

**ΜΕΛΕΤΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΣΥΛΛΟΓΗΣ
ΚΑΙ ΚΑΤΑΓΡΑΦΗΣ ΔΕΔΟΜΕΝΩΝ ΜΕ Arduino
(Development of an Arduino-based data acquisition
system)**

Πτυχιακή Εργασία των

ΓΕΩΡΓΙΑΔΗΣ ΑΝΔΡΕΑΣ (Α.Ε.Μ 1709)

ΚΛΕΦΤΟΓΙΩΡΓΗΣ ΡΟΥΣΣΗΣ ΧΡΗΣΤΟΣ (Α.Ε.Μ. 1990)

Επιβλέπων: Μαδεμλής Ιωάννης,
Εργαστηριακός Συνεργάτης

ΣΕΡΡΕΣ, 2015

ΠΕΡΗΛΗΨΗ

Η παρούσα πτυχιακή εργασία πραγματεύεται τη μελέτη της αναπτυξιακής υπολογιστικής πλατφόρμας ARDUINO. Η πλατφόρμα ARDUINO στηρίζεται σε μικροελεγκτή Atmel και αποτελεί ένα ολοκληρωμένο αναπτυξιακό σύστημα ανοιχτού λογισμικού και ευέλικτης προσαρμογής υλικού (open-source hardware), προσφέροντας δυνατότητες υλοποίησης εφαρμογών σε πολλούς τομείς της ηλεκτρονικής. Στο πλαίσιο της εργασίας θα μελετήσουμε την αρχιτεκτονική και τον προγραμματισμό του Arduino και θα σχεδιάσουμε κ θα υλοποιήσουμε το υλικό και λογισμικό για την ανάπτυξη ενός συστήματος που θα καταγράφει θερμοκρασία, ατμοσφαιρική πίεση και φωτεινότητα περιβάλλοντος.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	0
ΠΕΡΙΕΧΟΜΕΝΑ	1
ΕΙΣΑΓΩΓΗ	2
ΙΣΤΟΡΙΑ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΗ Arduino	5
ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ Arduino UNO Rev.3	6
ΑΚΡΟΔΕΚΤΕΣ ΡΕΥΜΑΤΟΣ ΑΝΑΛΟΓΙΚΗ ΚΑΙ ΨΗΦΙΑΚΟΙ ΑΚΡΟΔΕΚΤΕΣ	8
ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ Arduino	9
ΑΛΛΑ ΜΟΝΤΕΛΑ ΤΟΥ Arduino	11
ΚΕΦΑΛΑΙΟ 2 :ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ	12
2.10 ΟΛΟΚΛΗΡΩΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΤΟΥ ARDUINO	25
2.11 ΣΕΙΡΙΑΚΗ ΟΘΟΝΗ (SERIAL MONITOR)	26
2.12 Η ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	27
2.13 ΒΑΣΙΚΕΣ ΔΟΜΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	28
2.14 ΨΗΦΙΑΚΕΣ ΑΚΙΔΕΣ (DIGITAL PINS)	32
2.15 ΑΝΑΛΟΓΙΚΕΣ ΑΚΙΔΕΣ ΕΙΣΟΔΟΥ (ANALOG INPUT PINS)	33
2.16 ΥΠΟΣΤΗΡΙΞΗ ΒΙΒΛΙΟΘΗΚΩΝ (LIBRARIES)	34
ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ	38
ΚΕΦΑΛΑΙΟ 3: ΠΕΙΡΑΜΑ	39
ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ	41
ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	42
ΒΙΒΛΙΟΓΡΑΦΙΑ	45

ΕΙΣΑΓΩΓΗ

Ο μικροελεγκτής (microcontroller) είναι ένας τύπος επεξεργαστή, μια παραλλαγή ουσιαστικά του μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Ποιο συγκεκριμένα, Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα

Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). 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 προσυναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Τα παραπάνω δεν ακούγονται πρωτότυπα. Υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ποια είναι η ειδοποιός διαφορά; Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί να κατασκευαστεί από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές ακόμα και για εμπορικούς σκοπούς και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα που χρησιμοποιεί το Arduino σε κατασκευές άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας. Γενικά, τα Projects στον εν λόγω Μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP). Το Arduino χρησιμοποιεί τώρα ένα ειδικά προγραμματιζόμενο Atmega382 αντί του chip FTDI ώστε αυτό να επιτρέπει τόσο την πιο γρήγορη ταχύτητα μεταφοράς όσο και τη γρήγορη σειριακή επικοινωνία.

Ο μικροεπεξεργαστής ενός Arduino συνήθως προγραμματίζεται εκ των προτέρων ώστε να παρέχει κάποιο φορτωτή εκκίνησης (BootLoader). Ο φορτωτής

εκκίνησης υπάρχει ώστε να απλοποιεί την διαδικασία της αποθήκευσης των προγραμμάτων στην Flash Memory του Arduino μέσω σειριακής USB θύρας.

Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι ανεκτίμητη γνώση σε όλους.

Βασικά Πλεονεκτήματα πλατφόρμας Arduino:

- Οικονομική: Η πλατφόρμα Arduino αποτελεί οικονομική λύση διότι είναι φθηνότερη. Επιπλέον, είναι αρχιτεκτονικά ανοιχτή και μπορεί ο οποιοσδήποτε να την αναπτύξει από μόνος του.
- Μεταφέρσιμη: Σε σχέση με τις υπάρχουσες πλατφόρμες στο εμπόριο η πλατφόρμα Arduino παρέχει πλήρη μεταφερσιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.
- Επεκτάσιμη: Το υλικό και το λογισμικό της πλατφόρμας Arduino είναι ανοιχτά και ελεύθερα για όλους. Καθημερινά, χιλιάδες υποστηρικτές του ελεύθερου λογισμικού αναπτύσσουν διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Παράλληλα, τόσο η αρχιτεκτονική όσο και το υλικό της πλατφόρμας εξελίσσονται συνεχώς.

Παρακάτω ακολουθούν μερικές από τις πλατφόρμες Arduino που έχουν αναπτυχθεί και όπου η κάθε μία είτε αποτελεί εξέλιξη κάποιας άλλης, είτε έχει αναπτυχθεί για κάποιο συγκεκριμένο σκοπό :

Arduino Uno	▪ Arduino Stamp
Arduino Diecimila	▪ Arduino Fio
Arduino Duemilanove	▪ Arduino NG
Arduino Mega1280	▪ Arduino NG+

- Arduino Mega2560
- Extreme
- Arduino Nano
- Arduino USB
- Arduino Arduino Mini
- Arduino Bluetooth
- LilyPad
- Arduino
- Serial Arduino

Ακολουθεί ένας πίνακας όπου περιέχει για τις πιο τυπικές πλατφόρμες Arduino τα βασικά χαρακτηριστικά όσο αφορά το υλικό τους μέρος.

Πλατφόρμα Arduino	Μικροελεγκτής Atmel AVR	Flash KiB	EEPROM KiB	SRAM KiB	Ψηφιακές Επαφές E / E	PWM	Αναλογικές Επαφές Εισόδου
Diecimila	ATmega168	16	0.5	1	14	6	6
Duemilanove	ATmega168/328	16	0.5	1	14	6	6
Uno	ATmega328	32	1	2	14	6	6
Mega	ATmega1280	128	4	8	54	14	16
Fio	ATmega328P	32	1	2	14	6	8
Mega 2560	ATmega2560	256	4	8	54	14	16

ΙΣΤΟΡΙΑ ΤΟΥ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ “Arduino”

Το 2005, στην Ivrea της Ιταλίας (η ιστοσελίδα της εταιρείας υπολογιστών Olivetti), ένα έργο άρχισε να δημιουργείται, μια συσκευή για τον έλεγχο σχεδίων, χτισμένο από μαθητές με λιγότερα έξοδα από ό, τι με άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνη τη στιγμή. Από τον Μάιο του 2011, περισσότερα από 300.000 Arduino ήταν «στην άγρια φύση». Εφευρέτες Massimo Banzi και David Cuartielles, ονόμασαν το έργο τους Arduin of Ivrea. Το «Arduino» είναι επίσης ένα ιταλικό όνομα, που σημαίνει «γενναίος φίλος».

Το έργο Arduino είναι μια παραγόμενη έκδοση της πλατφόρμας ανοικτού κώδικα Wiring Platform. Ο Κολομβιανός καλλιτέχνης και προγραμματιστής Hernando Barragan δημιούργησε συρμάτωση «Wiring» ως μια Μεταπτυχιακή διπλωματική εργασία στο Interaction Design Institute Ivrea υπό την εποπτεία του Massimo Banzi και του Casey Reas. Η Καλωδίωση «Wiring» βασίστηκε στην επεξεργασία και το ολοκληρωμένο περιβάλλον ανάπτυξης που είχε δημιουργηθεί από τον Casey Reas και τον Ben Fry.

Το Arduino χτίστηκε γύρω από το έργο της καλωδίωσης «Wiring» του Hernando Barragan. Η Καλωδίωση ήταν η διπλωματική εργασία του Hernando στο Interaction Design Institute Ivrea. Επρόκειτο να είναι μια ηλεκτρονική έκδοση της επεξεργασίας που χρησιμοποιήθηκε σε προγραμματιστικό περιβάλλον και ήταν η βάση για την σύνταξη επεξεργασίας. Αυτό ήταν υπό την εποπτεία του Hernando και του Massimo Banzi, ιδρυτές ενός Arduino. Ήταν κομμάτι της φαντασίας ότι θα υπήρχαν Arduino χωρίς καλωδιώσεις και ότι δεν θα υπάρχει καλωδίωση χωρίς επεξεργασία. Η επεξεργασία σίγουρα δεν θα υπήρχε χωρίς τη γλώσσα προγραμματισμού Design By Numbers και τον John Maeda.

