

# **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

«ΥΛΟΠΟΙΗΣΗ ΑΣΥΡΜΑΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕ  
ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΩΝ»

ΒΥΣΑΝΣΙΩΤΗΣ ΣΤΑΥΡΟΣ Α.Μ. 10139

ΚΑΡΒΟΥΝΙΔΟΥ ΑΓΓΕΛΙΚΗ Α.Μ. 10057

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ ΤΑΤΣΗΣ ΓΕΩΡΓΙΟΣ

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κατάλογος Πινάκων.....	3
Κατάλογος Εικόνων.....	4
Περίληψη.....	5
Λέξεις κλειδιά.....	5
Κεφάλαιο 1 – Εισαγωγή.....	6
1.1 Ιστορικά.....	7
1.2 Μικροελεγκτές.....	7
1.3 Γενικές πληροφορίες Arduino UNO.....	10
1.4 Χαρακτηριστικά Arduino.....	10
1.5 Hardware.....	11
1.6 Τροφοδοσία.....	12
1.7 Επεξεργαστής.....	12
1.8 Ακροδέκτες Arduino UNO.....	13
1.9 Λογισμικό.....	14
Κεφάλαιο 2 – Μεθοδολογία υλοποίησης ασύρματης επικοινωνίας.....	16
2.1 Υλικά.....	16
2.2 Σχεδιασμός κυκλώματος.....	18
2.3 Προγραμματισμός. Κυκλώματος.....	21
Κεφάλαιο 3 – Συμπέρασμα.....	26
Βιβλιογραφία.....	27

# ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Χαρακτηριστικά Arduino.....	10
----------------------------------------	----

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Arduino Micro.....	8
Εικόνα 2: Arduino Mega.....	8
Εικόνα 3: Arduino Diecimila.....	8
Εικόνα 4: Arduino Duemilanova.....	8
Εικόνα 5: Arduino Leonardo.....	9
Εικόνα 6: Arduino Nano.....	9
Εικόνα 7: Arduino Mini.....	9
Εικόνα 8: Arduino LilyPad.....	9
Εικόνα 9: Arduino UNO.....	10
Εικόνα 10: Hardware Arduino.....	11
Εικόνα 11: Διασύνδεση των υποδοχών του Arduino.....	12
Εικόνα 12: Ψηφιακοί ακροδέκτες του Arduino.....	13
Εικόνα 13: Αναλογικοί ακροδέκτες του Arduino.....	14
Εικόνα 14: Παράδειγμα περιβάλλοντος του προγράμματος Arduino IDE.....	15
Εικόνα 15: Arduino Uno Rev3.....	16
Εικόνα 16: Πομπός.....	17
Εικόνα 17: Δέκτης.....	17
Εικόνα 18: Καλώδια.....	17
Εικόνα 19: Breadboard.....	18
Εικόνα 20: Υλοποιημένο κύκλωμα.....	18
Εικόνα 21: Ολοκληρωμένο κύκλωμα πομπού.....	19
Εικόνα 22: Ολοκληρωμένο κύκλωμα δέκτη.....	20
Εικόνα 23: Κεραία πομπού.....	21

## **ΠΕΡΙΛΗΨΗ**

Με την υλοποίηση ασύρματης επικοινωνίας με χρήση μικροελεγκτών θα ασχοληθούμε στην παρακάτω εργασία. Οι μικροελεγκτές με τους οποίους θα υλοποιήσουμε την ασύρματη επικοινωνία μας είναι οι arduino. Με την βοήθεια ενός πομπού και ενός δέκτη οι οποίοι θα είναι συνδεδεμένοι με τον arduino θα καταφέρουμε να στέλνουμε δεδομένα μεταξύ τους. Οι arduino όσο του πομπού όσο και του δέκτη θα είναι προγραμματισμένη για να επιτύχουμε την αποστολή και την λήψη και επίσης και την ανίχνευση τυχών λάθους. Συγκεκριμένα η γλώσσα προγραμματισμού θα είναι η C++.

## **ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**

Ασύρματη επικοινωνία, μικροελεγκτών, arduino, πομπός, δέκτης, αποστολή, λήψη, ανίχνευση λάθους, γλώσσα προγραμματισμού C++.

# Κεφάλαιο 1<sup>ο</sup> -Εισαγωγή

Η ανάγκη του ανθρώπου για επικοινωνία τον έχει ωθήσει στην δημιουργία μικροεφαρμογών. Αυτό γίνεται για να εξυπηρετήσει τις ανάγκες που έχει ο καθένας μας και γιατί το κόστος είναι πολύ μικρότερο.

Σκοπός της εργασίας είναι η δημιουργία ασύρματης επικοινωνίας με την χρήση μικροελεγκτών, ο οποίος θα είναι ο Arduino Uno. Πιο συγκεκριμένα θα αναφερθούμε στην αποστολή δεδομένων από το πομπό, την λήψη τους από το δέκτη και την ανίχνευση τυχών λάθους κατά την αποστολή. Η επικοινωνία θα γίνεται ανάμεσα σε δυο υπολογιστές οι οποίοι θα συνδέονται με τους μικροελεκτές μέσω καλωδίου Usb.

Αρχικά θα αναλύσουμε τα κομμάτια από τα οποία αποτελείτε το σύστημα μας και έπειτα θα παρουσιάσουμε κάποια παραδείγματα από τα οποία θα φαίνετε η πλήρης λειτουργία του συστήματος μας.

## 1.1 Ιστορικά

Το 2005 ο Massimo Banzi και ο David Cueartielles ξεκίνησαν να φτιάχνουν μια συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα που υπήρχαν εκείνη την περίοδο. Οι ιδρυτές ονόμασαν την συσκευή από τον Arduino iverea και ξεκίνησαν να παράγουν πλακέτες σε ένα εργαστήριο στην Ιβρέα κωμόπολη του Τορίνο της Ιταλίας .

Το Arduino είναι μια διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοιχτού κώδικα και προγραμματίζετε χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++.

## 1.2 Μικροελεγκτές

Μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα που περιλαμβάνει έναν επεξεργαστή, μνήμη και είσοδο έξοδο οι οποίες είναι προγραμματιζόμενες. Η μνήμη όπως και η μνήμη RAM είναι ενσωματωμένες στο τσιπ. Ο μικροελεγκτής έχει εφαρμογή σε συστήματα και συσκευές αυτομάτου ελέγχου, όπως αυτοκινούμενες μηχανές, μηχανές γραφείου, βιοιατρικές συσκευές οι οποίες είναι εμφυτευμένες στο ανθρώπινο σώμα, και σε πολλές άλλες.

Ο μικροελεγκτής έχει μικρό κόστος παραγωγής και είναι μικρός σε μέγεθος και για αυτο το λόγο μπορεί να έχει ψηφιακό έλεγχο σε πολλές συσκευές και διεργασίες.

