

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1617

## **ΣΥΣΤΗΜΑ ΣΥΝΑΓΕΡΜΟΥ ΜΕ ΧΡΗΣΗ ARDUINO**

ΣΠΟΥΔΑΣΤΕΣ : ΒΑΣΙΛΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΡΙΣΤΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΑΤΡΑ 2017



## Πρόλογος

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η υλοποίηση ενός συστήματος συναγερμού σε χώρο. Το ενσωματωμένο σύστημα κατασκευάστηκε με την αρχιτεκτονική Arduino. Το Arduino συνδέεται με μια οθόνη Icd και ένα πληκτρολόγιο ώστε να μπορούμε να ενεργοποιούμε και να απενεργοποιούμε τον συναγερμό ελέγχοντας τι πληκτρολογούμε στην οθόνη μας. Έχοντας συνδέσει δύο διακοπτάκια και έναν αισθητήρα κίνησης πετυχαίνουμε την προσομοίωση πραγματικών παγίδων ανθρωποθυρίδας και παραθύρων. Με την βοήθεια ενός buzzer και του led πετυχαίνουμε την προσομοίωση της σειρήνας και του φάρου αντίστοιχα.

## Περίληψη

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός συστήματος συναγερμού σε χώρο με την χρήση του Arduino σε προγραμματιστικό περιβάλλον Arduino IDE. Το Arduino είναι ένας μικροελεγκτής ο οποίος περιλαμβάνει ένα chip ATmega. Η εργασία αποτελείται από δύο μέρη, το πρακτικό και το προγραμματιστικό μέρος. Συγκεκριμένα για το προγραμματιστικό κομμάτι χρησιμοποιήθηκε γλώσσα προγραμματισμού Wiring, ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++. Στο πρακτικό κομμάτι έγινε ο προγραμματισμός της πλακέτας Arduino Due. Επίσης χρησιμοποιήθηκαν διακόπτες για να προσομοιωθούν τα καίρια σημεία ενός κλειστού χώρου όπως π.χ. μια πόρτα εισόδου, ένα παράθυρο, ένα αισθητήρα παρουσίας καθώς επίσης έγινε και χρήση ενός πληκτρολογίου και μιας LCDοθόνης.

# Περιεχόμενα

---

## Κεφάλαιο 1

1.	Arduino.....	1
1.1.	Τι είναι το Arduino.....	2
1.2.	Λόγοι επιλογής Arduino.....	2
1.3.	Συγκεκριμένα για το Arduino Due.....	4
1.3.1.	Γενικά.....	4
1.3.2.	Τεχνικά χαρακτηριστικά.....	5
1.3.3.	Οφέλη του Arm core.....	5
1.3.4.	Τροφοδοσία.....	6
1.3.5.	Μνήμη.....	7
1.3.6.	Είσοδοι/Εξοδοι.....	7
1.3.7.	Επικοινωνία.....	9
1.3.8.	Usb προστασία από υπερένταση.....	11

## Κεφάλαιο 2

2.1.	Εξαρτήματα που χρησιμοποιήθηκαν.....	13
2.1.1.	Led.....	13
2.1.2.	Αντιστάσεις.....	14
2.1.3.	Breadboard.....	15
2.1.4.	Switch.....	16
2.1.5.	Tip120.....	17
2.1.6.	Pir.....	17
2.1.7.	Buzzer.....	19
2.1.8.	Lcd.....	19
2.1.9.	Keypad.....	20
2.2.10.	Ποτενσιόμετρο.....	21

### Κεφάλαιο 3

3.1. Εγκατάσταση προγράμματος Arduino.....	23
3.2. Περιβάλλον ανάπτυξης.....	24
3.3. Δομή προγράμματος.....	25
3.4. Εγκατάσταση βιβλιοθηκών.....	26
3.5. Επικοινωνία Arduino με περιβάλλον ανάπτυξης.....	27

### Κεφάλαιο 4

4. Πειραματικό μέρος.....	29
4.1. Συνδεσμολογία.....	30
4.2. Κώδικας.....	37
Βιβλιογραφία .....	52

## *Home Alarm System*

# Κεφάλαιο 1<sup>ο</sup>

---

## 1. Arduino



Εικόνα 1: Arduino Due

## 1.1. Τι είναι το Arduino

Το Arduino είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή, εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες, το διάγραμμα και οι πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

## 1.2. Λόγοι επιλογής Arduino

Υπάρχουν πολλοί άλλοι μικροελεγκτές που διατίθενται για την εξυπηρέτηση των αναγκών μας, όπως είναι οι Parallax Basic Stamp, Netmedia του BX-24, Phidgets, Handyboard του MIT και πολλοί άλλοι προσφέρουν παρόμοια λειτουργικότητα. Όλα αυτά τα εργαλεία καλύπτονται κατά ένα πολύ μεγάλο μέρος τους, από τον προγραμματιζόμενο μικροελεγκτή AVR της ATME1 το οποίο είναι ένα πακέτο εύκολο στη χρήση. Το Arduino απλοποιεί τη διαδικασία της εργασίας με μικροελεγκτές αλλά προσφέρει ακόμα και κάποια πλεονέκτημα για τους εκπαιδευτικούς, τους μαθητές και τους ενδιαφερόμενους ερασιτέχνες σε σχέση με άλλα συστήματα.

Αυτά είναι:

### Ø Λίγα έξοδα

Οι Arduino πλακέτες είναι σχετικά φθηνές σε σύγκριση με άλλες πλακέτες μικροελεγκτών. Οι πιο φθηνές εκδόσεις της ενότητας Arduino μπορούν να



συναρμολογηθούν με το χέρι, ακόμη και οι προ-συναρμολογημένες ενότητες **Arduino** κοστίζουν λιγότερο από 25 €.

### Ø Cross-platform

Το **Arduino** λογισμικό τρέχει σε **Windows**, **Macintosh OSX** και **Linux** λειτουργικά συστήματα. Τα περισσότερα συστήματα μικροελεγκτή περιορίζονται στα **Windows**.

### Ø Απλό και ξεκάθαρο περιβάλλον προγραμματισμού

Το περιβάλλον προγραμματισμού **Arduino** είναι εύκολο στη χρήση για αρχάριους, αλλά είναι και αρκετά ευέλικτο για προχωρημένους χρήστες να το εκμεταλλευτούν. Επίσης για τους εκπαιδευτικούς, είναι μια χρήσιμη βάση το περιβάλλον προγραμματισμού , έτσι ώστε οι μαθητές να μάθουν και μέσα από αυτό το περιβάλλον θα είναι πιο εξοικειωμένοι με την εμφάνιση και την αίσθηση του **Arduino**.

### Ø Open source και επεκτάσιμο λογισμικό

Το λογισμικό **Arduino** δημοσιεύεται ως εργαλείο ανοικτού κώδικα, που διατίθεται για την επέκταση από έμπειρους προγραμματιστές. Η γλώσσα μπορεί να επεκταθεί σε **C ++** βιβλιοθήκες και οι άνθρωποι που ξέρουν και κατανοούν τις τεχνικές λεπτομέρειες μπορούν να ασχοληθούν στο **Arduino** με τη γλώσσα προγραμματισμού **AVR-C** επί των οποίων βασίζεται. Ομοίως, μπορείτε να προσθέσετε τον κωδικό **AVR-C** απευθείας στο **Arduino** τα προγράμματά σας.

### Ø Το Arduino βασίζεται σε μικροελεγκτές της Atmel

Τα **sketch** δημοσιεύονται με την άδεια **Creative Commons**, κάποιιοι έμπειροι προγραμματιστές μπορούν να κάνουν το δικό τους **sketch**, επεκτείνοντάς και βελτιώνοντάς το. Ακόμα και σχετικά άπειροι χρήστες μπορούν να δημιουργήσουν

οποιοδήποτε κύκλωμα πάνω στη **breadboard**, προκειμένου να κατανοήσουμε πώς λειτουργεί και να εξοικονομήσουμε χρήματα.

### 1.3. ΣΥΓΚΕΚΡΙΜΕΝΑ ΓΙΑ ΤΟ ARDUINO DUE

#### 1.3.1. Γενικά

Το Arduino Due είναι μια πλακέτα μικροελεγκτή με βάση την CPU Atmel SAM3X8E ARM Cortex-M3 . Είναι η πρώτη πλακέτα Arduino που βασίζεται σε έναν 32-bit πυρήνα ARM μικροελεγκτή. Έχει 54 ψηφιακές ακίδες εισόδου / εξόδου (εκ των οποίων 12 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 12 αναλογικές εισόδους, 4 UARTs (hardware σειριακές θύρες), ένα ρολόι 84 MHz, μια σύνδεση OTG USB , 2 DAC (ψηφιακό σε αναλογικό) , 2 TWI, μια υποδοχή ρεύματος, μια κεφαλίδα SPI, μια κεφαλίδα JTAG, ένα κουμπί επαναφοράς και ένα κουμπί διαγραφής.

Σε αντίθεση με τις περισσότερες πλακέτες Arduino, η πλακέτα Arduino τρέχει σε 3.3V. Η μέγιστη τάση που οι ακίδες I / O μπορούν να αντέξουν είναι 3.3V. Εφαρμόζοντας τάσεις υψηλότερες από 3.3V σε κάθε pin I / O θα μπορούσε να βλάψει την πλακέτα.

Η πλακέτα περιέχει όλα όσα χρειάζονται για τη στήριξη του μικροελεγκτή. Απλά συνδέστε το σε έναν υπολογιστή με ένα καλώδιο micro-USB ή με έναν μετασχηματιστή AC σε DC ή μια μπαταρία για να ξεκινήσετε.

### 1.3.2. Τεχνικά χαρακτηριστικά

#### AVR Arduino microcontroller

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

### 1.3.3. Οφέλη του ARM core

- Ένας πυρήνας 32-bit, ο οποίος επιτρέπει λειτουργίες σε 4 byte ευρέων δεδομένων μέσα σε ένα ενιαίο ρολόι της CPU.
- Ρολόι CPU χρονισμένο στα 84Mhz.
- 96 KBytes της SRAM.
- 512 KBytes μνήμης Flash για τον κώδικα
- Ο ελεγκτής DMA, μπορεί να ανακουφίσει την CPU με το να κάνει στη μνήμη μακροπρόθεσμες εντατικές εργασίες.

### 1.3.4. Τροφοδοσία

Το Arduino DUE μπορεί να τροφοδοτείται μέσω της υποδοχής USB ή με εξωτερικό τροφοδοτικό. Η πηγή ενέργειας επιλέγεται αυτόματα.

Εξωτερική ισχύς (χωρίς καλώδιο-USB) μπορεί να προέλθει είτε από ένα AC-σε-DC μετασχηματιστή ή μια μπαταρία.