Μια πλακέτα που αποτελείται από έναν 8-bit μικροελεγκτής Atmel AVR με συμπληρωματικά στοιχεία για τη διευκόλυνση του προγραμματισμού και της ενσωμάτωσης σε άλλα κυκλώματα. Μια σημαντική πτυχή του Arduino είναι ο τυπικός τρόπος ότι σύνδεσμοι εκτίθενται, επιτρέποντας τη CPU να συνδεθεί με μια ποικιλία εναλλακτικές πρόσθετες λειτουργικές μονάδες, γνωστές ως ασπίδες “shield”. Ορισμένες ασπίδες “shield” επικοινωνούν με την πλακέτα του Arduino άμεσα μέσω διαφόρων ακίδων, αλλά πολλές ασπίδες είναι μεμονωμένα καθορισμένες μέσω ενός σειριακού διαύλου I²C, επιτρέποντας πολλές ασπίδες, να στοιβάζονται και να χρησιμοποιούνται παράλληλα. Το επίσημο Arduino χρησιμοποίησε τη σειρά megaAVR και ειδικά τους ATmega8, ATmega168, ATmega328, ATmega1280 και ATmega2560. Πολλοί από τους άλλους επεξεργαστές έχουν χρησιμοποιηθεί από συμβατά Arduino. Οι περισσότερες πλακέτες περιλαμβάνουν μια γραμμική ρύθμιση 5 volt και μια 16 MHz crystal oscillato (ή κεραμικό συντονιστή σε μερικές παραλλαγές), αν και μερικά σχέδια

όπως το LilyPad εκτελείται στα 8 MHz και βασίζεται από τους ρυθμιστές τάσης πλακέτας λόγω ειδικής φόρμας-παράγοντα με περιορισμούς. Ο Arduino μικροελεγκτής είναι επίσης προ-προγραμματισμένος με έναν φορτωτή εκκίνησης που απλοποιεί την αποστολή των προγραμμάτων στη on-chip flash memory, σε σύγκριση με άλλες συσκευές που χρειάζεται συνήθως ένας εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, όταν χρησιμοποιείτε λογισμικό στοίβας Arduino, όλες οι σανίδες “shield” προγραμματίζονται μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος που αυτό υλοποιείται διαφέρει από την έκδοση υλικού. Σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για μετατροπή μεταξύ επιπέδων RS232 και TTL-επίπεδου σημάτων. Αυτές οι πλακέτες Arduino προγραμματίζονται μέσω USB και υλοποιείτε χρησιμοποιώντας τις μάρκες USB-σε-σειριακό προσαρμογέα όπως η FTDI FT232. Μερικές παραλλαγές, όπως το Arduino Mini και η ανεπίσημη Boarduino, χρησιμοποιούν ένα αποσπώμενο προσαρμογέα USB-σεσειριακή ή καλώδιο, Bluetooth ή άλλες μεθόδους.

Συγκεκριμένα για την υλοποίηση αυτής της πτυχιακής εργασίας χρησιμοποιήσαμε το Arduino Uno R3 λόγω του χαμηλού κόστους και της άμεσης διαθεσιμότητας, διότι όσον αφορά την σύνδεση του μικροεπεξεργαστή θα χρησιμοποιούσαμε περίπου την ίδια διαδικασία.

▪ **ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ “ARDUINO UNO R3”**

Ο Arduino είναι ένα εργαλείο που βοηθάει, θα λέγαμε, τους ηλεκτρονικούς υπολογιστές να έρθουν πιο κοντά στο φυσικό περιβάλλον ούτως ώστε να ελεγχθούν και να ληφθούν μετρήσεις φυσικών περιβαλλοντικών καταστάσεων. Είναι, με λίγα λόγια, ο μεσολαβητής μεταξύ ηλεκτρονικών υπολογιστών και περιβάλλοντος. Πρόκειται, ουσιαστικά, για μια φυσική υπολογιστική πλατφόρμα ανοιχτής πηγής, η οποία βασίζεται σε μια απλή πλακέτα μικροελεγκτή και σε ένα περιβάλλον ανάπτυξης σύμφωνα με το οποίο το λογισμικό γράφεται στην πλακέτα. Οι Arduino μπορούν να χρησιμοποιηθούν για τη δημιουργία αλληλεπιδραστικών αντικειμένων, αφού δέχονται δεδομένα από μια ποικιλία από εισόδους, όπως αισθητήρες ή διακόπτες, και ταυτόχρονα είναι σε θέση να ελέγξουν ένας μεγάλο εύρος από διόδους φωτός, από μηχανές ή άλλες φυσικές εξόδους. Οι εφαρμογές τους μπορούν να χρησιμοποιηθούν μεμονωμένα ή και με σύνδεση με Η/Υ μέσω του κατάλληλου λογισμικού. Η σύνδεση της πλακέτας μπορεί να γίνει με ευκολία από τον ίδιο το χρήστη ή και να είναι ήδη πραγματοποιημένη από την εταιρεία ώστε ο χρήστης να την αγοράσει έτοιμη, ενώ παράλληλα το IDE λογισμικό βρίσκεται διαθέσιμο δωρεάν στο διαδίκτυο ώστε ο χρήστης να μπορεί να το κατεβάσει χωρίς καμία καθυστέρηση ή χρηματική επιβάρυνση.

Κάθε πλατφόρμα οποιασδήποτε εταιρείας όμοια με τον Arduino χρησιμοποιείται για να συγκεντρωθούν όλες οι προγραμματιστικές λεπτομέρειες, που ενδεχομένως να προκαλούν μια σύγχυση στο χρήστη, σε ένα εύκολο και έτοιμο προς χρήση πακέτο. Ο Arduino, όμως, δεν απλοποιεί μόνο τη διαδικασία χρήσης μικροελεγκτή, αλλά ταυτόχρονα προσφέρει και κάποια επιπλέον πλεονεκτήματα για καθηγητές, μαθητές και αρχάριους σε σχέση με τις υπόλοιπες πλατφόρμες:

□Χαμηλό κόστος (Ακόμα και οι έτοιμες πλακέτες δεν κοστίζουν περισσότερο από 40.00 ευρώ)

□Εύρος πλατφορμών ανάπτυξης (Windows, MAC και Linux) □Απλό και ξεκάθαρο περιβάλλον προγραμματισμού.

□Εκτενές και ανοιχτής πηγής λογισμικό (Δυνατότητα επέκτασης της γλώσσας προγραμματισμού μέσω της χρήσης βιβλιοθηκών C/C++)

□Εκτενές και ανοιχτής πηγής συσκευές (Βασίζεται στον μικροελεγκτή ATMEGA8 και στον ATMEGA168 της εταιρείας Atmel)

Όσον αφορά την αρχιτεκτονική του το Arduino είναι μια πλατφόρμα βασισμένη στον ATMEGA328. Περιέχει 14 ψηφιακούς ακροδέκτες I/O εισόδου-εξόδου, 6 αναλογικές εισόδους και 6 ψηφιακές εξόδους έναν ταλαντωτή των 16 MHz, υποδοχή σύνδεσης με USB καλώδιο, υποδοχή σύνδεσης με ρεύμα, δυνατότητα εντός του κυκλώματος σειριακού προγραμματισμού-ICSP (In-Circuit Serial Programming) και ένα κουμπί επανεκκίνησης (reset) σε περίπτωση που βραχυκυκλώσει χωρίς την θέλησή μας . Εμπεριέχει όλα όσα χρειάζονται για την υποστήριξη ενός μικροελεγκτή. Το μόνο που απαιτείται είναι η σύνδεσή του με έναν ηλεκτρονικό υπολογιστή μέσω ενός καλωδίου USB ή τροφοδοσία του μέσω ενός μετασχηματιστή τάσης για τη μετατροπή μεταξύ εναλλασσόμενου και ασυνεχούς ρεύματος (adaptor AC-to-DC-Alternating Current to Direct Current) ή μιας μπαταρίας.

Οι πλατφόρμες Arduino διαθέτουν τρεις βασικές μνήμες:

- Flash memory (32 Kbytes) στην οποία τοποθετείται κάθε φορά το πρόγραμμα που πρόκειται να εκτελεστεί καθώς και ο φορτωτής εκκίνησης που διευκολύνει την διαδικασία του προγραμματισμού της πλατφόρμας.
- SRAM memory (στατική μνήμη τυχαίας προσπέλασης των 2 Kbytes) η οποία χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται.
- EEPROM memory (1 Kbytes) στην οποία αποθηκεύονται οι τιμές των μεταβλητών όταν η πλατφόρμα σβήσει(OFF). Χρησιμοποιείται για την αποθήκευση ρυθμίσεων και άλλων παραμέτρων ανάμεσα στα Reset του Arduino.

Πρέπει να προστεθεί, η μνήμη Flash και η μνήμη EEPROM είναι σταθερές (οι πληροφορίες παραμένουν μετά την απενεργοποίηση του ρεύματος). Η μνήμη SRAM είναι ασταθής και οι πληροφορίες χάνονται όταν εναλλάσσεται το ρεύμα.

Επειδή δεν υπάρχει πολύ διαθέσιμη SRAM, αν τελειώσει, το πρόγραμμα μπορεί να αποτύχει με απροσδόκητους τρόπους. Μπορεί να φαίνεται ότι φορτώνει με επιτυχία, αλλά δεν τρέχει, ή τρέχει παράξενα. Για να ελεγχθεί εάν αυτό συμβαίνει, μπορούν να μειωθούν τα σχόλια ή οι σειρές ή άλλες δομές δεδομένων στο sketch (χωρίς να αλλάξει ο κώδικας). Εάν λειτουργεί με επιτυχία στη συνέχεια, κατά πάσα πιθανότητα έχει εξαντληθεί η SRAM. Ένας τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα είναι αν

υπάρχουν πίνακες αναζήτησης ή άλλοι μεγάλοι πίνακες, τότε μπορεί να χρησιμοποιηθεί ο μικρότερος τύπος δεδομένων που είναι αναγκαίος για να αποθηκευτούν οι τιμές που χρειάζονται.

- **Ακροδέκτες ρεύματος**

Οι Ακροδέκτες ρεύματος αποτελούνται από (νούμερο 10 στην παρακάτω εικόνα)

3.3V: 3.3Volts έξοδο

5V: 5 Volts έξοδο

GND: 2 γειώσεις

VIN: Η τάση εισόδου στην πλακέτα Arduino όταν είναι με μια εξωτερική πηγή ενέργειας (σε αντιδιαστολή με 5 volt από τη σύνδεση USB ή άλλη ρυθμιζόμενη πηγή ενέργειας).

- **Αναλογικοί ακροδέκτες**

Αναλογικοί ακροδέκτες (νούμερο 11 στην παρακάτω εικόνα) είναι οι συντελεστές όπου θα συνδέσετε το αναλογικά συστατικά, όπως τα ποτενσιόμετρα και άλλους αισθητήρες. Αν και οι ψηφιακές εισοδοί / έξοδοι λειτουργούν μόνο με 0 και 1 τιμές, οι αναλογικές inputs λειτουργούν με τιμές 0 έως 1023.

