνώσεις προγραμματισμού μπορεί εύκολα να την χρησιμοποιήσει και να κατασκευάσει εφαρμογές ρομποτικής και συστήματα αυτοματισμού.

Το Arduino αποτελείται από έναν μικροεπεξεργαστή, τον ATmega της Atmel και έχει την δυνατότητα να δεχθεί μονάδες εισόδου / εξόδου. Οι μονάδες εισόδου / εξόδου χωρίζονται σε Ψηφιακές και Αναλογικές. Υπάρχει μεγάλη πληθώρα συσκευών, συμβατές με τις πλακέτες Arduino. Κάποιες από αυτές είναι, οι αισθητήρες θερμοκρασίας, υγρασίας, δύναμης, απόστασης, γυροσκόπια, αισθητήρες επιτάχυνσης, πίεσης και άλλοι. Επίσης με το Arduino μπορούμε να ελέγξουμε κινητήρες ακόμη και leds, φώτα (220v) και ότι άλλο μπορούμε να σκεφτούμε ώστε να δημιουργήσουμε το δικό μας project.

Το Arduino μπορούμε να το προγραμματίσουμε από τον υπολογιστή μας μέσω της σειριακής θύρας που υποστηρίζει ο μικροεπεξεργαστής ATmega από την θύρα USB του υπολογιστή. Η σειριακή αυτή σύνδεση (Serial over Usb) χρησιμοποιείται για την μεταφορά προγραμμάτων από τον υπολογιστή προς την πλακέτα Arduino αλλά και το αντίστροφο για την μεταφορά των δεδομένων, που λαμβάνει το Arduino από της συσκευές προς τον υπολογιστή. Το λογισμικό λειτουργίας του Arduino προσφέρεται δωρεάν από την εταιρία μέσω της ιστοσελίδας της.

Το Arduino έλαβε μία τιμητική αναφορά το 2006 στο ψηφιακό τμήμα κοινοτήτων στην κατηγορία ARS Electronica Prix.[5]



Εικόνα 1.1: Επίσημο λογότυπο Arduino

1.3 Ιστορικά στοιχεία

Το 2005, στην Ivrea της Ιταλίας άρχισε να δημιουργείται ένα έργο, μια συσκευή για τον έλεγχο των φοιτητικών διαδραστικών σχεδίων σχεδιασμού, που ήταν λιγότερο δαπανηρή από τα άλλα συστήματα που ήταν διαθέσιμα εκείνη τη στιγμή. Εφευρέτες είναι ο Massimo Banzi και ο David Cuartielles, ο πρώτος ονόμασε αυτό το κομμάτι υλικού Arduino προς τιμήν του βασιλιά Ιταλίας Arduin της Ivrea. Το πρόγραμμα Arduino είναι μια διακλάδωση της πλατφόρμας ανοιχτής πηγής καλωδίωσης και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη σε καλωδιώσεις (σύνταξη και βιβλιοθήκες), παρόμοια με τη C ++ με κάποιες μικρές απλουστεύσεις και τροποποιήσεις και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) με βάση την επεξεργασία.

Το Arduino χτίστηκε γύρω από το έργο καλωδίωσης του Hernando Barragan. Η καλωδίωση ήταν έργο διατριβής του Hernando στο Ινστιτούτο Σχεδίασης Αλληλεπίδρασης Ivrea που ήταν υπό την εποπτεία του Hernando και του Massimo Banzi. Προοριζόταν να είναι μια ηλεκτρονική έκδοση της επεξεργασίας που χρησιμοποίησε το περιβάλλον προγραμματισμού και σχεδιάστηκε μετά τη σύνταξη επεξεργασίας.

Επί του παρόντος, υπάρχουν περισσότεροι από 200 διανομείς προϊόντων Arduino σε όλο τον κόσμο. Περίπου το 80% των ανθρώπων που αγοράζουν αυτό το προϊόν προέρχονται από τις Ηνωμένες Πολιτείες και την Ευρώπη. Το ενδιαφέρον για αυτό το προϊόν αυξάνεται στις αγορές της Κίνας, της Ινδίας και της Νότιας Αμερικής. Με τα χρόνια, έχουν δημιουργηθεί νέα σχέδια του Arduino. Ο αρχικός σχεδιασμός ονομάζεται Arduino Uno. Μερικά από τα σχέδια του Arduino είναι τα Arduino Mega, Arduino Nano, LilyPad Arduino και Arduino Ethernet. Το Arduino κέρδισε δημοσιότητα συνεργάζοντας με την Google. Η Google κυκλοφόρησε το Android ADK ή το κιτ ανάπτυξης αξεσουάρ, το οποίο βασίζεται στο Arduino. Ένα άτομο μπορεί να δημιουργήσει μια εφαρμογή Android που χρησιμοποιεί την κάμερα του τηλεφώνου, αισθητήρες κίνησης, οθόνη αφής και σύνδεση στο διαδίκτυο. Φαίνεται ότι το Arduino δημιουργεί ένα νέο, φθηνότερο τρόπο προγραμματισμού. [6]

1.4 Μοντέλα Arduino

Μερικά από τα βασικά και πιο γνωστά Arduino.

Arduino UNO Rev3



Arduino Leonardo (+headers)



Arduino YŦN



Arduino DUE



Arduino Mega2560 Rev3



Arduino Esplora



Arduino Ethernet Rev3 WITH PoE



Arduino Ethernet Rev3 WITHOUT PoE



Arduino Ethernet WITH PoE +
USB2SERIAL



Arduino UNO SMD Rev3



Arduino ADK Rev3



Arduino Leonardo



Arduino Nano



Arduino Mini 05



Arduino UNO to MKR Bundle



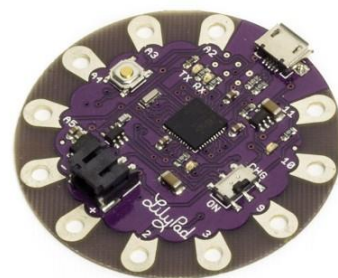
ARDUINO INDUSTRIAL 101



LILYPAD ARDUINO SIMPLESNAP



LILYPAD ARDUINO USB



Πίνακας 1.1: Διάφορα μοντέλα Arduino[7]

1.5 Shields Arduino

Shields (ασπίδα) είναι τα εξαρτήματα που συνδέονται απευθείας με όλα τα pins μιας πλακέτας Arduino. Η έννοια του shield στην αρχιτεκτονική Arduino, είναι η ενσωμάτωση επιπλέον υλικού (hardware) στο μικροελεγκτή που του προσδίδει μια νέα ιδιότητα, κυρίως σε θέματα επικοινωνίας. Με την προσαρμογή του υλικού αυτού η επικοινωνία από σειριακή (μέσω USB) μετατρέπεται στην αντίστοιχη που διακρίνει το shield.

Παρακάτω βλέπουμε κάποια Shields.[7]

- Arduino WiFi Shield

Το Arduino WiFi Shield συνδέει το Arduino σε ασύρματο δίκτυο.



Εικόνα 1.2: Arduino WiFi Shield

- Arduino Ethernet Shield

Το Arduino Ethernet Shield συνδέει το Arduino στο διαδίκτυο με ένα RJ45 καλώδιο.



Εικόνα 1.3: Arduino Ethernet Shield

- Arduino Wireless SD Shield

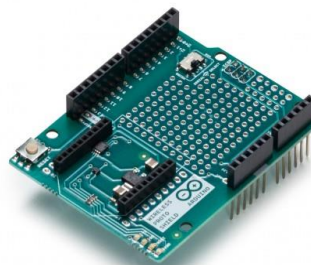
Το Wireless SD Shield επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 30 μέτρα περίπου σε εσωτερικούς χώρους και σε εξωτερικούς χώρους ως 100 μέτρα περίπου. Η μονάδα περιλαμβάνει και μια θύρα υποδοχής SD.



Εικόνα 1.4: Arduino Wireless SD Shield

- Arduino Wireless Proto Shield

Το Wireless Proto Shield επιτρέπει στο Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 30 μέτρα περίπου σε εσωτερικούς χώρους και σε εξωτερικούς χώρους ως 100 μέτρα περίπου. Η μονάδα δεν περιλαμβάνει θύρα υποδοχής SD.



Εικόνα 1.5: Arduino Wireless Proto Shield

- Arduino Motor Shield

Το Arduino Motor Shield επιτρέπει την οδήγηση δύο DC κινητήρων από την ίδια συσκευή, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά.



Εικόνα 1.6: Arduino Motor Shield

- Arduino GSM Shield 2 (Antenna Connector)

Το Arduino GSM Shield 2 επιτρέπει σε ένα board Arduino να συνδεθεί στο διαδίκτυο, να στείλει / λάβει κλήσεις φωνής και να στείλει / λάβει μηνύματα SMS. Χρησιμοποιεί ένα radio modem M10 από την Quectel.



Εικόνα 1.7: Arduino GSM Shield 2 (Antenna Connector)

- Arduino MKR CAN Shield

Το MKR CAN SHIELD συνδέεται εύκολα σε ένα διαύλου τύπου CAN (Bus Controller Network). Το MKR CAN μπορεί να απλοποιήσει τη σύνδεση των πλακών MKR με βιομηχανικά συστήματα και ειδικά με εφαρμογές αυτοκινήτων. Ανοίγει ένα νέο σύνολο πιθανών εφαρμογών όπως έξυπνα οχήματα, αυτόνομα αυτοκίνητα και αεροσκάφη.



Εικόνα 1.8: Arduino MKR CAN Shield

1.6 Σύγκριση μεταξύ των μοντέλων Arduino

Παρακάτω βλέπουμε πίνακες διαφορετικών τύπων Arduino.

Όνομα	Επεξεργαστής	Τάση λειτουργίας/εισόδου	Ταχύτητα CPU	Αναλογικοί είσοδοι/έξοδοι
Uno	ATmega328P	5V / 7-12V	16 MHz	6/0
101	Intel Curie	3.3V / 7-12V	32 MHz	6/0
Robot	ATmega32u4	5V	16 MHz	6/0
Mega 2560	ATmega2560	5V / 7-12V	16 MHz	16/0
Zero	ATSAMD21G18	3.3V / 7-12V	48 MHz	6/1
Due	ATSAM3X8E	3.3V / 7-12V	84 MHz	12/2
Mega ADK	ATmega2560	5V / 7-12V	16 MHz	16/0
Pro	ATmega168 ATmega328P	3.3V / 3.35-12V 5V / 5-12V	8 MHz 16 MHz	6/0
Yun	ATmega32U4 AR9331 Linux	5V 3.3V	16 MHz 400 MHz	12/0
Ethernet	ATmega328P	5V / 7-12V	16 MHz	6/0
Gemma	ATiny85	3.3V / 4-16V	8 MHz	1/0
Lilypad	ATmega168V ATmega328P	2.7-5.5V 2.7-5.5V	8 MHz	6/0

Πίνακας 1.2: Σύγκριση διαφορετικών τύπων Arduino A' [8]

Όνομα	Ψηφιακοί IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UAR T
Uno	14/6	1	2	32	Regular	1
101	14/4	-	24	196	Regular	-
Robot	20/6	1 KB (ATmega32u4)	2.5 KB (ATmega328P)	32 KB / 4KB	1	1
Mega 2560	54/15	4	8	256	Regular	4
Zero	14/10	-	32	256	2 Micro	2
Due	54/12	-	96	512	2 Micro	4
Mega ADK	54/15	4	8	256	Regular	4
Pro	14/6	512 Bytes 1 kB	1 2	16 32	-	1
Yun	20/7	1	2.5 16 MB	32 64 MB	Micro	1
Ethernet	14/4	1	2	32	Regular	-
Gemma	3/2	0.5	0.5	8	Micro	0
Lilypad	14/6	512 Bytes	1	16	-	-

Πίνακας 1.3: Σύγκριση διαφορετικών τύπων Arduino B' [8]

1.7 Εκδόσεις Arduino

- Τον Σεπτέμβριο του 2006 ανακοινώθηκε το Arduino Mini .
- Τον Οκτώβρη του 2008 ανακοινώθηκε το Arduino Duemilanove. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328.
- Τον Μάρτιο του 2009 ανακοινώθηκε το Arduino Mega. Είναι βασισμένο στο Atmel ATmega1280.
- Από τον Μάιο του 2011 πάνω από 300.000 Arduino ήταν σε χρήση σε όλο τον κόσμο.
- Τον Ιούλιο του 2012 ανακοινώθηκε το Arduino Leonardo. Είναι βασισμένο στο Atmel ATmega32u4.
- Τον Οκτώβριο του 2012 ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM Cortex-M3.
- Τον Νοέμβριο του 2012 ανακοινώθηκε το Arduino Micro. Είναι βασισμένο στο Atmel ATmega32u4.
- Τον Μάιο του 2013 ανακοινώθηκε το Arduino Robot. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες
- Τον Μάιο του 2013 ανακοινώθηκε το Arduino Yun. Είναι βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν WiFi που συνδύαζε το Arduino με το Linux.[6]

1.8 Πλεονεκτήματα Arduino

- ✓ Μπορεί να τρέξει αυτόνομο από έναν υπολογιστή και έχει ενσωματωμένη μνήμη.
- ✓ Μπορεί να λειτουργήσει με ψηφιακά και αναλογικά ηλεκτρονικά σήματα, αισθητήρες και ενεργοποιητές.
- ✓ Επικοινωνεί με τον υπολογιστή με USB ή με Bluetooth.
- ✓ Τροφοδοτείται από εξωτερική πηγή τροφοδοσίας, μπαταρία ή από την USB του υπολογιστή.
- ✓ Το Arduino είναι ένα εργαλείο που μπορούμε να κατασκευάσουμε με την έννοια ότι θα ελέγχει συσκευές του φυσικού κόσμου σε αντίθεση με έναν κοινό ηλεκτρονικό υπολογιστή.
- ✓ Μικρό κόστος υλικών.[5]

Με βάση αυτήν την εισαγωγή για τα είδη των πλακετών Arduino που υπάρχουν επιλέξαμε να υλοποιήσουμε την εργασία μας στις πλακέτες Arduino Mega 2560 Rev3 και Arduino Nano, λόγω του μεγέθους μνήμης και του πλήθους των ακροδεκτών τους. Έτσι στο επόμενο κεφάλαιο παρουσιάζονται οι πλακέτες Mega και Nano.

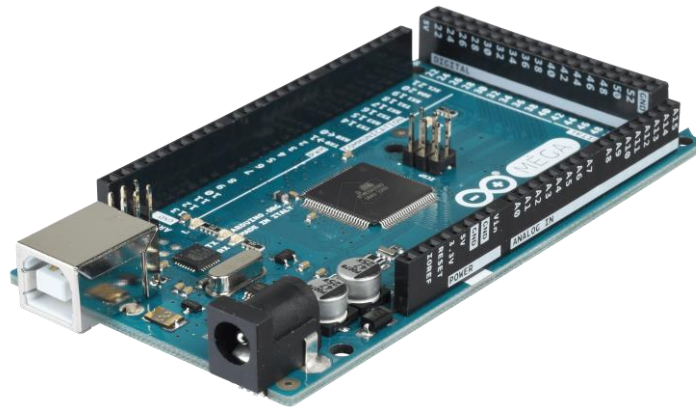
Κεφάλαιο 2 - Η πλακέτα Arduino Mega και Nano

2.1 Πληροφορίες για το Arduino Mega 2560

Το Arduino Mega είναι η πιο εξελιγμένη πλακέτα της τεχνολογίας Arduino και προτείνεται για περίπλοκες κατασκευές που απαιτούν μεγαλύτερη μνήμη και περισσότερες εισόδους/εξόδους. Το μοντέλο Mega 2560 της οικογένειας Arduino είναι το πιο διαδεδομένο και είναι συμβατό με πλήθος αισθητήρων και επεκτάσεων, βασίζεται στην ίδια αρχιτεκτονική με την πλακέτα Uno και η βασική της διαφορά είναι η χωρητικότητα της μνήμης και το πλήθος των εισόδων εξόδων για την σύνδεση με εξωτερικές συσκευές. Η έκδοση 3 (Rev3) είναι η τελευταία που κυκλοφορεί και είναι πλήρως συμβατή με καινούργιες και παλιές πλακέτες επέκτασης.

Το Arduino Mega Rev3 βασίζεται στον μικροελεγκτή ATmega2560 της Atmel. Είναι μια ολοκληρωμένη πλακέτα που περιέχει ότι χρειάζεται για να μπορεί να προγραμματιστεί και να λειτουργήσει συνδέοντάς την με ένα απλό καλώδιο USB στον ηλεκτρονικό υπολογιστή ή με ένα τροφοδοτικό στην πρίζα, επίσης μπορεί να λειτουργήσει με απλή μπαταρία.

Το Arduino Mega είναι συμβατό μετά περισσότερα shields που έχουν κυκλοφορήσει για το Arduino αλλά όχι με το Ethernet Shield το οποίο είναι ένα αρκετά σημαντικό μειονέκτημα για όσους θέλουν να φτιάξουν εφαρμογές με πρόσβαση στο Internet ή σε κάποιο άλλο δίκτυο.[9]



Εικόνα 2.1: Arduino mega 2560 Rev3

2.1.1 Τροφοδοσία Arduino Mega

Το Arduino Mega 2560 μπορεί να τροφοδοτηθεί με ρεύμα από τον ηλεκτρονικό υπολογιστή μέσω της σύνδεσης USB ή από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής των 2.1mm με 7V έως 12V DC από την είσοδο της τροφοδοσίας χωρίς να υπάρχει κίνδυνος ηλεκτροπληξίας. Ο Mega 2560 διαφέρει από όλους τους προηγούμενους του, γιατί δεν χρησιμοποιεί FTDI (USB to Serial) ως chip οδηγού, αλλά χρησιμοποιεί τον Atmega8U2 της Atmel που έχει προγραμματιστεί ως USB to Serial μετατροπέας, με αποτέλεσμα πολύ μεγαλύτερες ταχύτητες στην σειριακή σύνδεση.