Μπορεί να λειτουργήσει σε μια εξωτερική τροφοδοσία 6 έως 20 βολτ. Εάν παρέχεται με λιγότερο από 7V, ωστόσο, η ακίδα των 5V μπορεί να παρέχει λιγότερο από πέντε βολτ και η πλακέτα μπορεί να είναι ασταθής. Εάν τροφοδοτείτε με περισσότερα από 12V, ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να προκαλέσει ζημιά στην πλακέτα. Η συνιστώμενη τάση κυμαίνεται 7 έως 12 βολτ.

Οι **ακίδες ισχύος** είναι οι εξής:

**Vin.** Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιούμε μια εξωτερική πηγή ενέργειας (σε αντίθετα με τα 5 βολτ από τη σύνδεση USB ή άλλες οργανωμένες πηγή ενέργειας).

**5V** .Αυτή η ακίδα εξάγει μια ρυθμιζόμενη τάση 5V από τον ρυθμιστή της πλακέτας. Η πλακέτα μπορεί να τροφοδοτηθεί είτε από την υποδοχή ρεύματος DC (7 - 12V), την υποδοχή USB (5V), ή την ακίδα VIN της πλακέτας(7-12V). Προμηθεύοντας τάση μέσω των 5V ή 3.3V ακίδων παρακάμπει το ρυθμιστή και μπορεί να βλάψει την πλακέτα

**3V3** . είναι μια παροχή 3,3 volt η οποία παράγεται από τον ρυθμιστή της πλακέτας. Η μέγιστη κατανάλωση ρεύματος είναι 800 mA. Ο ρυθμιστής παρέχει επίσης την παροχή ηλεκτρικού ρεύματος στο μικροελεγκτή SAM3X.

**GND** . ακίδες εδάφους.

**IOREF** . Αυτή η ακίδα στην πλακέτα Arduino παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Μια κατάλληλη ρυθμισμένη shield μπορεί να διαβάσει την

τάση pin IOREF και επιλέγει την κατάλληλη πηγή ενέργειας ή να ενεργοποιήσει μεταφραστές τάσης στις εξόδους για την εργασία με την 5V ή 3.3V.

### 1.3.5. Μνήμη

Η SAM3X έχει 512 KB (2 ξεχωριστούς χώρους των 256 KB) της μνήμης flash για την αποθήκευση του κώδικα. Ο bootloader είναι προεγκατεστημένος στο εργοστάσιο από την Atmel και αποθηκεύεται σε μια ειδική μνήμη ROM. Η διαθέσιμη SRAM είναι 96 KB σε δύο συνεχόμενους χώρους των 64 KB και 32 KB.

Είναι δυνατόν να διαγράψουμε τη μνήμη Flash της SAM3X με το κουμπί διαγραφής. Αυτό θα αφαιρέσει το τρέχον φορτωμένο σκίτσο από την MCU. Για διαγραφή, πατήστε και κρατήστε πατημένο το κουμπί Διαγραφής για μερικά δευτερόλεπτα, ενώ η πλακέτα μας τροφοδοτείται.

### 1.3.6. Είσοδοι και έξοδοι

Digital I / O: ακίδες από 0 να 53

Κάθε μια από τις 54 ψηφιακές ακίδες της due μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος, χρησιμοποιώντας `pinMode ()`, `digitalWrite ()`, και `digitalRead ()` λειτουργίες. Λειτουργούν στα 3,3 βολτ. Κάθε ακίδα μπορεί να παρέχει πηγή ρεύματος 3 mA ή 15 mA, ανάλογα με την ακίδα, ή να δεχτεί ρεύμα 6 mA ή 9 mA, ανάλογα με την ακίδα. Έχουν επίσης μια εσωτερική αντίσταση pull-up 100 KOhm. Επιπλέον, μερικές καρφίτσες έχουν εξειδικευμένες λειτουργίες:

**Serial:** 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)

Χρησιμοποιούνται για τη λήψη (RX) και να διαβιβάσουν σειριακά δεδομένα (TX) TTL (με επίπεδο 3,3 V). Οι ακίδες 0 και 1 συνδέονται με τις αντίστοιχες ακίδες του ATmega16U2 USB-σε-TTL σειριακού chip.

**PWM:** Pins 2 έως 13

Παρεχουν 8-bit PWM έξοδο με την `analogWrite()` λειτουργία. Η ανάλυση του PWM μπορεί να αλλάξει με τη λειτουργία `analogWriteResolution()`.

**SPI:** SPI ακίδες (header ICSP σε άλλους πίνακες Arduino)

Αυτές οι ακίδες υποστηρίζουν την επικοινωνία SPI, χρησιμοποιώντας την βιβλιοθήκη SPI. Οι ακίδες SPI προεξέχουν πάνω στην κεντρική πλακέτα (6 ακίδες), οι οποίες είναι φυσικά συμβατές με το Uno, Leonardo και Mega2560. Οι ακίδες SPI μπορούν να χρησιμοποιηθούν μόνο για την επικοινωνία με άλλες συσκευές SPI, όχι για τον προγραμματισμό του SAM3X με την τεχνική In-Circuit-Serial-Programming. Η SPI του Due διαθέτει επίσης προηγμένα χαρακτηριστικά που μπορούν να χρησιμοποιηθούν με τις Εκτεταμένες μεθόδους SPI για Due.

**"L" LED:** pin 13

Υπάρχει ένα ενσωματωμένο LED που συνδέεται με την ψηφιακή ακίδα 13. Όταν η ακίδα είναι high, η ενδεικτική λυχνία είναι αναμμένη, όταν η ακίδα είναι low, αυτή είναι απενεργοποιημένη. Είναι επίσης δυνατόν να χειριστούμε την φωτεινότητα του LED επειδή η ψηφιακή ακίδα 13 είναι επίσης μια PWM έξοδος.

**Αναλογικοί είσοδοι:** καρφίτσες από A0 έως A11

Το Due έχει 12 αναλογικές εισόδους, καθεμία από τις οποίες μπορεί να παρέχει 12 bits ανάλυσης (δηλαδή 4096 διαφορετικές τιμές). Από προεπιλογή, η ανάλυση των μετρήσεων ορίζεται σε 10 bits, για συμβατότητα με άλλες πλακέτες Arduino. Είναι δυνατόν να αλλάξει η ανάλυση του ADC με `analogReadResolution()`. Οι αναλογικοί

είσοδοι του Due μετρούν από το ground σε μια μέγιστη τιμή των 3.3V. Εφαρμόζοντας πάνω από 3.3V σε ακίδες του DUE θα βλάψει το τσιπ SAM3X. Η λειτουργία analogReference () αγνοείται στο Due.

Άλλες καρφίτσες στον πίνακα:

- **AREF**

Είναι τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με analogReference () .

- **Reset**

Τοποθετήστε το στο LOW για να επαναφέρετε τον μικροελεγκτή. Συνήθως χρησιμοποιείται για να προσθέσουμε ένα κουμπί επαναφοράς στις Shields που μπλοκάρουν την πλακέτα.

### 1.3.7. Επικοινωνία

Το Arduino Due έχει μια σειρά από εγκαταστάσεις για την επικοινωνία με έναν υπολογιστή, ένα άλλο Arduino ή άλλους μικροελεγκτές, και διάφορες συσκευές όπως κινητά τηλέφωνα, τάμπλετ, φωτογραφικές μηχανές και ούτω καθεξής. Η SAM3X παρέχει ένα hardware UART και τρεις hardware USARTs για TTL (3.3V) σειριακή επικοινωνία.

Η θύρα **προγραμματισμού** συνδέεται με μια ATmega16U2, η οποία παρέχει μια εικονική θύρα COM για το λογισμικό σε έναν συνδεδεμένο υπολογιστή (να αναγνωρίσει τη συσκευή, στα Windows μηχανήματα θα χρειαστείτε ένα αρχείο .inf, αλλά στα OSX και Linux μηχανήματα θα αναγνωρίσει τη πλακέτα ως θύρα COM αυτόματα ). Η 16U2 συνδέεται επίσης με το UART hardware SAM3X. Οι σειριακές ακίδες RX0 και TX0 παρέχουν σειριακή σε USB επικοινωνία για τον προγραμματισμό της πλακέτας μέσω του μικροελεγκτή ATmega16U2. Το λογισμικό του Arduino

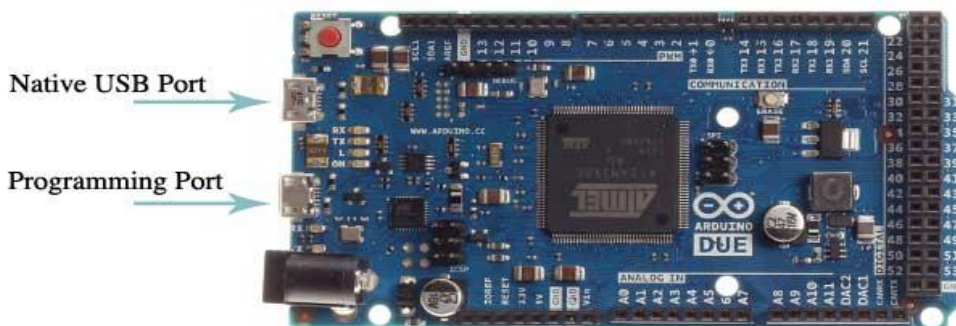
περιλαμβάνει μια σειριακή οθόνη η οποία επιτρέπει απλά δεδομένα κειμένου να αποστέλλονται προς και από τη πλακέτα. Τα LED RX και TX στη πλακέτα θα αναβοσβήνουν όταν τα δεδομένα μεταδίδονται μέσω του τσιπ και τη USB σύνδεση ATmega16U2 στον υπολογιστή (αλλά όχι για σειριακή επικοινωνία στις ακίδες 0 και 1).

Η **Native** θύρα USB είναι συνδεδεμένη με το SAM3X. Επιτρέπει για σειριακή (CDC) επικοινωνία μέσω USB. Αυτό παρέχει μια σειριακή σύνδεση με το Serial Monitor ή άλλες εφαρμογές στον υπολογιστή σας. Επιτρέπει επίσης στην μίμηση ενός πληκτρολόγιου ή ποντικιού USB σε ένα συνδεδεμένο υπολογιστή.

### Προγραμματισμός

Η Due μπορεί να προγραμματιστεί με [το λογισμικό Arduino \(IDE\)](#).

Φορτώνοντας σκίτσα στο SAM3X είναι διαφορετικό από ό,τι άλλοι μικροελεγκτές AVR που βρίσκονται σε άλλες πλακέτες Arduino, επειδή η μνήμη flash θα πρέπει να διαγράφεται πριν από το νέο προγραμματισμό.