- **Ψηφιακοί ακροδέκτες**

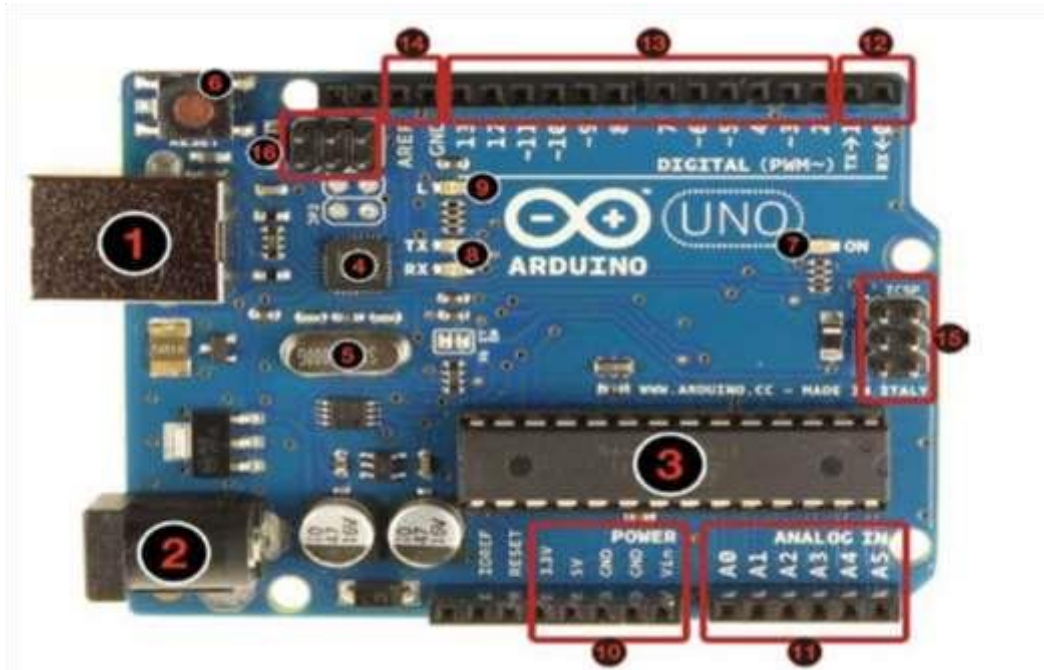
Ψηφιακοί ακροδέκτες (νούμερο 13 στην παρακάτω εικόνα) μπορεί να λειτουργήσουν ως εισροές ή εκροές και να ορίσετε πώς θα λειτουργήσουν με την pinMode λειτουργία ().

Οι πείροι που έχουν το "~" μπροστά από τα νούμερα είναι PWM (Pulse Width Modulation) έξοδοι, και μπορούν να μιμηθούν αναλογικής εξόδου με λειτουργία analogWrite (). Όταν χρησιμοποιείτε αυτή τη λειτουργία, διαμορφώνετε το πλάτος του παλμού, μεταβάλλοντας τη συχνότητα max και χαμηλών παλμών (0 ή 1), μιμούμενοι ένα αναλογικό παλμό.

▪ Συνοπτική περιγραφή **Arduino**

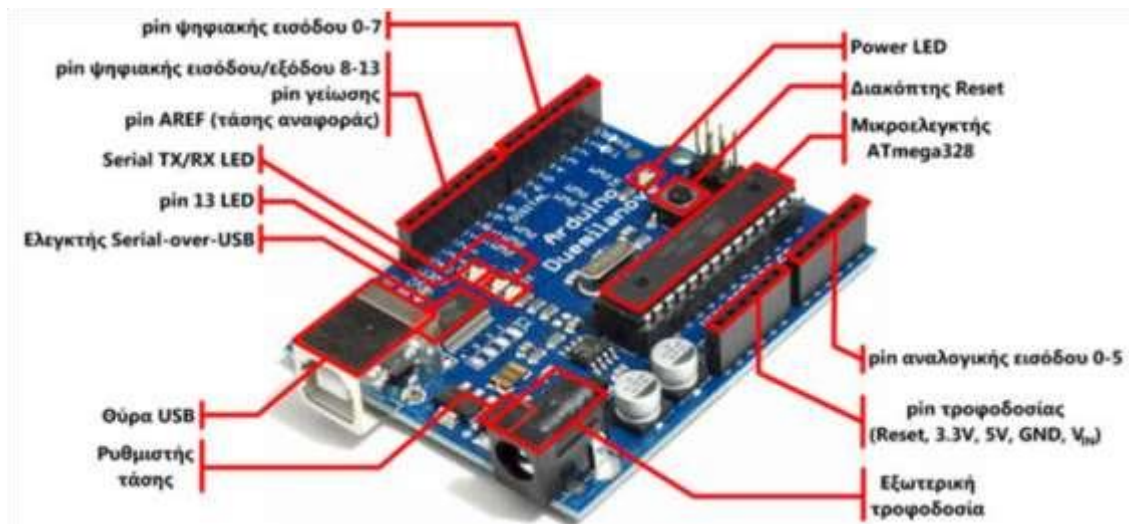
Το Arduino Uno R3 παρουσιάζει τα εξής χαρακτηριστικά:

- Μικροελεγκτής ATMEGA328
- Τάση λειτουργίας 5V
- Τάση εισόδου (συνίσταται) 7-12V
- Όρια τάσης εισόδου 6-20V
- Ψηφιακοί ακροδέκτες I/O 14(6 εκ των οποίων PWM έξοδο)
- Αναλογικοί ακροδέκτες εισόδου 6
- Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη 40mA
- Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V 50mA
- Μνήμη flash 32KB(ATMEGA328)
- Μνήμη SRAM 2KB(ATMEGA328)
- Μνήμη EEPROM 1KB(ATMEGA328)
- Ταχύτητα ρολογιού 16MHz



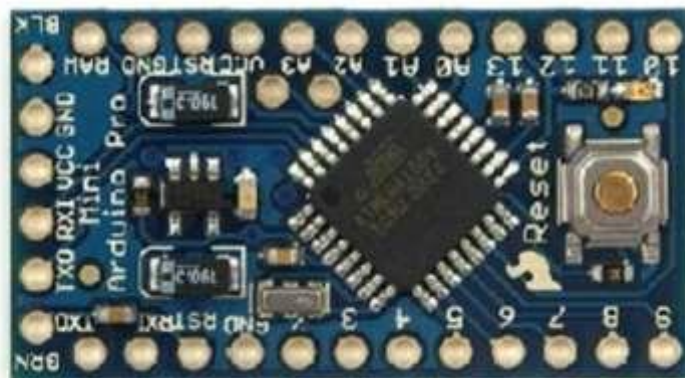
- 1- USB υποδοχή
- 2-υποδοχή τροφοδοσίας,
- 3-επεξεργαστή
- 4-chip επικοινωνίας
- 5-16 MHz κρυστάλλου
- 6-κουμπί επαναφοράς
- 7-Led ON
- 8-TX / NX Leds
- 9 -Led
- 10 -ακροδέκτες ρεύματος
- 11 -Αναλογικές Είσοδοι
- 12 -TX και RX ακροδέκτες
- 13 -Ψηφιακές είσοδοι / έξοδοι. μπροστά από τους αριθμούς είναι για PWM εξόδους.
- 14 -Ισόγειο και AREF ακροδέκτες. ➤ 15-ICSP για ATmega328
- 16 -ICSP για διασύνδεση USB

Ο αριθμός των ακίδων (ψηφιακών in / out και αναλογικών) ποικίλλουν ανάλογα με το μοντέλο.



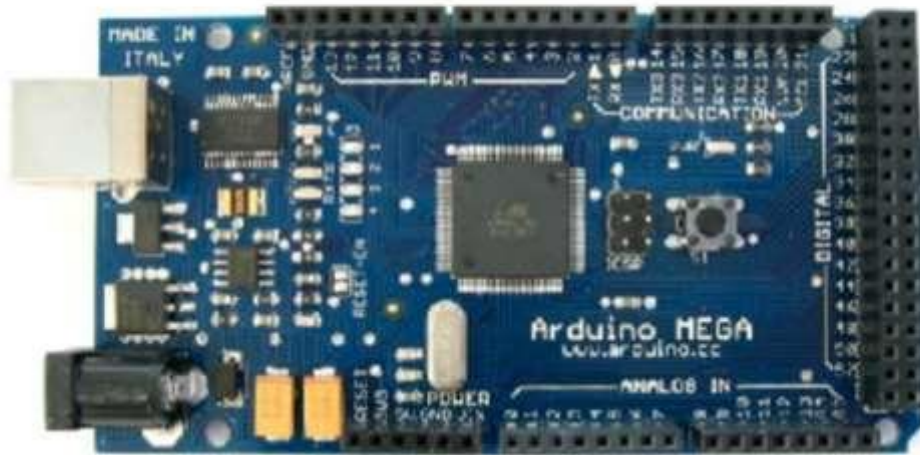
▪ ΑΛΛΑ ΜΟΝΤΕΛΑ

Arduino Pro Mini



- Microcontroller: ATmega168
- Operating Voltage: 3.3V or 5V (depending on model)
- Input Voltage : 3.35 -12 V or 5 - 12 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins : 6
- DC Current: 40 Ma
- Flash Memory: 16 KB
- SRAM: 1 KB
- EEPROM: 512 bytes
- Clock Speed: 8 MHz or 16 MHz

Arduino Mega



- Microcontroller: ATmega1280
- Operating Voltage: 5V
- Input Voltage: 6-20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- Analog Input Pins: 16
- DC Current: 40 mA
- Flash Memory: 128 KB
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 MHz

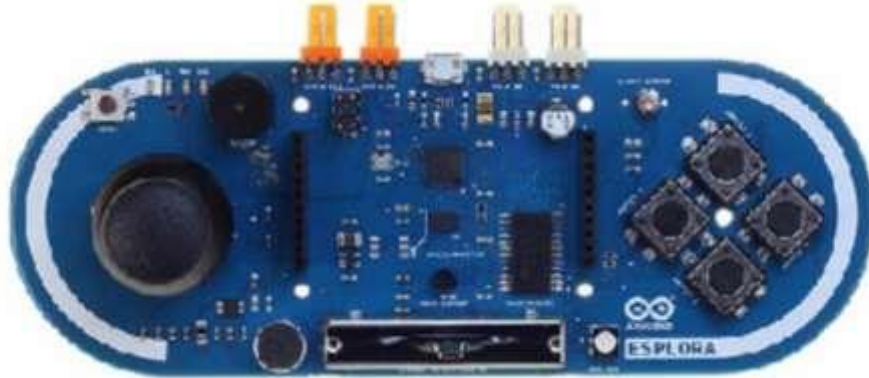
Arduino LilyPad



- Microcontroller: ATmega168V or ATmega328V
- Operating Voltage: 2.7-5.5 V
- Input Voltage: 2.7-5.5 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current: 40 mA
- Flash Memory: 16 KB

SRAM: 1 KB ○ EEPROM: 512 bytes ○ Clock Speed: 8 MHz

Arduino Esplora



- Analog joystick ○ 4 push-buttons ○ Arduino Esplora ○ Linear potentiometer slider
- Microphone ○ Light sensor ○ Temperature sensor ○ Three-axis accelerometer ○ Buzzer can produce square-waves. ○ RGB led bright LED
- 2 TinkerKit Inputs to connect the TinkerKit sensor modules with the 3-pin connectors.
- 2 TinkerKit Outputs to connect the TinkerKit actuator modules with the 3-pin connectors.
- TFT display connector connector for an optional color LCD screen, SD card
 - **ΕΠΕΚΤΑΣΕΙΣ**

Arduino Ethernet Shield



Επιτρέπει τη σύνδεση του Arduino με δίκτυο.

Arduino WiFi Shield



Επιτρέπει τη σύνδεση του Arduino με ασύρματα δίκτυα.