Ορισμένοι χρησιμοποιούν λέξεις τεσσάρων bit και λειτουργούν σε συχνότητες χρονισμού από 4 khz ώστε να έχουν χαμηλή κατανάλωση ισχύος. Μερικοί έχουν την δυνατότητα να είναι σε αδράνεια και να έρχονται πάλι σε λειτουργία με το πάτημα ενός κουμπιού. Κάτι το οποίο τους κάνει πολύ χρήσιμους γιατί λειτουργούν για μεγάλα χρονικά διαστήματα με την χρήση μιας μπαταρίας. Η digital signal processor (DSP) παίζουν σημαντικό ρόλο στην απόδοση του συστήματος με γρήγορο ρολόι αλλά αυτοί έχουν μεγαλύτερη κατανάλωση. Στις παρακάτω εικόνα παρουσιάζονται κάποια μοντέλα που κυκλοφορούν στην αγορά. Για την συγκεκριμένη εργασία το μοντέλο που επιλέχτηκε είναι το Arduino Uno.



**Εικόνα 1 :** Arduino Micro.



**Εικόνα 2:** Arduino Mega.



**Εικόνα 3:** Arduino Diecimila.



**Εικόνα 4:** Arduino Duemilanova.





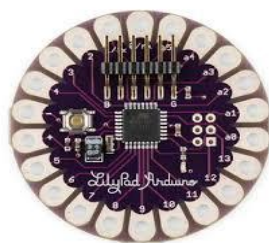
**Εικόνα 5:** Arduino Leonardo.



**Εικόνα 6:** Arduino Nano.



**Εικόνα 7:** Arduino Mini.



**Εικόνα 8:** Arduino LilyPad.



Εικόνα 9: Arduino UNO

### 1.3 Γενικές πληροφορίες Arduino Uno

Για την υλοποίηση της εργασίας, χρησιμοποιήθηκε ένας μικροελεγκτής Arduino. Το Arduino είναι μια υπολογιστική πλατφόρμα ανοιχτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους η οποίες προγραμματίζονται με τη γλώσσα Wiring (στην πραγματικότητα είναι η γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιείτε για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά μπορεί να συνδεθεί και ε τον υπολογιστή μέσω προγραμμάτων όπως Flas, Processing, MaxMSP.

### 1.4 Χαρακτηριστικά του Arduino Uno

Εφόσον για την συγκεκριμένη εργασία έχει επιλεγεί για μικροελεγκτής το Arduino, ο παρακάτω πίνακας μας δείχνει τα χαρακτηριστικά του.

Μικροελεγκτής	ATMEGA328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όριο τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14, (PWM έξοδοι)

Αναλογική ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάση 3.3V	50mA
Μνήμη Flas	32kB (ATMEGA328)
Μνήμη SRAM	2Kb (ATMEGA328)
Μνήμη EEPROM	1kB (ATMEGA328)
Ταχύτητα ρολογιού	16MHz (ATMEGA328)

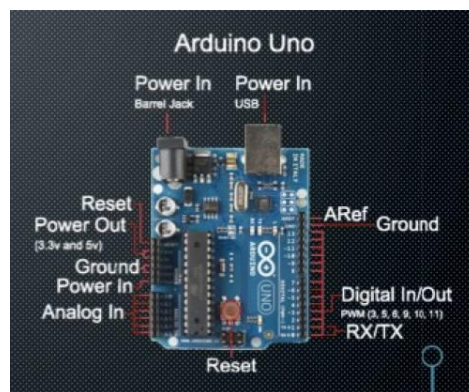
**Πίνακας 1:** Χαρακτηριστικά Arduino

## 1.5 Hardware

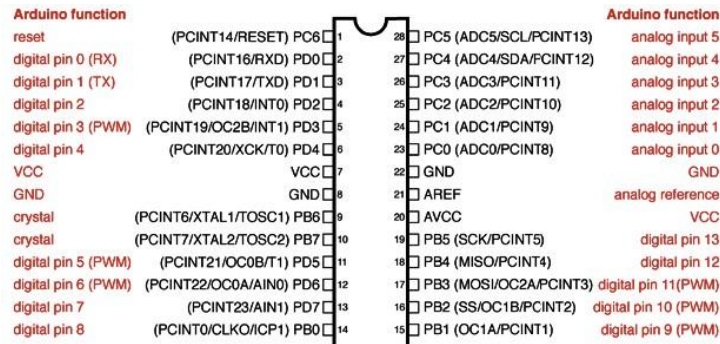
Η πλακέτα περιλαμβάνει τον ενσωματωμένο μικροεπεξεργαστή του συστήματος, ψηφιακές υποδοχές που χρησιμοποιούνται σαν εισοδοι και έξοδοι, αναλογικές υποδοχές που χρησιμοποιούνται σαν εισοδοι μόνο, σταθεροποιητή τάσης, εξόδους τροφοδοσίας 3.3V και 5V, κουμπί επαναφοράς (reset), και υποδοχές για την γείωση(ground).

Επίσης έχει εισόδους για την τροφοδοσία του συστήματος η οποία γίνεται με USB ή από κάποια εξωτερική πηγή μέσω ενός κυλινδρικού κονέκτορα (Barrel jack).

Το Arduino Uno έχει ένα επαναφερόμενο παθητικό ηλεκτρικό εξάρτημα το resettable polyfuse το οποίο προστατεύει τις θύρες USB για τυχόν βραχυκύκλωμα.



**Εικόνα 10:** Hardware Arduino.



Εικόνα 11: Διασύνδεση των υποδοχών του Arduino.

## 1.6 Τροφοδοσία

Η τροφοδοσία του Arduino μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι από τον υπολογιστή με σύνδεση USB και ο δεύτερος από εξωτερική τροφοδοσία μέσω μιας υποδοχής φίς των 2.1mm που βρίσκεται στην κάτω αριστερή γωνία.

Η εξωτερική τροφοδοσία μπορεί να είναι κάποιος μετασχηματιστής εναλλασσόμενου ρεύματος σε συνεχές είτε κάποια μπαταρία. Η μπαταρία συνδέεται την υποδοχή ground και Vin του κονέκτορα τροφοδοσίας, ενώ ο μετασχηματιστής στην κυλινδρική υποδοχή.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

- Vin: Η τάση εισόδου της πλακέτας, όταν χρησιμοποιείτε εξωτερική πηγή ενέργειας. Η τροφοδοσία γίνεται μέσω αυτό του ακροδέκτη.
- 5V: Η τάση που χρησιμοποιείται από τα διάφορα μέρη της πλακέτας και τον μικροελεγκτή είναι 5V. Η τάση αυτή είναι 5V που δίνει η σύνδεση USB είτε η ρυθμισμένη τάση που δίνετε μέσω Vin.
- 3.3V: Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI. Το όριο άντλησης ρεύματος είναι 50mA.
- GND: Είσοδοι γείωσης.
- IOREF: Η υποδοχή αυτή παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής.