Εικόνα 2: Native port και programming port

**Θύρα προγραμματισμού:** Για να χρησιμοποιήσετε αυτή τη θύρα, επιλέξτε "Arduino Due (Programming Port)" ως πλακέτα στο Arduino IDE. Συνδέστε τη θύρα προγραμματισμού του Due (βρίσκεται πλησιέστερα προς την υποδοχή ρεύματος DC)



στον υπολογιστή σας. Η θύρα προγραμματισμού χρησιμοποιεί το 16U2 ως τσιπ USB-to-serial συνδέεται με το πρώτο UART του SAM3X (RX0 και TX0). Η 16U2 έχει δύο ακροδέκτες που συνδέονται με το Reset και τις ακίδες Διαγραφής του SAM3X. Ανοίγοντας και κλείνοντας τη θύρα προγραμματισμού που συνδέεται σε 1200bps ενεργοποιεί μια διαδικασία «hard erase» του τσιπ SAM3X, ενεργοποιώντας τις Erase και Reset ακίδες της SAM3X πριν από την επικοινωνία με το UART. Αυτή είναι η συνιστώμενη θύρα για τον προγραμματισμό του Due. Είναι πιο αξιόπιστο από το "Soft Erase" που εμφανίζεται στην Native θύρα.

**Native Θύρα:** Για να χρησιμοποιήσετε αυτή τη θύρα, επιλέξτε "Arduino Due (Native USB Port)" ως πλακέτα στο Arduino IDE. Η Native θύρα USB συνδέεται απευθείας με το SAM3X. Συνδέστε την Native θύρα USB του Due (βρίσκεται πλησιέστερα προς το κουμπί reset) στον υπολογιστή σας. Ανοίγοντας και κλείνοντας την Native Θύρα στα 1200bps ενεργοποιεί μια διαδικασία «soft erase»: η μνήμη flash διαγράφεται και η πλακέτα ξαναρχίζει με τον bootloader. Εάν η MCU συνετρίβη για κάποιο λόγο, είναι πιθανό ότι η soft erase δεν θα λειτουργήσει καθώς αυτή η διαδικασία συμβαίνει εξ ολοκλήρου σε λογισμικό για την SAM3X. Ανοίγοντας και κλείνοντας τη native θύρα σε διαφορετικό ρυθμό baud δεν θα κάνει reset η SAM3X. Σε αντίθεση με άλλες πλακέτες Arduino που χρησιμοποιούν avrdude για το φόρτωμα, το Due βασίζεται σε bossac.

### 1.3.8. USB προστασία από υπερένταση

Το Arduino Due έχει δυνατότητα επαναφοράς polyfuse που προστατεύει τις θύρες USB του υπολογιστή σας από βραχυκύκλωμα και υπερένταση. Αν και οι περισσότεροι υπολογιστές παρέχουν τη δική τους εσωτερική προστασία, η ασφάλεια παρέχει ένα επιπλέον επίπεδο προστασίας. Εάν περισσότερα από 500 mA εφαρμοστούν στη θύρα USB, η ασφάλεια θα σπάσει αυτόματα τη σύνδεση μέχρι το βραχυκύκλωμα ή υπερφόρτωση να αφαιρεθεί.

## Home Alarm System

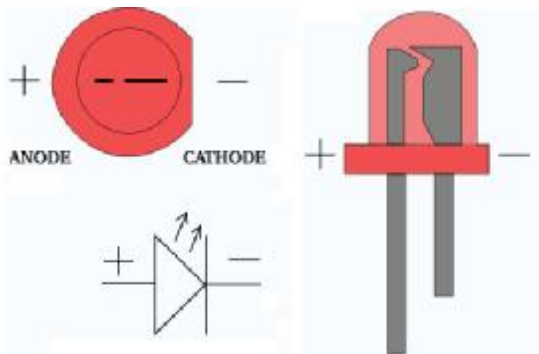
## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

---

Εξαρτήματα που χρησιμοποιήθηκαν στην παρούσα πτυχιακή και ανάλυση αυτών:

### 2.1.1. Led

**Δίοδος Εκπομπής Φωτός, (LED, Light Emitting Diode)**, αποκαλείται ένας ημιαγωγός ο οποίος εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης (forward-biased). Όταν μια κατάλληλη τάση εφαρμόζεται στα ηλεκτρόδια, τα ηλεκτρόνια είναι σε θέση να απελευθερώνουν ενέργεια με τη μορφή φωτονίων. Τα Led έχουν πολλά πλεονεκτήματα σε σχέση με το φως πυρακτώσεως, όπου κάποια από αυτά είναι η χαμηλότερη κατανάλωση ενέργειας, μεγαλύτερη διάρκεια ζωής, μικρότερο μέγεθος, και ταχύτερη εναλλαγή. Για την σωστή σύνδεση των led τοποθετούμε το + της τροφοδοσίας στο μακρύ ακροδέκτη και το - στον άλλο ακροδέκτη.



Εικόνα 3: Led

### 2.1.2. Αντιστάσεις (Resistors)

Η **ηλεκτρική αντίσταση** (R) είναι ένα ηλεκτρολογικό/ηλεκτρονικό εξάρτημα το οποίο χρησιμοποιείται σε διάφορα κυκλώματα για τον έλεγχο της ροής του ρεύματος. Η μονάδα μέτρησης της ηλεκτρικής αντίστασης στο Διεθνές Σύστημα Μονάδων (SI), είναι το Ωμ(Ohm). Οι τιμές των αντιστάσεων παρουσιάζουν ανοχή σε σχέση με αυτές που υπολογίζονται λαμβάνοντας υπόψη τα χρώματα τους. Έτσι, οι κατασκευαστές, έχουν προσθέσει μια τέταρτη ζώνη χρώματος, την ανοχή, για να δηλώσουν την ακρίβεια της αντίστασης.



**Εικόνα 4: Αντίσταση**

Ο χρωματικός κώδικας αποτελεί μέσο για τον υπολογισμό της ωμικής αντίστασης καθώς και της ανοχής της. Οι αντιστάσεις χρησιμοποιούν τον χρωματικό κώδικα για να γίνει εύκολη η ανάγνωση τους.

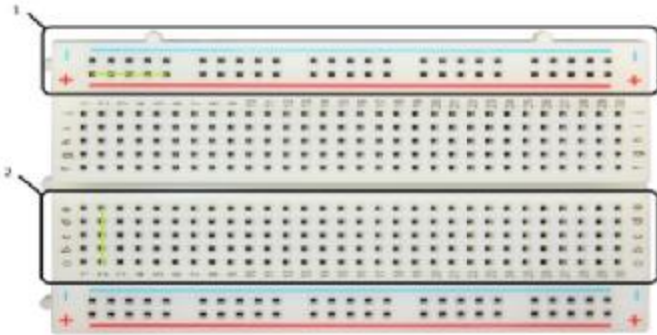
Ο πίνακας του χρωματικού κώδικα που εφαρμόζεται για κάθε αντίσταση φαίνεται στο παρακάτω σχήμα:

## Home Alarm System

Χρώμα	1η λωρίδα	2 <sup>η</sup> λωρίδα	3 <sup>η</sup> λωρίδα	4 <sup>η</sup> λωρίδα(ανοχή)	Θερμικός Συντελεστής
Καφέ	1	1	$\times 10^1$	$\pm 1\%$ (F)	100ppm
Κόκκινο	2	2	$\times 10^2$	$\pm 2\%$ (G)	50ppm
Πορτοκαλί	3	3	$\times 10^3$		15ppm
Κίτρινο	4	4	$\times 10^4$		25ppm
Πράσινο	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Μπλε	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Μωβ	7	7	$\times 10^7$	$\pm 1\%$ (B)	
Γκρι	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
Λευκό	9	9	$\times 10^9$		
Χρυσάφι			$\times 0.1$	$\pm 5\%$ (J)	
Ασημί			$\times 0.01$	$\pm 10\%$ (K)	
Κανένα				$\pm 20\%$ (M)	

### 2.1.3. Breadboard

Το Breadboard είναι μια κατασκευή το οποίο μας επιτρέπει να συνδέουμε ηλεκτρονικά εξαρτήματα χωρίς κάποια συγκόλληση. Το υλικό του αρχικά ήταν ένα επεξεργασμένο κομμάτι ξύλου το οποίο κατά δεκαετία του 1970 το χρησιμοποιούσαν για τον τεμαχισμό του ψωμιού. Από εκεί προέρχεται και ο όρος Breadboard . Ένα σύγχρονο breadboard αποτελείται από ένα διάτρητο πλαστικό μπλοκ που περιέχει πολυάριθμα κλιπάκια ελατηρίου, τα οποία είναι φτιαγμένα από κασίτερο, φωσφόρο, ασημί ή ακόμα και νικέλιο. Τα κλιπάκια συχνά αποκαλούνται ως σημεία επαφής. Το νούμερο των σημείων επαφής συχνά δίνεται από το χαρτί των οδηγιών που συμπεριλαμβάνεται με το breadboard.



Εικόνα 5: Breadboard

#### 2.1.4. Switch

Οι **διακόπτες** έχουν σημεία με τα οποία συνδέονται με το κύκλωμα τα οποία ονομάζονται **ακροδέκτες**. Κάθε διακόπτης έχει δύο καταστάσεις, την κατάσταση που είναι **κλειστός** και την κατάσταση που είναι **ανοιχτός**. Όταν ένας διακόπτης είναι ανοιχτός δεν επιτρέπει τη διέλευση ηλεκτρικού ρεύματος μεταξύ των ακροδεκτών του, ενώ όταν είναι κλειστός επιτρέπει τη διέλευση ηλεκτρικού ρεύματος μεταξύ των ακροδεκτών του. Ο διακόπτης διατηρεί την κατάσταση στην οποία βρίσκεται, ενώ αυτή μεταβάλλεται μόνο από εξωτερικούς του στοιχείου παράγοντες, όπως είναι το πάτημα ενός κουμπιού. Κάθε κλειστός διακόπτης μπορεί να ανοίξει, ενώ κάθε ανοιχτός διακόπτης μπορεί να κλείσει.

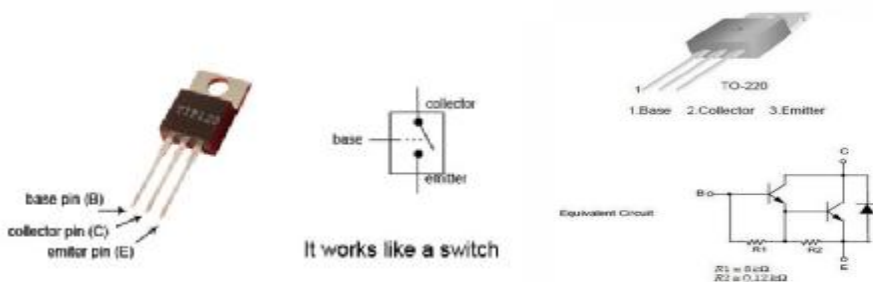
Για να διέλθει ηλεκτρικό ρεύμα μέσω ενός διακόπτη, πρέπει να είναι κλειστός και να εφαρμοστεί στους ακροδέκτες του διαφορά δυναμικού. Για να μη διέλθει ηλεκτρικό ρεύμα αρκεί να είναι ανοιχτός, αν και είναι πιθανό όταν είναι κλειστός να μη διαρρέεται από ηλεκτρικό ρεύμα, γιατί δεν υπάρχει τάση.