Arduino Motor Shield



Επιτρέπει τον έλεγχο 2 DC κινητήρων.

Arduino Bluetooth Shield



Δυνατότητα επικοινωνίας μέσω bluetooth.

Arduino GPS Shield



Δυνατότητα γεωεντοπισμού.

Arduino LCD Shield



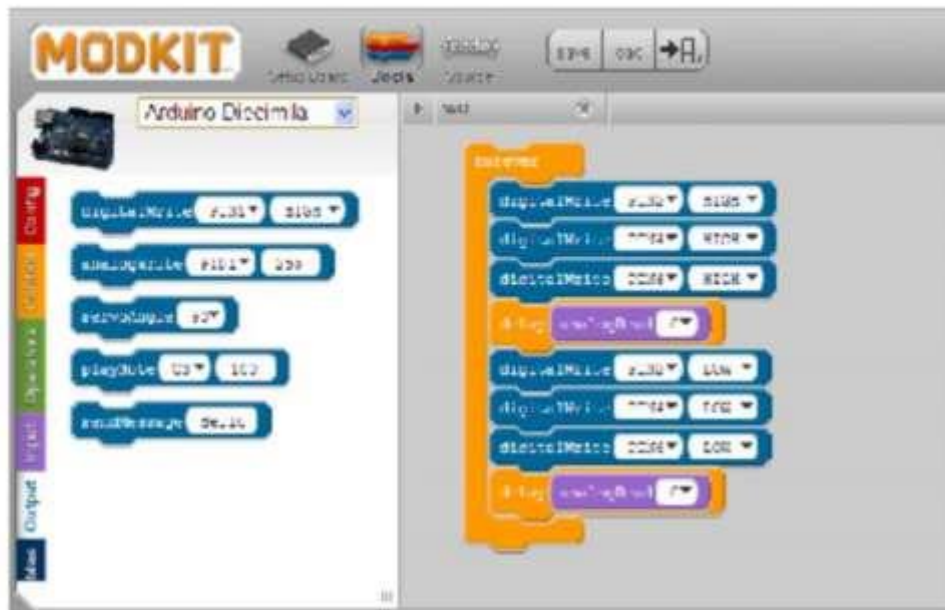
Εύκολη πρόσβαση σε οθόνη LCD. ▪ Προγραμματιστικά περιβάλλοντα

Arduino Software



- Βασίζεται στην C
- Windows/Linux/Mac OS
- Ελεύθερο – Ανοικτού κώδικα

Modkit



- Θυμίζει scratch

- Windows/Linux/Mac OS
- Free online – pay to download

Minibloq



- Γραφικό περιβάλλον
- Windows/Linux/OLPC/Mac OS

Εγκατάσταση του προγράμματος

Για να γίνει σωστή εγκατάσταση του προγράμματος, πρέπει να ακολουθηθεί μια σειρά από βήματα, ανάλογα με το λειτουργικό σύστημα που διαθέτει. Στην περίπτωση μας θα εγκατασταθεί σε λειτουργικό σύστημα των Windows XP.

1. Πλακέτα Arduino και καλώδιο USB

Στην παρούσα πτυχιακή εργασία επιλέχθηκε να χρησιμοποιηθεί η πλακέτα Arduino Uno. Θα χρειαστούμε ένα καλώδιο USB για να συνδεθούν πλακέτα και υπολογιστής.



2. Περιβάλλον Arduino Μεταφορτώνουμε δωρεάν την τελευταία έκδοση Arduino-1.0 από την ιστοσελίδα <http://arduino.cc/en/Main/Software>.

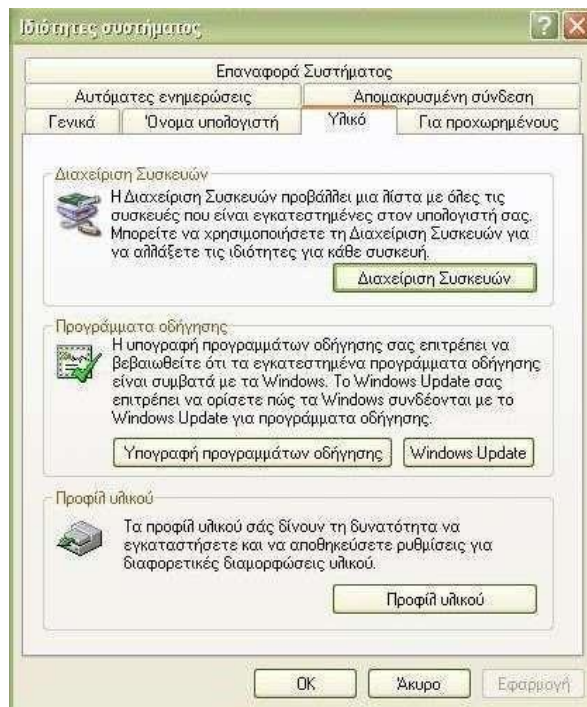
3. Σύνδεση της πλακέτα στον υπολογιστή

Συνδέουμε την πλακέτα Arduino Uno στον υπολογιστή χρησιμοποιώντας το καλώδιο USB. Παρατηρούμε ότι το LED της πλακέτας ανάβει.

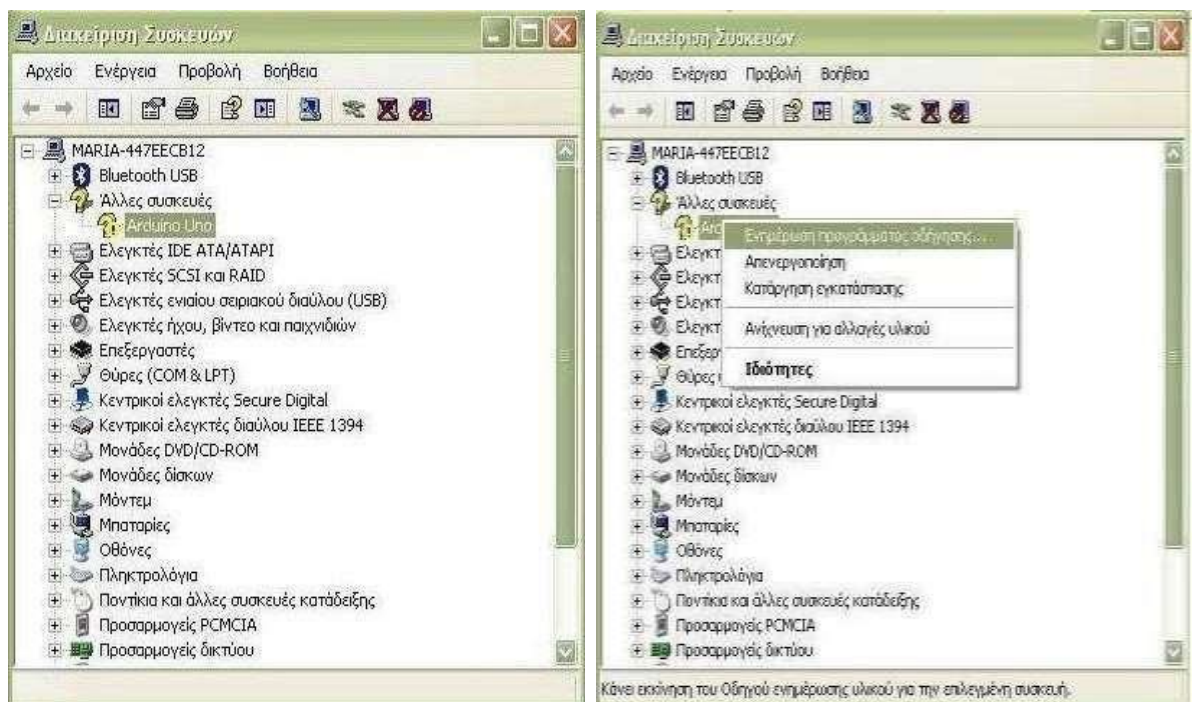
4. Εγκατάσταση του προγράμματος

+ Κάνουμε κλικ στο μενού Έναρξη, και ανοίγουμε τον Πίνακα Ελέγχου.

+ Από τον Πίνακα Ελέγχου, μεταβαίνουμε στο «Σύστημα» και ακολούθως «Υλικό» και ανοίγουμε τη διαχείριση συσκευών.



+ Βλέπουμε στις συσκευές το όνομα Arduino Uno. Κάνουμε δεξί κλικ και επιλέγουμε το «Ενημέρωση προγράμματος οδήγησης».



+ Ξεκινάει εγκατάσταση λογισμικού για το Arduino.



+ Κάνουμε εγκατάσταση τα drivers στον υπολογιστή μας.

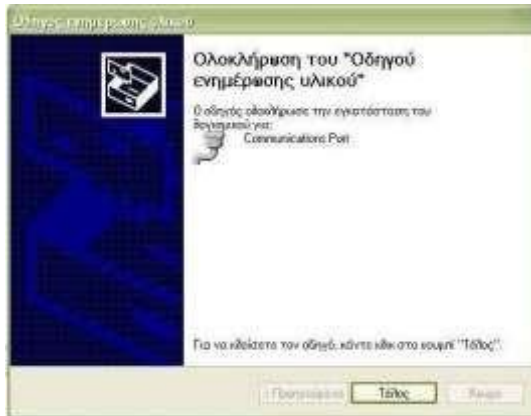


+ Περιμένουμε μέχρι να τελειώσει η εγκατάσταση λογισμικού για το Arduino.



+ Τελειώνοντας παρατηρούμε ότι στις Θύρες (COM & LPT) εμφανίστηκε το Serial Port COM 15 για το Arduino που θα χρησιμοποιήσουμε. Οπότε το Arduino έχει προγραμματιστεί στη σειριακή θύρα 15.