## 1.7 Επεξεργαστής

Ο επεξεργαστής της πλακέτας είναι ο ATmega328 της Atmel. Λειτουργεί στα 5V, ενώ η ταχύτητα χρονισμού είναι στα 16MHz το οποίο οφείλετε στον κρυσταλλικό ταλαντωτή. Διαθέτει Flash memory 32Kb, 2Kb STRAM και 1Kb EEPROM. Καταλαβαίνουμε ότι ο επεξεργαστής διαθέτει μικρό αποθηκευτικό χώρο, σε περίπτωση που θέλουμε μεγαλύτερο χώρο μπορούμε να το καταφέρουμε αυτό με κάποιο εξωτερικό μέσο όπως για παράδειγμα κάρτα SD.

## 1.8 Ακροδέκτες Arduino Uno

Κάθε μικροελεγκτής Arduino διαθέτει εισόδους και εξόδους οι οποίοι αλληλεπιδρούν με το περιβάλλον και το εξαρτήματα. Το Arduino Uno διαθέτει 20 ακροδέκτες (pin), από τους οποίους 14 είναι ψηφιακοί και 6 αναλογικοί.

### 1. Ψηφιακοί ακροδέκτες

Καθένας από τους ψηφιακούς υποδοχής λειτουργεί όσο σαν είσοδος όσο και σαν έξοδος με την χρήση συναρτήσεων `pinMode()`, `digitalWrite()`, `digitalRead()`. Η λειτουργία γίνεται στα 5V. Κάθε υποδοχή μπορεί να παρέχει και να δέχεται ρεύμα μέγιστη τιμή 40mA και έχει εσωτερικά αντίσταση pull-up της τάξης των 20-50KOhms για προστασία.

Παρακάτω θα δούμε και κάποιες άλλες λειτουργίες των υποδοχών.

- 0 (RX) και 1 (TX): Χρησιμοποιούνται για τη σειριακή λήψη (RX) και μετάδοση (TX).
- External Interrupts: 2 and 3. Οι υποδοχείς 2 και 3 μπορούν να χρησιμοποιηθούν για την ενεργοποίηση κάποιας μεταβολής του συστήματος..
- PWM: 3, 5, 6, 9, 10,11. Παρέχουν έξοδο 8-bit διαμόρφωσης PWM με χρήση της συνάρτησης `analogWrite()`.
- SPI:10(SS), 11(MOSI), 12(MISO), 13(SCK). Οι υποδοχές αυτοί παρέχουν σειριακή επικοινωνία με χρήση βιβλιοθήκης SPI (serial peripheral interphase).
- LED:13. Στην υποδοχή αυτή υπάρχει εκ κατασκευής ένα LED.



Εικόνα 12: Ψηφιακοί ακροδέκτες του Arduino.

## 2. Αναλογικοί ακροδέκτες

Οι 6 αναλογικοί ακροδέκτες του Arduino λειτουργούν μόνο ως είσοδοι. Κάθε υποδοχέας έχει ενσωματωμένο ένα μετατροπέα A/D με παροχή ανάλυσης 10bit κάτι που σημαίνει ότι μπορεί να εμφανίσει 1024 διαφορετικές τιμές. Είναι ρυθμισμένα να μετρούν από 0 έως 5V, με την ικανότητα να αλλάξει το άνω όριο χρησιμοποιώντας τον υποδοχέα AREF και την συνάρτηση analogReference().

Παρακάτω θα δούμε κάποιες άλλες λειτουργίες των αναλογικών υποδοχέων.

- TWI: A4 or SDA pin and A5 or SCL pin. Υποστηρίζουν την επικοινωνία TWI χρησιμοποιώντας την βιβλιοθήκη Wire.
- AREF: τροφοδοσία με την τάση αναφοράς για τη αναλογικές εισόδους.
- Reset: επαναφέρει τον μικροελεγκτή.



**Εικόνα 13:** Αναλογικοί ακροδέκτες Arduino.

### 1.9 Λογισμικό

Για να επιτύχουμε την ασύρματη επικοινωνία θα πρέπει να προγραμματίσουμε το Arduino. Αυτό θα το επιτύχουμε συνδέοντας το Arduino με τον υπολογιστή μέσω του καλωδίου usb και στη συνέχεια μέσω του προγράμματος Arduino IDE θα προγραμματίσουμε το Arduino. Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Μπορεί να μεταγλωττίζει και να φορτώνει το ολοκληρωμένο πρόγραμμα στην πλακέτα με το πάτημα ενός κουμπιού. Η γλώσσα προγραμματισμού που χρησιμοποιούμε για να προγραμματίσουμε το Arduino είναι η C/C++.



```
othoni | Arduino 1.0.5-r2
File Edit Sketch Tools Help
othoni.g
Serial.print("header size "); Serial.println(tap, DEC);
bapWidth = read32(f);
bapHeight = read32(f);

if (read16(f) != 1)
    return false;

bapDepth = read16(f);
Serial.print("bitdepth "); Serial.println(bapDepth, DEC);

Serial.print("compression "); Serial.println(tap, DEC);

return true;
}

//*****

// These read data from the SD card file and convert them to big endian
// (the data is stored in little endian format!)

// LITTLE ENDIAN!
uint16_t read16(File sf) {
    uint16_t d;
    uint8_t b;
    b = f.read();
    d = f.read();
    d <<= 8;
    d |= b;
    return d;
}

// LITTLE ENDIAN!
uint32_t read32(File sf) {
    uint32_t d;
    uint16_t b;

    b = read16(f);
    d = read16(f);
    d <<= 16;
}
```

Done uploading.  
Binary sketch size: 20,770 bytes (of a 32,256 byte maximum)

276 Arduino Uno on COM7

Εικόνα 14: Παράδειγμα περιβάλλοντος του προγράμματος Arduino IDE

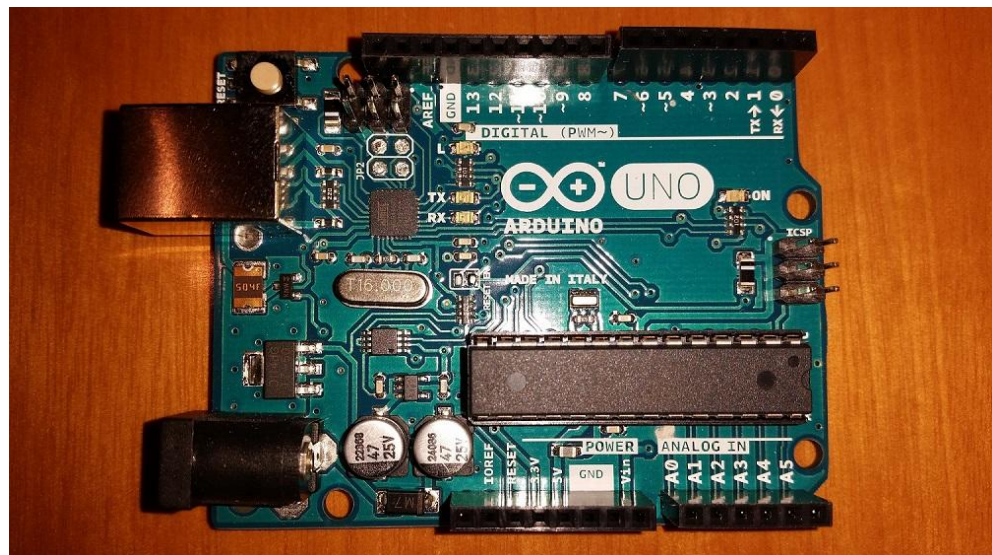
## Κεφάλαιο 2<sup>ο</sup> – Μεθοδολογία υλοποίησης ασύρματης επικοινωνίας