Εικόνα 6: Toggle switch

### 2.1.5. Tip120

Το TIP 120 είναι ένα NPN Darlington τρανζίστορ με ένα τρέχον κέρδος 1000, είναι μια καλή επιλογή για τη διασύνδεση μεγαλύτερου ρεύματος ή υψηλότερα φορτία τάσης σε ένα Arduino ή άλλους μικροελεγκτές. Μπορεί να χρησιμοποιηθεί σαν διακόπτης για τον έλεγχο DC κινητήρες, LED, buzzer με ελάχιστη κατανάλωση ρεύματος από τον μικροελεγκτή.



Εικόνα 7: Tip120

### 2.1.6. Pir

Ένας **παθητικός υπέρυθρος αισθητήρας ( PIR )** είναι ένας ηλεκτρονικός αισθητήρας που μετρά υπέρυθρο (IR) φως που ακτινοβολείτε από τα αντικείμενα στο οπτικό του πεδίο. Πιο συχνά χρησιμοποιούνται για ανίχνευση κίνησης .

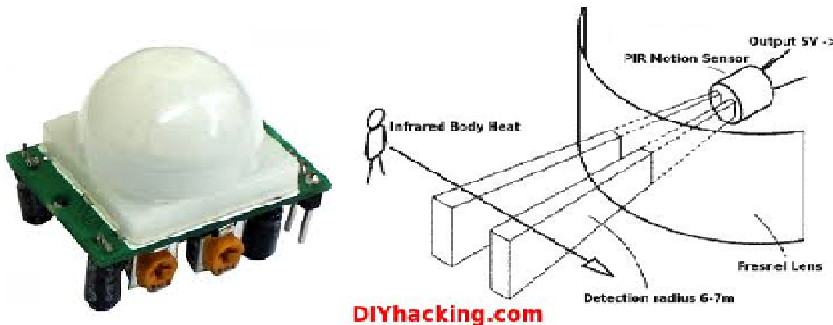
Όλα τα αντικείμενα με θερμοκρασία πάνω από το απόλυτο μηδέν εκπέμπουν θερμότητα με τη μορφή ακτινοβολίας. Συνήθως αυτή η ακτινοβολία δεν είναι ορατή στο ανθρώπινο μάτι , επειδή ακτινοβολεί σε υπέρυθρα μήκη κύματος, αλλά μπορεί να ανιχνευθεί με ηλεκτρονικές συσκευές σχεδιασμένες για ένα τέτοιο σκοπό.

Ο όρος παθητική (passive) στην περίπτωση αυτή αναφέρεται στο γεγονός ότι οι συσκευές PIR δεν δημιουργούν ή ακτινοβολούν οποιαδήποτε ενέργεια για τον σκοπό

της ανίχνευσης. Εργάζονται εξ ολοκλήρου από την ανίχνευση της ενέργειας που εκπέμπεται από άλλα αντικείμενα.

Η λειτουργία:

Ένας αισθητήρας PIR ανιχνεύει αλλαγές στην ποσότητα της υπέρυθρης ακτινοβολίας που προσπίπτει επάνω σε αυτό, το οποίο ποικίλλει ανάλογα με την θερμοκρασία και τα χαρακτηριστικά της επιφάνειας των αντικειμένων μπροστά από τον αισθητήρα. Όταν ένα αντικείμενο, όπως ένας άνθρωπος, περνά μπροστά από το φόντο, η θερμοκρασία σε αυτό το σημείο στο οπτικό πεδίο του αισθητήρα του θα αυξηθεί από την θερμοκρασία δωματίου μέχρι τη θερμοκρασία του σώματος, και στη συνέχεια πάλι πίσω. Ο αισθητήρας μετατρέπει την προκύπτουσα μεταβολή στην εισερχόμενη υπέρυθρη ακτινοβολία σε μια αλλαγή στην τάση εξόδου, και αυτό προκαλεί την ανίχνευση. Αντικείμενα της ίδιας θερμοκρασίας αλλά με διαφορετικά χαρακτηριστικά επιφανείας μπορεί επίσης να έχουν μια διαφορετική εκπομπή υπέρυθρης ακτινοβολίας και έτσι τη μετακίνησή τους σε σχέση με το φόντο μπορεί να πυροδοτήσει τον ανιχνευτή.



Εικόνα 8: Pir sensor



### 2.1.7. Buzzer

Το buzzer είναι ένα ολοκληρωμένο κύκλωμα τροφοδοτούμενο από DC τάση που παράγει ηχητικά κύματα στο ακουστικό φάσμα συχνοτήτων. Χρησιμοποιείται ευρέως σε υπολογιστές, εκτυπωτές, συναγερμούς, τηλέφωνα και άλλες ηχητικές συσκευές. Περιέχει έναν πιεζοκρύσταλλο, το οποίο ταλαντώνεται με διαφορετική συχνότητα ανάλογα με την τάση εισόδου(pwm)



Εικόνα 9 : Buzzer

### 2.1.8. Lcd

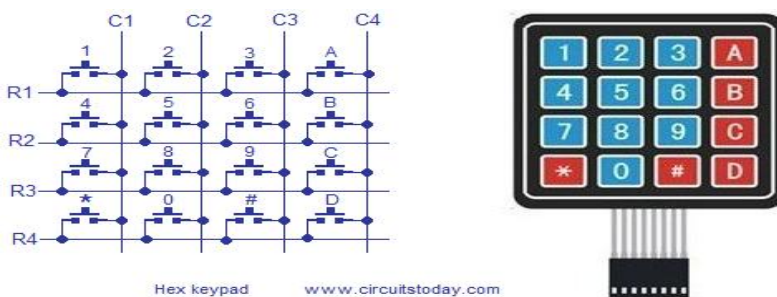
Η **οθόνη υγρών κρυστάλλων (LCD)** είναι μία επίπεδη οθόνη ή μία άλλη ηλεκτρονικά διαμορφωμένη οπτική συσκευή, η οποία χρησιμοποιεί τις ιδιότητες φωτός των υγρών κρυστάλλων. Οι υγροί κρύσταλλοι δεν εκπέμπουν απευθείας το φώς, αντί αυτού χρησιμοποιούν έναν οπίσθιο φωτισμό (**backlight**) ή ανακλαστήρα, έτσι ώστε να παράξει έγχρωμες ή μονόχρωμες εικόνες. Οι οθόνες αυτές είναι ικανές να προβάλλουν αυθαίρετες εικόνες (όπως κάνει μια κοινή οθόνη ενός ηλεκτρονικού υπολογιστή) ή σταθερές εικόνες με χαμηλή περιεκτικότητα πληροφοριών, που μπορούν είτε να προβάλλονται είτε να αποκρύπτονται, όπως για παράδειγμα η εμφάνιση λέξεων και ψηφίων.



Εικόνα 10: Lcd 16x2

### 2.1.9. Keypad

Το keypad Arduino 4\*4 πρόκειται για 16 μπουτόν τα οποία χωρίζονται σε 4 γραμμές και 4 στήλες. Στην κάθε γραμμή συνδέετε το ένα πόδι από όλα τα μπουτόν της γραμμής και βγάζουν μια κοινή έξοδο. Αντίστοιχα στις στήλες συνδέετε το άλλο πόδι από όλα τα μπουτόν της στήλης και βγάζουν μια κοινή έξοδο. Έτσι για παράδειγμα με το πάτημα του πλήκτρου 6 ενεργοποιείτε η C3(στήλη) και η R2 (Σειρά).Ενώ με το πάτημα του πλήκτρου 9 η C3(στήλη) και η R3(Σειρά).



Εικόνα 11: Keypad 4x4

### 2.1.10. Ποτενσιόμετρο

Το **ποτενσιόμετρο** είναι αναλογικό ηλεκτρονικό εξάρτημα, που χρησιμοποιείται στα κυκλώματα ως μεταβλητή αντίσταση. Αποτελείται από αγωγική πλάκα σχήματος  $\Omega$ , πάνω στην οποία γυρίζει, με τη βοήθεια ενός στροφέα, μια επαφή. Ανάλογα με την απόσταση της επαφής από την είσοδο του ρεύματος στο ποτενσιόμετρο μεταβάλλεται και η αντίσταση. Στην πραγματικότητα είναι ένας καταμεριστής τάσης, μόνο που έχει μεταβλητή σχέση καταμερισμού και χρησιμεύει σε μετρήσεις χαμηλών τάσεων. Πρακτικά αποτελείται από μια μεταβλητή αντίσταση πάνω στη οποία κινείται ένας δρομέας, η κίνηση του οποίου ρυθμίζεται από τον χρήστη.



Εικόνα 12: Ποτενσιόμετρο

## Home Alarm System

## ΚΕΦΑΛΑΙΟ 3ο

### 3.1 Εγκατάσταση προγράμματος Arduino

Για να ξεκινήσουμε να γράφουμε τον κώδικα μας πρέπει πρώτα να κατεβάσουμε και να εγκαταστήσουμε το Arduino IDE, το οποίο υπάρχει δωρεάν στο site του Arduino <http://www.arduino.cc/en/Main/Software>

Το Arduino IDE είναι διαθέσιμο για λειτουργικά Windows, Mac, και Linux 32bits ή 64bits. Εμείς θα χρησιμοποιήσουμε Windows λειτουργικό. Εφόσον κατεβάσουμε το συμπιεσμένο αρχείο που φαίνεται στην εικόνα το μόνο που μένει είναι να το κάνουμε αποσυμπίεση και να το εγκαταστήσουμε βήμα-βήμα. Έπειτα με διπλό κλικ στο εικονίδιο που έχει εμφανιστεί στην επιφάνεια εργασίας του υπολογιστή μας, θα μπορούμε να μπούμε και να ξεκινήσουμε να γράφουμε το πρόγραμμα μας.



Εικόνα 13: Συμπιεσμένο αρχείο

### 3.2 Περιβάλλον ανάπτυξης του Arduino.

Το περιβάλλον ανάπτυξης είναι μια εφαρμογή γραμμένη σε Java. Έχει σχεδιαστεί για να εισάγει τον προγραμματισμό και στους νέους, που δεν είναι εξοικειωμένοι με την ανάπτυξη προγραμματισμού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης, ο συνδυασμός αγκυλών και είναι σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα μόνο με ένα κλικ. Τα προγράμματα είναι γραμμένα σε γλώσσα C ή C++. Το περιβάλλον ανάπτυξης του Arduino αποτελείται από το μενού, την εργαλειοθήκη στην οποία τρέχουμε το πρόγραμμα στο Arduino, της καρτέλες που έχουμε ανοικτές, τον επεξεργαστή κειμένου στον οποίο γράφουμε το πρόγραμμα και την κονσόλα μηνυμάτων στην οποία βλέπουμε αν έχει περάσει το πρόγραμμα επιτυχώς ή αν έχει λάθη.