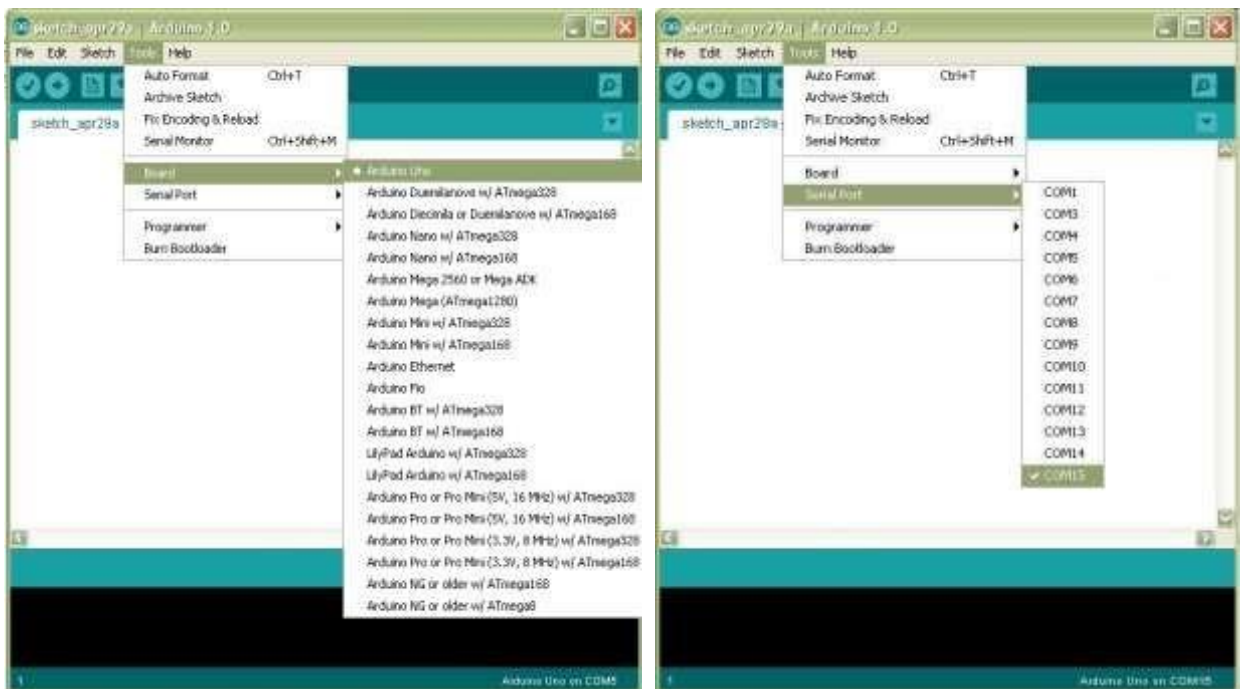
5. Έναρξη της εφαρμογής Arduino

Κάνουμε διπλό κλικ στην εφαρμογή Arduino.exe



6. Επιλογή Board και Σειριακής θύρας

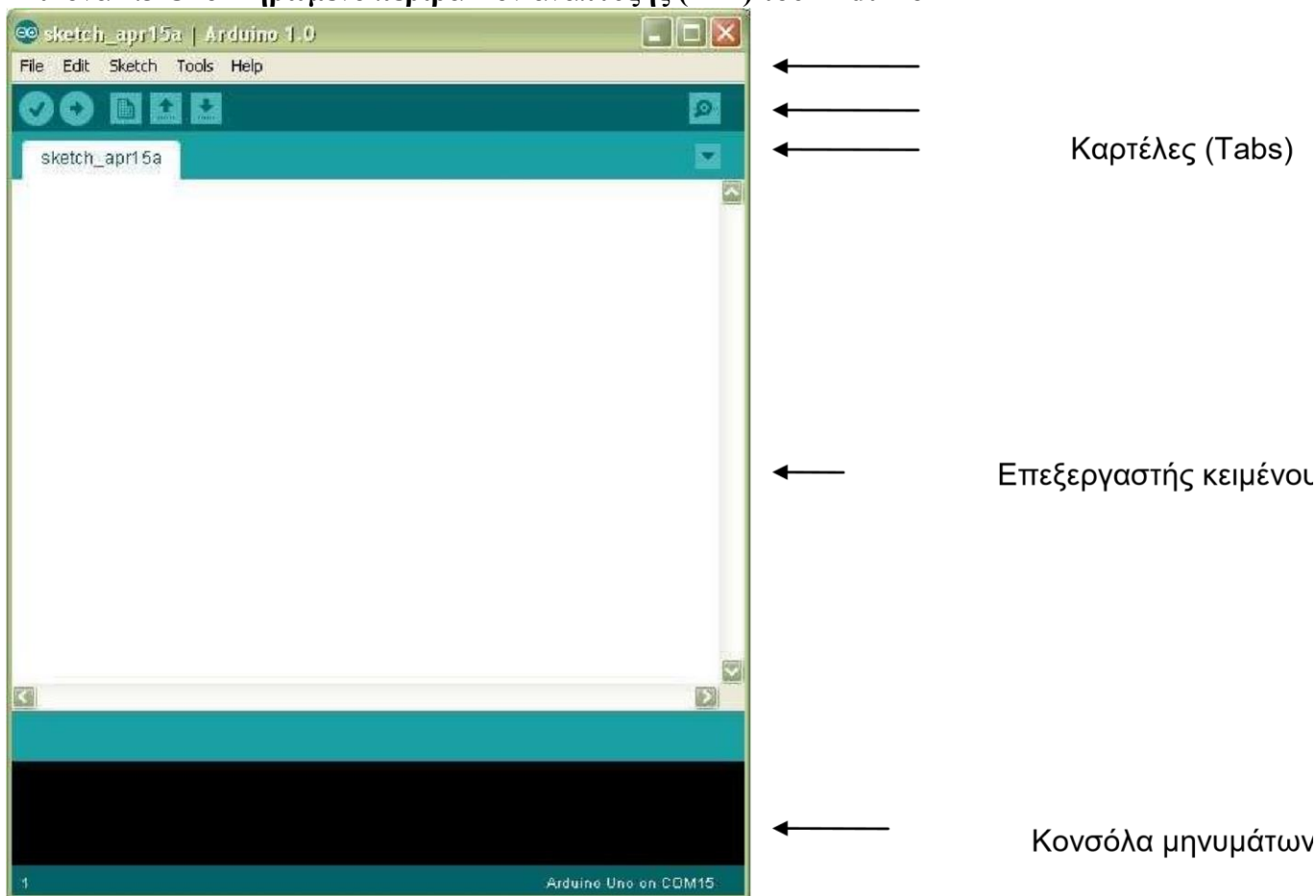
Από το περιβάλλον ανάπτυξης του, από το μενού Tools επιλέγουμε για Board το Arduino Uno και για Σειριακή θύρα το COM15.



2.10 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino Το περιβάλλον ανάπτυξης Arduino περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους.

Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης.

Εικόνα 2.3 Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino



Τα κουμπιά της γραμμής εργαλείων:



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, κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο



Save(Αποθήκευση): Αποθηκεύει το sketch



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

2.11 Σειριακή οθόνη (Serial Monitor) Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino. Πιο συγκεκριμένα, η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί send ή πατώντας το Enter. Επίσης, στο κάτω μέρος της σειριακής οθόνης, μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (baud) από την λίστα που εμφανίζεται ανάλογα με την τιμή που θα επιλεγθεί στο προγραμματισμό του Arduino με το Serial.begin().