Σε αυτό το κεφάλαιο θα μιλήσουμε και θα αναλύσουμε ποιο πολύ τα υλικά τα οποία χρησιμοποιήσαμε για την επίτευξη του έργου (hardware) και για το προγραμματιστικό μέρος.

### 2.1 Τα υλικά

Τα υλικά που θα χρειαστούμε για να επιτύχουμε την ασύρματη επικοινωνία μας είναι τα εξής:

- 2 **Arduino UNO Rev3**



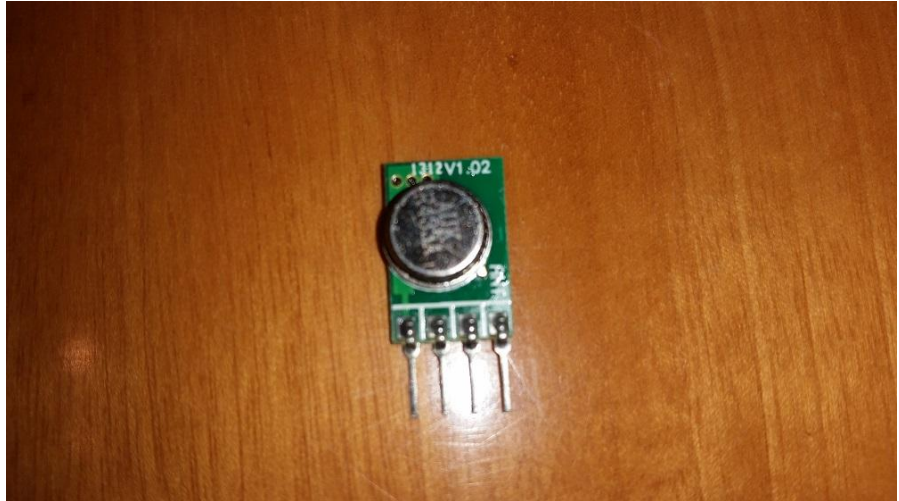
Εικόνα 15: Arduino UNO Rev3

Η συχνότητα λειτουργίας του arduino είναι στα 16MHz

$$\text{Περίοδος: } T = \frac{1}{f} = \frac{1}{16 \cdot 10^6} = 0.0625 \cdot 10^{-6} \rightarrow T = 62.5 \text{ ns}$$