Δίπλα από τα pins αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER.

Η λειτουργία του καθενός έχει ως εξής:

- IOREF: παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής.
- RESET: όταν γειωθεί (σε οποιοδήποτε από τα 3 pins με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- 3.3V: μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial over USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- 5V: μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την <<φέρει>> στα 5V.
- GND: Το pin της γείωσης τροφοδοσίας.
- Vin: έχει διπλό ρόλο, σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή των 2.1mm. Αν όμως έχουμε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, μπορούμε να χρησιμοποιήσουμε αυτό το pin για να τροφοδοτήσουμε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7 - 12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.[9]

2.1.2 Ακροδέκτες Arduino Mega

Αναλυτικά η πλακέτα διαθέτει 54 ψηφιακές εισόδους ή εξόδους (14 από αυτές μπορεί να χρησιμοποιηθούν σαν PWM εξόδους), 16 αναλογικές εισόδους, 4 σειριακές θύρες, 1 θύρα USB (τύπου B) για τον προγραμματισμό και την τροφοδοσία της πλακέτας, 1 είσοδο τροφοδοσίας που μπορεί να χρησιμοποιηθεί εναλλακτικά για τροφοδοσία από τροφοδοτικό πρίζας ή από απλή μπαταρία, 1 υποδοχή ICSP και τέλος κουμπί για το reset της πλακέτας.

Συγκεκριμένα:

- Τα pins αναλογικής εισόδου A0-A15 λειτουργούν πανομοιότυπα με του δομικού διαγράμματος.
- Τα ψηφιακά pins 2-13, τα ψηφιακά pins 22-49, τα communication pins 0-1 και τα communication pins 14-21, χρησιμοποιούνται σαν ψηφιακές εισοδοι ή εξοδοι αλλά και σαν εισοδοι γενικού σκοπού με χρήση των εντολών pinMode(), digitalRead(), digitalWrite(). Κάθε pin περιέχει μία εσωτερική pull-up αντίσταση 20-50 kOhms η οποία μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί όταν τα ψηφιακά pins που λειτουργούν σαν εισοδοι, με την βοήθεια της εντολής digitalWrite(). Παρ' όλα αυτά όμως, τα ψηφιακά pins και τα communication pins έχουν κι άλλες λειτουργίες (κάθε ψηφιακό pin μπορεί να επιτελέσει μόνο μία λειτουργία κάθε φορά).

Πιο συγκεκριμένα:

- Τα ψηφιακά pins 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) και 21 (interrupt 2), μπορούν να χρησιμοποιηθούν ώστε να ενεργοποιήσουν τους αναφερθείσας εξωτερικούς διακόπτες (interrupts) με τη βοήθεια της εντολής attachInterrupt().
- Τα ψηφιακά pins 2-13 και 44-46, μπορούν να χρησιμοποιηθούν για να δώσουν μία 8-bits PWM (Pulse-width-modulation) έξοδο με χρήση της εντολής analogWrite().
- Το ψηφιακό pin 7 (RST), δεν πρέπει να χρησιμοποιείται σαν ψηφιακή είσοδος ή έξοδος αφού χρησιμοποιείται για την επικοινωνία με το FTDI.
- Τα communication pins 0-1 και 14-19, λειτουργούν και σαν ψηφιακά pins αλλά μπορούν να λειτουργήσουν και σαν pins σειριακής επικοινωνίας με χρήση της εντολής Serial.begin (UARTs). Τα pins RX λαμβάνουν σειριακά δεδομένα και τα pins TX μεταδίδουν σειριακά δεδομένα. Πιο συγκεκριμένα οι σειριακές θύρες είναι οι εξής:

- Serial 0: 0 (RX) και 1 (TX)
- Serial 1: 19 (RX) και 18 (TX)
- Serial 2: 17 (RX) και 16 (TX)
- Serial 3: 15 (RX) και 14 (TX)

Η σειριακή επικοινωνία επιτυγχάνεται με την βοήθεια ενός chip ATMEGA8U2 το οποίο συνδέει μέσω USB τα UART, δηλαδή τους συνδυασμούς RX & TX pins (serial 0,1,2,3) και

παράγει μία εικονική θύρα COM στο λειτουργικό σύστημα του υπολογιστή. Όταν μεταδίδονται δεδομένα από το ATMEGA8U2 chip και την θύρα USB στον υπολογιστή τότε αναβοσβήνουν οι λυχνίες LED TX, RX εκτός αν μεταδίδονται σειριακά δεδομένα μέσω των pins 0 και 1 οπότε οι λυχνίες LED TX, RX δεν αναβοσβήνουν.

- Τα pins 20 (SDA) και 21 (SCL), επιτρέπουν την επικοινωνία με το πρωτόκολλο Two Wire Protocol (TWI) ή I2C το οποίο δημιουργήθηκε από την Philips για σύνδεση περιφερειακών χαμηλής ταχύτητας. Το πρωτόκολλο TWI βασίζεται στην σειριακή ασύγχρονη επικοινωνία και χρησιμοποιεί δύο αμφίδρομες γραμμές επικοινωνίες οι οποίες έχουν pull-up αντιστάσεις, την Serial Data Line (SDL) και την Serial Clock (SCL). Οι συσκευές που μπορούν να συνδεθούν στον δίαυλο I2C ανέρχονται σε 128. Μία από αυτές θα είναι Master και θα ελέγχει τις υπόλοιπες ενώ οι υπόλοιπες θα είναι Slave.

- Τα pins 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) υποστηρίζουν επικοινωνία με πρωτόκολλο SPI με χρήση της βιβλιοθήκης SPI library. Το πρωτόκολλο SPI (Serial Peripheral Interface Bus), το οποίο δημιουργήθηκε από την Motorola, επιτρέπει την σειριακή σύγχρονη επικοινωνία μεταξύ ολοκληρωμένων σε πλήρως αμφίδρομη επικοινωνία.

- Το AREF ή Analog Reference pin, προγραμματίζεται μέσω της εντολής analogReference(type) με την ονομαστική τάση που χρησιμοποιείται όταν υπάρχει κάποια αναλογική είσοδος όπως περιγράφεται παραπάνω.

- Η λυχνία LED L, είναι συνδεδεμένη με το pin 13 και όταν αυτό έχει τιμή ορίσματος HIGH τότε το λαμπάκι ανάβει ενώ όταν έχει τιμή ορίσματος LOW το λαμπάκι παραμένει σβηστό.

- Η θύρα USB, για σύνδεση με τον υπολογιστή.

- Το chip MAX3421E ή FTDI, το οποίο χρησιμεύει ώστε να προγραμματίζεται το Arduino μέσω της σύνδεσης με καλώδιο USB. Το FTDI επικοινωνεί με την πλακέτα μέσω του διαύλου επικοινωνίας SPI και χρησιμοποιεί γι' αυτήν την επικοινωνία τα ψηφιακά pins 7 (RST), 50 (MISO), 51 (MOSI), 52 (SCK).

- Η δίοδος D1, που προστατεύει τον μικροελεγκτή από το ρεύμα.

- Ο μικροελεγκτής (IC1), που διαθέτει η πλακέτα Arduino Mega 2560, ο οποίος είναι το chip ATMEGA2560 της εταιρείας Atmel.

- Με το ICSP (in-circuit serial programming), προγραμματίζεται απευθείας ο μικροελεγκτής της πλατφόρμας, συνδέοντάς τον με ένα καλώδιο στον H/Y.

- Η λυχνία LED PWR, ενεργοποιείται όταν η πλακέτα τεθεί σε λειτουργία.

Όπως φαίνεται, σε αυτήν την πλακέτα δεν υπάρχουν SV1 pins αφού η επιλογή της τροφοδοσίας γίνεται αυτόματα μόλις συνδέσουμε την πλακέτα στον υπολογιστή μέσω USB ή μόλις την συνδέσουμε με εξωτερική τροφοδοσία μέσω Barrel jack ή μέσω μπαταρίας με το pin VIN. [9]

2.1.4 Μνήμες Arduino mega

Ο μικροεπεξεργαστής ATmega2560 έχει τρεις κατηγορίες μνημών. Διαθέτει flash memory στην οποία αποθηκεύονται τα Arduino sketch, SRAM (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει και EEPROM η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών.

- Μνήμη Flash 256 KB, από τα οποία 8 KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 254KB χρησιμοποιούνται για την αποθήκευση των προγραμμάτων.
- Μνήμη SRAM 8 KB, που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύσουν μεταβλητές, πίνακες και άλλα κατά το runtime. Αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος σταματήσει ή γίνει reset.
- EEPROM 4 KB, μπορεί να χρησιμοποιηθεί για εγγραφή και ανάγνωση δεδομένων χωρίς datatype ανά byte από το πρόγραμμα κατά το runtime. Η EEPROM έχει όριο ζωής καθώς δε μπορεί να επαναπρογραμματιστεί για περισσότερες από 10.000 φορές και τέλος δεν χάνει τα περιεχόμενα της όταν σταματήσει η παροχή ρεύματος ή γίνει reset.[9]

2.2 Πληροφορίες για το Arduino Nano

Το Arduino Nano αποτελεί την πιο μικρή πλακέτα της τεχνολογίας Arduino και είναι η μικρή έκδοχή του Arduino Uno. Το Arduino Nano βασίζεται στον μικροελεγκτή ATmega328 της Atmel. Η πλακέτα αυτή είναι ακριβώς η ίδια στην λειτουργία με τον Arduino Uno σε μικρότερο μέγεθος και πλήρως συμβατή ώστε να τοποθετηθεί σε πλακέτες δοκιμών (breadboard). Είναι μια ολοκληρωμένη πλακέτα που περιέχει ότι χρειάζεται για να μπορεί να προγραμματιστεί και να λειτουργήσει συνδέοντάς την με ένα απλό καλώδιο Mini-B USB στον ηλεκτρονικό υπολογιστή .[10]



Εικόνα 2.2: Arduino Nano

2.2.1 Τροφοδοσία Arduino Nano

Το Arduino Nano λειτουργεί σε χαμηλή τάση, με τροφοδοσία 5V DC από την είσοδο του USB.[8]

2.2.2 Ακροδέκτες Arduino Nano

Η πλακέτα διαθέτει 14 ψηφιακές εισόδους ή εξόδους (6 από αυτές μπορεί να χρησιμοποιηθούν σαν PWM εξόδους), 6 αναλογικές εισόδους, 1 θύρα Mini-B USB για τον προγραμματισμό και την τροφοδοσία της πλακέτας, 1 υποδοχή ICSP και τέλος κουμπί για το reset της πλακέτας

Αναλυτικά:

Κάθε ένα από τα 14 ψηφιακές pins του Nano μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος χρησιμοποιώντας λειτουργίες pinMode (), digitalWrite () και digitalRead (). Λειτουργούν στα 5V. Κάθε pin μπορεί να παρέχει ή να λαμβάνει μέγιστο 40 mA και έχει εσωτερική αντίσταση pull-up (αποσυνδεδεμένη από προεπιλογή) 20-50 kOhms. Επιπλέον, μερικά pins έχουν εξειδικευμένες λειτουργίες:

- Serial: 0 (RX) και 1 (TX). Χρησιμοποιείται για τη λήψη (RX) και (TX)TTL για σειριακά δεδομένα. Τα pins αυτά συνδέονται με τους αντίστοιχους ακροδέκτες του σειριακού τσιπ FTDI USB-to-TTL.
- External Interrupts: 2 και 3. Αυτά τα pins μπορούν να διαμορφωθούν για να προκαλέσουν διακοπή σε χαμηλή τιμή, άνοδο ή πτώση άκρης ή αλλαγή τιμής. Για λεπτομέρειες, ανατρέξτε στη λειτουργία attachInterrupt ().
- PWM: 3, 5, 6, 9, 10 και 11. Παρέχετε έξοδο PWM 8-bit με τη λειτουργία analogWrite ().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτά τα pins υποστηρίζουν την επικοινωνία SPI, η οποία δεν περιλαμβάνεται αυτή τη στιγμή στη γλώσσα Arduino.
- LED: 13. Υπάρχει μια ενσωματωμένη λυχνία LED συνδεδεμένη στον ψηφιακό ακροδέκτη 13. Όταν ο ακροδέκτης είναι HIGH τιμή, η λυχνία LED είναι αναμμένη, όταν ο ακροδέκτης είναι LOW, είναι απενεργοποιημένη.

Το Nano διαθέτει 8 αναλογικές εισόδους, καθεμία από τις οποίες παρέχει 10bits (δηλαδή 1024 διαφορετικές τιμές). Από προεπιλογή, μετρούν από το έδαφος σε 5V, αν και είναι δυνατή η αλλαγή του ανώτερου άκρου της περιοχής τους χρησιμοποιώντας τη λειτουργία analogReference (). Τα αναλογικά pins 6 και 7 δεν μπορούν να χρησιμοποιηθούν ως ψηφιακά.

Επιπλέον, μερικά έχουν εξειδικευμένη λειτουργικότητα:

- 2C: 4 (SDA) και 5 (SCL). Υποστηρίζει την επικοινωνία I2C (TWI) χρησιμοποιώντας τη Wire library.
- AREF: Τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με την εντολή analogReference ().
- Reset: Έχει ως αποτέλεσμα την επανεκκίνηση του Arduino Nano.[10]

2.2.3 Μνήμες Arduino Nano

Η μνήμη Flash του Arduino Nano που μπορεί να αποθηκεύσει το πρόγραμμα (sketch) είναι 32KB από τα οποία 2KB χρησιμοποιούνται από το firmware του Arduino.

SRAM: 2KB

EEPROM: 1 KB [10]

Στην συνέχεια θα αναφερθούμε στο προγραμματιστικό περιβάλλον του Arduino, ώστε να καταλάβουμε τα βήματα που χρειάζονται για τον προγραμματισμό και την υλοποίηση του κώδικα.

Κεφάλαιο 3 – Περιβάλλον ανάπτυξης του Arduino

3.1 Λογισμικό Arduino IDE

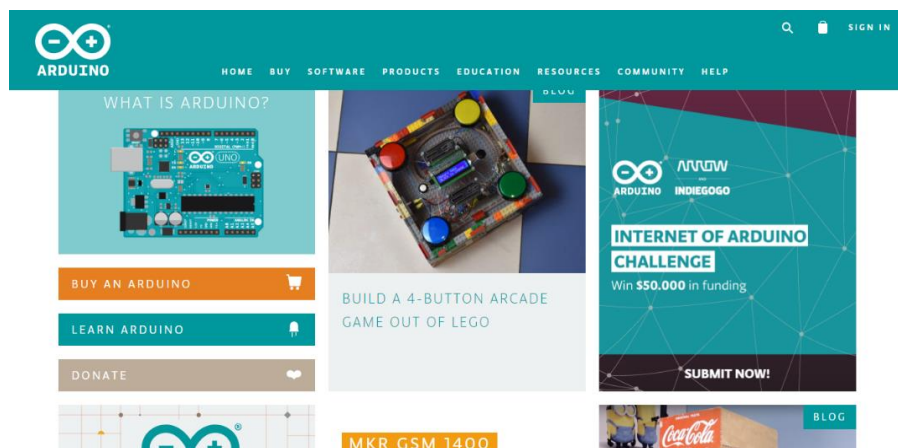
Το Arduino είναι βασισμένο σε γλώσσα C και συγκεκριμένα παρέχει:

- Ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση.
- Έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και εύκολο χειρισμό μέσα από τον κώδικα των εξαρτημάτων που έχουν συνδεθεί στο Arduino
- Τον compiler για την μεταγλώττιση των sketch.
- Ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά την επιλογή στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch.
- Την επιλογή να ανεβάσει το μεταγλωττισμένο sketch στο Arduino.