Εικόνα 2.4: Serial Monitor

2.12 Η Δομή του προγράμματος

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
//δήλωση μεταβλητών
void setup ()
{
    //αρχικοποιήσεις
}
void loop ()
{
    //Κώδικας
}
```

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.

2.13 Βασικές δομές και λειτουργίες προγραμματισμού

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθεί ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

Δομές ελέγχου ροής

if	(δομή ελέγχου μίας συνθήκης)
if ... else	(δομή ελέγχου πολλαπλών συνθηκών)
for	(δομή επαναληπτικού ελέγχου συνθήκης)
while	(δομή επαναληπτικού ελέγχου συνθήκης)
do ... while	(δομή επαναληπτικού ελέγχου συνθήκης)
switch ... case	(δομή ελέγχου περιπτώσεων)
break	(εντολή διακοπής μιας επαναληπτικής δομής)
continue	(εντολή παράλειψης της τρέχουσας επανάληψης)
return	(εντολή επιστροφής από μία συνάρτηση)
goto	(εντολή μετάβασης σε κάποιο σημείο του κώδικα)

Αριθμητικοί τελεστές

□ =	(τελεστής εκχώρησης)
□ +	(τελεστής πρόσθεσης)
□ -	(τελεστής αφαίρεσης)
□ *	(τελεστής πολλαπλασιασμού)

- / (τελεστής διαίρεσης)
- % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

Λογικοί τελεστές

- && (λογική σύζευξη)
- || (λογική διάζευξη)
- ! (λογική άρνηση)

Δυαδικοί τελεστές

- & (δυαδική σύζευξη)
- | (δυαδική διάζευξη)
- ^ (δυαδική αποκλειστική διάζευξη)
- ~ (δυαδική άρνηση)
- << (δυαδική αριστερή ολίσθηση)
- >> (δυαδική δεξιά ολίσθηση)

Τελεστές αύξησης και μείωσης

- ++ (αύξηση κατά μία ακέραιη μονάδα)
- -- (μείωση κατά μία ακέραιη μονάδα)

Σύνθετοι τελεστές

- +=, -=, *=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
- &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

Τελεστές σύγκρισης

- == (ισότητα)
- != (ανισότητα)
- < (μικρότερο)
- > (μεγαλύτερο)
- <= (μικρότερο ή ίσο)
- >= (μεγαλύτερο ή ίσο)

Τελεστές δεικτών

- * (τελεστής απόκτησης περιεχομένου)
- & (τελεστής απόκτησης διεύθυνσης)

Σταθερές

HIGH	(τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)
LOW	(τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)
false	(λογικό επίπεδο ψεύδους σε μία συνθήκη)
true	(λογικό επίπεδο αλήθειας σε μία συνθήκη)
INPUT	(χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)
OUTPUT	(χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)
A0, ..., A5	(συμβολοσταθερές για τις αναλογικές επαφές εισόδου)

Τύποι δεδομένων

boolean	(λογική δυαδική τιμή)
char	(προσημασμένος χαρακτήρας 8 ψηφίων)
unsigned char	(μη προσημασμένος χαρακτήρας 8 ψηφίων)
byte	(μη προσημασμένος χαρακτήρας 8 ψηφίων)
int	(προσημασμένος ακέραιος αριθμός 16 ψηφίων)
unsigned int	(μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
word	(μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
long	(προσημασμένος ακέραιος αριθμός 32 ψηφίων)
unsigned long	(μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
float, double	(αριθμός κινητής υποδιαστολής απλής ακρίβειας)
String	(αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)
▪	Ως αλφαριθμητικό μπορεί να θεωρηθεί και ο πίνακας χαρακτήρων

Συναρτήσεις μετατροπής τύπων

- char(), byte()
- int(), word(), long()
- float(), double()

Συναρτήσεις εισόδου και εξόδου

- pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

Συναρτήσεις ψηφιακής εισόδου και εξόδου

- digitalWrite() (γράφει σε μία ψηφιακή επαφή έξοδο)
- digitalRead() (διαβάζει από μία ψηφιακή επαφή είσοδο)

Συναρτήσεις αναλογικής εισόδου και εξόδου

- `analogReference()` (ορίζει την τάση αναλογικής αναφοράς)
- `max()` (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
- `min()` (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
- `abs()` (επιστρέφει την απόλυτη τιμή ενός αριθμού)
- `constrain()` (ελέγχει για υπερχειλίση ή υποχειλίση ορίων)
- `map()` (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- `analogWrite()` (γράφει PWM σήματα σε μία επαφή εξόδου)
- `analogRead()` (διαβάζει από μία αναλογική επαφή εισόδου)

Προηγμένες συναρτήσεις εισόδου και εξόδου

- `tone()` (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
- `noTone()` (διακόπτει την παραγωγή τετραγωνικών σημάτων)
- `shiftOut()` (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
- `pulseIn()` (επιστρέφει την διάρκεια σε μs ενός παλμού HIGH ή LOW)

Συναρτήσεις χρόνου

- `millis()` (διάρκεια εκτέλεσης του προγράμματος σε ms)
- `micros()` (διάρκεια εκτέλεσης του προγράμματος σε μs)
- `delay()` (παύση προγράμματος - η διάρκεια δίδεται σε ms)
- `delayMicroseconds()` (παύση προγράμματος - η διάρκεια δίδεται σε μs)

Μαθηματικές και Τριγωνομετρικές συναρτήσεις

- `pow()` (επιστρέφει το αποτέλεσμα μίας δύναμης)
- `sqrt()` (επιστρέφει την ρίζα ενός αριθμού)
- `sin()` (υπολογίζει το ημίτονο ενός αριθμού)
- `cos()` (υπολογίζει το συνημίτονο ενός αριθμού)
- `tan()` (υπολογίζει την εφαπτομένη ενός αριθμού)

Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών

- `random()` (δίδεται ένας νέος αριθμός από την γεννήτρια)
- `randomSeed()` (θέτει τον σπόρο της γεννήτριας παραγωγής)

Συναρτήσεις επεξεργασίας δυαδικών αριθμών

- `lowByte()` (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
- `highByte()` (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
- `bitRead()` (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)

- `bitWrite()` (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)
- `bitSet()` (γράφει την τιμή 1 σε κάποιο ψηφίο μιας μεταβλητής)
- `bitClear()` (γράφει την τιμή 0 σε κάποιο ψηφίο μιας μεταβλητής)
- `bit()` (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

- `attachInterrupt()` (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
- `detachInterrupt()` (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

- `interrupts()` (ενεργοποιεί τα σήματα διακοπής)
- `noInterrupts()` (απενεργοποιεί τα σήματα διακοπής)

Υποστήριξη σειριακής επικοινωνίας

- `Serial` (αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους)

2.14 Ψηφιακές ακίδες (*Digital pins*)

Οι ακίδες αυτές στο 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 ακίδων του Arduino (Atmega), μπορεί να ρυθμιστεί και να χρησιμοποιηθεί, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακές ακίδες. Οι συναρτήσεις ψηφιακής εισόδου και εξόδου είναι οι παρακάτω:

`pinMode()`: Ρυθμίζει τη συγκεκριμένη ακίδα να συμπεριφέρεται ως είσοδος/ έξοδος.

Σύνταξη: `pinMode(pin, mode)`

Παράμετροι: `pin`: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.

`mode`: INPUT/OUTPUT

digitalWrite(): Γράφει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε μια ψηφιακή ακίδα. Αν η ακίδα έχει ρυθμιστεί ως έξοδος με την συνάρτηση pinMode(), τότε η τάση της θα καθορίσει στην αντίστοιχη τιμή: 5V για HIGH

και 0V για LOW. Αν η ακίδα έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην συνάρτηση digitalWrite() θα ενεργοποιήσει μια εσωτερική pullup-αντίσταση των 20 K ενώ γράφοντας LOW θα την απενεργοποιήσει.

Σύνταξη: digitalWrite(pin,value)

Παράμετροι:

pin: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.

Value: INPUT/OUTPUT

digitalRead(): Διαβάζει την τιμή από μια συγκεκριμένη ψηφιακή ακίδα, που είναι είτε HIGH είτε LOW.

Σύνταξη: digitalRead(pin)

Παράμετροι:

pin: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.

Επιστρέφει: HIGH/LOW

2.15 Αναλογικές ακίδες εισόδου (Analog input pins) Οι ελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό μετατροπέα 6 καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10 bit, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας για τους περισσότερους χρήστες Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου. Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

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

Σύνταξη: analogWrite(pin, value)

Παράμετροι: pin: Ο αριθμός της ακίδας της οποίας θα γράψει επάνω
value: ο κύκλος λειτουργίας μεταξύ 0 και 255

analogRead(): Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

Σύνταξη: analogRead(pin)

Παράμετροι: pin: Ο αριθμός της αναλογικής ακίδας εισόδου από όπου θα διαβάζει

Επιστέφει: ακέραιο από 0 έως 1023