Εικόνα 14: Περιβάλλον ανάπτυξης

Κουμπιά εργαλειοθήκης:



Verify/Compile (Έλεγχος/ Μεταγλώττιση): Έλεγχος για λάθη στον κώδικα



Upload: Ανέβασμα του κώδικα στον μικροελεγκτή.



New(Νέο): Δημιουργεί ένα νέο Sketch



Open(Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα Sketch.



Save(Αποθήκευση): Αποθηκεύει το Sketch



Serial Monitor(Σειριακή οθόνη): Ανοίγει την σειριακή οθόνη ώστε να μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο

### 3.3 Η δομή του προγράμματος.

Ένα πρόγραμμα Arduino πρέπει να έχει την παρακάτω δομή. Πρέπει να αποτελείται από ένα void setup και ένα void loop. Στο void setup() γράφουμε και δηλώνουμε τις μεταβλητές μας, ενώ στο void loop γράφουμε το κώδικα μας.

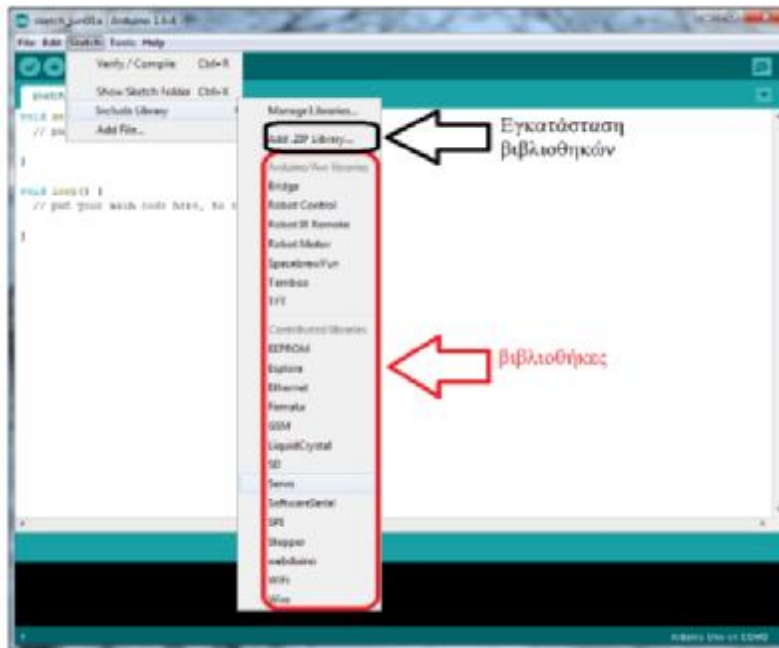
Δήλωση μεταβλητών

```
void setup() {  
  // Αρχικοποίησης  
}  
void loop() {  
  // Κώδικας  
}
```

### 3.4 Εγκατάσταση βιβλιοθηκών στο πρόγραμμα

Ένα άλλο σημαντικό εργαλείο είναι οι βιβλιοθήκες. Υπάρχει ήδη με την εγκατάσταση του Arduino IDE μια μεγάλη γκάμα βιβλιοθηκών έτοιμες για χρήση. Οι βιβλιοθήκες αποτελούνται από έναν κώδικα που μας καθιστά εύκολο να συνδέσουμε μια οθόνη έναν αισθητήρα κ.τ.λ. Για να τρέξουμε τον κώδικα μιας βιβλιοθήκης πρέπει πρώτα να την καλέσουμε στο πρόγραμμα μας, πηγαίνοντας στο `sketch->include library->` και επιλέγουμε την βιβλιοθήκη που χρειαζόμαστε, όπως φαίνεται στην εικόνα 21. Στην περίπτωση όμως που δεν υπάρχει η βιβλιοθήκη που χρειαζόμαστε θα πρέπει να την κατεβάσουμε από το διαδίκτυο και να την εγκαταστήσουμε πηγαίνοντας `sketch->include library-> Add Zip Library` να την βρούμε και να κάνουμε `open`. Μετά την εγκατάσταση θα έχει εμφανιστεί στις βιβλιοθήκες και θα μπορούμε να την χρησιμοποιούμε.





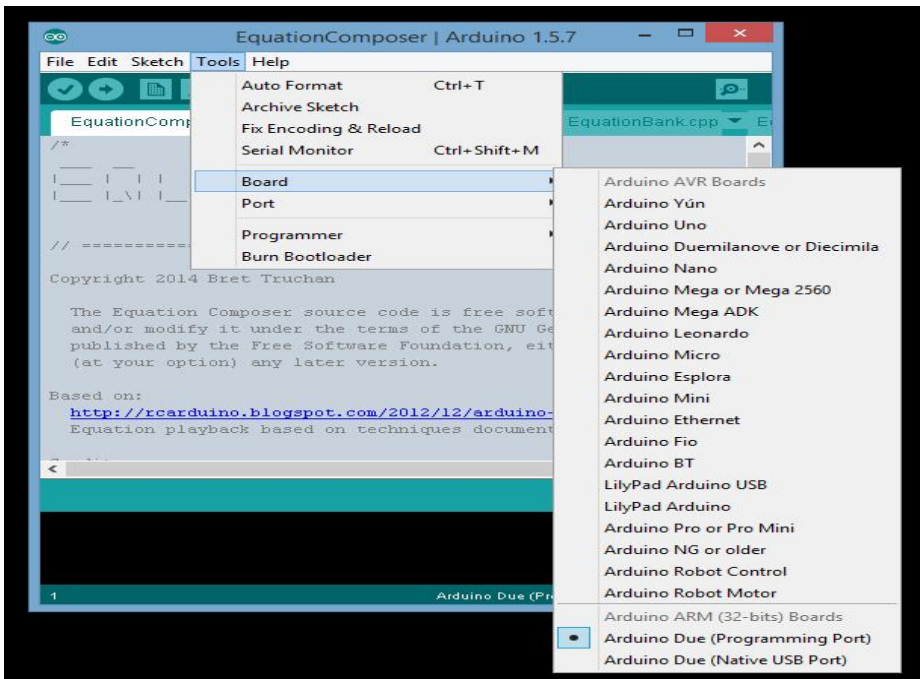
Εικόνα 15: Βιβλιοθήκες

### 3.5 Επικοινωνία Arduino με περιβάλλον ανάπτυξης.

Για την σωστή επικοινωνία του Arduino με το περιβάλλον ανάπτυξης πρέπει να γίνουν κάποιες διαδικασίες. Στην δικιά μας περίπτωση θα χρησιμοποιήσουμε την πλακέτα Arduino Due. Έπειτα συνδέουμε το καλώδιο USB στον υπολογιστή και στην υποδοχή του Arduino(programming port). Για να αναγνωρίσει το πρόγραμμα ποιο Arduino χρησιμοποιούμε και σε ποια θύρα USB το έχουμε συνδέσει, πρέπει να το ρυθμίσουμε εμείς. Για την ρύθμιση του Arduino Due πάμε στο μενού, Tools και επιλέγουμε στο Board το Arduino Due(programming port). Για την σειριακή θύρα πάμε πάλι στο μενού, Tool και επιλέγουμε το Port για τον δικό μας υπολογιστή είναι το COM1.

## Home Alarm System

Σε κάποια windows χρειάζεται να κάνετε εγκατάσταση τους Drivers ώστε να αναγνωρίσει η σειριακή θύρα το Arduino, όμως στην δικιά μας περίπτωση δεν χρειάστηκε.



Εικόνα 16: Programming port Arduino Due

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>

---

### 4. Πειραματικό μέρος

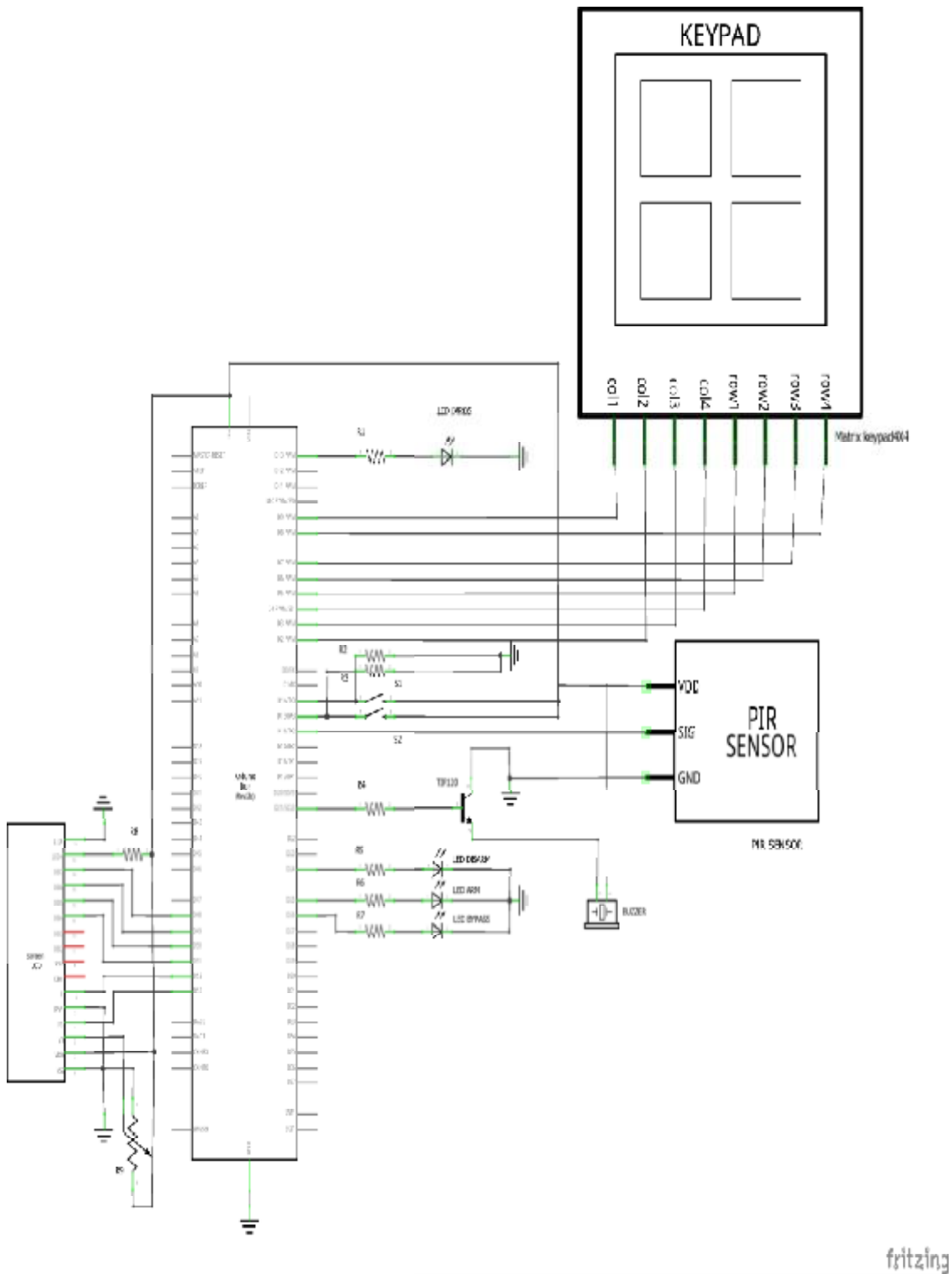
Στο πειραματικό μέρος θα ασχοληθούμε με τον συναγερμό σε χώρο. Πιο συγκεκριμένα θα έχουμε την δυνατότητα να ενεργοποιούμε και να απενεργοποιούμε τον συναγερμό εισάγοντας τον σωστό κωδικό πρόσβασης από το πληκτρολόγιο. Κατά την ενεργοποίηση θα έχουμε δυο επιλογές ενεργοποίησης. Η πρώτη θα μας τοποθετεί στην λειτουργία ARM, με την οποία θα ενεργοποιούνται όλες οι παγίδες του συναγερμού μας(πόρτα εισόδου, παράθυρο, pir sensor) και θα περιμένουν την ανίχνευση κάποιας κίνησης για την πυροδότηση της σειρήνας και του φάρου μας. Η δεύτερη λειτουργία θα μας τοποθετεί στην κατάσταση BY PASS η οποία λειτούργει όπως η ARM με την μονή διάφορα ότι θα θέτει τον pir sensor εκτός λειτουργίας ώστε να μπορούμε να κινούμαστε ελεύθερα μέσα στον χώρο και ταυτόχρονα να είμαστε ασφαλισμένοι με τις περιφερειακές παγίδες(πόρτα εισόδου, παράθυρο).Με την χρήση 2 διακοπών θα προσομοιώσουμε τις παγίδες στην πόρτα εισόδου και παραθύρου. Επίσης θα υπάρχει μια lcd οθόνη στην οποία θα μπορούμε να βλέπουμε την πληκτρολόγηση του κωδικού μας με μορφή αστερίσκου(\*) για να μπορούμε να καταλαβαίνουμε τι πληκτρολογούμε.