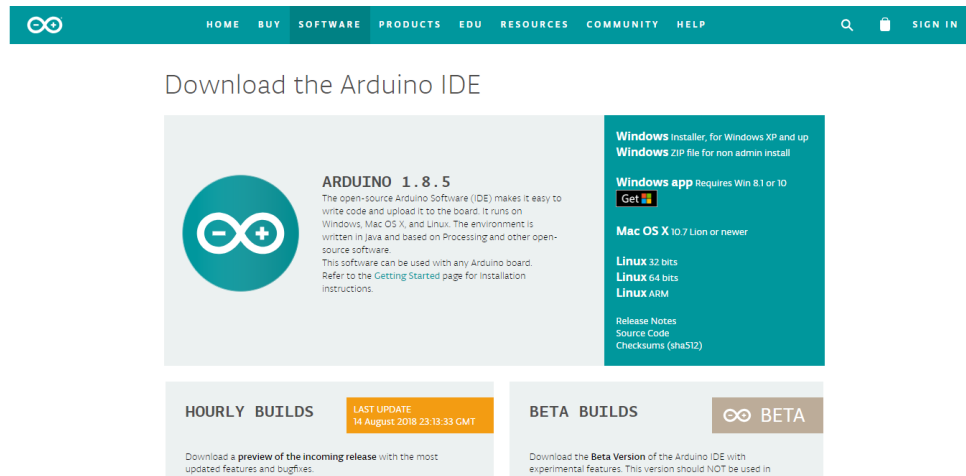
3.2 Εγκατάσταση του Arduino IDE

Για να εγκαταστήσουμε το Arduino IDE σε περιβάλλον Windows ακολουθούμε τα εξής βήματα:

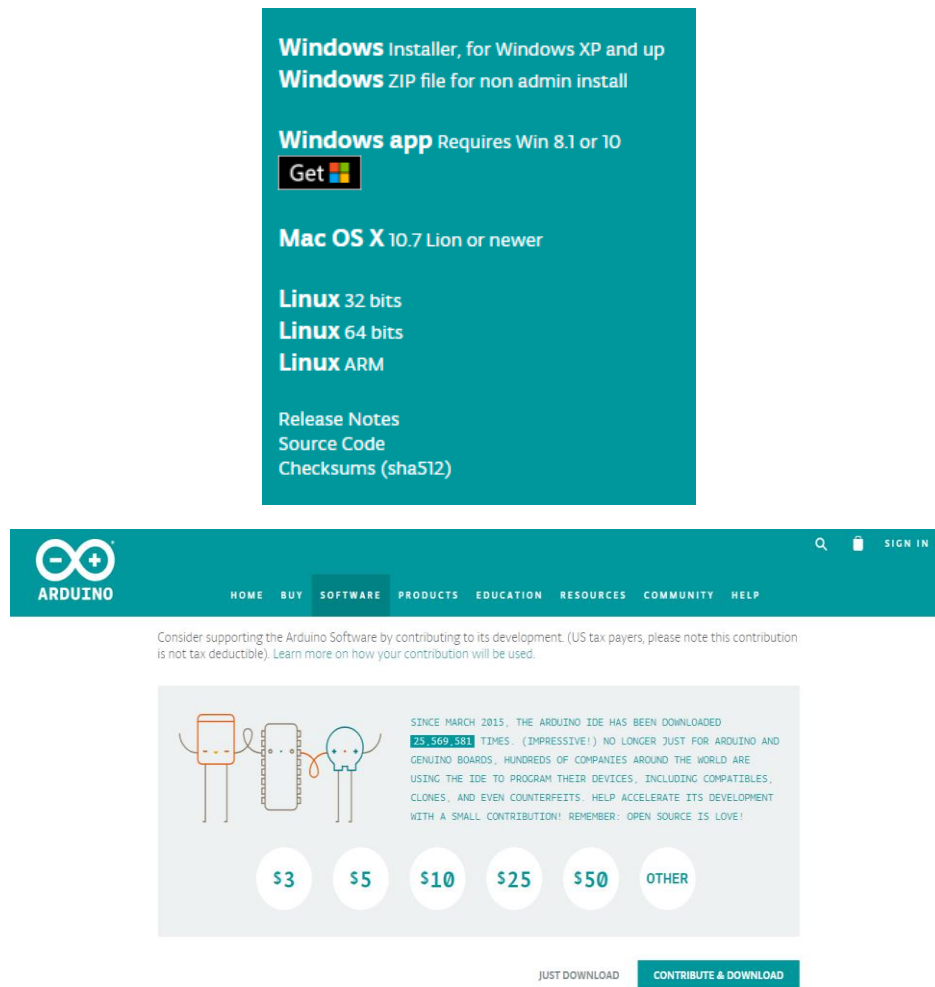
1. Χρησιμοποιώντας έναν browser όπως π.χ. ο Chrome ή ο Mozilla επισκεπτόμαστε την επίσημη σελίδα του Arduino(<https://www.arduino.cc/>).



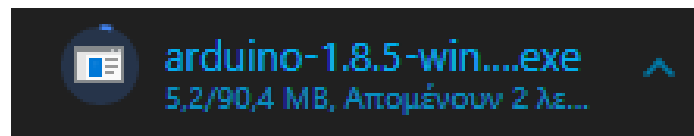
2. Στην συνέχεια επιλέγουμε την κατηγορία SOFTWARE και μετά DOWNLOADS.



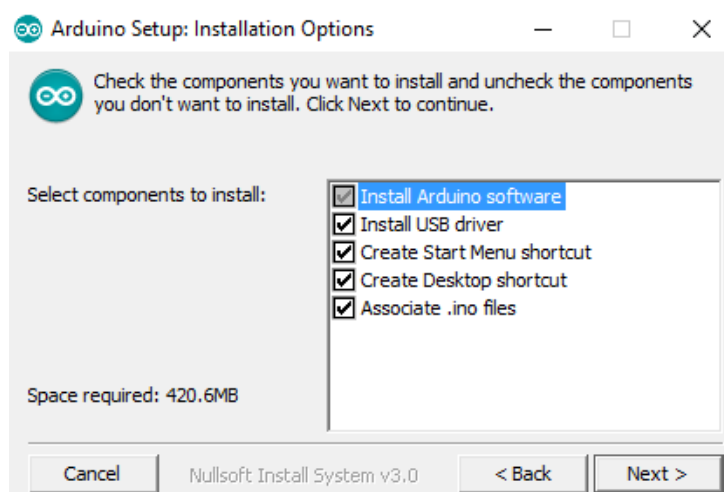
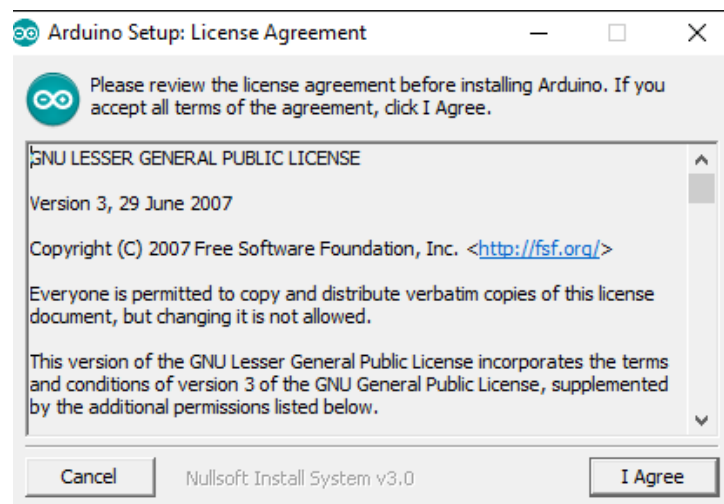
3. Έπειτα από το παράθυρο που εμφανίζεται στα δεξιά της οθόνης επιλέγουμε το Windows Installer και στην συνέχεια την επιλογή Just Download.

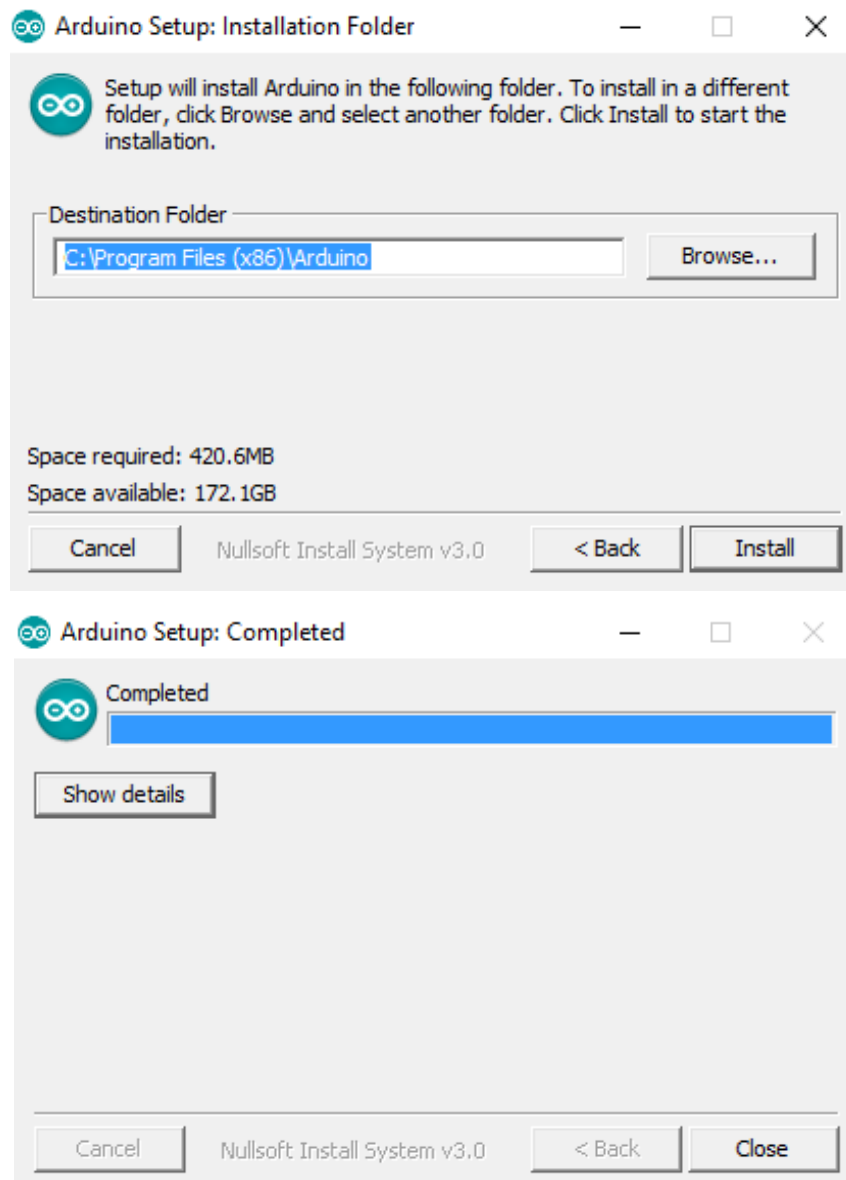


4. Γίνεται η λήψη του αρχείου.



5. Μετά την ολοκλήρωση της λήψης κάνουμε διπλό κλικ στο αρχείο και εμφανίζονται τα ακόλουθα παράθυρα διαλόγου στα οποία επιλέγουμε I Agree, Next, Install και Close όταν ολοκληρωθεί η εγκατάσταση.





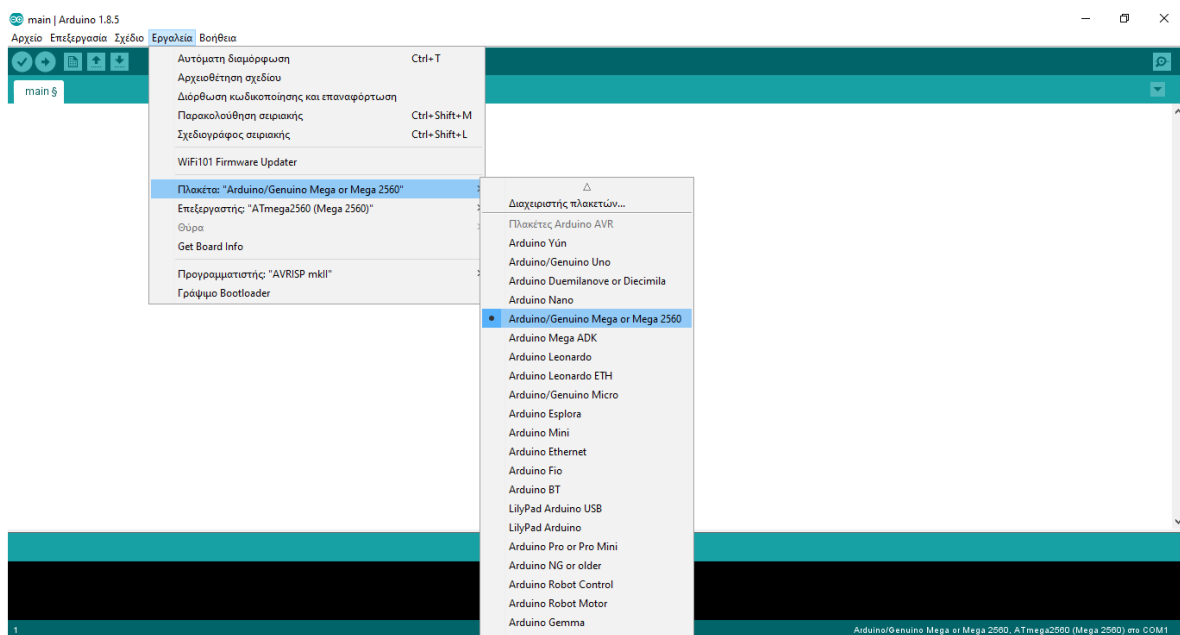
6. Η εγκατάσταση έχει ολοκληρωθεί με επιτυχία και το εικονίδιο της συντόμευσης για το Arduino IDE έχει εμφανιστεί στην επιφάνεια εργασίας μας.



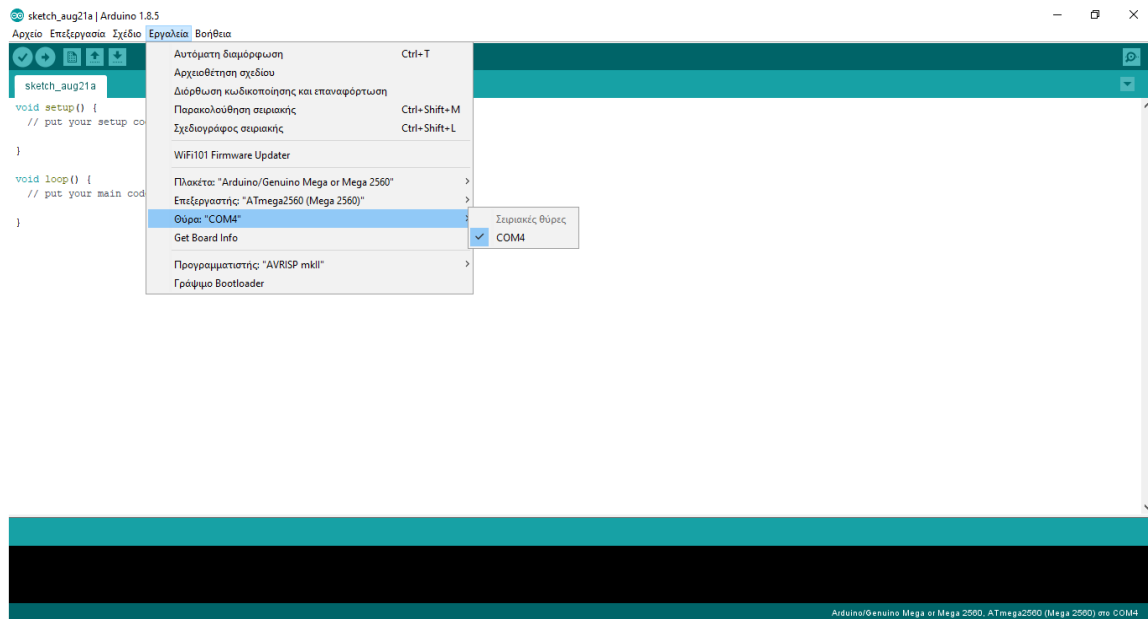
7. Η εφαρμογή μπορεί να εκτελεστεί πατώντας διπλό “κλικ” στο παραπάνω εικονίδιο.



8. Ολοκληρώνοντας την διαδικασία εγκατάστασης του IDE επιλέγουμε από την γραμμή εργαλείων Εργαλεία>> Πλακέτα: “Arduino/Genuino Mega or Mega2560”.



9. Στην συνέχεια Εργαλεία>> Θύρα: “COM4”.



3.3 Έλεγχος προγράμματος σε επίπεδο Compiler

• Συνάρτηση Setup

Το Setup είναι η πρώτη συνάρτηση που διαβάζεται σε ένα πρόγραμμα Arduino και εκτελείται μόνο μία φορά. Η βασική λειτουργία της συνάρτησης αυτής είναι:

- Η αρχικοποίηση των μεταβλητών που πρόκειται να χρησιμοποιηθούν κατά την εκτέλεση του προγράμματος αλλά και
- Η ανάθεση τιμών και σταθερών στην πλακέτα, που δεν πρόκειται να μεταβληθούν στην συνέχεια.

• Συνάρτηση Loop

Η συνάρτηση Loop είναι ο πυρήνας του κώδικα που πρόκειται να εκτελεστεί στην πλακέτα του Arduino. Η συνάρτηση αυτή, σε αντίθεση με την συνάρτηση Setup, εκτελείται ατέρμονα μέχρι να πατηθεί από τον χρήστη το κουμπί Reset ή να παρουσιαστεί κάποια πτώση τάσης.