2.16 Υποστήριξη Βιβλιοθηκών (Libraries)

Η χρήση βιβλιοθηκών προσφέρουν περισσότερο λειτουργικότητα σε συνεργασία με το υλικό και τον χειρισμό των δεδομένων. Για να χρησιμοποιηθεί μια βιβλιοθήκη σε ένα sketch, μπορεί να επιλεγεί από το μενού *Sketch* → *Import Library*. Αυτό θα εισάγει μια ή περισσότερες βιβλιοθήκες #include δηλώσεις στην κορυφή του sketch. Επειδή οι βιβλιοθήκες φορτώνονται στην πλακέτα με το sketch, αυξάνουν το μέγεθος του χώρου που καταλαμβάνεται. Εάν ένα sketch δεν χρειάζεται πλέον μια βιβλιοθήκη, αλλά μπορούμε να την διαγράψουμε από την κορυφή του κώδικα.

Για την εγκατάσταση των βιβλιοθηκών που δεν υπάρχουν ήδη στο λογισμικό, μπορεί να δημιουργηθεί ένας κατάλογος με την ονομασία libraries (βιβλιοθήκες), μέσα στον κατάλογο του sketchbook. Στην συνέχεια αποσυμπιέζουμε τη βιβλιοθήκη εκεί.

Παρακάτω ακολουθούν μερικές από τις βιβλιοθήκες που υποστηρίζονται από το Arduino.

Επικοινωνίας (δικτύωση και πρωτόκολλα):

- Messenger - για την επεξεργασία κειμένου με βάση τα μηνύματα από τον υπολογιστή.
- NewSoftSerial - βελτιωμένη έκδοση της βιβλιοθήκης SoftwareSerial.
- OneWire - συσκευές ελέγχου (της Dallas Semiconductor) που χρησιμοποιούν το πρωτόκολλο one Wire.
- PS2Keyboard - διαβάζει χαρακτήρες από ένα πληκτρολόγιο PS2.
- Simple Message System - στέλνει μηνύματα μεταξύ Arduino και του υπολογιστή
- SSerial2Mobile - αποστολή μηνυμάτων κειμένου ή e-mail χρησιμοποιώντας ένα κινητό τηλέφωνο (μέσω εντολών AT μέσω σειράς λογισμικού)
- Webduino - επεκτάσιμη βιβλιοθήκη web server (για χρήση με το Arduino Ethernet Shield)
- X10 - Αποστολή σημάτων X10 μέσω γραμμών εναλλασσόμενου ρεύματος
- Xbee - για την επικοινωνία με XBees σε λειτουργία API
- SerialControl - Τηλεχειριστήριο άλλες Arduino πάνω από μια σειριακή σύνδεση
- Servo – για τον έλεγχο κινητήρων τύπου Servo.

Ανίχνευσης:

- Capacitive Sensing - δύο ή περισσότερες ακίδες σε αισθητήρες πυκνωτή
- Debounce - για την ανάγνωση θορυβώδη ψηφιακών εισόδων (π.χ. από τα κουμπιά).

20

Εμφάνιση και LED:

- Improved LCD library - διορθώνει σφάλματα αρχικοποίησης LCD στην επίσημη Arduino LCD βιβλιοθήκη.
- GLCD – γραφικές ρουτίνες για LCD με βάση την KS0108 ή ισοδύναμο chipset.
- LedControl - για τον έλεγχο των LED ή επτά τμημάτων οθόνες με MAX7221 ή MAX7219.
- LedControl - μια εναλλακτική λύση στη βιβλιοθήκη Matrix για την οδήγηση με πολλαπλούς LED.
- LedDisplay - τον έλεγχο της HCMS-29xx οθόνη LED.

Συχνότητα παραγωγής ήχου:

- Tone - αναπαράγει κύματα ήχου συχνότητας στο παρασκήνιο σε κάθε καρφίτσα του μικροελεγκτή.

Κινητήρες και PWM:

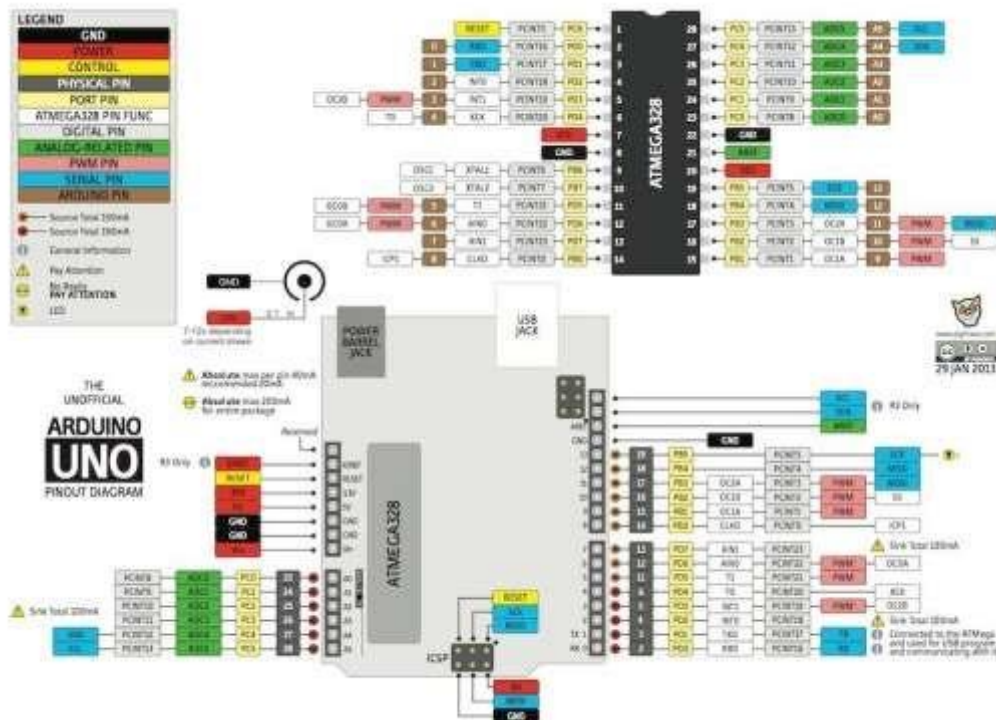
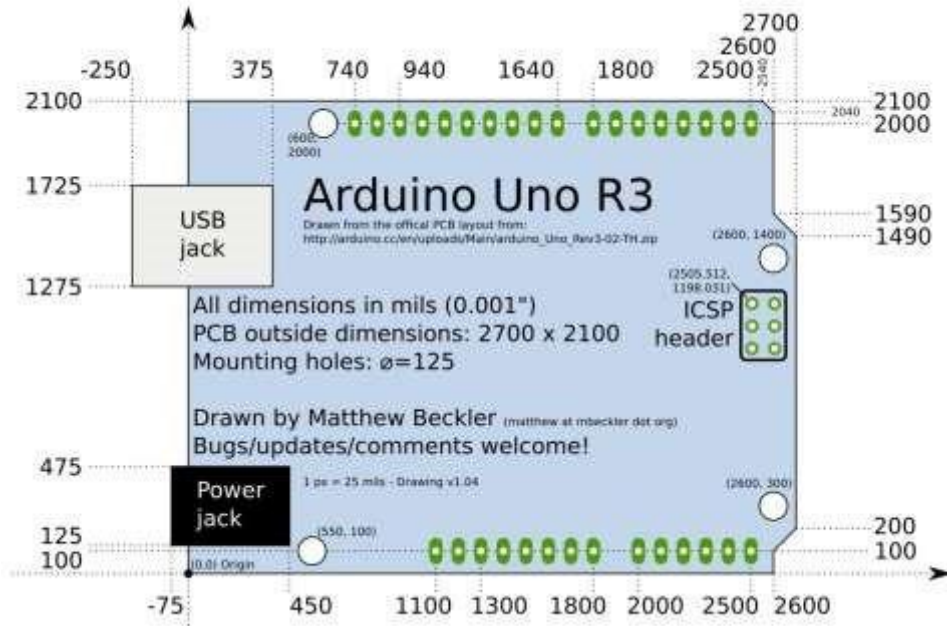
- TLC5940 - 16 κανάλι 12 bit PWM ελεγκτή.

Χρονοδιάγραμμα:

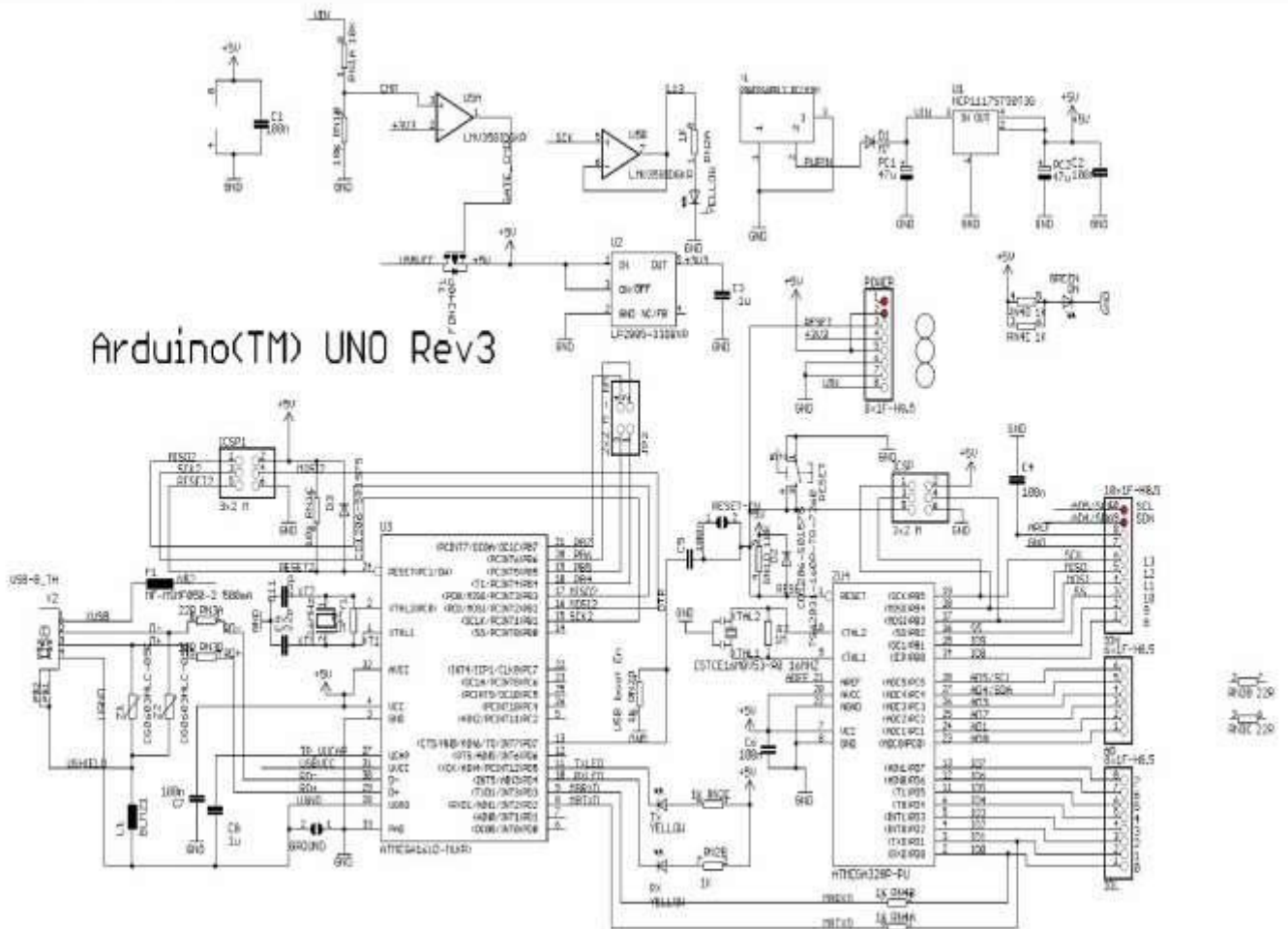
- DateTime - μια βιβλιοθήκη για την παρακολούθηση της τρέχουσας ημερομηνίας και ώρας.
- MsTime2 - χρησιμοποιεί το χρονόμετρο διακοπής 2 για να ενεργοποιήσει μια δράση κάθε χιλιοστά του δευτερολέπτου N.

Βοηθητικά προγράμματα:

Streaming - μια μέθοδος για την απλοποίηση δηλώσεων εκτύπωσης.




Arduino function			Arduino function		
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)





- Φθηνό σε σχέση με τους ανταγωνιστές
- Open source hardware
- Επεκτασιμότητα
- Πολυπληθής κοινότητα που το υποστηρίζει
- Πλήθος tutorials και βοηθημάτων



Χρειάζεται εξοικείωση με βασικές έννοιες ηλεκτρονικών κυκλωμάτων
Έντονη hands-on εμπειρία

Πείραμα:

Στο πείραμα θα κατασκευάσουμε ένα κύκλωμα με τρία αισθητήρια, τα οποία θα είναι τα εξής: αισθητήριο θερμοκρασίας LM35, αισθητήριο πίεσης MPX4250AP και μια φωτοαντίσταση με μία δίοδο **LED** (Light Emitting Diodes) όπου θα ελέγχουμε την φωτεινότητα της. Στο παρακάτω σχήμα μπορούμε να δούμε αναλυτικά την συνδεσμολογία του κάθε υλικού πάνω στο ράστερ και στην συνέχεια την διασύνδεση του με το Arduino μας.

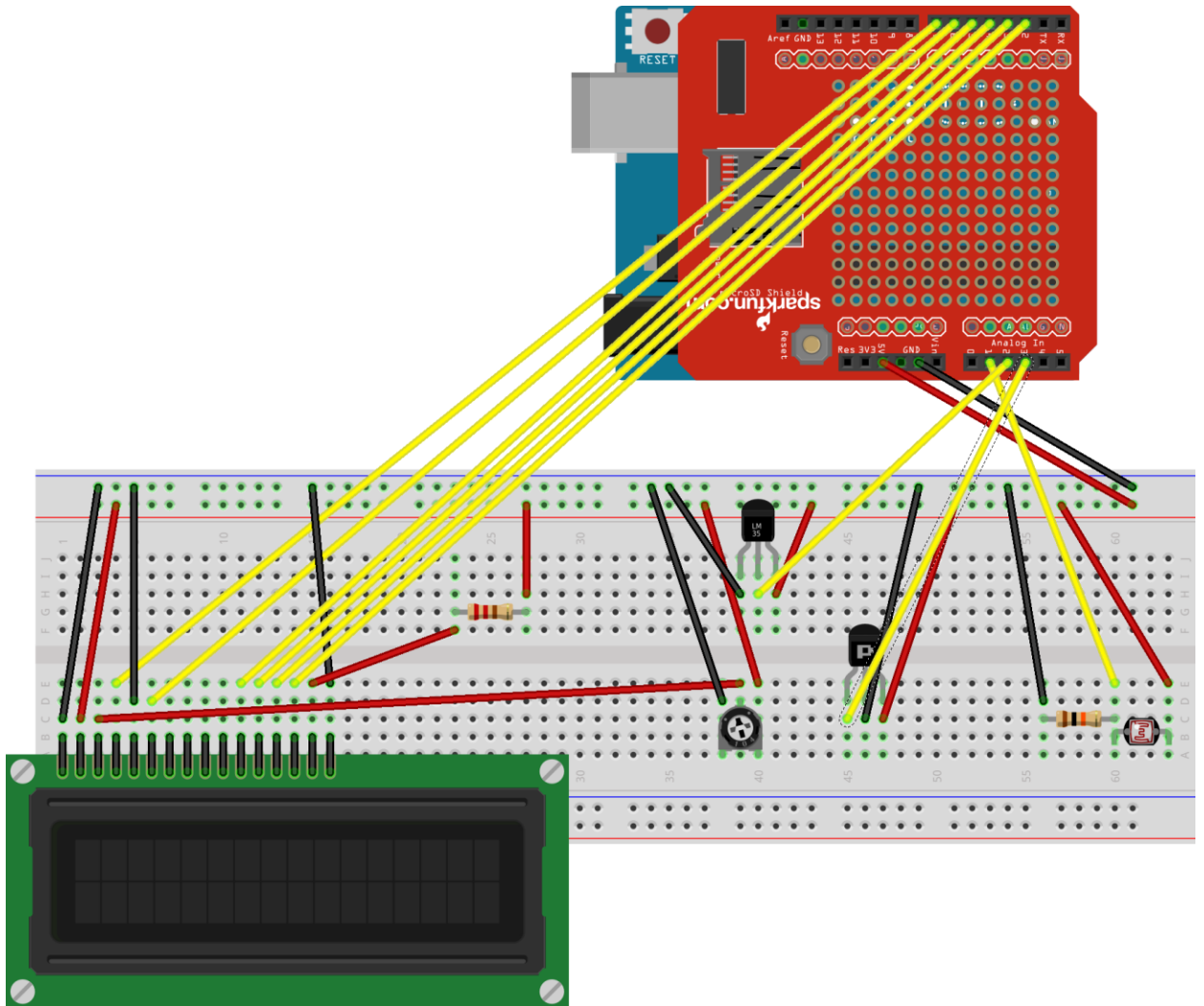
Αρχίζοντας με το LM35, και τον τρόπο που κάναμε την συνδεσμολογία του, βλέπουμε ότι έχουμε συνδέσει την τροφοδοσία του και την γείωση του στα κατάλληλα ποδαράκια του υλικού, και για καταφέρουμε μεγαλύτερη ακρίβεια στις τιμές μας, έχουμε τοποθετήσει μια αντίσταση ανάμεσα στο ποδαράκι της γείωσης και το ποδαράκι που μας δίνει τις μετρήσεις μας, και έτσι φιλτράρουμε τυχόν παρεμβολές από την γείωση μας ειδικότερα αν δεν έχουμε ένα σταθερό τροφοδοτικό και το τροφοδοτούμε από τον υπολογιστή μας. Στην περίπτωση που δεν θα τοποθετούσαμε μία τέτοια αντίσταση βλέπουμε τις τιμές της θερμοκρασίας μας να αλλάζουν ανορθόδοξα χωρίς να μας δίνουν κάποια σωστή τιμή. Εδώ πρέπει να πούμε ότι το συγκεκριμένο υλικό μας δίνει αναλογική έξοδο για αυτό και το έχουμε συνδέσει τις αναλογικές εισόδους του Arduino μας. Η χρήση του δεν είναι για εργασίες υψηλής ακρίβειας διότι και το ίδιο δεν είναι κατασκευασμένο για τέτοιου είδους εργασίες. Είναι για την μέτρηση θερμοκρασίας με απόκλιση 1 τουλάχιστον βαθμό κελσίου. Είναι ιδανικό για ποικίλες εργασίες και προτζεκτ διότι είναι ένα πολύ φτηνό εξάρτημα και πολύ διαδεδομένο στην αγορά. Ορίζουμε την είσοδο του στον κώδικα μας και με μία εξίσωση παίρνουμε τα κατάλληλα αποτελέσματα όπου μπορούμε να τα εξάγουμε στον υπολογιστή μας μέσω της κονσόλας επικοινωνίας του Arduino. Έτσι έχουμε την δυνατότητα να χρησιμοποιήσουμε αυτήν την πληροφορία για να κάνουμε κάποια άλλη ενέργεια αυτόματα, πχ να ανοίξουμε έναν ανεμιστήρα όταν η θερμοκρασία είναι υψηλή, ή να ανοίξουμε την θέρμανση όταν είναι χαμηλή.

Η φωτοαντίσταση μας είναι το δεύτερο υλικό με το οποίο θα ασχοληθούμε σε αυτή την εργασία. Είναι ένα υλικό το οποίο έχει τη ιδιότητα όπως λέει και η ονομασία του, να αλλάζει τη εσωτερική του αντίσταση με την αλλαγή της φωτεινότητας. Στην συγκεκριμένη εργασία το έχουμε συνδέσει παράλληλα με μία αντίσταση για να μας δίνει πιο σταθερές τιμές, και με αυτόν τον τρόπο μπορούμε να ελέγχουμε το led μας με μεγαλύτερη ευκολία. Το led έχει συνδεθεί με τις ψηφιακές εξόδους του Arduino μας και σύμφωνα με τον κώδικα μας δυναμώνει η φωτεινότητα του κάθε φορά που πέφτει φως στην φωτοαντίσταση μας, και το αντίθετο κάθε φορά που καλύπτουμε την

φωτοαντίσταση και δεν την βλέπει το φως. Έχουμε μία μικρή καθυστέρηση λόγω του επεξεργαστή μας, διότι δεν μπορεί να κάνει multitasking όπως για παράδειγμα ο υπολογιστής σου και με αποτέλεσμα να πρέπει να τρέχει κάθε γραμμή κώδικα την μία μετά την άλλη. Βέβαια για την εργασία μας αυτό δεν δημιουργεί κανένα απολύτως πρόβλημα. Η φωτοαντίσταση είναι και αυτή ένα πολύ φτηνό εξάρτημα και μπορεί να χρησιμοποιηθεί σε ποικίλες εργασίες για το σπίτι ή το αυτοκίνητο μας.

Το τρίτο υλικό που χρησιμοποιήσαμε είναι το MPX4250AP το οποίο είναι ένας μετρητής ατμοσφαιρικής πίεσης, πολύ κοινός και γνωστός στην ηλεκτρονική. Η συνδεσμολογία του είναι απλή, διαβάζοντας πάντα το εγχειρίδιο του, βλέπουμε ποια ποδαράκια είναι για την τροφοδοσία, την γείωση και τα δεδομένα μας, οπότε και τα συνδέουμε αναλόγως στο Arduino μας. Τα δεδομένα και εδώ είναι αναλογικά, και με αυτό εννοούμε ότι είναι ρεύμα που μετράμε στα πίν και από αυτό μπορούμε να καταλάβουμε τις διακυμάνσεις στις τιμές πίεσης που του ασκείτε. Ο τρόπος λειτουργίας του είναι απλός και συνδέοντας τα ποδαράκια του στο Arduino μας παίρνουμε τις ανάλογες μετρήσεις και από εκεί μπορούμε εύκολα να τις απεικονίσουμε στον υπολογιστή μας ή σε κάποια οθόνη ή να κάνουμε κάποια ενέργεια με αυτές. Μπορούμε να αλλάξουμε τα αποτελέσματα από bar,psi, κτλ. απλώς αλλάζοντας ένα μικρό κομμάτι του κώδικα με της αναλογίες που παίρνουμε σε volt και την μετατροπή τους στο αντίστοιχο μέτρο πίεσης. Σε κάποιες περιπτώσεις μπορεί να έχουμε τιμές που ανεβοκατεβαίνουν, και αυτό μπορούμε να το λύσουμε με έναν πυκνωτή ανάμεσα από τα ποδαράκια γείωσης και δεδομένων που θα λειτουργεί σαν φίλτρο. Στην προκειμένη περίπτωση δεν είχαμε κάποιο τέτοιο θέμα με της μετρήσεις μας. Είναι επίσης και αυτό ένα φτηνό εξάρτημα με ποικίλες δυνατότητες και επειδή είναι υψηλής ακριβείας, το κάνει χρήσιμο ακόμα και σε μηχανήματα ιατρική.

Σχηματικό διάγραμμα



fritzing