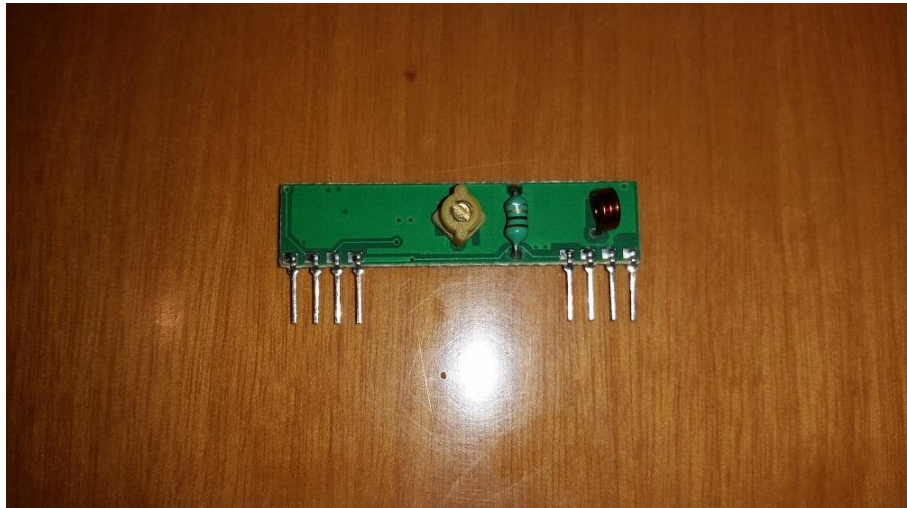


- Έναν πομπό: RF Link Transmitter - 434MHz



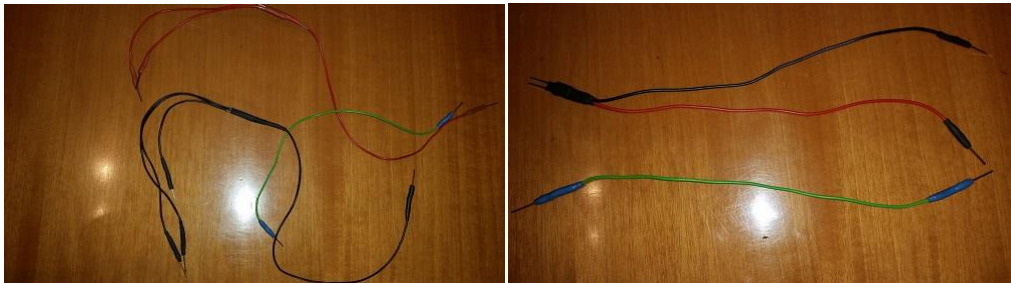
Εικόνα 16: Πομπός

- Έναν δέκτη: RF Link Receiver - 434MHz

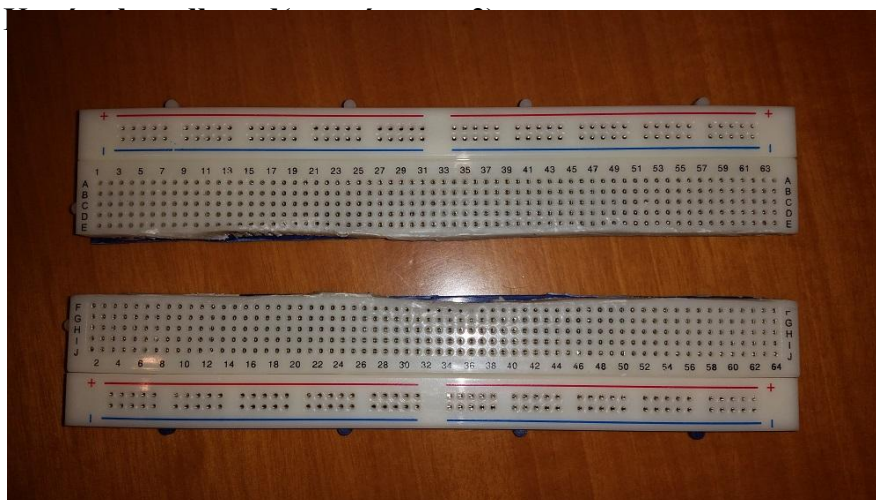


Εικόνα 17: Δέκτης

- Μερικά καλώδια



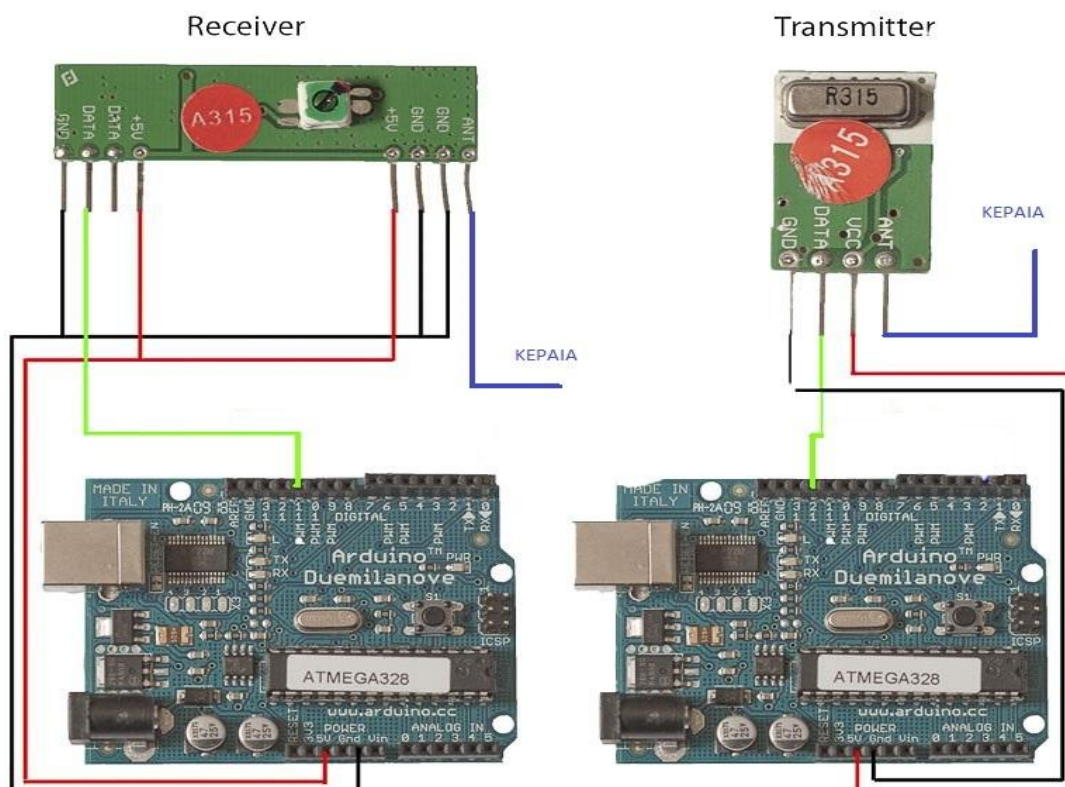
Εικόνα 18: Καλώδια



Εικόνα 19: Breadboard

## 2.2 Σχεδιασμός κυκλώματος

Σε αυτή την ενότητα θα αναλύσουμε την συνδεσμολογία του κυκλώματος που υλοποιήσαμε και αυτό μπορούμε να το παρατηρήσουμε στην εικόνα 20.



Εικόνα 20: Υλοποιημένο κύκλωμα

Η συνδεσμολογία του πομπού έχει ως εξής:

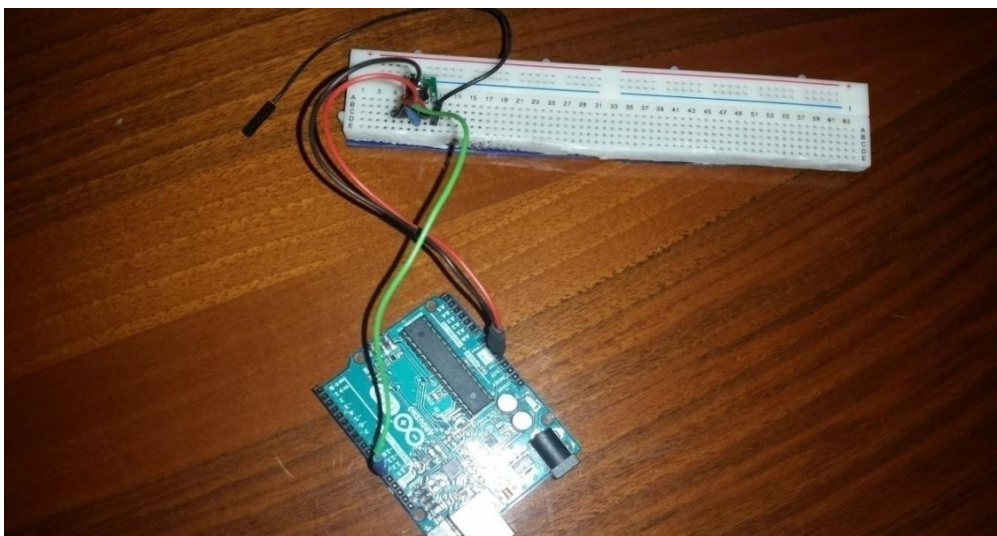
- Το πράσινο καλώδιο είναι για τα δεδομένα και το τοποθετούμε στη μεριά των ψηφιακών ακροδεκτών στην υποδοχή 12
- Το μαύρο καλώδιο είναι για την γείωση το οποίο όπως φαίνεται και στην εικόνα 20 το τοποθετούμε στην υποδοχή η οποία αναγράφει GND
- Το κόκκινο καλώδιο είναι για την παροχή ισχύος του πομπού το οποίο και αυτό με την σειρά του το τοποθετούμε στην υποδοχή η οποία αναγράφει 5V
- Το 4<sup>ο</sup> ποδαράκι του πομπού δεν το συνδέουμε με κάποιο καλώδιο διότι είναι η κεραία του
- Το μπλε καλώδιο είναι η κεραία μας  $\lambda/4$

Η συνδεσμολογία για τον δέκτη έχει ως εξής:

- Το πράσινο καλώδιο είναι για τα δεδομένα και όπως για τον πομπό έτσι και για τον δέκτη το συνδέουμε στη μεριά των ψηφιακών ακροδεκτών αλλά αυτή τη φορά το τοποθετούμε στην υποδοχή 11
- Το μαύρο καλώδιο είναι για την γείωση το οποίο όπως βλέπουμε και στην εικόνα 20 το συνδέουμε σε τρεις μεριές του δέκτη μας και στη συνέχεια το τοποθετούμε και αυτό στην υποδοχή με την ένδειξη GND
- Το κόκκινο καλώδιο είναι για την παροχή ισχύος. Το συνδέουμε σε δύο μεριές και το τοποθετούμε στην υποδοχή που με την ένδειξη 5V
- Το 3<sup>ο</sup> ποδαράκι είναι και αυτό για τα δεδομένα αλλά δεν το χρησιμοποιούμε και το 8<sup>ο</sup> είναι η κεραία μας η οποία θα λαμβάνει
- Το μπλε καλώδιο είναι η κεραία μας  $\lambda/4$

Η τελική μορφή που θα έχει το κύκλωμα του πομπού και του δέκτης θα είναι η εξής:

#### • ΠΟΜΠΟΣ



**Εικόνα 21:** Ολοκληρωμένο κύκλωμα πομπού

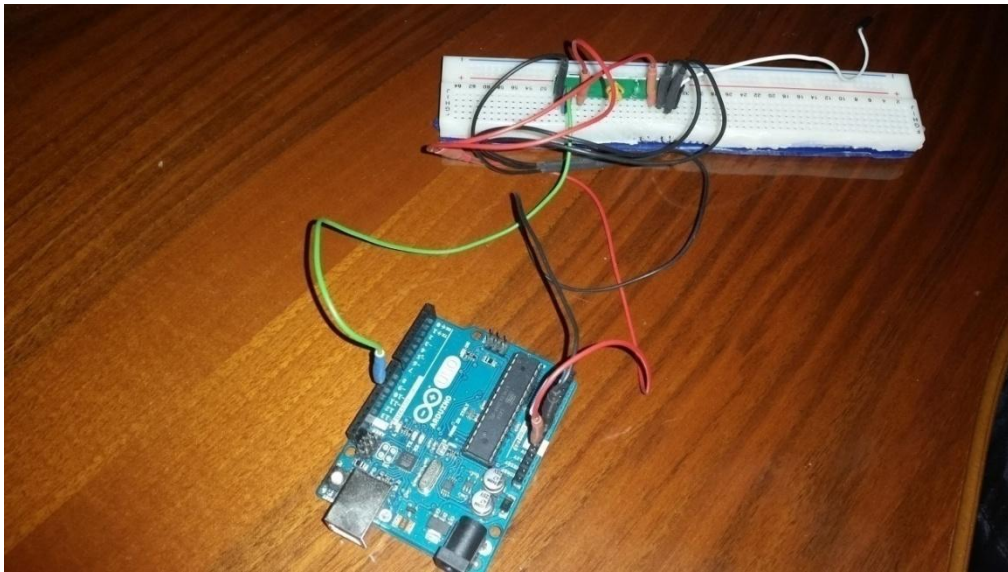


Ο πομπός είναι μια ηλεκτρονική συσκευή που συνήθως με την βοήθεια μιας κεραίας εκπέμπει σήματα που περιέχουν πληροφορίες. Δέχεται κάποια δεδομένα τα οποία τα μετασχηματίζει ή τα κωδικοποιεί για να μεταδοθούν σε κάποιο κανάλι ή μέσω μετάδοσης (π.χ. ο κενός χώρος, δισύρματα καλώδια , οπτικές ίνες κ.α.)

Τα χαρακτηριστικά του πομπού είναι:

- Η συχνότητα η οποία είναι στα 433.92MHz(min=433.420MHz και max=434.420MHz)
- Η διαμόρφωση η οποία είναι ASK.  
ASK είναι ο τύπος διαμόρφωσης σήματος όπου ψηφιακά δεδομένα παρουσιάζονται ως αλλαγές στο πλάτος ενός φέροντος σήματος.
- Ρυθμός δεδομένων:4800bps.
- Τάση τροφοδοσίας:5V (min=4.9V και max=5.1V)
- Τύπος κυκλώματος: συντονισμένο LC.
- Απόσταση καναλιών:+\_500KHz

#### • ΔΕΚΤΗΣ



**Εικόνα 22:** Ολοκληρωμένο κύκλωμα δέκτη

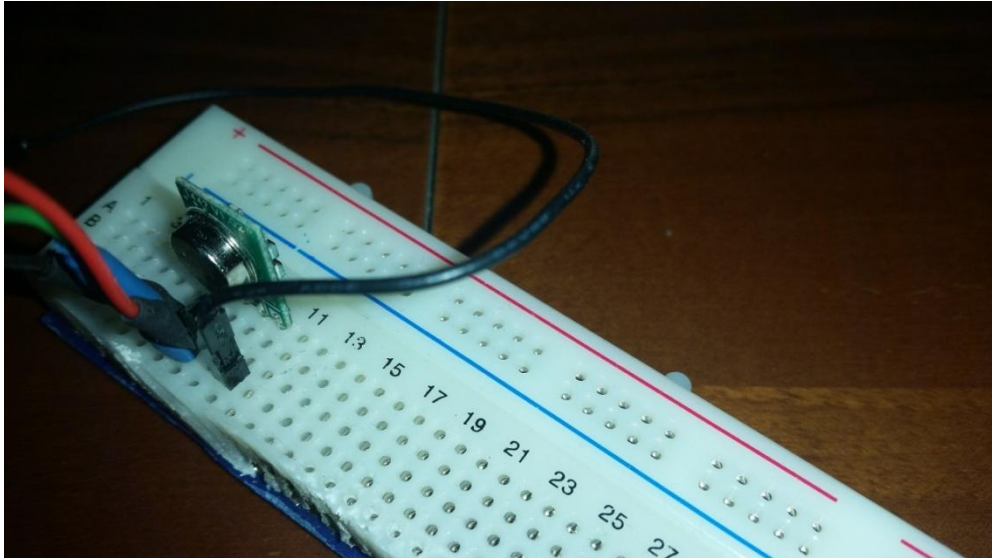
Ο δέκτης κάνει την αντίστροφη εργασία από τον πομπό, δηλαδή μετατρέπει τα δεδομένα του καναλιού μετάδοσης στα πρωτογενή που έχει στείλει ο πομπός.

Τα χαρακτηριστικά του δεκτή είναι:

- Συχνότητα: 433.92MHz
- Διαμόρφωση: ASK
- Ρυθμός δεδομένων:8Kbps

➤ Τάση τροφοδοσίας: 1.5~12V

- **Κεραία πομπού**



**Εικόνα 23:** Κεραία

Κεραία είναι μια διάταξη που χρησιμοποιείται για να εκπέμπει ή να λαμβάνει ηλεκτρομαγνητική ακτινοβολία. Ουσιαστικά η κεραία μετατρέπει σε ηλεκτρομαγνητική ενέργεια που οδεύει σε μια γραμμή μεταφοράς ή ένα κυματοδηγό σε ηλεκτρομαγνητικό κύμα στο χώρο και αντίστροφα.