#### 4.1. Συνδεσμολογία

Στο επόμενο βήμα πραγματοποιούμε την συνδεσμολογία. Για την συνδεσμολογία θα χρειαστούμε:

1. 1 Arduino Due
2. 1 Lcd οθόνη
3. 1 Breadboard
4. 4 Led (2 κόκκινα, 1 κίτρινο, 1 πράσινο).
5. 1 Buzzer
6. 1 Tip120
7. 1 Keypad
8. 2 Διακόπτες
9. 8 Αντιστάσεις(7 των 220Ω και 1 των 1ΚΩ).
10. 1 Ποτενσιόμετρο 10ΚΩ
11. Καλώδια

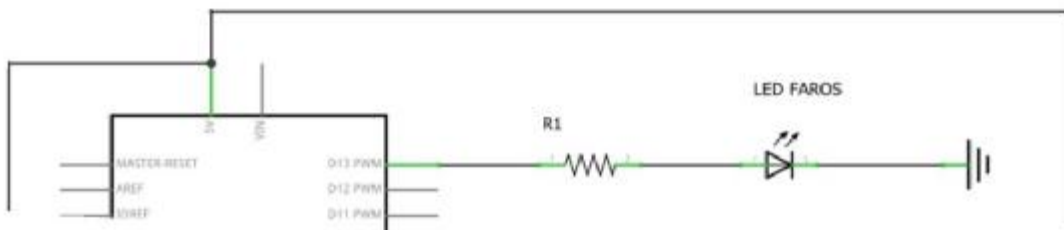
## Home Alarm System



Εικόνα 17: Συνδεσμολογία συναγερμού

### Ø pin13 φάρος

Στο pin13 συνδέεται μια αντίσταση 220Ω και σε σειρά η άνοδος του led(φάρος),η κάθοδος συνδέετε με το gnd.



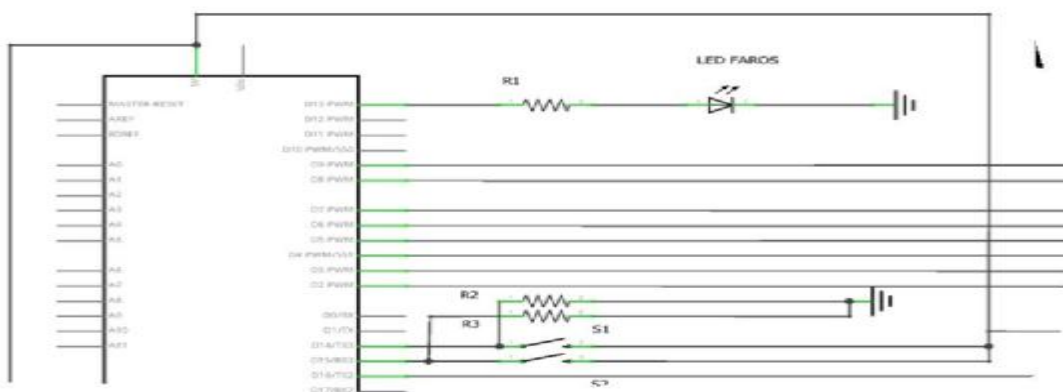
Εικόνα 18: Συνδεσμολογία φάρου

### Ø pin14 switch

Στην έξοδο του 14 συνδέεται μια αντίσταση 220Ω και ταυτόχρονα το ένα άκρο του διακόπτη (s1).Το άλλο μέρος της αντίστασης συνδέεται με το gnd ενώ το άλλο άκρο του διακόπτη συνδέεται με τα 5volt.

### Ø pin15 switch

Στην έξοδο του 15 συνδέεται μια αντίσταση 220Ω και ταυτόχρονα το ένα άκρο του διακόπτη (s2).Το άλλο μέρος της αντίστασης συνδέεται με το gnd ενώ το άλλο άκρο του διακόπτη συνδέεται με τα 5volt.



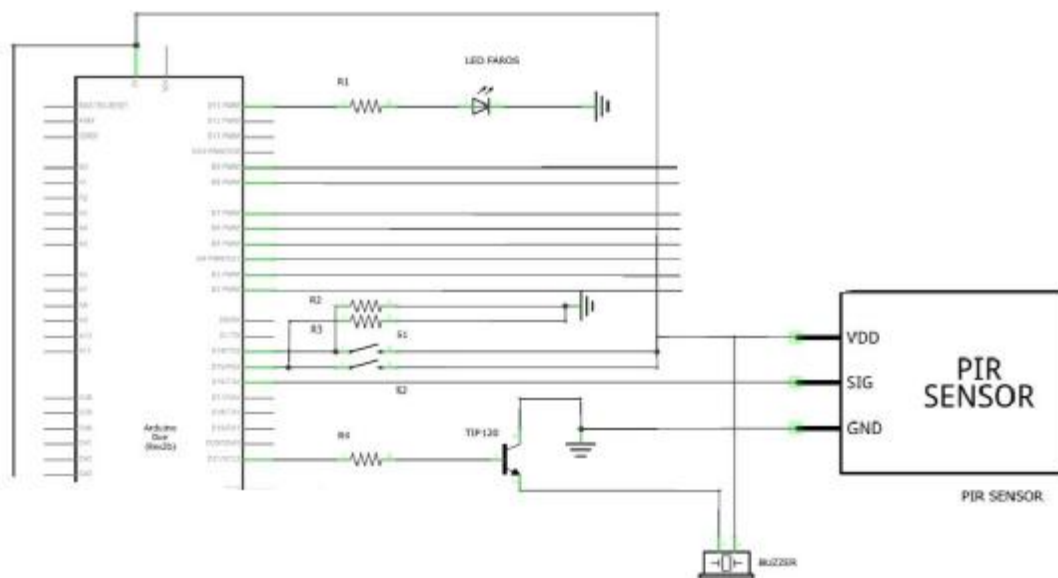
Εικόνα 19: Συνδεσμολογία διακόπτη πόρτας και παραθύρου.

Ø pin16 pir

Το pin16 συνδέεται απευθείας στο SIG του pir αισθητήρα μας. Τα VDD και GND του pir αισθητήρα συνδέονται στα +5V και gnd της πλακέτας αντίστοιχα.

Ø pin21 tip120

Το tip120 το χρησιμοποιούμε για την οδήγηση του buzzer. Το pin21 συνδέεται με μια αντίσταση 1 KΩ ,το άλλο άκρο της αντίστασης συνδέεται στην βάση του τρανζίστορ.



Εικόνα 20: Συνδεσμολογία pir sensor και buzzer.

Ø pin24 disarm

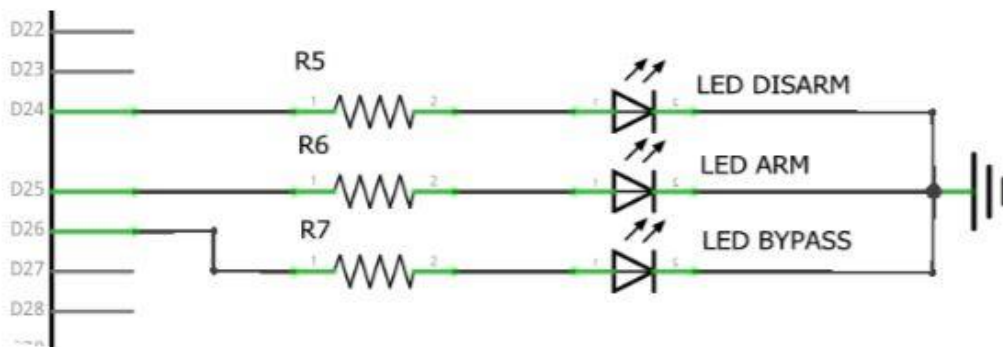
Στο pin24 συνδέεται μια αντίσταση 220Ω και σε σειρά η άνοδος του led(),η κάθοδος συνδέετε με το gnd.

Ø pin25 ledarm

Στο pin25 συνδέεται μια αντίσταση 220Ω και σε σειρά η άνοδος του led(),η κάθοδος συνδέετε με το gnd.

### Ø pin26 bypass

Στο pin26 συνδέεται μια αντίσταση 220Ω και σε σειρά η άνοδος του led(),η κάθοδος συνδέετε με το gnd.



Εικόνα 21: Συνδεσμολογία led(arm,disarm,bypass).