```
#include <SPI.h>
#include <Wire.h>
#include <SD.h>
#include <LiquidCrystal.h>
#include "RTCLib.h"

RTC_DS1307 RTC;
Sd2Card card;
SdVolume volume;
SdFile root;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
const int chipSelect = 10;
File logFile;

// Pressure Sensor
const int MPX4250AP_pin = 3;
double MPX4250AP_read_value = 0;
float pressurekPa = 0;
// Temperature Sensor
const int temperatureSensorPin = 2;
// Light Sensor
const int lightSensorPin = 1;
int lightLevel;

void setup () {
  pinMode(10, OUTPUT);
  Serial.begin(57600);
  Wire.begin();
  RTC.begin();

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");

    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    return;
  }
}
```

```

}
Serial.println("card initialized.");
lcd.begin(16, 2);
// print the type of card
Serial.print("\nCard type: ");
switch(card.type()) {
  case SD_CARD_TYPE_SD1:
    Serial.println("SD1");
    lcd.print("SD1");
    break;
  case SD_CARD_TYPE_SD2:
    Serial.println("SD2");
    lcd.print("SD2");
    break;
  case SD_CARD_TYPE_SDHC:
    Serial.println("SDHC");
    lcd.print("SDHC");
    break;
  default:
    Serial.println("Unknown");
}
}

void loop () {
  DateTime now = RTC.now();
  showAndWriteMeasurements();
  // Εδώ ρυθμίζω το διαστήμα στο οποίο παίρνει τιμές από τους αισθητήρες
  delay(10000);
}

void showAndWriteMeasurements() {
  MPX4250AP_read_value = analogRead(MPX4250AP_pin);
  pressurekPa = (MPX4250AP_read_value*(.00488)/(.022)+20);
  lcd.setCursor(0, 1); lcd.print("P:");
  lcd.setCursor(2, 1); lcd.print(pressurekPa);

  lcd.setCursor(0,0); lcd.print("C:");
  float celsius = (analogRead(temperatureSensorPin)/1024.0)*500;
  lcd.setCursor(2,0); lcd.print(celsius);

  lcd.setCursor(10,1); lcd.print("L:");

```

```
lightLevel = analogRead(lightSensorPin);  
lcd.setCursor(12,1); lcd.print(lightLevel);
```

```
lcd.setCursor(8,0);  
  lcd.print(millis());  
  
  logFile = SD.open("log.txt", FILE_WRITE);  
  if (logFile) {  
    DateTime now = RTC.now();  
    logFile.print(now.year(), DEC);  
    logFile.print('/');  
    logFile.print(now.month(), DEC);  
    logFile.print('/');  
    logFile.print(now.day(), DEC);  
    logFile.print(' ');  
    logFile.print(now.hour(), DEC);  
    logFile.print(':');  
    logFile.print(now.minute(), DEC);  
    logFile.print(':');  
    logFile.print(now.second(), DEC);  
    logFile.println();  
  
    logFile.print("Pressure: ");  
    logFile.println(pressurekPa);  
    logFile.print("Celsius: ");  
    logFile.println(celsius);  
    logFile.print("Light level:");  
    logFile.println(lightLevel);  
  
    logFile.close();  
    Serial.println("Measurements written to log file");  
  } else {  
    Serial.println("error opening log file");  
  }  
}
```

- <http://www.oomlout.com/a/products/ardx/>
 - <http://www.arduino.cc/>
 - <http://el.wikipedia.org/wiki>
 - <http://creativec0d1ng.blogspot.gr/2012/09/introduction-to-arduino-part-1.html>
 - <http://arduino.cc/en/Main/Products?from=Main.Hardware>
 - <http://shieldlist.org/>
 - <http://arduino.cc/en/main/software>
 - <http://www.modk.it>
 - <http://blog.minibloq.org/>
 - <http://blogs.sch.gr/ioarvanit/2012/12/17/%CF%84%CE%BF-arduino-%CE%B1%CF%80%CF%8C-%CF%84%CE%B7%CE%BD-%CF%83%CE%BA%CE%BF%CF%80%CE%B9%CE%AC%CE%B5%CE%BD%CF%8C%CF%82-newbie/>
- Βιβλία και σημειώσεις που χρησιμοποιήθηκαν:

- [A] Δρ. Βολογιαννίδης Σταύρος (2009). Ευφυής Έλεγχος, Θεωρία και Εφαρμογής.
[B] Banzi, M. (2009). *Getting Started with Arduino*. O'Reilly.

Ιστοσελίδες που χρησιμοποιήθηκαν:

Arduino

- [1] <http://arduino.cc/en/Guide/Environment?from=Tutorial.Bootloader>
[2] <http://arduino.cc/en/Guide/Windows>
[3] <http://arduino.cc/en/Reference/HomePage> [4]
<http://arduino.cc/en/Main/ArduinoBoardUno> [5]
<http://arduino.cc/en/Tutorial/Memory>