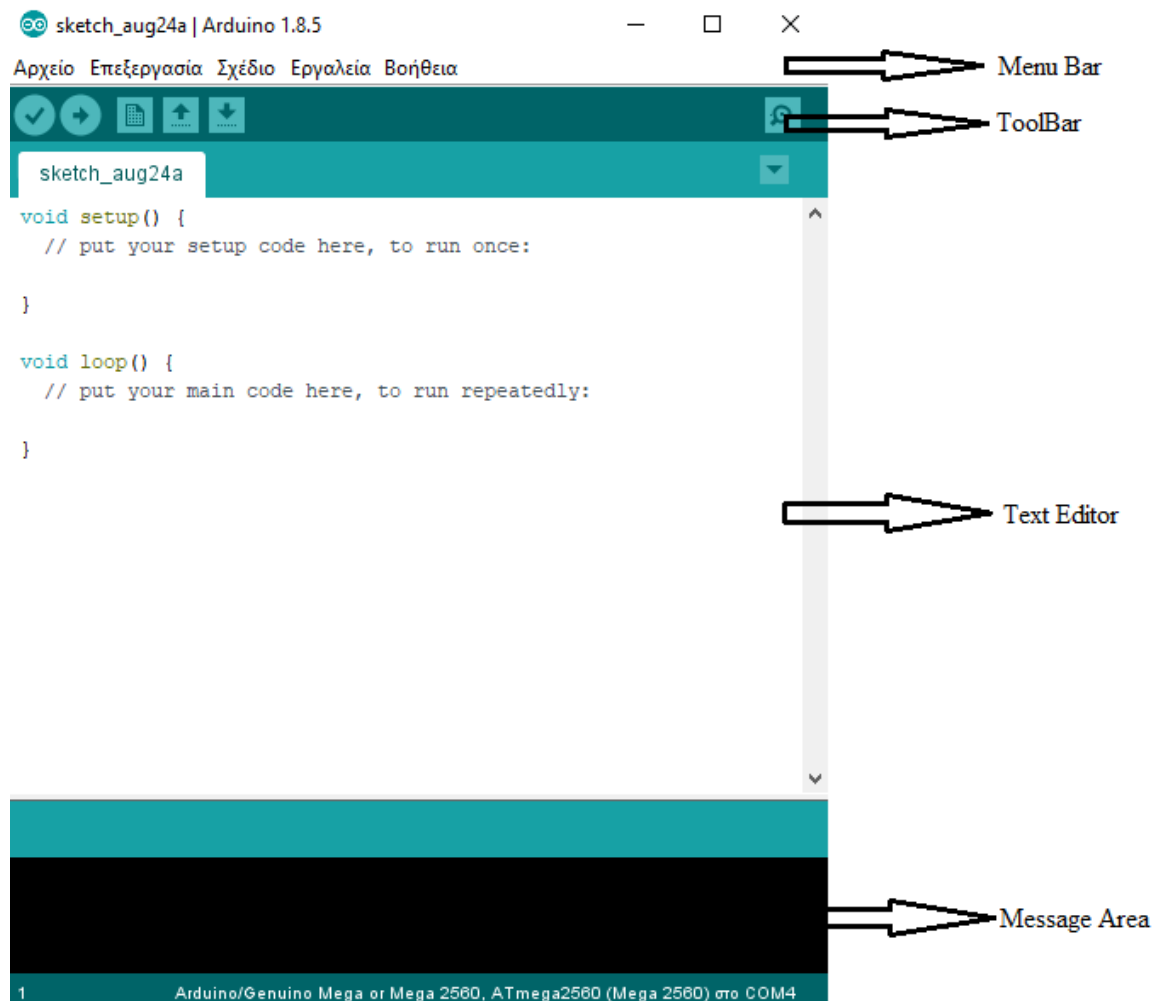
• Bootloader-Μεταφορά δεδομένων και προγράμματος στον Arduino

Οι μικροεπεξεργαστές προγραμματίζονται συνήθως μέσω ενός programmer εκτός αν υπάρχει κάποιο κομμάτι firmware σε αυτούς που τους επιτρέπει να εγκαταστήσουν καινούργιο firmware χωρίς την χρήση κάποιου εξωτερικού programmer. Αυτό ονομάζεται bootloader. Εάν ο χρήστης επιθυμεί να αξιοποιήσει όλη την χωρητικότητα της μνήμης flash

του τσιπ ή να αποφύγει την καθυστέρηση από τον bootloader, μπορεί να τρέξει τα sketches του χρησιμοποιώντας κάποιον εξωτερικό programmer.

3.4 Το GUI του Arduino IDE

Όπως φαίνεται κι από την εικόνα το GUI χωρίζεται σε 4 βασικά μέρη τα οποία είναι:



Εικόνα 3.1: Το GUI του Arduino IDE

- Menu bar: Η μπάρα αυτή περιέχει drop-down μενού για όλα τα εργαλεία, τις λειτουργίες και τις πληροφορίες που αξιοποιούνται από το πρόγραμμα. Η μπάρα αυτή βρίσκεται στην κορυφή της οθόνης και περιέχει τις επιλογές:

- 1.1 Αρχείο: Η επιλογή αυτή περιέχει λειτουργίες όπως η αποθήκευση, η φόρτωση, το άνοιγμα sketches κ.λπ.

1.2 Επεξεργασία: Η επιλογή αυτή περιέχει τις λειτουργίες της αντιγραφής, επικόλλησης και συναρτήσεις αναζήτησης.

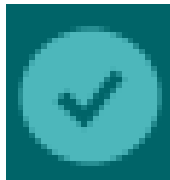
1.3 Σχέδιο: Η επιλογή αυτή περιλαμβάνει συναρτήσεις πιστοποίησης του σχεδίου (sketch) πριν την μεταφόρτωση του στην πλακέτα.

1.4 Εργαλεία: Η επιλογή αυτή διαθέτει μια πληθώρα συναρτήσεων κι εντολών που επιτρέπουν στον χρήστη να καθορίσει την πλακέτα του Arduino που πρόκειται να χρησιμοποιηθεί αλλά και την θύρα USB.

1.5 Βοήθεια: Η επιλογή αυτή περιέχει υπερσυνδέσεις για διάφορα ζητήματα που αφορούν θέματα σχετικά με το Arduino και το IDE.

2. Toolbar: Η εργαλειοθήκη περιέχει συγκεκριμένα κουμπιά που χρησιμοποιούνται για την συγγραφή των sketches. Η μορφή και η λειτουργία των επιλογών αυτών περιγράφεται στην συνέχεια.

2.1 Επικύρωση: Με την επιλογή αυτή γίνεται έλεγχος του κώδικα που έχει γραφτεί από εμάς, με βάση το λογισμικό του Arduino. Η μεταγλώττιση (Compile) του προγράμματος εξασφαλίζει ότι ο κώδικας δεν έχει συντακτικά λάθη, χωρίς όμως να ελέγχεται από το λογισμικό η ύπαρξη λογικών λαθών. Επομένως, με την διαδικασία αυτή ελέγχεται η εγκυρότητα αλλά όχι η αξιοπιστία του προγράμματος του χρήστη.



2.2 Ανέβασμα: Η επιλογή αυτή στέλνει το sketch στην πλακέτα του Arduino που είναι συνδεδεμένος με τον υπολογιστή. Με την διαδικασία αυτή εκτελείται αυτόματα η μεταγλώττιση του κώδικα του χρήστη πριν αυτός φορτωθεί στην πλακέτα.



2.3 Δημιουργία: Η επιλογή αυτή δημιουργεί ένα καινούργιο sketch.



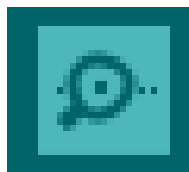
2.4 Άνοιγμα: Η επιλογή αυτή ανοίγει ένα υπάρχον sketch.



2.5 Αποθήκευση: Με την επιλογή αυτή γίνεται αποθήκευση του κώδικα του τρέχοντος sketch.



2.6 Serial monitor: Με την επιλογή αυτή παραχωρείται η δυνατότητα να παρατηρήσουμε τα δεδομένα που αποστέλλονται από και προς την πλακέτα του Arduino.



3. Text editor: Πρόκειται για το τμήμα του παραθύρου στο οποίο συγγράφουμε τον κώδικα του sketch που θέλουμε να καταχωρήσουμε στην πλακέτα. Πρόκειται για ένα κλασικό παράθυρο συγγραφής κειμένου εμπλουτισμένο με μερικά έξτρα χαρακτηριστικά, που στόχο έχουν να μας βοηθήσουν να συγγράψουμε επιτυχώς τον κώδικα του προγράμματός μας. Για τον λόγο αυτό το λογισμικό του Arduino χρησιμοποιεί διαφορετικούς χρωματικούς κώδικες για τις αντίστοιχες δομές του sketch.

4. Message area: Στο τμήμα αυτό του GUI εμφανίζονται μηνύματα σχετικά με σφάλματα (errors), προειδοποιήσεις (warnings) κατά την μεταγλώττιση του κώδικα μας, καθώς και πληροφορίες σχετικές με την πλακέτα του Arduino και την θύρα USB στην οποία είναι συνδεδεμένη.

3.5 Βασικές εντολές και η δομή του προγράμματος

Για την υλοποίηση ενός προγράμματος στην πλακέτα Arduino χρησιμοποιούνται οι παρακάτω βασικές δομές.[1][3]

• Τύποι δεδομένων

void: Ορισμός συνάρτησης που δεν επιστρέφει κάποια τιμή.

boolean: Λογικοί δυαδική τιμή (true/false).

char: Προσημασμένος χαρακτήρας 1 byte.

unsigned char: Μη προσημασμένος χαρακτήρας 1 byte.

byte: Μη προσημασμένος χαρακτήρας 8 bits.

int: Προσημασμένος ακέραιος αριθμός 2 bytes.

unsigned int: Μη προσημασμένος ακέραιος αριθμός 2 bytes.

word: Μη προσημασμένος ακέραιος αριθμός 2 bytes.

long: Προσημασμένος ακέραιος αριθμός 4 bytes.

unsigned long: Μη προσημασμένος ακέραιος αριθμός 4 bytes.

short: Προσημασμένος ακέραιος αριθμός 2 bytes.

float: Αριθμός κινητής υποδιαστολής 4 bytes.

double: Αριθμός κινητής υποδιαστολής 8 bytes.

string-char array: Πίνακας προσημασμένων χαρακτήρων 1 byte.

String-object: Σύνθετη κλάση-δομή πίνακα προσημασμένων χαρακτήρων 1 byte (καταλαμβάνει περισσότερη μνήμη).

array: Πίνακας (συλλογή μεταβλητών) δεδομένων.

• Σταθερές

HIGH: Τιμή υψηλής στάθμης.

LOW: Τιμή χαμηλής στάθμης.

False: Τιμή λογικού 0 (λάθους).

True: Τιμή λογικού 1 (αληθείας).

INPUT: Ορισμός ενός ακροδέκτη ως είσοδο.

OUTPUT: Ορισμός ενός ακροδέκτη ως έξοδο.

• Συναρτήσεις μετατροπής τύπων

char(): Μετατροπή μεταβλητής σε δεδομένα τύπου char.

byte(): Μετατροπή μεταβλητής σε δεδομένα τύπου byte.

int(): Μετατροπή μεταβλητής σε δεδομένα τύπου int.

word(): Μετατροπή μεταβλητής σε δεδομένα τύπου word.

long(): Μετατροπή μεταβλητής σε δεδομένα τύπου long.

float(): Μετατροπή μεταβλητής σε δεδομένα τύπου float.

• Μεταβλητό πεδίο δεδομένων

static: Μεταβλητή με ορατότητα στο πλαίσιο μιας συνάρτησης.

volatile: Μεταβλητή που φορτώνεται κατευθείαν από την μνήμη RAM κι όχι από κάποιον καταχωρητή αποθήκευσης. Στον προγραμματισμό του Arduino χρησιμοποιείται για να διευθετήσει ρουτίνες διακοπών.

const: Μεταβλητή που μπορεί απλά να διαβαστεί αλλά όχι να αλλαχθεί.

• Utilities

sizeof(): Επιστροφή του αριθμού από bytes ενός τύπου δεδομένων (μεταβλητής ή πίνακα).

PROGMEM: Αποθήκευση των δεδομένων στην μνήμη flash αντί της SRAM.

• Δομές ελέγχου ροής

if: Δομή ελέγχου υλοποίησης μιας συνθήκης.

if ... else: Δομή ελέγχου υλοποίησης πολλαπλών συνθηκών.

for: Δομή επαναληπτικού ελέγχου συνθήκης με δεδομένο αριθμό επαναλήψεων (πεπερασμένο ή ατέρμονα).

switch case: Δομή συνθήκης επιλογής ανάλογα με την τιμή κάποιας μεταβλητής.

while: Δομή επαναληπτικού ελέγχου συνθήκης όσο η μεταβλητή ελέγχου παραμένει true.

do...while: Δομή επαναληπτικού ελέγχου συνθήκης όσο η μεταβλητή ελέγχου παραμένει true. Ο βρόγχος θα εκτελεστεί τουλάχιστον μία φορά.

break: Εντολή διακοπής εκτέλεσης μιας επαναληπτικής δομής.

continue: Εντολή παράκαμψης μιας επανάληψης και συνέχιση εκτέλεσης της επόμενης.

return: Εντολή ολοκλήρωσης μιας συνάρτησης κι επιστροφή της τιμής που έχει οριστεί.

go to: Εντολή μετάβασης της ροής εκτέλεσης ενός προγράμματος σε κάποιο συγκεκριμένο σημείο του κώδικα (η εντολή αυτή αποφεύγεται σε δομημένα προγράμματα).

• Δεσμευμένοι τελεστές

; : Ολοκλήρωση εκτέλεσης μιας εντολής.

{ } : Δομή εισαγωγής κι εκτέλεσης ενός συνόλου εντολών.

// : Σχόλιο μίας μόνο γραμμής.

/**/ : Σχόλια που εκτείνονται σε περισσότερες της μίας γραμμής.

#define : Ορισμός σταθερών τιμών μεταβλητών.

#include : Εισαγωγή βιβλιοθήκης.

• Αριθμητικοί τελεστές

= : Τελεστής απόδοσης τιμής σε μεταβλητή.

+ : Τελεστής πρόσθεσης.

- : Τελεστής αφαίρεσης.

* : Τελεστής πολλαπλασιασμού.

/ : Τελεστής διαίρεσης.

% : Τελεστής modulo (υπόλοιπο διαίρεσης).

• Τελεστές σύγκρισης

== : Τελεστής ισότητας.

!= : Τελεστής ανισότητας.

< : Τελεστής μικρότερο από.

> : Τελεστής μεγαλύτερο από.

<= : Τελεστής μικρότερο ή ίσο.

>= : Τελεστής μεγαλύτερο ή ίσο.

• Λογικοί τελεστές

&& Τελεστής λογικού AND.

|| Τελεστής λογικού OR.

! Τελεστής λογικού NOT.

• Δυαδικοί τελεστές

& Δυαδικός τελεστής AND.

| Δυαδικός τελεστής OR.

^ Δυαδικός τελεστής XOR.

~ Δυαδικός τελεστής NOT.

<< Τελεστής αριστερής ολίσθησης.

>> Τελεστής δεξιάς ολίσθησης.

• Σύνθετοι τελεστές

++ Τελεστής αύξησης κατά 1.

-- Τελεστής ελάττωσης κατά 1.

+= Τελεστής σύνθετης πρόσθεσης ($x+=y$ ισοδυναμεί με $x=x+y$).

-= Τελεστής σύνθετης αφαίρεσης ($x-=y$ ισοδυναμεί με $x=x-y$).

= Τελεστής σύνθετου πολλαπλασιασμού ($x=y$ ισοδυναμεί με $x=x*y$).

/= Τελεστής σύνθετης διαίρεσης ($x/=y$ ισοδυναμεί με $x=x/y$).

%= Τελεστής σύνθετου modulo ($x\%=y$ ισοδυναμεί με $x=x\%y$).

&= Τελεστής σύνθετου AND ($x\&y$ ισοδυναμεί με $x=x\&y$).

|= Τελεστής σύνθετου OR ($x|=y$ ισοδυναμεί με $x=x|y$).

• Συναρτήσεις ψηφιακών εισόδων-εξόδων

I/O pinMode(): Ορισμός ενός ακροδέκτη (pin) ως είσοδο ή έξοδο.

digitalWrite(): Εγγραφή σε μια ψηφιακή επαφή εξόδου της λογικής στάθμης HIGH ή LOW.

digitalRead(): Ανάγνωση της λογικής τιμής ενός ψηφιακού ακροδέκτη.

• Συναρτήσεις αναλογικών εισόδων-εξόδων I/O

analogReference(): Ορισμός του επιπέδου της τάσης αναφοράς για τις αναλογικές εισόδους.

analogRead(): Ανάγνωση της αναλογικής τιμής ενός ακροδέκτη εισόδου.

analogWrite(): Εγγραφή μιας αναλογικής τιμής σε έναν ακροδέκτη εξόδου.

• Προηγμένες συναρτήσεις εισόδου-εξόδου I/O

tone(): Δημιουργία ενός τετραγωνικού σήματος δεδομένης συχνότητας.

noTone(): Διακοπή της παραγωγής τετραγωνικών σημάτων.

shiftOut(): Ολίσθηση των ψηφίων μιας μεταβλητής σε έναν ακροδέκτη εξόδου.

shiftIn(): Ολίσθηση των ψηφίων μιας μεταβλητής από έναν ακροδέκτη εισόδου.

pulseIn(): Επιστροφή της διάρκειας ενός παλμού (HIGH/LOW) σε ms.

• Συναρτήσεις χρόνου

millis(): Επιστροφή της διάρκειας εκτέλεσης του προγράμματος σε ms.

micros(): Επιστροφή της διάρκειας εκτέλεσης του προγράμματος σε ms.

delay(): Καθυστέρηση της εκτέλεσης του προγράμματος σε ms.

delayMicroseconds(): Καθυστέρηση της εκτέλεσης του προγράμματος σε ms.

• Μαθηματικές συναρτήσεις

min(): Επιστροφή του ελάχιστου ανάμεσα σε 2 αριθμούς.

max(): Επιστροφή του μέγιστου ανάμεσα σε 2 αριθμούς.

abs(): Επιστροφή της απόλυτης τιμής ενός αριθμού.

constrain(): Περιορισμός του εύρους διακύμανσης ενός αριθμού στα επιθυμητά όρια.

map(): Μετασχηματισμός ενός αριθμού δεδομένου εύρους σε διαφορετικό εύρος.

pow(): Επιστροφή του αποτελέσματος ύψωσης σε δύναμη μιας μεταβλητής.

sqrt(): Επιστροφή του αποτελέσματος της τετραγωνικής ρίζας μιας μεταβλητής.

• Τριγωνομετρικές συναρτήσεις

sin(): Υπολογισμός του ημιτόνου μιας μεταβλητής (σε radians).

cos(): Υπολογισμός του συνημίτονου μιας μεταβλητής (σε radians).

tan(): Υπολογισμός της εφαπτομένης μιας μεταβλητής (σε radians).

• Συναρτήσεις χαρακτήρων

isAlphaNumeric(): Επιστροφή

true: αν ένας χαρακτήρας είναι λατινικός χαρακτήρας ή αριθμός.

isAlpha(): Επιστροφή true αν ένας χαρακτήρας είναι λατινικός χαρακτήρας.

isAscii(): Επιστροφή true αν ένας χαρακτήρας είναι αριθμός Ascii.

isWhitespace(): Επιστροφή true αν ένας χαρακτήρας αποτελεί white space (ένα σύνολο χαρακτήρων που απαρτίζουν οριζόντια ή κάθετα τον χώρο στην τυπογραφία).

isControl(): Επιστροφή true αν ένας χαρακτήρας αποτελεί χαρακτήρα ελέγχου.

isDigit(): Επιστροφή true αν ένας χαρακτήρας είναι ψηφίο.

isGraph(): Επιστροφή true αν ένας χαρακτήρας είναι εκτυπώσιμος.

isLowerCase(): Επιστροφή true αν ένας χαρακτήρας είναι γραμμένος με “μικρά” γράμματα.

isPrintable(): Επιστροφή true αν ένας χαρακτήρας είναι εκτυπώσιμος.

isPunct(): Επιστροφή true αν ένας χαρακτήρας είναι σημείο στίξης.

isSpace(): Επιστροφή true αν ένας χαρακτήρας αντιστοιχεί στον χαρακτήρα κενού.

isUpperCase(): Επιστροφή true αν ένας χαρακτήρας είναι γραμμένος με κεφαλαία γράμματα.

isHexadecimalDigit(): Επιστροφή true αν ένας χαρακτήρας αποτελεί έγκυρο δεκαεξαδικό ψηφίο.