Στην δική μας περίπτωση θέλουμε να εκπέμπει ηλεκτρομαγνητική ενέργεια και έτσι τοποθετούμε στον 4<sup>ο</sup> ακροδέκτη του πομπού ένα καλώδιο. Το καλώδιο αυτό θα είναι η κεραία μας με μήκος  $\frac{\lambda}{4}$ .

Εύρεση μήκους κεραίας:

Η συχνότητα του πομπού είναι στα 434MHz=434.000.000Hz

$$\lambda = c * T \rightarrow \lambda = \frac{c}{f} = \frac{3 * 10^8 \frac{m}{s}}{434 * 10^6 Hz} = 0.00691 * 10^2 \rightarrow \lambda = 69,1mm$$

$$\frac{\lambda}{4} = \frac{69,1mm}{4} \approx 1,7cm$$

Άρα το μήκος της κεραίας θα πρέπει να είναι περίπου **1,7cm**.

### 2.3 Προγραμματισμός κυκλωμάτων

Σε αυτή την ενότητα θα σας παρουσιάσουμε και θα αναλύσουμε τον κώδικα του πομπού και του δεκτή έτσι όπως τον υλοποιήσαμε για στέλνει δεδομένα ο πομπός στον δεκτή.

Για την υλοποίηση του κώδικα χρησιμοποιήσαμε την βιβλιοθήκη VirtualWire.

Η VirtualWire είναι μια βιβλιοθήκη οι όποια χρησιμοποιείτε για το Arduino. Παρέχει την δυνατότητα για την αποστολή συντόμων μηνυμάτων χωρίς αντιμετώπιση, ανάλυση ή αναγνώριση. Χρησιμοποιεί ASK ( διαμόρφωση μετατόπισης πλάτους). Υποστηρίζει μια σειρά από φθηνούς ραδιοφωνικούς πομπούς και δέκτες. Το μόνο που απαιτεί είναι μετάδοση και λήψη δεδομένων.

Παρακάτω θα δούμε και κάποιες εντολές της virtualWire.

- **ΠΟΜΠΟΣ**

```
#include <VirtualWire.h> /* Δηλώνουμε την βιβλιοθήκη που θα
χρησιμοποιήσουμε*/

const int ledPin = 13; /*Δηλώνουμε το ledPin στην υποδοχή 13*/

int Data_from_Serial; /* Ορίζουμε ακέραιο*/

char Data_from_Serial_CharMsg[10]; /* Πίνακας χαρακτήρων 10
στοιχείων*/

void setup() {

pinMode(ledPin,OUTPUT); /* Το ορίζουμε ως έξοδο */

Serial.begin(115200); /* Ορίζω τη σειριακή θύρα */

/* Ρύθμιση της βιβλιοθήκης */
vw_setup(1000); /* Ρυθμίζει το ρυθμό δεδομένων που στέλνει η
βιβλιοθήκη (όσο μικρότερο το νούμερο τόσο λιγότερα
σφάλματα) */
vw_set_tx_pin(12); /* Ρυθμίζει πιο Pin θα χρησιμοποιεί από την
πλακέτα για να στείλει τα δεδομένα */
Serial.println("Transmitter started."); /* Εκτυπώνει στη σειριακή θύρα το
μήνυμα που έχουμε και πάει στην
επόμενη γραμμή */
Serial.println("Type a message to send: ");

}

void loop() {

if(Serial.available()) /* Αν είναι ελεύθερη η σειριακή διάβασε και
αποθήκευσε στην Data_from_Serial */
{
```

```

Data_from_Serial = Serial.read();

itoa(Data_from_Serial,Data_from_Serial_CharMsg,10); /* Μετατρέπει τον
ακέραιο σε
χαρακτήρα */

Serial.print((char)Data_from_Serial); /* Εκτυπώνει στη σειριακή
Data_from_serial αλλά το έχει
μετατρέψει σε χαρακτήρα*/

delay(10); /* Αναμονή 10ms */

digitalWrite(13, true); /* Κάνω το Pin 13 ένα (ανάβει το LED) */
vw_send((uint8_t *)Data_from_Serial_CharMsg,
strlen(Data_from_Serial_CharMsg)); /*
vw_wait_tx(); /* Περιμένει μέχρι να σταλεί το μήνυμα και όταν γίνει
πάει παρακάτω*/
digitalWrite(13, false); /* Κάνει το Pin 13 μηδέν (σβήνει το LED) */

if (Data_from_Serial == 10) /* Όταν ισούται με 10 έχει πατηθεί το
ENTER, άρα πρέπει να αρχίσει να
στέλνεται το μήνυμα */
{
Serial.println("Message sent. "); /* Εκτυπώνει στη σειριακή θύρα το
μήνυμα που έχουμε και πάει στην
επόμενη γραμμή */

Serial.print("Type a message to send: ");
}

delay(200); /* Αναμονή 200ms */
}
}

```

- **ΔΕΚΤΗΣ**

```

#include <VirtualWire.h> /* Δηλώνουμε την βιβλιοθήκη που θα
χρησιμοποιήσουμε*/

int ledPin = 13; /* Δηλώνουμε το ledPin στην υποδοχή 13*/

int Data_Received; /* Δηλώνουμε το Data_Received ως ακέραιο */

```

```

char Data_Received_CharMsg[10]; /* Πίνακας χαρακτήρων 10
στοιχείων*/

bool show=true; /* Χρησιμοποιείται για να εμφανίζει μόνο μια φορά το
Data_Received */

void setup() {
Serial.begin(115200); /* Ορίζω τη σειριακή θύρα */

pinMode(ledPin, OUTPUT); /* Το ορίζουμε ως έξοδο */

/* Ρύθμιση της βιβλιοθήκης */
vw_setup(1000); /* Ρυθμίζει το ρυθμό δεδομένων που στέλνει η
βιβλιοθήκη */
vw_set_rx_pin(11); /* Ρυθμίζει πιο Pin θα χρησιμοποιεί από την
πλακέτα για να δέχεται τα δεδομένα */
vw_rx_start(); /* Αρχίζει ο δέκτης να τρέχει */

Serial.println("Receiver started"); /* Γράφει στη σειριακή θύρα ότι ο
δέκτης έχει αρχίσει να λειτουργεί */

}
void loop(){ /* Εδώ γράφουμε τον κώδικα που θα επαναλαμβάνεται*/

uint8_t buf[VW_MAX_MESSAGE_LEN]; /* Ορίζουμε τον πίνακα που
αποθηκεύονται τα δεδομένα*/

uint8_t buflen = VW_MAX_MESSAGE_LEN; /* Αποθηκεύουμε το
μέγιστο μήκος μηνύματος
ως buflen */

/* Η παρακάτω εντολή εκτελείται μόνο όταν έχει λάβει μήνυμα. Στην
συγκεκριμένη περίπτωση έναν χαρακτήρα */
if (vw_get_message(buf, &buflen))
{
int i;
digitalWrite(13, true); /* Ανάβει το LED όταν λαμβάνει ένα μήνυμα*/

for (i = 0; i < buflen; i++) /* Εντολή επανάληψης*/
{
Data_Received_CharMsg[i] = char(buf[i]); /* Αποθηκεύει στο