### Ø pin2-9 keypad

Το πληκτρολόγιο συνδέεται ως εξής:

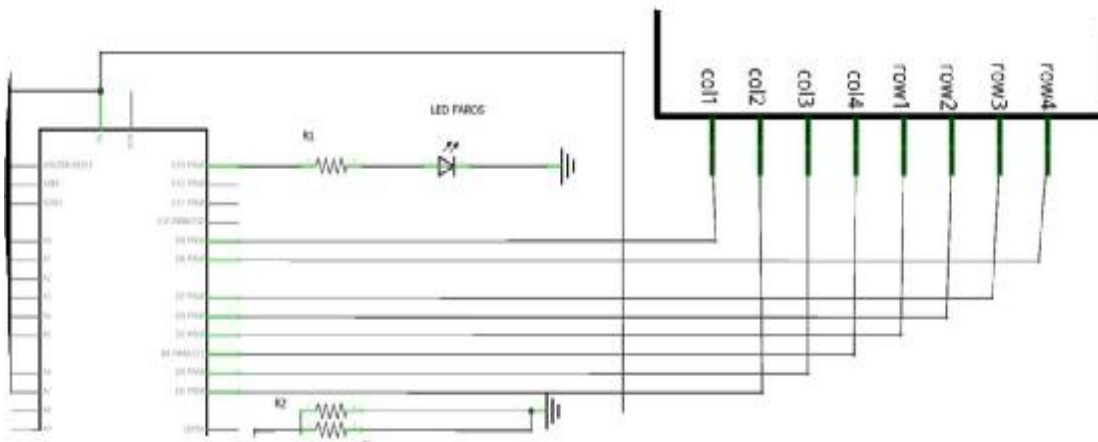
pin2-> col2    pin3-> col3

pin4-> col4    pin5-> row1

pin6-> row2    pin7-> row3

pin8-> row4    pin9-> col1





Εικόνα 22: Συνδεσμολογία πληκτρολογίου.

### Ø pin48-53 lcd

Lcd1: Συνδέετε με τη γείωση και το ένα άκρο του ποτενσιόμετρου.

Lcd2: Συνδέετε με τα 5volts και το άλλο άκρο του ποτενσιόμετρου.

Lcd3: Συνδέετε με τη μεσαία λήψη του ποτενσιόμετρου.

Lcd4: Συνδέετε με το pin53

Lcd5: Συνδέετε με το gnd

Lcd6: Συνδέετε με το pin52

Lcd7: -

Lcd8: -

Lcd9: -

Lcd10: -

Lcd11: Συνδέετε με το pin51

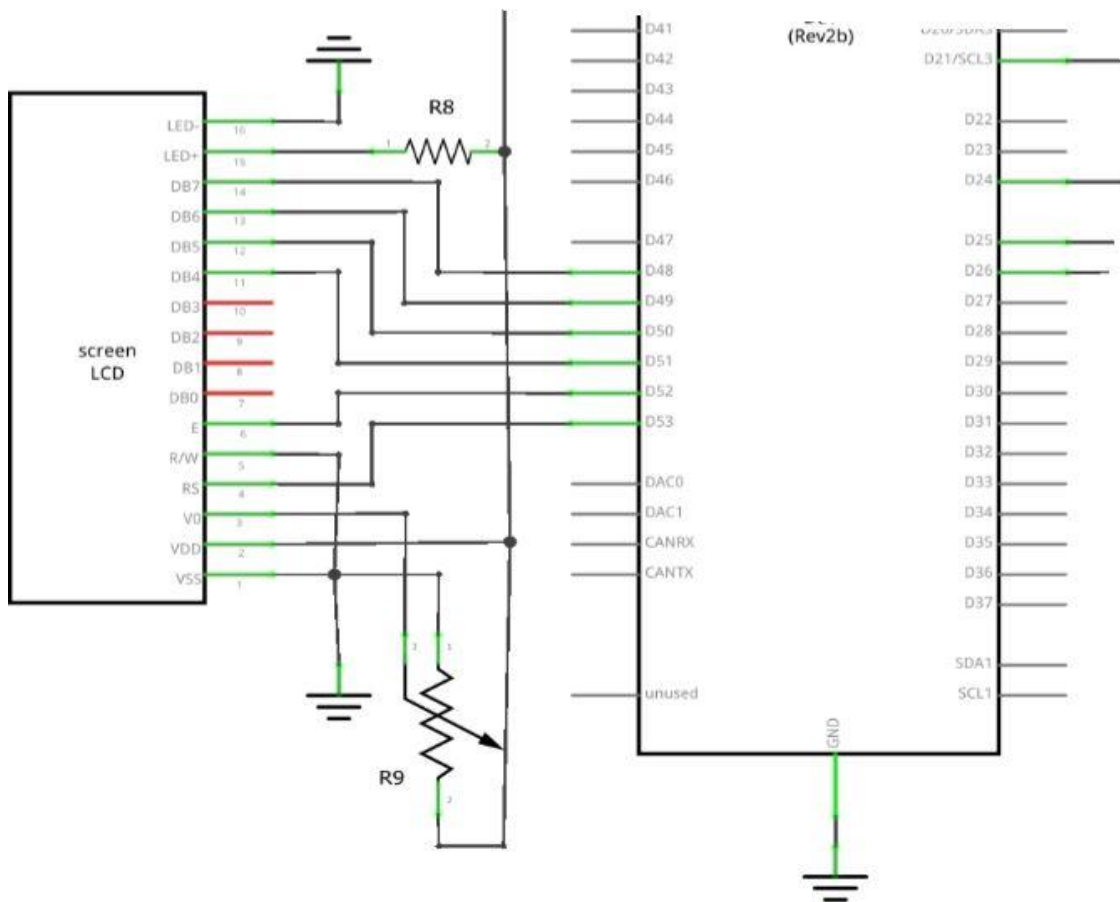
Lcd12: Συνδέετε με το pin50

Lcd13: Συνδέετε με το pin49

Lcd14: Συνδέετε με το pin48

Lcd15: Συνδέετε με το ένα άκρο μιας αντίστασης 220Ω και το άλλο άκρο της αντίστασης με τα +5V

Lcd16: Συνδέετε με το gnd



Εικόνα 23: Συνδεσμολογία Lcd οθόνης.

## 4.2. Κώδικας

Όταν τελειώσουμε με την συνδεσμολογία είμαστε έτοιμοι να πάμε να γράψουμε τον κώδικα στο πρόγραμμα, το οποίο θα περάσουμε στο Arduino Due. Θα δούμε μερικά βασικά μέρη του κώδικα, για να κατανοήσουμε την λειτουργία του. Αρχικά πρέπει να καλέσουμε της βιβλιοθήκες που θα χρησιμοποιήσουμε, οι οποίες είναι:

**Με το #include και δίπλα την αντίστοιχη βιβλιοθήκη, την καλούμε στο πρόγραμμα μας.**

1. #include<Password.h>  
`//http://playground.arduino.cc/uploads/Code/Password.zip //tells to use password library`
2. #include<Keypad.h>  
`//http://www.arduino.cc/playground/uploads/Code/Keypad.zip //tells to use keypad library`
3. #include <LiquidCrystal.h>

Προετοιμάζει τη βιβλιοθήκη με τους αριθμούς από τις ακίδες διασύνδεσης που θα χρησιμοποιήσει η lcd οθόνη

4. `LiquidCrystal lcd(53, 52, 51, 50, 49, 48);`

Ορίζει τον κωδικό πρόσβασης ως 0000

5. `Password password = Password("0000"); //password to unlock, can be changed`

Ορίζουμε τις γραμμές και τις στήλες του πληκτρολογίου, στην περίπτωση μας θα έχουμε 4 γραμμές και 4 στήλες

6. `const byte ROWS = 4; // Four rows`

```
7. const byte COLS = 4; // Four columns
```

**Ορίζουμε τον χάρτη του πληκτρολόγιου μας**

```
8. char keys[ROWS][COLS] = {
9.   {
10.  '1','2','3','A'  }
11.  ,
12.  {
13.  '4','5','6','B'  }
14.  ,
15.  {
16.  '7','8','9','C'  }
17.  ,
18.  {
19.  '*', '0', '#', 'D'  }
20. };
```

**Ορίζουμε που θα συνδεθούν στην πλακέτα μας οι γραμμές και οι στήλες του πληκτρολογίου μας**

```
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
```

```
21. byte rowPins[ROWS] = {8, 7, 6, 5};
```

```
// Connect keypad COL0, COL1, COL2 and COL3 to these Arduino pins.
```

```
22. byte colPins[COLS] = {4, 3, 2, 9};
```

### Δημιουργούμε το Keypad

```
23. Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

### Αρχικοποίηση και σύνδεση pin

```
24.int const ledPin = 13; // LED connected to digital pin 13
25.int const magnetw = 14; // WINDOWS connected to digital pin 14
26.int const magnetd = 15; // DOOR connected to digital pin 15
27.int const pir = 16; // PIR connected to digital pin 16
28.int const pzSpeaker = 21; // SPEAKER connected to digital pin 21
29.int disarmlamp =24; // disarmlamp connected to digital pin 24
30.int armlamp =25; // armlamp connected to digital pin 25
31.int bypasslamp =26; // bypasslamp connected to digital pin 26
```

### Μεταβλητές που θα αλλάζουν τιμή στο πρόγραμμα

```
32.int Counter = 0; // counter for pir alarm
33.int Counter1 = 0; // counter for windows alarm
34.int Counter2 = 0; // counter for door alarm
35.int var =0; //counter for door entrance
36.int var1 =0; //counter for the activation alarm
37.int diakoptis=1;
38.int diakoptis2=1;
39.int k = 0; // variable to store the read value
40.int l = 0; // variable to store the read value
41.int m = 0; // variable to store the read value
42.int c=0; // variable to store the read value
43.int e=0; // variable to store the read value
44.int d=0; // variable to store the read value
45.void setup(){
```

```
46.Serial.begin(9600); // serial communication setup at 9600
```

#### Ορίζει τις στήλες και τις σειρές της οθόνης μας

```
47.lcd.begin(16, 2);
```

#### Είσοδοι

```
48.pinMode(pir, INPUT); // sets the digital pin as input
```

```
49.pinMode(magnetd, INPUT); // sets the digital pin as input
```

```
50.pinMode(magnetw, INPUT); // sets the digital pin as input
```

#### Έξοδοι

```
51.pinMode(armlamp, OUTPUT); // sets the digital pin as output
```

```
52.pinMode(bypasslamp, OUTPUT); // sets the digital pin as output
```

```
53.pinMode(disarmlamp, OUTPUT); // sets the digital pin as output
```

```
54.pinMode(ledPin, OUTPUT); // sets the digital pin as output
```

```
55.pinMode(pzSpeaker, OUTPUT); // sets the digital pin as output
```

```
56.delay(200); // waits for 200 ms
```

```
57.keypad.addEventListener(keypadEvent); //add an event listener
```

```
58.delay(200); // waits for 200 ms
```

```
59.}
```

```
60.void loop(){
```

#### Διαβάζει το πάτημα του πληκτρολογίου

```
61.keypad.getKey();
```

### Διαβάζει την τιμή εισόδου

```
62.int k = digitalRead(magnetw); // read the input pin
63.int l = digitalRead(pir); // read the input pin
64.int m = digitalRead(magnetd); // read the input pin
```

### Αν ο διακοπτής είναι LOW ενεργοποιείται η armlamp