• Συναρτήσεις τυχαίων αριθμών

randomSeed(): Αρχικοποίηση της γεννήτριας ψευδοτυχαίων αριθμών.

random(): Υπολογισμός ψευδοτυχαίων αριθμών.

- **Συναρτήσεις επεξεργασίας δυαδικών αριθμών**

lowByte(): Επιστροφή του δεξιότερου byte μιας μεταβλητής.

highByte(): Επιστροφή του αριστερότερου byte μιας μεταβλητής.

bitRead(): Ανάγνωση ενός συγκεκριμένου bit από έναν αριθμό.

bitWrite(): Εγγραφή ενός συγκεκριμένου bit σε έναν αριθμό.

bitSet(): Εγγραφή λογικού 1 σε ένα συγκεκριμένο bit ενός αριθμού.

bitClear(): Εγγραφή λογικού 0 σε ένα συγκεκριμένο bit ενός αριθμού.

bit(): Υπολογισμός της τιμής ενός συγκεκριμένου bit (ανάλογα με το όρισμα που θα δεχτεί θα υπολογίσει την αντίστοιχη δύναμη του 2).

- **Συναρτήσεις ρουτινών εξυπηρέτησης εξωτερικών διακοπών**

attachInterrupt(): Ενεργοποίηση μιας ρουτίνας εξυπηρέτησης διακοπής.

detachInterrupt(): Απενεργοποίηση μιας ρουτίνας εξυπηρέτησης διακοπής.

- **Συναρτήσεις ενεργοποίησης/απενεργοποίησης διακοπών**

interrupts(): Ενεργοποίηση της δυνατότητας εμφάνισης διακοπών.

noInterrupts(): Απενεργοποίηση της δυνατότητας εμφάνισης διακοπών.

- **Συναρτήσεις επικοινωνίας**

Serial(): Ένα σύνολο χρήσιμων συναρτήσεων που επιτρέπουν την σειριακή επικοινωνία της πλακέτας.

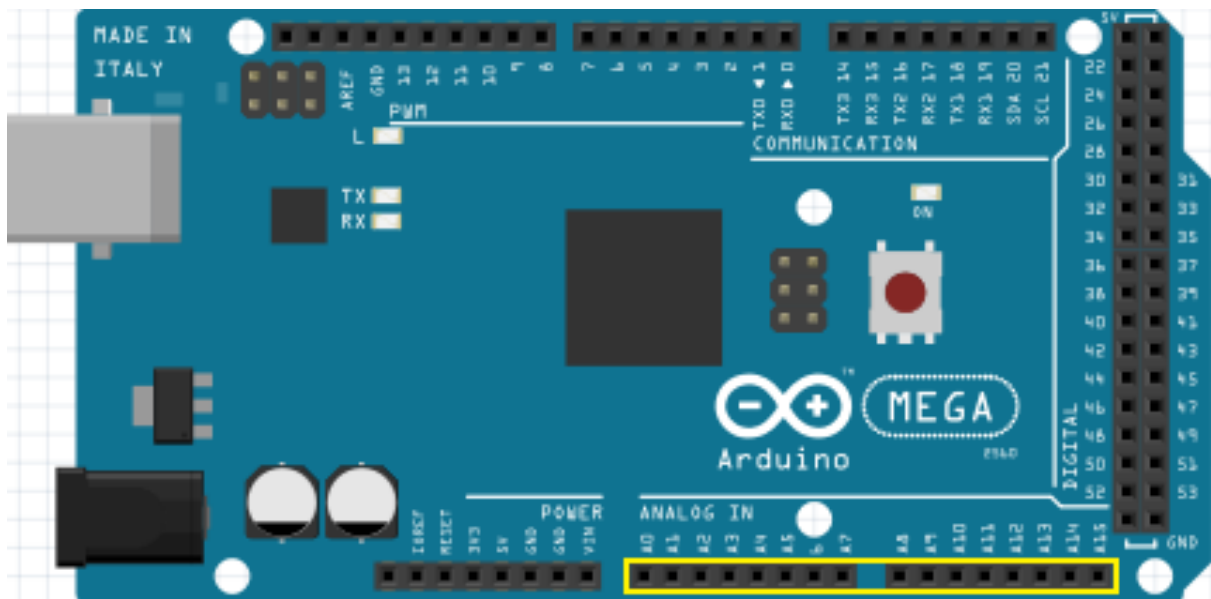
Stream(): Ένα σύνολο συναρτήσεων που ορίζει τις συναρτήσεις ανάγνωσης του Arduino.

3.6 Συναρτήσεις ακροδεκτών

3.6.1 Συναρτήσεις Αναλογικών ακροδεκτών

Οι συναρτήσεις ακροδεκτών απαριθμήθηκαν στην παράγραφο 3.5. Στην παράγραφο αυτή δίνονται οι απαραίτητες διευκρινήσεις για τον χειρισμό των εν λόγω συναρτήσεων.

Οι ακροδέκτες του Arduino μπορούν να ρυθμιστούν είτε ως είσοδοι είτε ως έξοδοι, όμως από προεπιλογή είναι ρυθμισμένες ως είσοδοι. Επίσης, αξίζει να σημειωθεί, ότι η πλειοψηφία των αναλογικών ακροδεκτών του Arduino (Atmega), μπορούν να ρυθμιστούν και να χρησιμοποιηθούν, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακοί ακροδέκτες.



Σχήμα 3.1: Αναλογικοί ακροδέκτες του Arduino Mega

Οι ελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο A/D μετατροπέα έξι καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10 bits, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία των αναλογικών ακροδεκτών για τους περισσότερους χρήστες Arduino είναι να διαβάσει αναλογικούς αισθητήρες, οι αναλογικοί ακροδέκτες έχουν επίσης όλες τις λειτουργίες των ακροδεκτών γενικής χρήσης εισόδου/εξόδου.

Οι συναρτήσεις αναλογικών εισόδων/εξόδων είναι οι παρακάτω:

analogRead(): Διαβάζει την τιμή από έναν συγκεκριμένο αναλογικό ακροδέκτη.

Σύνταξη: analogRead(pin)

Παράμετροι:

pin: Ο αριθμός του αναλογικού ακροδέκτη εισόδου από όπου θα διαβαστεί η τιμή.
Επιστρέφει: Ακέραιο αριθμό από 0 έως 1023.

analogWrite()

Γράφει μια αναλογική τιμή (PWM) σε έναν ακροδέκτη. Μπορεί να χρησιμοποιηθεί για παράδειγμα για να ανάψει ένα LED σε διάφορες φωτεινότητες ή να οδηγήσει ένα κινητήρα σε διάφορες ταχύτητες. Μετά από μια κλήση της analogWrite(), ο ακροδέκτης θα δημιουργήσει ένα σταθερό τετραγωνικό κύμα καθορισμένου κύκλου λειτουργίας μέχρι την επόμενη κλήση της analogWrite() ή μια κλήση της digitalWrite() ή digitalRead() για τον ίδιο ακροδέκτη. Η συχνότητα του σήματος PWM είναι περίπου 490 Hz. Στις περισσότερες πλατφόρμες Arduino η συνάρτηση αυτή λειτουργεί στα pins 3, 5, 6, 9, 10 και 11.

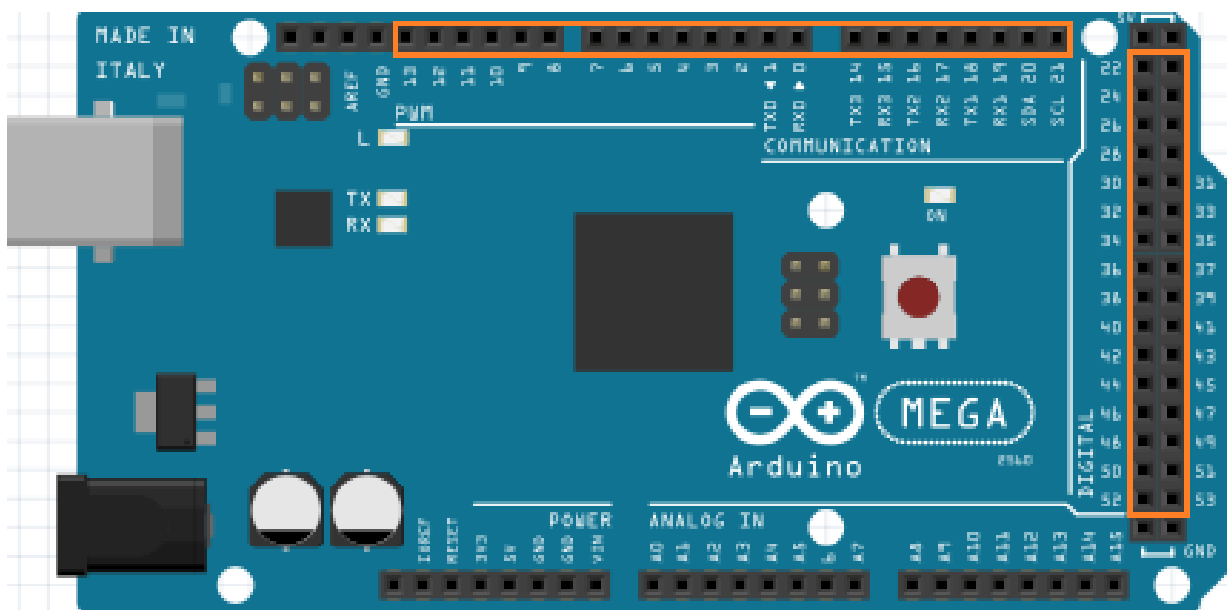
Σύνταξη: analogWrite(pin, value)

Παράμετροι:

pin: Ο αριθμός του ακροδέκτη στον οποίο θα εγγραφεί το PWM κύμα.

value: ο κύκλος λειτουργίας μεταξύ 0 και 255.

3.6.2 Συναρτήσεις Ψηφιακών ακροδεκτών



Σχήμα 3.2: Ψηφιακοί ακροδέκτες του Arduino Mega

Οι συναρτήσεις ψηφιακών εισόδων/εξόδων καταγράφονται παρακάτω:

pinMode()

Ορίζει έναν συγκεκριμένο ακροδέκτη (pin) να συμπεριφέρεται ως είσοδος/ έξοδος.

Σύνταξη: pinMode(pin, mode)

Παράμετροι:

- pin: Ο αριθμός του ακροδέκτη που επιθυμούμε να μεταβάλουμε.
- mode: INPUT/OUTPUT/INPUT_PULLUP

digitalWrite ()

Ορίζει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε έναν ψηφιακό ακροδέκτη. Αν ο ακροδέκτης έχει ρυθμιστεί ως έξοδος με την συνάρτηση pinMode(), τότε η τάση του θα καθοριστεί στην αντίστοιχη τιμή: 5V (ή 3.3V για τις πλακέτες που λειτουργούν με 3.3 V) για HIGH και 0V για LOW. Αν ο ακροδέκτης έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην συνάρτηση digitalWrite() θα ενεργοποιηθεί μια εσωτερική pull-up αντίσταση ενώ γράφοντας LOW θα απενεργοποιηθεί.

Σύνταξη: digitalWrite(pin, value)

Παράμετροι:

pin: Ο αριθμός του ακροδέκτη στον οποίο θα καταγραφεί το επιθυμητό αποτέλεσμα.

Value: HIGH/LOW

digitalRead()

Διαβάζει την τιμή ενός συγκεκριμένου ψηφιακού ακροδέκτη, που είναι είτε HIGH είτε LOW.

Σύνταξη: digitalRead(pin)

Παράμετροι:

pin: Ο αριθμός του ακροδέκτη που θα αναγνωστεί.

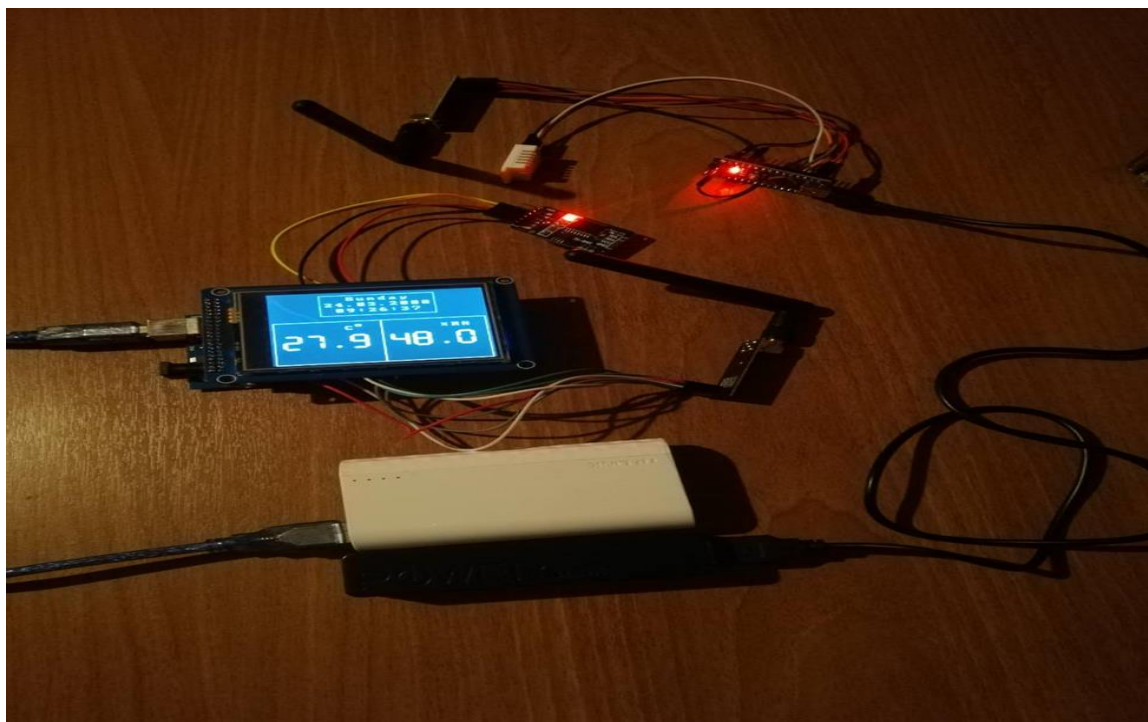
Επιστρέφει: HIGH/LOW[9]

Μετά την ανάλυση του περιβάλλοντος προγραμματισμού, ακολουθεί στο επόμενο κεφάλαιο η περιγραφή των εξαρτημάτων που χρησιμοποιήθηκαν και η μεταξύ τους συνδεσμολογία.

Κεφάλαιο 4 – Υλοποίηση της εργασίας

4.1 Λειτουργία του συστήματος

Αρχικά το σύστημα μας παίρνει μετρήσεις απομακρυσμένα από έναν αισθητήρα θερμοκρασίας και υγρασίας. Στην συνέχεια αυτές τις μετρήσεις τις μεταδίδει ασύρματα μέσω WiFi σε μία κεντρική μονάδα και τις εμφανίζει στην οθόνη. Ως αισθητήρα χρησιμοποιήσαμε τον DHT22, τα δεδομένα τα επεξεργάζεται ο Arduino Nano και τα στέλνει ασύρματα μέσω της NRF24L01+ Long Range Wireless πλακέτας στο Arduino Mega 2560 όπου πάνω σε αυτό έχουμε συνδεδεμένο μία πλακέτα απεικόνισης του πραγματικού χρόνου και την οθόνη που εμφανίζονται τα δεδομένα μας, δηλαδή η θερμοκρασία, η υγρασία, η ημερομηνία και η ώρα. Όταν γίνει η τροφοδοσία των δύο πλακετών στη οθόνη εμφανίζονται οι πληροφορίες του συγγραφέα μέσα σε ένα διάστημα 2 δευτερολέπτων και στην συνέχεια εμφανίζετε η ημερομηνία, η ώρα, η θερμοκρασία και η υγρασία. Μόλις ενεργοποιήσουμε το Arduino Mega κάθε 5 δευτερόλεπτα γίνεται ενημέρωση της θερμοκρασίας και της υγρασίας, αν όμως μέσα σε αυτά τα 5 δευτερόλεπτα για κάποιον λόγο(π.χ. λόγο εμβέλειας) χαθεί η σύνδεση μεταξύ των δύο πλακετών στην οθόνη εμφανίζεται το μήνυμα, “no connection”. Την οθόνη την έχουμε χωρίσει σε τρία τμήματα, το πάνω τμήμα που εμφανίζεται η ημερομηνία και η ώρα, το κάτω αριστερό τμήμα που εμφανίζεται η θερμοκρασία σε βαθμούς κελσίου και τέλος στα δεξιά του τμήματος της οθόνης η υγρασία. Στην συνέχεια γίνεται η ανάλυση των επιμέρους εξαρτημάτων και υλικών και η συνδεσμολογία τους.