```



```
Data_Received_CharMsg  
τον πίνακα χαρακτήρων */  
}
```

```
Data_Received_CharMsg[bufLen] = '\0';
```

```
Data_Received = atoi(Data_Received_CharMsg); /* Μετατρέπει τον  
πίνακα σε int */
```

```
if (show) Serial.print("Data received: "); /* Εκτυπώνει ότι το μήνυμα  
ελήφθη */
```

```
Serial.print(char(Data_Received)); /* Το μετατρέπει σε χαρακτήρα */
```

```
digitalWrite(13, false); /* Το Led σβήνει και περιμένει το επόμενο  
μήνυμα */
```

```
show = false;
```

```
if (Data_Received == 10) /* Τερματίζει και εμφανίζει ότι έλαβε το  
μήνυμα */
```

```
{
```

```
show = true;
```

```
Serial.println("Message received.");
```

```
}
```

```
}
```

```
}
```

## Κεφάλαιο 3<sup>ο</sup> – Συμπέρασμα

Η πτυχιακή εργασία που παρουσιάστηκε στις παραπάνω επιμέρους παραγράφους, υλοποιείσαι μια ασύρματη επικοινωνία με χρήση μικροελεγκτών. Η χρήση των μικροελεγκτών και πιο συγκεκριμένα για την δική μας περίπτωση ο Arduino Uno, μας βοηθάει να δημιουργήσουμε τις δίκες μας αυτόνομες ηλεκτρονικές συσκευές και να τις προγραμματίσουμε ανάλογα με τα ενδιαφέροντα και τις ανάγκες μας.

Επίσης το Arduino δεν απευθύνετε μόνο σε προγραμματιστές και ηλεκτρονικούς αλλά σε όποιον ενδιαφέρεται για την αλληλεπίδραση περιβάλλοντος και συσκευής. Δεν χρειάζεται να έχει άψογες γνώσεις πάνω στον προγραμματισμό γιατί το Arduino είναι μια πλακέτα ανοιχτού κώδικα και μπορεί να προγραμματιστεί γνωρίζοντας κάποια βασικά για τον προγραμματισμό.

Τέλος, το κόστος του είναι πολύ μικρό. Για την δημιουργία μιας εφαρμογής χωρίς Arduino το κόστος θα ήταν διπλάσιο ίσως και τριπλάσια. Το χαμηλό κόστος μας βοηθάει να δημιουργήσουμε τις δίκες μας εφαρμογές να καλύψουμε τις ανάγκες μας και να κάνουμε πιο εύκολη την καθημερινότητα μας. Ας πειραματιστούμε λοιπόν!

# Βιβλιογραφία

- [1] Εφαρμογές τηλεπικοινωνιακών διατάξεων,  
<http://brain.ee.auth.gr/dokuwiki/doku.php?id=arduino:arduino>, Ιανουάριος 2016
- [2] Πομπός, <https://el.wikipedia.org/wiki/Πομπός>, Ιανουάριος 2016
- [3] Arduino, <https://el.wikipedia.org/wiki/Arduino>, Ιανουάριος 2016
- [4] Διπλωματική εργασία, Σχεδιασμός και υλοποίηση συστήματος έλεγχου πρόσβασης με RFID του Αυγέρης Γεώργιος  
[https://arch.ict.e.uowm.gr/mdasyg/pdfs/avgeris\\_thesis.pdf](https://arch.ict.e.uowm.gr/mdasyg/pdfs/avgeris_thesis.pdf), Ιανουάριος 2016
- [5] Deltahacker, <http://deltahacker.gr/arduino-intro/>, Ιανουάριος 2016
- [6] Διπλωματική εργασία, Οικιακός αυτοματισμός με χρήση μικροελεγκτή του Μαρτίνης Στέλιος  
[http://www.icsd.aegean.gr/website\\_files/diplomatikes/msc/55660114.PDF](http://www.icsd.aegean.gr/website_files/diplomatikes/msc/55660114.PDF),  
Ιανουάριος 2016
- [7] GitHub, <https://github.com/arduino/Arduino>, Ιανουάριος 2016
- [8] Εφαρμογή, <https://www.arduino.cc>, Ιανουάριος 2016



- [9] Receiver, RWS-371-6\_433.92MHz\_ASK\_RF\_Receiver\_Module\_Data\_Sheet.pdf ,  
Ιανουάριος 2016



- [10] Transmitter, TWS-BS-3\_433.92MHz\_ASK\_RF\_Transmitter\_Module\_Data\_Sheet.pdf ,  
Ιανουάριος 2016



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ.

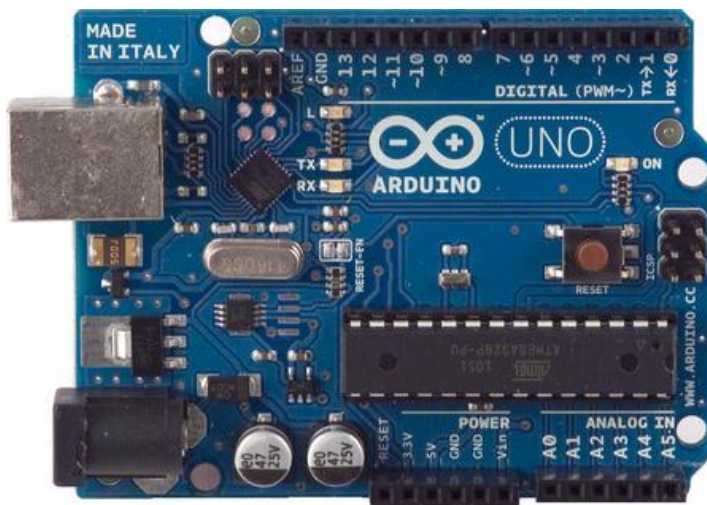
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**Πτυχιακή εργασία**

«Υλοποίηση ασύρματης επικοινωνίας με χρήση μικροελεγκτών».

ΒΥΣΑΝΣΙΩΤΗΣ ΣΤΑΥΡΟΣ

ΚΑΡΒΟΥΝΙΔΟΥ ΑΓΓΕΛΙΚΗ



ΣΥΜΒΟΥΛΟΣ ΚΑΘΗΓΗΤΗΣ ΤΑΤΣΗΣ ΓΕΩΡΓΙΟΣ