```
65.if( diakoptis == LOW){ // do stuff if the condition is true
66.digitalWrite(disarmlamp,LOW); //sets the disarmlamp off
67.digitalWrite(bypasslamp,LOW); //sets the bypasslamp off
68.digitalWrite(armlamp,HIGH); //sets the armlamp on
```

### Αν το l ή το c γίνει HIGH ενεργοποιείτε η σειρά για την μέτρηση που έχει το counter

//PIR

```
69.if (l == HIGH || c==HIGH ){
70.if (Counter < 201 ){ // do stuff if the condition is true
71.digitalWrite(pzSpeaker,HIGH); // sets the pzSpeaker on
72.c= 1; //sets c= 1
73.Counter++; //add counter +1 for every loop
74.delay(25);
```

### Μετά από 201 μετρήματα απενεργοποιείτε η σειρά

// turns on the LED every four button pushes by

// checking the modulo of the button push counter.

// the modulo function gives you the remainder of

// the division of two numbers:

```
75. if(Counter % 201== 0){ // do stuff if the condition is true
76.digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off
77.}
```

**Ανά τέσσερις μετρήσεις ενεργοποιείτε το ledPin και έπειτα κλείνει**

```
78.if (Counter % 4== 0) { //every 4 number of counter
79.digitalWrite(ledPin, HIGH); // sets the LED on
80.}
81.else {
82.digitalWrite(ledPin, LOW); //sets the LED off
83.}
84.}
85.}
```

//PARATHIRA

**Αν το k ή το e γίνει HIGH ενεργοποιείτε η σειρήνα για την μέτρηση που έχει το counter**

```
86.if (k == HIGH || e==HIGH){
87.if (Counter1 < 201){
88.digitalWrite(pzSpeaker,HIGH); // sets the pzSpeaker on
89.e=1; //sets e= 1
90.Counter1++; //add counter1 +1 for every loop
91.delay(25);
```



**Μετά από 201 μετρήματα απενεργοποιείτε η σειράνα**

```
// turns on the LED every four button pushes by  
// checking the modulo of the button push counter.  
// the modulo function gives you the remainder of  
// the division of two numbers:
```

```
92.if(Counter1 % 201== 0){  
93.digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off  
94.}
```

**Ανά τέσσερις μετρήσεις ενεργοποιείτε το ledPin και έπειτα κλείνει**

```
95.if (Counter1 % 4== 0) {  
96.digitalWrite(ledPin, HIGH); // sets the LED on  
97.}  
98.else {  
99.digitalWrite(ledPin, LOW); //sets the LED off  
100.  }  
101.  }  
102.  }
```

//PORTA

**Αν το m ή το d γίνει HIGH ενεργοποιείτε η μέτρηση χρόνου εισόδου**

```
103.    if (m== HIGH || d==HIGH){ // do stuff if the condition is true  
104.    d=1; // sets d=1  
105.    if (var <1000){
```

```
106.    var++; //add var +1 for every loop
107.    delay(5);

//Serial.println(var);

108.    }
109.    else if (var>=1000 ){
```

**Ενεργοποιείτε η σειρήνα για την μέτρηση που έχει το counter**

```
110.    if (Counter2 < 201){
111.        digitalWrite(pzSpeaker,HIGH); // sets the pzSpeaker on
112.        Counter2++; //add counter2 +1 for every loop
113.        delay(25);
```

**Μετά από 201 μετρήματα απενεργοποιείτε η σειρήνα**

// turns on the LED every four button pushes by

// checking the modulo of the button push counter.

// the modulo function gives you the remainder of

// the division of two numbers:

```
114.    if(Counter2 % 201== 0){
115.        digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off
116.    }
```

**Ανά τέσσερις μετρήσεις ενεργοποιείτε το ledPin και έπειτα κλείνει**

```
117.    if (Counter2 % 4== 0) {
118.        digitalWrite(ledPin, HIGH); // sets the LED on
```

```
119.    }
120.    else {
121.    digitalWrite(ledPin, LOW); //sets the LED off
122.    }
123.    }
124.    }
125.    }
126.    }
```

**Αν ο διακοπτις2 είναι low ενεργοποιείτε η bypasslamp**

```
127.    else if(diakoptis2 ==LOW){ // do stuff if the condition is true
128.    digitalWrite(disarmlamp,LOW); //sets the disarmlamp off
129.    digitalWrite(armlamp,HIGH); //sets the armlamp on
130.    digitalWrite(bypasslamp,HIGH); //sets the bypasslamp on
```

//PARATHIRA

**Αν το κ ή το ε γίνει HIGH ενεργοποιείτε η σειρήνα για την μέτρηση που έχει το counter**

```
131.    if (k == HIGH || e==HIGH){ // do stuff if the condition is true
132.    if (Counter1 < 201){
133.    digitalWrite(pzSpeaker,HIGH); // sets the pzSpeaker on
134.    e=1; //sets e=1
135.    Counter1++//add counter1 +1 for every loop
136.    delay(25);
```

**Μετά από 201 μετρήματα απενεργοποιείτε η σειράνα**

```
// turns on the LED every four button pushes by
```

```
// checking the modulo of the button push counter.
```

```
// the modulo function gives you the remainder of
```

```
// the division of two numbers:
```

```
137.     if(Counter1 % 201== 0){
138.     digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off
139.     }
```

**Ανά τέσσερις μετρήσεις ενεργοποιείτε το ledPin και έπειτα κλείνει**

```
140.     if (Counter1 % 4== 0) {
141.     digitalWrite(ledPin, HIGH); // sets the LED on
142.     }
143.     else
144.     digitalWrite(ledPin, LOW); //sets the LED off
145.     }
146.     }
147.     }
```

```
//PORTA
```

**Αν το m ή το d γίνει HIGH ενεργοποιείτε η μέτρηση χρόνου εισόδου**

```
148.     if (m== HIGH || d==HIGH){ // do stuff if the condition is true
149.     d=1; //sets d=1
150.     if (var <1000){
```

```
151.    var++; //add var +1 for every loop
152.    delay(5);
153.    }
154.    else if (var>=1000 ){
```

**Ενεργοποιείτε η σειρήνα για την μέτρηση που έχει το counter**

```
155.    if (Counter2 < 201){
156.    digitalWrite(pzSpeaker,HIGH); // sets the pzSpeaker on
157.    Counter2++; //add counter2 +1 for every loop
158.    delay(25);
```

**Μετά από 201 μετρήματα απενεργοποιείτε η σειρήνα**

```
// turns on the LED every four button pushes by
// checking the modulo of the button push counter.
// the modulo function gives you the remainder of
// the division of two numbers:
```

```
159.    if(Counter2 % 201== 0){
160.    digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off
161.    }
```

**Ανά τέσσερις μετρήσεις ενεργοποιείτε το ledPin και έπειτα κλείνει**

```
162.    if (Counter2 % 4== 0) {
163.    digitalWrite(ledPin, HIGH); // sets the LED on
```

```
164.    }
165.    else {
166.    digitalWrite(ledPin, LOW); //sets the LED off
167.    }
168.    }
169.    }
170.    }
171.    }
```

**Αν ο διακοπτής γίνει HIGH τότε ενεργοποιείτε η disarmlamp και ο συναγερμός μας είναι απενεργοποιημένος**

```
172.    else if ( diaktotis==HIGH){ // do stuff if the condition is true
173.    digitalWrite(disarmlamp,HIGH); //sets the disarmlamp on
174.    digitalWrite(armlamp,LOW); //sets the armlamp off
175.    digitalWrite(bypasslamp,LOW); //sets the bypasslamp off
176.    digitalWrite(ledPin, LOW); //sets the LED off
177.    digitalWrite(pzSpeaker,LOW); // sets the pzSpeaker off
178.    Counter=0; //sets counter=0
179.    Counter1=0; //sets counter1=0
180.    Counter2=0; //sets counter2=0
181.    var =0; //sets var=0
182.    c=0; //sets c=0
183.    e=0; //sets e=0
184.    d=0; //sets d=0

//    Serial.println(Counter);

185.    }
186.    }
```

### Περιπτώσεις πληκτρολογίου

```
187. void keypadEvent(KeypadEvent eKey){
188.   switch (keypad.getState()){
189.     case PRESSED:
190.       lcd.print("*");
191.       Serial.print("Enter:");
192.       Serial.println(eKey);
193.       delay(10);
194.       switch (eKey){
195.         case '*': lcd.clear();checkPassword(); delay(1);break;
196.         case '#': lcd.clear(); password.reset();delay(1); break;
197.         case 'D':lcd.clear();bypass();delay(1); break;
198.         default: password.append(eKey); delay(1);
199.       }
200.     }
201.   }
```

### Κωδικός πρόσβασης

#### Περίπτωση κανονικής ενεργοποίησης

```
202. void checkPassword(){
203.   if (password.evaluate()){ //if password is right open
204.     Serial.println("disarm");

//Add code to run if it works

205.     password.reset();
206.     diakoptis=1; //sets diakoptis=1
207.     var1=0; //sets var1=0
208.   }
209.   else{
```

```
210.    if (var1 <999){
211.        for(var1=0; var1<1000;){
212.            var1++; //add var1 +1 for every loop
213.            delay(5);

// Serial.println(var1);

214.        }
215.    }
216.    if (var1=1000){
217.        Serial.println("arm"); //if passwords wrong keep lock
218.        password.reset();
219.        diakoptis=0; //sets diakoptis=0
220.        diakoptis2=1; //sets diakoptis2=1
221.    }
222. }
223. }
```

### Περίπτωση Bypass ενεργοποίησης

```
224.    void bypass(){
225.        if (password.evaluate()){ //if password is right open
226.            Serial.println("disarm");

//Add code to run if it works

227.        password.reset();
228.        diakoptis2=1; //sets diakoptis2=1
229.        var1=0; //sets var=0
230.    }
231.    else{
232.        if (var1 <999){
233.            for(var1=0; var1<1000;){
```



## Home Alarm System

```
234.    var1++; //add var1 +1 for every loop
235.    delay(5);
236.    }
237.    }
238.    if (var1=1000){
239.        Serial.println("arm"); //if passwords wrong keep locke
240.        password.reset();
241.        diakoptis2=0; //sets diakoptis2=0
242.        diakoptis=1; //sets diakoptis=1
243.    }
244.    }
245.    }
```

Βιβλιογραφία:

[www.arduino.cc](http://www.arduino.cc)

[www.epalsite.com](http://www.epalsite.com)

[www.wikipedia.org](http://www.wikipedia.org)

[www.fritzing.org](http://www.fritzing.org)

[www.open-electronics.org](http://www.open-electronics.org)

[www.learn.adafruit.com/](http://www.learn.adafruit.com/)

[www.instructables.com](http://www.instructables.com)