Εικόνα 4.1: Το αποτέλεσμα της λειτουργίας του συστήματος

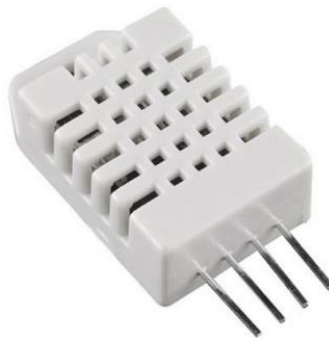
4.2 Υλικά

- 1 x Arduino Mega 2560 Rev3
- 1 x Arduino Nano
- 1 x DHT22
- 1 x DS3231 RTC
- 2 x NRF24L01+ Long Range Wireless
- 1 x TFT LCD Shield v2.0
- 1 x TFT_320QDT_9341
- 21 x Female To Female Dupont Jumper Wires Cable

Η εργασία μας αποτελείται από τα εξής υλικά, το Arduino Mega 2560 R3, το Arduino Nano, τον αισθητήρα DHT22, ένα ρολόι χρονισμού DS3231 RTC, δύο κεραίες NRF24L01+ Long Range Wireless, μία οθόνη TFT LCD 9341 με ένα TFT LCD Shield v2.0, καλώδια και δύο power banks για την τροφοδοσία των πλακετών.

Για το Arduino Mega και Nano αναφερθήκαμε αναλυτικά στο δεύτερο κεφάλαιο, ας δούμε τώρα αναλυτικά για τα υπόλοιπα εξαρτήματα.

4.2.1 Αισθητήρας DHT22



Εικόνα 4.2: Αισθητήρας DHT22

Ο λόγος που επιλέξαμε τον συγκεκριμένο αισθητήρα είναι γιατί έχει μικρό μέγεθος και μικρή απόκλιση από τις πραγματικές τιμές θερμοκρασίας και υγρασίας, άρα η χρήση του εν λόγω αισθητήρα είναι η ιδανική δεδομένου του κόστους του. Λόγω αυτών των ιδιοτήτων του έχει χρησιμοποιηθεί σε πολλές εφαρμογές και υπάρχει πληθώρα βιβλιοθηκών στο διαδίκτυο.[11]

• Τεχνικά χαρακτηριστικά

	Υγρασία	Θερμοκρασία
Τάση	3,3 – 5,5 V	3,3 – 5,5 V
Έξοδος	Ψηφιακή	Ψηφιακή
Αισθητήρας	Πολυμερής Πυκνωτής	DS18B20
Εύρος	0 – 100% Rh	-40 – 80 C°
Ακρίβεια	2 – 5 %	± 0,5 C°
Δειγματοληψία	0,5 Hz	0,5 Hz

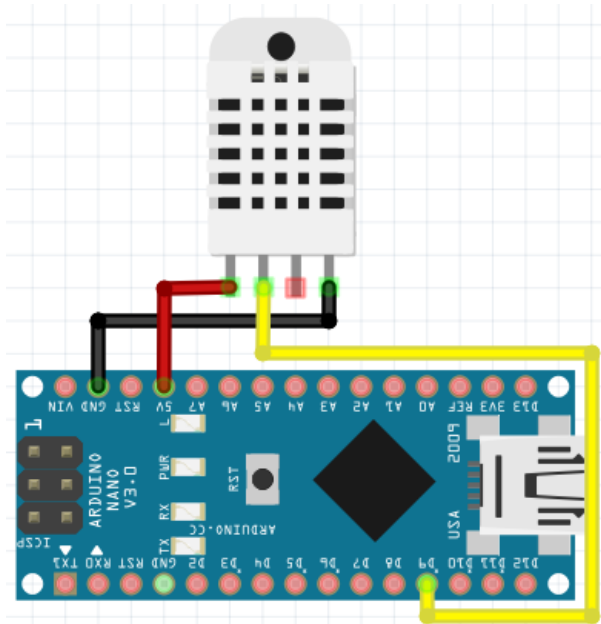
Πίνακας 4.1: Τεχνικά χαρακτηριστικά αισθητήρα DTH22

Κάθε δύο δευτερόλεπτα ο αισθητήρας αποθηκεύει προσωρινά 40 bit πληροφορίας τα οποία αντιστοιχούν σε:

- 8 bit για την τιμή της υγρασίας
- 8 bit για την ακέραια τιμή της υγρασίας
- 8 bit για την τιμή της θερμοκρασίας
- 8 bit για την ακέραια τιμή της θερμοκρασίας
- 8 bit για το bit ελέγχου ισοτιμίας των δεδομένων.[11]

• Συνδεσμολογία με την πλακέτα Arduino

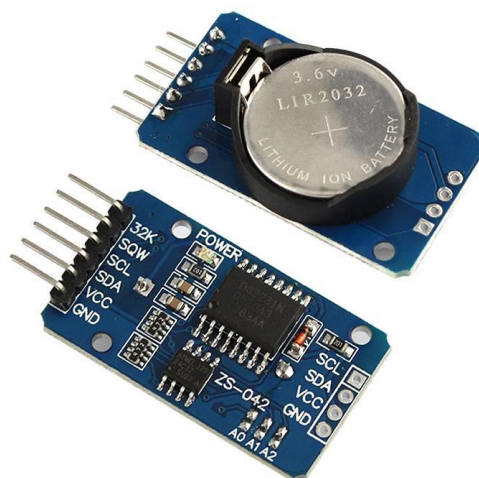
Η συνδεσμολογία μεταξύ του αισθητήρα και της πλακέτας Arduino είναι αρκετά απλή. Στον ακροδέκτη 1 η τάση (5v ή 3.3v), στον ακροδέκτη 4 του αισθητήρα συνδέεται η γείωση (ground) και στον ακροδέκτη 2 μία από τις ψηφιακές εισόδους του Arduino, στην εργασία αυτή έχει συνδεθεί στην θύρα 9. Στο σχήμα βλέπουμε τον τρόπο συνδεσμολογίας του αισθητήρα με την πλακέτα Arduino.[11]



Σχήμα 4.1: Συνδεσμολογία του αισθητήρα με την πλακέτα

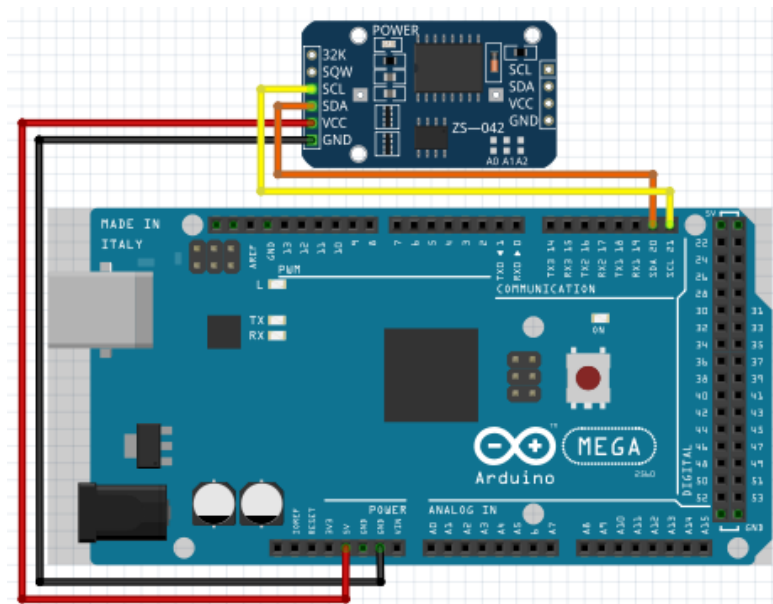
4.2.2 DS3231 RTC

Το ρολόι χρονισμού RTC (Real Time Clock) DS – 3231 είναι μία συσκευή παραγωγής πραγματικού χρόνου με εξαιρετική ακρίβεια. Η συσκευή ενσωματώνει μία είσοδο μπαταρίας διατηρώντας την ακριβή χρονομέτρηση όταν η κύρια τροφοδοσία έχει διακοπεί. Ο τύπος της μπαταρίας που χρησιμοποιείται είναι CR – 2032 σε τάση λειτουργίας τα 3V. Ο λόγος που χρησιμοποιήσαμε αυτό το ρολόι χρονισμού είναι γιατί θέλουμε να εμφανίζεται στην οθόνη η ώρα. [12]



Εικόνα 4.3: Ρολόι χρονισμού RTC DS-3241

- Συνδεσμολογία



Σχήμα 4.2: Συνδεσμολογία Ρολογιού RTC DS - 3231 με την πλακέτα Arduino

4.2.3 NRF24L01+ Long Range Wireless

Το NRF24L01+ το χρησιμοποιήσαμε για την επικοινωνία μεταξύ των δύο πλακετών μας. Χρησιμοποιεί το πομποδέκτη από το Nordic Semiconductor, το NRF24L01+. Αυτός ο πομποδέκτης IC λειτουργεί στη ζώνη 2.4GHz. Επίσης διαθέτει την κεραία 2.4G (2dB), με ταχύτητα μετάδοσης 250Kbps σε περιοχή ανοιχτού χώρου φτάνουν στην απόσταση επικοινωνίας περίπου 800-1000 μέτρα. Ο λόγος που χρησιμοποιήσαμε την συγκεκριμένη πλακέτα είναι η μεγάλη εμβέλεια που διαθέτει.[13]



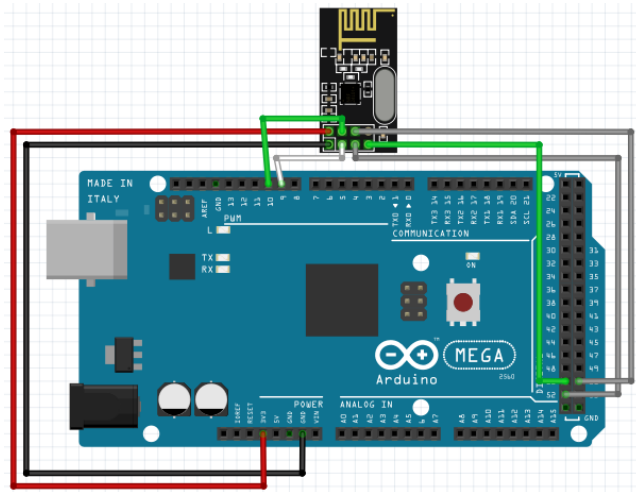
Εικόνα 4.4: NRF24L01+ Long Range Wireless

• Τεχνικά χαρακτηριστικά

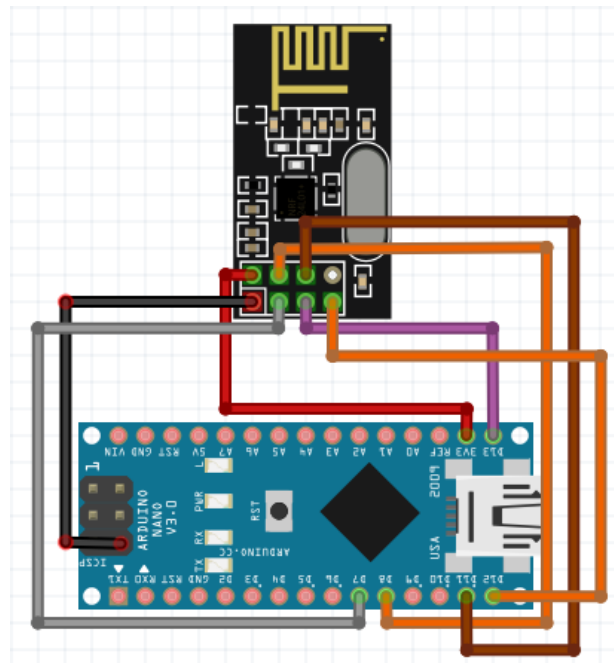
Τάση	3 - 3.6V (συνιστάται 3.3V)
Μέγιστη ισχύς εξόδου	+ 20dBm
Ρεύμα λειτουργίας εκπομπής	115mA
Τρόπος λήψης ρεύματος	45mA
Ρεύμα κατάστασης λειτουργίας	4.2uA
Λειτουργία ευαισθησίας 2Mbps σε λήψη	-92dBm
Λειτουργία ευαισθησίας 1Mbps σε λήψη	-95dBm
Λειτουργία ευαισθησίας 250kbps σε λήψη	-104dBm
Κέρδος PA	20Db
Κέρδος LNA	10Db
Σχήμα θορύβου LNA	2.6Db
Κέρδος κεραίας (κορυφή)	2Dbi
Ποσοστό 2MB (Περιοχή ανοιχτού χώρου)	520μ
Ποσοστό 1MB (Περιοχή ανοιχτού χώρου)	750μ
Ποσοστό 250Kb (Περιοχή ανοιχτού χώρου)	1000μ
Θερμοκρασία λειτουργίας	-20 έως 70 °C

Πίνακας 4.2: Τεχνικά χαρακτηριστικά NRF24l01+

- Συνδεσμολογία



Σχήμα 4.3: Συνδεσμολογία με την πλακέτα Arduino Mega



Σχήμα 4.4: Συνδεσμολογία με την πλακέτα Arduino Nano

4.2.4 Οθόνη TFT_320QDT_9341

Για την εργασία χρησιμοποιήσαμε μία οθόνη TFT LCD 3.2” με ανάλυση 320x240. Αναλυτικά έχει τα εξής χαρακτηριστικά:

- Η οθόνη αυτή είναι αφής και δέχεται SD card.
- LCD chip ελέγχου ILI9341.
- Chip πίνακα ελέγχου XPT2046.
- Υποστηρίζει διεπαφή δεδομένων 8 / 16bit.
- Είναι συμβατό με τα Arduino STM32, AVR και 8051.
- 16bit παράλληλη διασύνδεση για τη διασύνδεση οθόνης.
- Υποστήριξη διασύνδεσης διαύλου bus 6800 και 8080.
- Υποστήριξη διασύνδεσης διαύλου STM32 FSMC.
- Χαμηλή τάση ρεύματος, συμβατή με τροφοδοσία 5V και 3.3V.
- Ενσωματωμένο κύκλωμα ελέγχου οπίσθιου φωτισμού PWM, μπορεί να υποστηρίξει τον έλεγχο I/O και τον έλεγχο φωτισμού PWM.
- Υποστήριξη οριζόντιας και κάθετης οθόνης.
- Υποστήριξη 65K RGB χρώματα.[14]



Εικόνα 4.4: TFT_320QDT_9341

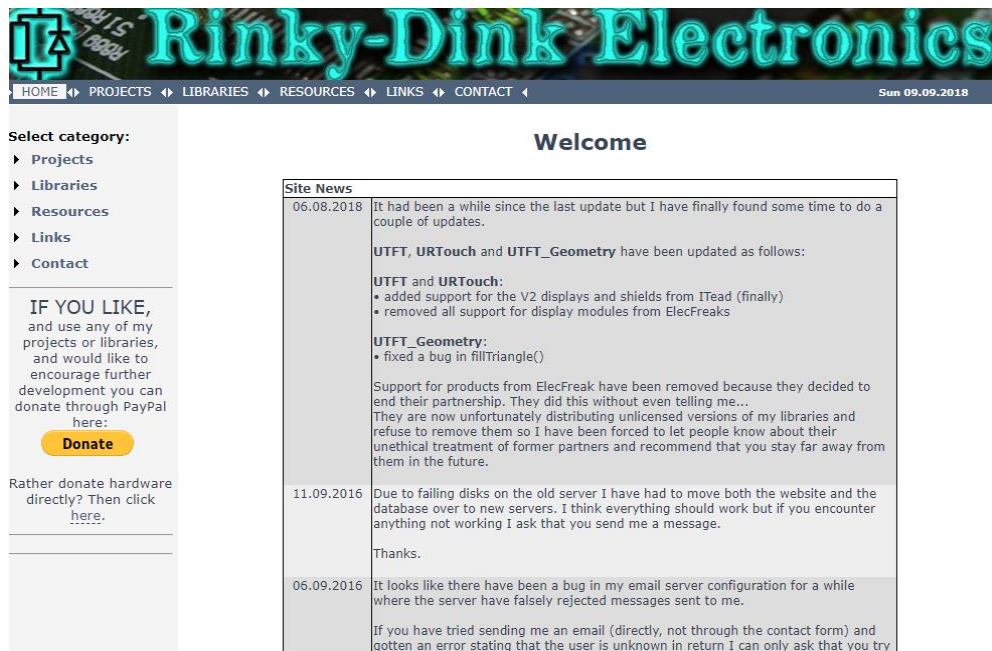
• Παράδειγμα κώδικα για την οθόνη

Το περιβάλλον Arduino μπορεί να επεκταθεί με τη χρήση βιβλιοθηκών, ακριβώς όπως οι περισσότερες πλατφόρμες προγραμματισμού. Οι δυνατότητες των βιβλιοθηκών είναι απεριόριστες γιατί ο καθένας μπορεί να φτιάξει δικές του βιβλιοθήκες για το Arduino.

Πριν αρχίσουμε να γράφουμε τον κώδικα μας θα πρέπει να επιλέξουμε τις βιβλιοθήκες που χρειαζόμαστε και αν αυτές δεν υπάρχουν προεγκατεστημένες τότε θα πρέπει να τις δημιουργήσουμε οι ίδιοι μας ή να τις κατεβάσουμε από το ίντερνετ. Για την οθόνη που χρησιμοποιούμε στην εργασία μας θα πρέπει να την κατεβάσουμε από το διαδίκτυο.

Παρακάτω θα δούμε τα βήματα που χρειάζονται για να την εγκαταστήσουμε και στην συνέχεια θα αναφερθούμε και σε κάποια παραδείγματα για το πώς μπορούμε να την χρησιμοποιήσουμε.

- Αρχικά ανοίγουμε έναν browser π.χ. το chrome και στην συνέχεια επισκεπτόμαστε την εξής ιστοσελίδα, <http://rinkydinkelectronics.com/>.



Select category:

- ▶ Projects
- ▶ Libraries
- ▶ Resources
- ▶ Links
- ▶ Contact

IF YOU LIKE,
and use any of my
projects or libraries,
and would like to
encourage further
development you can
donate through PayPal
here:
Donate

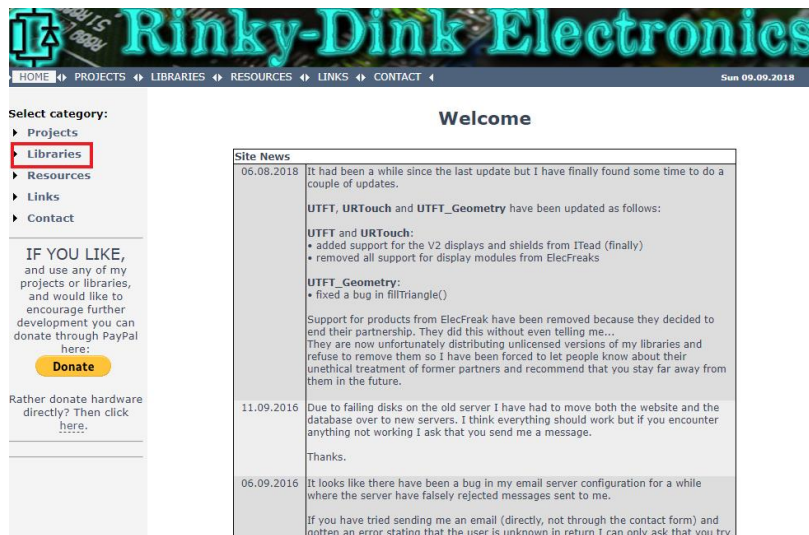
Rather donate hardware
directly? Then click
here.

Welcome

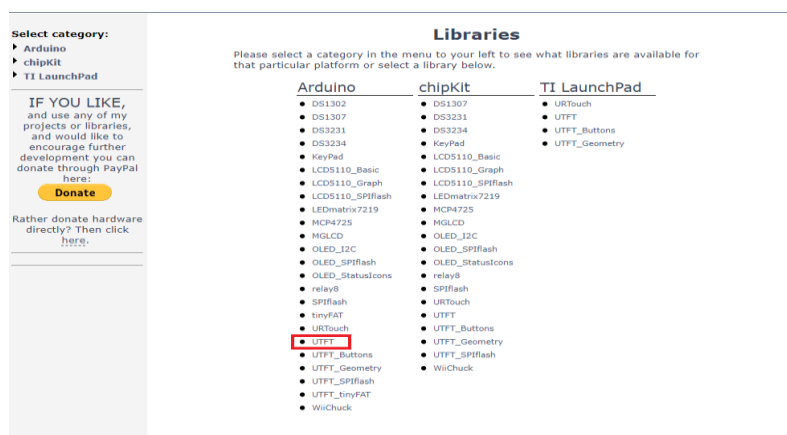
Site News

06.08.2018	It had been a while since the last update but I have finally found some time to do a couple of updates. UTFT, URTouch and UTFT_Geometry have been updated as follows: UTFT and URTouch: • added support for the V2 displays and shields from ITEAD (finally) • removed all support for display modules from ElecFreaks UTFT_Geometry: • fixed a bug in fillTriangle() Support for products from ElecFreak have been removed because they decided to end their partnership. They did this without even telling me... They are now unfortunately distributing unlicensed versions of my libraries and refuse to remove them so I have been forced to let people know about their unethical treatment of former partners and recommend that you stay far away from them in the future.
11.09.2016	Due to failing disks on the old server I have had to move both the website and the database over to new servers. I think everything should work but if you encounter anything not working I ask that you send me a message. Thanks.
06.09.2016	It looks like there have been a bug in my email server configuration for a while where the server have falsely rejected messages sent to me. If you have tried sending me an email (directly, not through the contact form) and gotten an error stating that the user is unknown in return I can only ask that you try

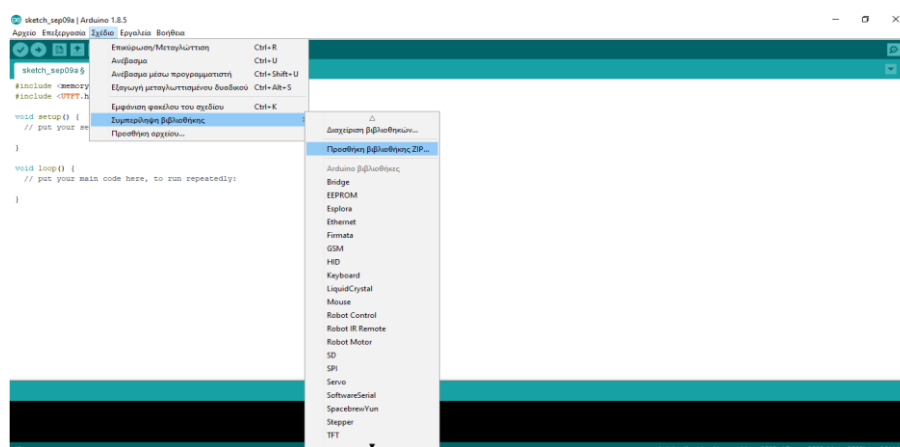
- Στην συνέχεια επιλέγουμε Libraries.



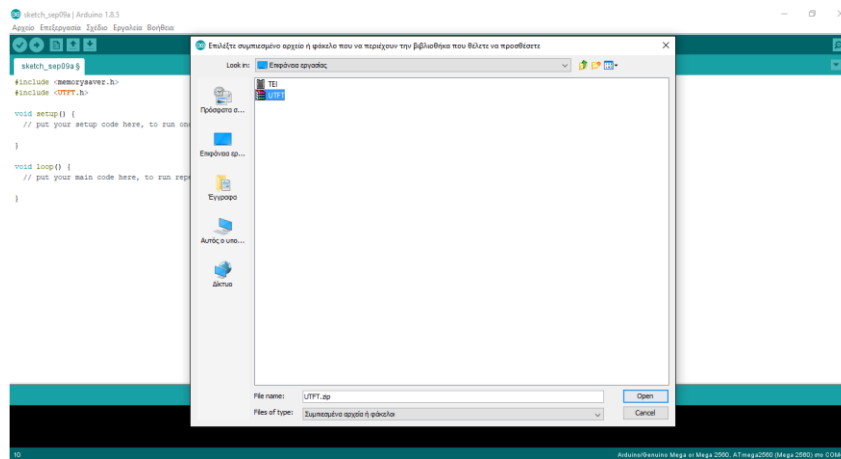
- Και τέλος επιλέγουμε UTFT και στην συνέχεια download.



- Στην συνέχεια και αφού κατέβει η βιβλιοθήκη μας στο περιβάλλον του Arduino επιλέγουμε το Σχέδιο το οποίο βρίσκεται στο Menu Bar, έπειτα Συμπερίληψη Βιβλιοθήκης και μετά Προσθήκη Βιβλιοθήκης ZIP όπως στην εικόνα παρακάτω.



- Τέλος, βρίσκουμε την βιβλιοθήκη σε μορφή zip που κατεβάσαμε και στην συνέχεια πατώντας Open εγκαθίστατε στο Arduino IDE .



• Παραδείγματα

```
void display_tables()
```

```
{  
    myGLCD.drawRect(5, 100, 160, 315); // Δημιουργία τετραγώνου  
    myGLCD.drawRect(161, 100, 315, 315); // Δημιουργία τετραγώνου  
}
```

```
void drawHomeScreen() {
```

```
    myGLCD.setBackColor(0,0,0); // Ορίζει το χρώμα φόντου όπου το κείμενο θα  
    εμφανιστεί σε μαύρο χρώμα.
```

```
    myGLCD.setColor(255, 255, 255); // Ορίζει το χρώμα σε λευκό
```

```
    myGLCD.setFont(BigFont); // Ορίζει την γραμματοσειρά.
```

```
    myGLCD.print("Arduino TFT Tutorial", CENTER, 10); // Εκτυπώνει τη  
    συμβολοσειρά στην οθόνη.
```

```
    myGLCD.clrScr(); // Καθαρισμός της οθόνης από τα υπάρχον μηνύματα.
```

```
}
```

4.2.5 TFT LCD Shield για το Arduino Mega 2560

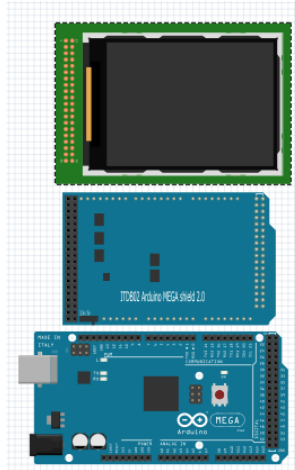
Αυτό το shield είναι συμβατό με το Arduino Mega 2560 και διευκολύνει τη σύνδεση μεταξύ της οθόνης και της πλακέτας. Λειτουργεί σε 3.3V και υποστηρίζει τη λειτουργία 16 bit. Η χρησιμοποίηση αυτού του shield στην εργασία μας διευκολύνει γιατί απαλλαγόμαστε από την χρήση πολλών καλωδίων για την σύνδεση της οθόνης και της πλακέτας και έτσι εξοικονομούμαι τα pins που χρειάζονται για την σύνδεσή τους.[15]



Εικόνα 4.6: TFT LCD Shield για το Arduino Mega2560

- **Συνδεσμολογία**

Η συνδεσμολογία είναι αρκετά απλή, αρχικά τοποθετούμε το shield πάνω στο Arduino Mega και στην συνέχεια τοποθετούμε την οθόνη πάνω στο shield.

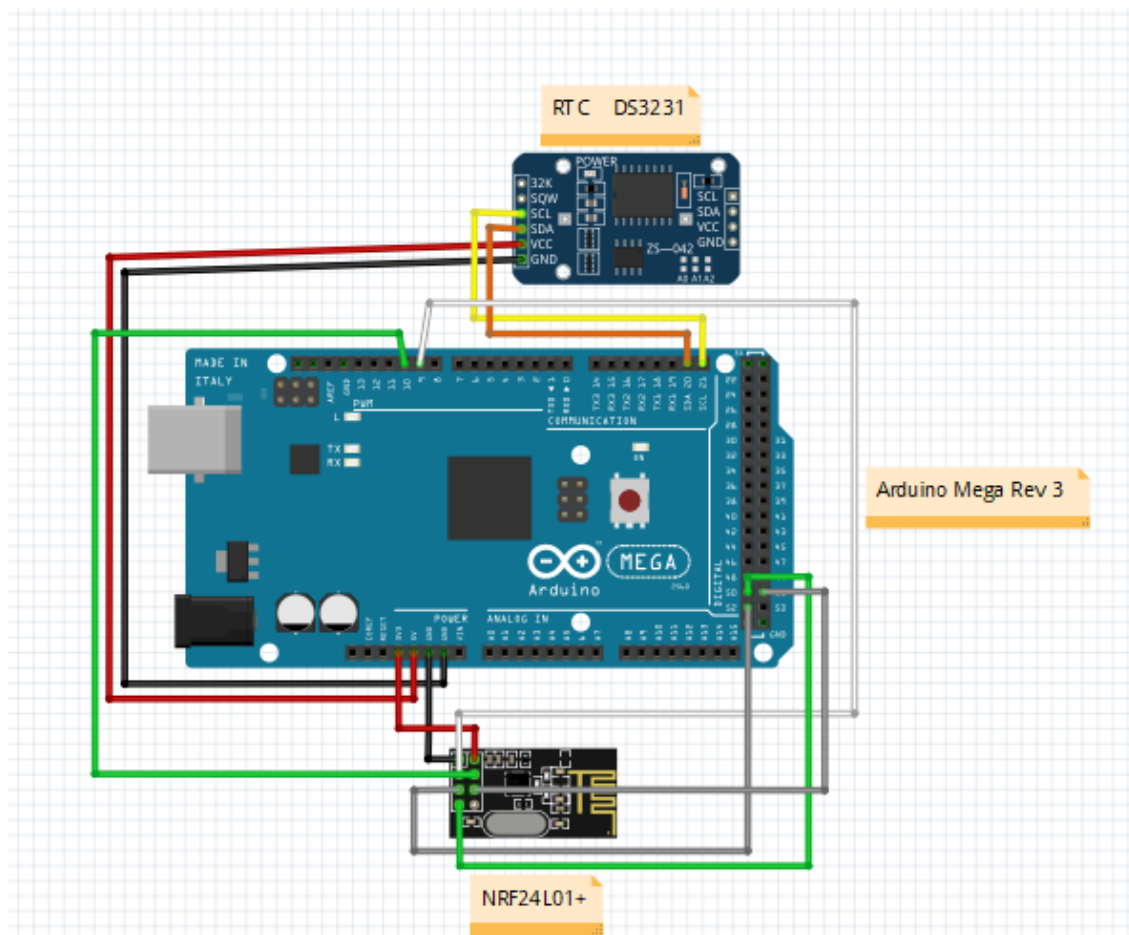


Σχήμα 4.5: Συνδεσμολογία οθόνης – shield – Arduino mega

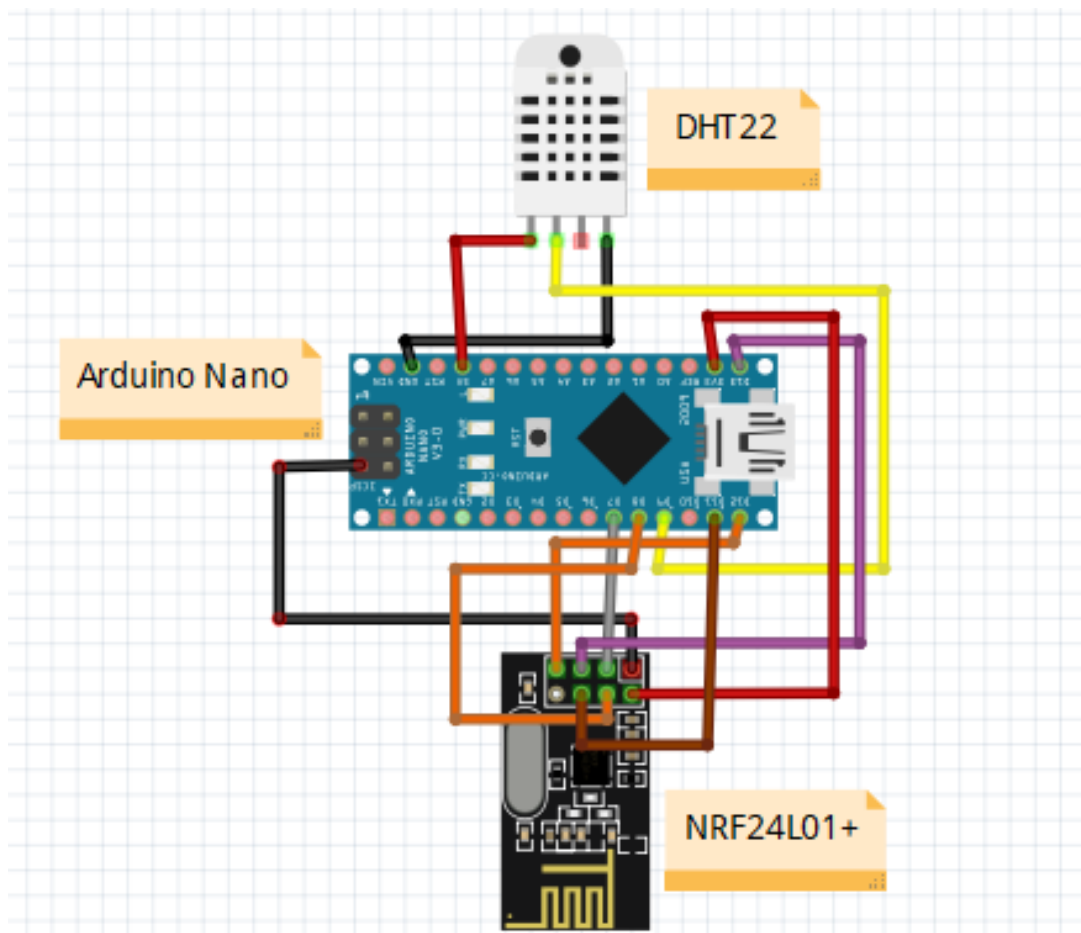
4.3 Συνδεσμολογία στο πρόγραμμα Fritzing

Το Fritzing άρχισε ως μια πρωτοβουλία της σχολής Εφαρμοσμένων επιστημών του Πότσδαμ στην Γερμανία. Η εφαρμογή Fritzing είναι ανοιχτού κώδικα και δημιουργήθηκε βάση της γλώσσας Processing και του μικροελεγκτή Arduino, έχει σαν στόχο να βοηθήσει σχεδιαστές, ερευνητές και ερασιτέχνες να κάνουν το βήμα από το αρχικό πρωτότυπο στο τυπωμένο κύκλωμα. [16]

Στο παρακάτω σχήμα βλέπουμε την συνδεσμολογία των υλικών για την δημιουργία του ολοκληρωμένου συστήματος με την βοήθεια του Fritzing. Η τροφοδοσία των δύο πλακετών γίνεται με power banks.



Σχήμα 4.6: Συνδεσμολογία Arduino Mega 2560



Σχήμα 4.7: Συνδεσμολογία Arduino Nano

Στο επόμενο κεφάλαιο ακολουθεί ο κώδικας για την λειτουργία του συστήματος.

Κεφάλαιο 5 - Κώδικας υλοποίησης της εργασίας

5.1 Κώδικας του Arduino Mega 2560 Rev3

```
1 /* Εισαγωγή βιβλιοθηκών */
2 #include <UTFT.h>
3 #include <DS3231.h>
4 #include <SPI.h>
5 #include <nRF24L01.h>
6 #include <RF24.h>
7
8
9 #if defined(__AVR__)
10 #include <avr/pgmspace.h>
11 #define fontdatatype const uint8_t
12 #elif defined(__PIC32MX__)
13 #define PROGMEM
14 #define fontdatatype const unsigned char
15 #elif defined(__arm__)
16 #define PROGMEM
17 #define fontdatatype const unsigned char
18 #endif
19
20
21 /* Προγραμματισμοί για την οθόνη υγρών κρυστάλλων */
22 fontdatatype SevenSegNumFontMDS[2396] PROGMEM = {
23     0x20, 0x2E, 0x2D, 0x0D,
24     0x00, 0x00, 0x00, [...] 0x00, 0x00, 0x00, // -
25     0x00, 0x00, 0x00, [...] 0x00, 0x00, 0x00, // . decimal point
26     0x00, 0x00, 0x00, [...] 0x00, 0x00, 0x00, // / = Blank space
27     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 0
28     0x00, 0x00, 0x00, [...] 0x00, 0x00, 0x00, // 1
29     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 2
30     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 3
31     0x00, 0x00, 0x00, [...] 0x00, 0x00, 0x00, // 4
32     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 5
33     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 6
34     0x00, 0xFF, 0xFE, [...] 0x80, 0x00, 0x00, // 7
35     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 8
36     0x00, 0xFF, 0xFE, [...] 0x80, 0xFE, 0x00, // 9
37 };
38
39 /* Ορίζουμε τις ψηφιακές εισόδους/εξόδους για τις συσκευές */
40 UTFT myGLCD(CTE32_R2, 38, 39, 40, 41); /* Οθόνη υγρών κρυστάλλων */
41 DS3231 rtc(SDA, SCL); /* Ρολόι πραγματικού χρόνου (Το πρωτόκολλο
42 επικοινωνίας με το ρολόι είναι I2C */
43 RF24 radio(9, 10); /* Συσκευή ασύρματης επικοινωνίας RX/TX */
44
45 unsigned long interval = 1000; /* Μεταβλητή για Χρονοκαθυστέρηση */
46 unsigned long previousMillis = 0; /* Μεταβλητή για Χρονοκαθυστέρηση
47 */
48 byte lastrecieve = 0;
49 byte timeout = 0;
50
```

```

51 extern uint8_t BigFont[];
52 extern unsigned short info[0x400];
53 const byte address[6] = "00001";
54 char buffer[12]; /* buffer για προσωρινή αποθήκευση */
55 char *p = NULL;
56
57
58 /* Η συνάρτηση setup εκτελείται μια μονό φορά στο ξεκίνημα του
59 μικροεπεξεργαστή και αρχικοποιεί τις συσκευές,
60 * όπως οθόνη, ραδιοεπικοινωνίες, αισθητήρες κ.λπ.
61 */
62 void setup()
63 {
64   randomSeed(analogRead(0));
65   rtc.begin(); /* Αρχικοποίηση του ρολογιού πραγματικού χρόνου */
66   radio.begin(); /* Αρχικοποίηση της μονάδας ραδιοεπικοινωνίας */
67   radio.openReadingPipe(0, address); /* Ορισμός του καναλιού που θα
68 γίνεται η επικοινωνία */
69   radio.setPALevel(RF24_PA_MIN);
70   radio.startListening();
71   myGLCD.InitLCD(); /* Αρχικοποίηση της οθόνης υγρών κρυστάλλων */
72   myGLCD.setFont(BigFont); /* Ορισμός της γραμματοσειράς για την
73 εμφάνιση των μηνυμάτων */
74   display_welcome(); /* Εκτέλεση της συνάρτησης για την εμφάνιση του
75 μηνύματος εισαγωγής */
76   display_datetime(); /* Εκτέλεση της συνάρτησης για την εμφάνιση
77 του μηνύματος ημερομηνίας/ώρας */
78   display_tables(); /* Εκτέλεση της συνάρτησης για την εμφάνιση των
79 τετραγώνων στην οθόνη */
80
81 }
82
83 /* Η συνάρτηση loop εκτελείται συνέχεια από τον μικροεπεξεργαστή και
84 εδώ βάζουμε όλες τις λειτουργίες
85 * που πρόκειται να κάνει το πρόγραμμα μας.
86 */
87
88 void loop()
89 {
90   unsigned long currentMillis = millis(); /* Αποθηκεύουμε τον
91 τρέχοντα χρόνο */
92
93   /* Ελέγχουμε αν έχει υπερβεί το χρονικό όριο που έχουμε θέσει το
94 οποίο είναι 1 δευτερόλεπτο
95   * και εκτελούμε τη συνάρτηση η οποία ανανεώνει τις πληροφορίες
96 της οθόνης
97   */
98   if ((unsigned long)(currentMillis - previousMillis) >= interval) {
99
100     if (timeout == 0) {
101       display_update_date();
102     }
103
104     display_temperature_humidity(); /* Εκτέλεση της συνάρτησης που
105 ενημερώνει τις τρέχουσες τιμές των αισθητήρων στην οθόνη */
106
107     previousMillis = millis(); /* Αποθηκεύουμε τον τρέχοντα χρόνο */

```

```

108 }
109
110
111 }
112
113 /* Βοηθητική συνάρτηση διαγραφής ενός χαρακτήρα μέσα από έναν πίνακα
114 χαρακτήρων */
115 void delete_char(char *str, int i) {
116     int len = strlen(str);
117
118     for (; i < len - 1 ; i++)
119     {
120         str[i] = str[i + 1];
121     }
122
123     str[i] = '\0';
124 }
125
126 /* Συνάρτηση εμφάνισης των τετραγώνων στην οθόνη */
127 void display_tables()
128 {
129     /* Εκτελούμε τις συναρτήσεις από τη βιβλιοθήκη του κατασκευαστή
130     για την εμφάνιση των τετραγώνων */
131     myGLCD.drawRect(5, 100, 160, 315);
132     myGLCD.drawRect(161, 100, 315, 315);
133 }
134
135 /* Συνάρτηση εμφάνισης των τιμών των αισθητήρων
136 θερμοκρασίας/υγρασίας */
137 void display_temperature_humidity(void)
138 {
139     char *temp = NULL;
140     char *hum = NULL;
141
142     /* Ελέγχουμε αν υπάρχει εισερχόμενο πακέτο από τη μονάδα της
143     ραδιοεπικοινωνίας */
144     if (radio.available()) {
145         memset(&buffer, 0, sizeof(buffer)); /* Καθαρίζουμε την προσωρινή
146         μνήμη ώστε να αποθηκεύσουμε το πακέτο */
147         radio.read(&buffer, sizeof(buffer)); /* Εκτελούμε τη συνάρτηση
148         από τη βιβλιοθήκη του κατασκευαστή για να λάβουμε το πακέτο και να
149         το υποθηκεύσουμε στον buffer */
150
151         if (timeout == 1) {
152             myGLCD.clrScr();
153             display_datetime();
154             display_tables();
155         }
156
157         timeout = 0;
158         /* Το εισερχόμενο πακέτο περιέχει σε μορφή string την
159         θερμοκρασία και την υγρασία σε μορφή (θερμοκρασία,υγρασία π.χ.
160         28,35) */
161         /* Ξεχωρίζουμε τις δυο τιμές από την κόμμα και τις αποθηκεύουμε
162         στις μεταβλητές */
163         p = strtok(buffer, ",");

```

```

165     temp = p;
166
167     delete_char(temp, 4);
168
169     p = strtok(NULL, ",");
170     hum = p;
171     delete_char(hum, 4);
172
173
174     if (timeout == 0) {
175         myGLCD.setFont(SevenSegNumFontMDS); /*Ορίζουμε τη
176 γραμματοσειρά για την οθόνη υγρών κρυστάλλων */
177         myGLCD.print(temp, 15, 145); /* Εμφανίζουμε την τιμή της
178 θερμοκρασίας */
179         myGLCD.print(hum, 170, 145); /* Εμφανίζουμε την τιμή της
180 υγρασίας */
181         myGLCD.setFont(BigFont); /*Ορίζουμε τη γραμματοσειρά για την
182 οθόνη υγρών κρυστάλλων */
183         myGLCD.print("C", 105, 115); /* Εμφανίζουμε το σύμβολο για τη
184 θερμοκρασία Celcius */
185         myGLCD.print("o", 120, 105);
186         myGLCD.print("%RH", 245, 115); /* Εμφανίζουμε το σύμβολο για
187 την υγρασία. Ποσοστό υγρασίας*/
188         lastrecieve = 0; /* Μηδενίζουμε την τιμή της τελευταίας λήψης
189 πακέτου */
190
191     }
192 }
193
194 lastrecieve++; /* Αυξάνουμε κατά ένα την μεταβλητή της τελευταίας
195 λήψης πακέτου */
196
197 /* Ελέγχουμε πότε έγινε τελευταία φορά λήψη πακέτου από την
198 απομακρυσμένη μονάδα,
199 * και αν δεν λάβουμε πακέτο σε λίγο χρονικό διάστημα εμφανίζουμε
200 στην οθόνη
201 * μήνυμα ότι δεν υπάρχει σύνδεση με την απομακρυσμένη μονάδα.
202 * Είτε γιατί η μονάδα είναι κλειστή, είτε γιατί έχουμε
203 απομακρυνθεί κατά πολύ και δεν επαρκεί το σήμα.
204 */
205 if (lastrecieve > 10) {
206     myGLCD.clrScr(); /* Καθαρισμός της οθόνης από τα υπάρχον
207 μηνύματα */
208     myGLCD.setColor(255, 0, 0); /* Αλλαγή χρωματισμού των γραμμών
209 */
210     myGLCD.print("No Connection", CENTER, 130); /* Εμφάνιση
211 μηνύματος */
212     timeout = 1;
213 }
214
215 }
216
217 /* Συνάρτηση εμφάνισης ημερομηνίας και ώρας από το ρολόι πραγματικού
218 χρόνου */
219 void display_update_date(void)
220 {

```

```

221 myGLCD.setFont(BigFont); /* Ορισμός γραμματοσειράς για την οθόνη
222 */
223 myGLCD.print(rtc.getDOWStr(), CENTER, 15); /* Εμφάνιση ονόματος
224 ημέρας της εβδομάδας */
225 myGLCD.print(rtc.getDateStr(), CENTER, 35); /* Εμφάνιση
226 ημερομηνίας */
227 myGLCD.print(rtc.getTimeStr(), CENTER, 55); /* Εμφάνιση ώρας */
228 }
229
230 /* Συνάρτηση εμφάνισης ημερομηνίας και ώρας από το ρολόι πραγματικού
231 χρόνου */
232 void display_datetime(void)
233 {
234 myGLCD.setColor(255, 255, 255); /* Ορίζουμε λευκό χρώμα για τα
235 γράμματα της οθόνης */
236 myGLCD.setBackgroundColor(0, 0, 0); /* Ορίζουμε μαύρο χρώμα για τον
237 φόντο της οθόνης */
238 myGLCD.drawRect(75, 10, 245, 80); /* Σχεδιάζουμε το τετράγωνο που
239 θα εμφανιστούν οι τιμές */
240 myGLCD.print(rtc.getDOWStr(), CENTER, 15);
241 myGLCD.print(rtc.getDateStr(), CENTER, 35);
242 myGLCD.print(rtc.getTimeStr(), CENTER, 55);
243 }
244
245 /* Συνάρτηση εμφάνισης μηνύματος εισαγωγής */
246 void display_welcome(void)
247 {
248 myGLCD.clrScr(); /* Καθαρισμός της οθόνης */
249 myGLCD.setColor(255, 255, 255); /* Ορίζουμε λευκό χρώμα για τα
250 γράμματα της οθόνης */
251 myGLCD.setBackgroundColor(0, 0, 255); /* Ορίζουμε χρώμα για τον φόντο
252 της οθόνης */
253 /* Εμφανίζουμε τα μηνύματα */
254 myGLCD.print("Ptixiaki Ergasia", CENTER, 93);
255 myGLCD.print("Rafail Mixalakakis", CENTER, 119);
256 myGLCD.print("A.M: 11211", CENTER, 148);
257 delay(2000); /* Χρονοκαθυστέρηση 2 δευτερόλεπτα */
258 myGLCD.clrScr(); /* Καθαρισμός της οθόνης */
259 }
260

```

5.2 Κώδικας του Arduino Nano

```
1  /* Εισαγωγή βιβλιοθηκών */
2
3  #include <DHT.h>
4  #include <SPI.h>
5  #include <nRF24L01.h>
6  #include <RF24.h>
7
8
9  /* Ορίζουμε τις ψηφιακές εισόδους/εξόδους */
10 #define DHTPIN 9 /* Θέση pin 9 για τον αισθητήρα DHT22 */
11 #define DHTTYPE DHT22 /* Τύπος αισθητήρα DHT22 */
12
13
14 DHT dht(DHTPIN, DHTTYPE);
15 RF24 radio(7, 8); /* Συσκευή ασύρματης ραδιοεπικοινωνίας Rx/TX */
16 const byte address[6] = "00001";
17
18
19 int chk;
20 float hum; /* Μεταβλητή υγρασίας */
21 float temp; /* Μεταβλητή θερμοκρασίας */
22 char buffer[12]; /* buffer για προσωρινή αποθήκευση */
23 char str_temp[6];
24 char str_hum[6];
25
26 /* Η συνάρτηση setup εκτελείται μια μονό φορά στο ξεκίνημα του
27 μικροεπεξεργαστή και αρχικοποιεί τις συσκευές,
28 όπως οθόνη, ραδιοεπικοινωνίες, αισθητήρες κ.λπ.
29 */
30
31 void setup()
32 {
33     dht.begin(); /* Αρχικοποίηση του αισθητήρα DHT22 */
34     radio.begin(); /* Αρχικοποίηση της μονάδας ραδιοεπικοινωνίας */
35     radio.openWritingPipe(address); /* Ορισμός του καναλιού που θα
36 γίνεται η επικοινωνία */
37     radio.setPALevel(RF24_PA_MIN);
38     radio.stopListening();
39 }
40
41
42 /* Η συνάρτηση loop εκτελείται συνέχεια από τον μικροεπεξεργαστή και
43 εδώ βάζουμε όλες τις λειτουργίες
44 που πρόκειται να κάνει το πρόγραμμα μας.
45 */
46 void loop()
47 {
48
49     hum = dht.readHumidity(); /* Εκτέλεση της συνάρτησης ανάγνωσης της
50 υγρασίας από τις βιβλιοθήκες του κατασκευαστή και αποθήκευση στη
51 μεταβλητή */
52     temp = dht.readTemperature(); /* Εκτέλεση της συνάρτησης ανάγνωσης
53 της θερμοκρασίας από τις βιβλιοθήκες του κατασκευαστή και αποθήκευση
54 στη μεταβλητή */
```

```

55  memset(&buffer, 0, sizeof(buffer)); /* Καθαρίζουμε την προσωρινή
56  μνήμη ώστε να υποθηκεύσουμε το πακέτο */
57  dtostrf(temp, 4, 2, str_temp); /* Μετατροπή της τιμής από
58  αριθμητικό σε αλφαριθμητικό */
59  dtostrf(hum, 4, 2, str_hum); /* Μετατροπή της τιμής από αριθμητικό
60  σε αλφαριθμητικό */
61  sprintf(buffer, "%s,%s", str_temp, str_hum); /* Αποθήκευση των
62  τιμών της θερμοκρασίας υγρασίας σε μορφή string για αποστολή του
63  πακέτου */
64  radio.write(&buffer, sizeof(buffer)); /* Συνάρτηση αποστολή του
65  πακέτου στη μονάδα ραδιοεπικοινωνίας από τις βιβλιοθήκες του
66  κατασκευαστή */
67  delay(5000); /* Χρονοκαθυστέρηση 5 δευτερόλεπτα */
68  }

```

ΚΕΦΑΛΑΙΟ 6 – Συμπεράσματα

Στην εργασία αυτή παρουσιάστηκε η υλοποίηση ενός συστήματος απομακρυσμένης καταγραφής της θερμοκρασίας και υγρασίας ενός χώρου με την χρήση των Arduino Mega και Arduino Nano, όπου η εμφάνιση των δεδομένων γίνεται σε μία οθόνη που είναι συνδεδεμένη με το Arduino Mega.

Τέλος, κάποιες μελλοντικές αναβαθμίσεις που θα μπορούσαν να γίνουν είναι η προσθήκη και κάποιων ακόμα αισθητήρων, όπως ένας αισθητήρας βαρομετρικής πίεσης, αισθητήρας υψομέτρου, αισθητήρας αερίων και η δυνατότητα αποστολής μέσω SMS ή email των τιμών των αισθητήρων παράδειγμα κάθε ώρα ή ακόμα και να επιλέγουμε μέσα σε ένα διάστημα την καταγραφή των μέσων όρων των τιμών αυτών ή ακόμα και ο απομακρυσμένος έλεγχος μέσω μιας εφαρμογής στο κινητό τηλέφωνό μας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Δρ. Γ. Σ. Τσελίκης, & Δρ. Ν. Δ. Τσελίκας (2012) C Από την θεωρία στην εφαρμογή, Β' Έκδοση. Αθήνα
2. Άγγελος Δ. Δημητριάδης (2001) Προγραμματίζοντας τον μικροελεγκτή AVR.Θεσσαλονίκη.
3. *Paul Deitel, & Harvey Deitel (2001) C: how to program, 7 έκδοση. New jersey.*

ΙΣΤΟΣΕΛΙΔΕΣ

4. <https://internetofthingsagenda.techtarget.com/definition/microcontroller>
5. <https://www.arduino.cc/en/Guide/Introduction#>
6. <https://arduinohistory.github.io/>
7. <https://www.arduino.cc/en/Main/Products>
8. <https://www.arduino.cc/en/Products/Compare>
9. <https://www.arduino.cc/en/Main/arduinoBoardMega/>
10. <https://store.arduino.cc/arduino-nano>
11. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
12. https://components101.com/sites/default/files/component_datasheet/DS3231%20Datash eet.pdf
13. <https://www.robotshop.com/media/files/pdf/datasheet-wir020.pdf>
14. http://wiki.sainsmart.com/index.php/3.2%22_TFT_Touch_LCD
15. <https://www.sainsmart.com/products/tft-lcd-adjustable-shield-for-arduino-mega2560>
16. <http://fritzing.org/home/>
17. <https://www.techopedia.com/definition/3641/microcontroller>

