Α.Ε.Ι. ΠΕΙΡΑΙΑ Τ.Τ

Φεβρουάριος 2017

Εκπαιδευτικό Ίδρυμα Πειραιά

Τμήμα Ηλεκτρολογίας

Υλοποίηση Ενός Συναγερμού Αυτοκίνητου Χρησιμοποιώντας Πλατφόρμα Arduino

Πτυχιακή Εργασία

Ελενής Γιώργος ΑΜ: 41919

Επιβλέπων: Βαρσαμής Χρήστος - Πλάτωνας

Αθήνα, Νοέμβριος 2017

Περιεχόμενα

ΕΙΣΑΓΩΓΗ

ARDUINO

Το Arduino είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και 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 πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία Digital Communities στο Prix Ars Electronica το 2006. Το 2005, ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο.

Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduin της Ivrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

ΚΕΦΑΛΑΙΟ 2

2.1 Γενικά

Πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περίεργος -για hardware- χαρακτηρισμός «ανοικτού κώδικα»). Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί – και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός – ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει.

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο.

Μικροελεγκτής – η καρδιά του Arduino

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

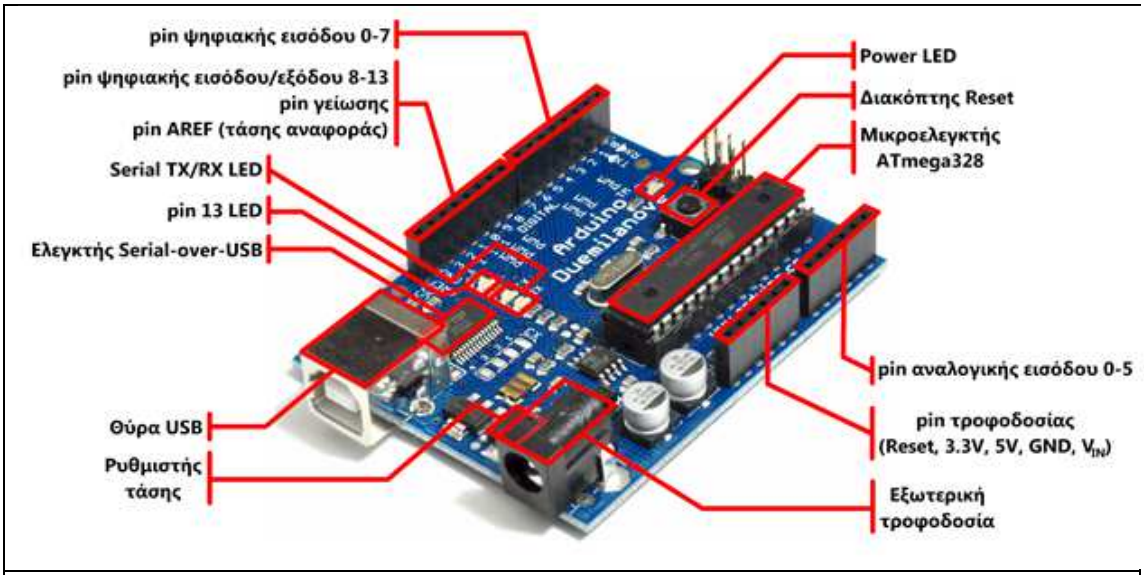
- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.

- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για

2.2 Μέρη ενός Arduino Uno

Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Εικόνα 1.1: Μέρη ενός arduino

Η ανατομία ενός Arduino Duemilanove. Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει

ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγου χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγου χάρη μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη).

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγου χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.

- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει *άμεσα* και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγου χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχετε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπές ψηφιακές έξοδοι.

Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρείτε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να

τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμάνετε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref η οποία, αν δεν κάνετε κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά σας μπορείτε να «διαβάσετε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V).

Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσετε το pin AREF με 3.3V και στην συνέχεια δοκιμάσετε να διαβάσετε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα σας επιστρέψει την τιμή 512.

Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

2.3 Χαρακτηριστικά του Arduino

Microcontroller:	ATmega328
Τάση λειτουργίας:	5V
Τάση εισόδου:	7-12V
Τάση εισόδου (όριο):	6-20V
Digital I/O Pins:	14 (εκ των οποίων 6 περιέχουν PWM εξόδους)
Analog Input Pins:	6
DC ρεύματος I/O Pin:	40 mA
DC τρέχουσα για 3.3V Pin:	50 mA
Flash Memory:	32 KB εκ των οποίων 0,5 KB που χρησιμοποιούνται από τον bootloader
SRAM:	2 KB
EEPROM:	1 KB
Clock Speed:	16 MHz

2.4 Βασικές μνήμες

Οι πλατφόρμες Arduino διαθέτουν τρείς 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 απενεργοποίησης του ρεύματος). Η μνήμη SRAM είναι ασταθής και οι πληροφορίες χάνονται όταν εναλλάσσεται το ρεύμα.

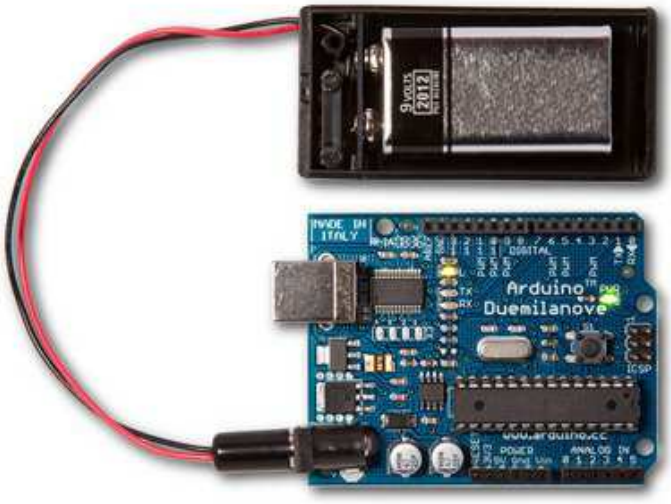
Επειδή δεν υπάρχει πολύ διαθέσιμη SRAM, αν τελειώσει, το πρόγραμμα μπορεί να αποτύχει με απροσδόκητους τρόπους. Μπορεί να φαίνεται ότι φορτώνει με επιτυχία, αλλά δεν τρέχει, ή τρέχει παράξενα. Για να ελεγχθεί εάν αυτό συμβαίνει, μπορούν να μειωθούν τα σχόλια ή οι σειρές ή άλλες δομές δεδομένων στο sketch (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.

2.5 Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φις των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.

Για εφαρμογές μακριά από τον υπολογιστή θα χρειαστείτε ένα μετασχηματιστή -σαν αυτό της εικόνας- για να τροφοδοτήσετε το Arduino με ρεύμα. Μπορείτε να χρησιμοποιήσετε ένα παλιό φορτιστή ή ένα τροφοδοτικό που σας έχει περισσέψει από κάποια άλλη συσκευή, αρκεί να παρέχει από 7 ως 12 Volt. Αν πάλι θέλετε μια εντελώς αυτόνομη εφαρμογή που δεν εξαρτάται ούτε από πρίζα, μια κοινή μπαταρία των 9 Volt συνδεδεμένη όπως στην εικόνα είναι η ιδανική τροφοδοσία.

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.



2.5.1 τροφοδοσία Arduino με μπαταριά 9v

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.

- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.

- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.

- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.

- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή του φις των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φις, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V),

πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

2.6 Ενσωματωμένα κουμπιά και LED

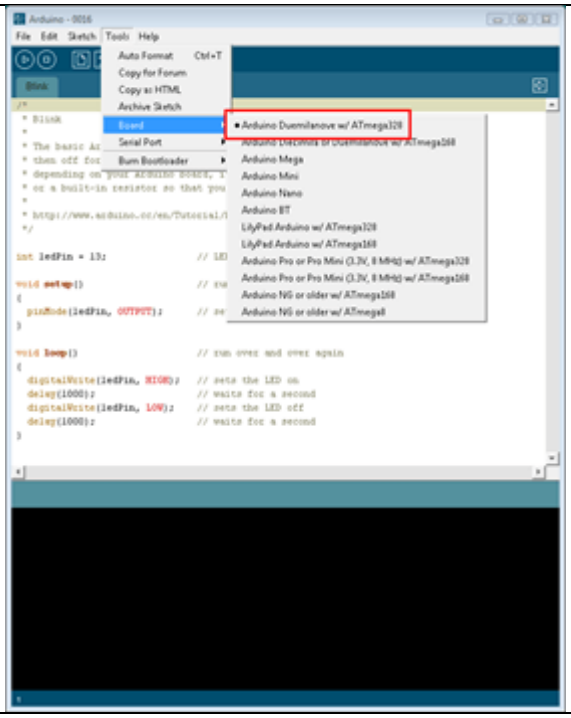
Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης.

Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής.

Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED (θα το δείτε αυτό στην συνέχεια όταν θα φτιάξετε την πρώτη εφαρμογή σας). Για να μπορείτε να το κάνετε αυτό από την πρώτη στιγμή, χωρίς να συνδέσετε τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχετε συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά σας, θα ανάψει αυτό το ενσωματωμένο LED.

2.7 Επικοινωνία



2.7.1Arduino IDE και σύνδεση με τον υπολογιστή

Ότι χρειάζεστε για την διαχείριση του Arduino από τον υπολογιστή σας το παρέχει το Arduino IDE, την τελευταία έκδοση του οποίου μπορείτε να κατεβάσετε από το επίσημο site για καθένα από τα τρία δημοφιλέστερα λειτουργικά συστήματα.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

Το κεντρικό παράθυρο του Arduino IDE χωρίζεται σε δύο μέρη. Στο πάνω μέρος γράφετε τα sketch σας και στο κάτω μέρος εμφανίζονται πιθανά λάθη κατά την διαδικασία της μεταγλώττισης. Αμέσως μετά την πρώτη του εκτέλεση, δηλώστε την έκδοση του Arduino σας (όπως φαίνεται στην εικόνα) και την εικονική σειριακή που χρησιμοποιεί.

Ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση, αρκετά έτοιμα παραδείγματα, μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino, τον compiler για την μεταγλώττιση των sketch σας, ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch σας και την επιλογή να ανεβάσετε το μεταγλωττισμένο sketch στο Arduino.

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σας σύστημα ως εικονική σειριακή θύρα.

Αν ο οδηγός του ελεγκτή Serial-over-USB είναι σωστά εγκατεστημένος στα Windows, το Arduino θα αναγνωρίζεται από τον Device Manager όπως στην εικόνα. Εκεί μπορείτε να δείτε και τον αριθμό της εικονικής σειριακής θύρας που του ανατέθηκε. Για την σύνδεση θα χρειαστείτε ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών. Για την αναγνώριση από το λειτουργικό θα χρειαστεί να εγκαταστήσετε τον οδηγό του FTDI chip (δηλαδή του ελεγκτή Serial-over-USB) ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE που κατεβάσατε. Την τελευταία έκδοση αυτού του οδηγού μπορείτε επίσης να κατεβάσετε για κάθε λειτουργικό σύστημα από το site της FTDI. Σημειώστε ότι στους τελευταίους πυρήνες του Linux υπάρχει εγγενής υποστήριξη του συγκεκριμένου ελεγκτή.

Αν όλα έγιναν σωστά, το κεντρικό παράθυρο του Arduino IDE θα εμφανιστεί όταν το εκτελέσετε και στο μενού Tools –> Serial Port θα πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyusb## για το Linux). Επιλέξτε αυτή την εικονική θύρα και στην συνέχεια επιλέξτε τον τύπο του Arduino σας (Arduino Duemilanove w/ ATmega328) από το μενού Tools –> Board.

Το Arduino είναι πλέον έτοιμο να δεχτεί τα sketch σας. Αν εμφανίστηκε οποιοδήποτε πρόβλημα διαβάστε τις αναλυτικές οδηγίες εγκατάστασης για κάθε λειτουργικό σύστημα στη διεύθυνση:

http://arduino.cc/en/Guide/HomePage.

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σας σύστημα ως εικονική σειριακή θύρα. Αν ο οδηγός του ελεγκτή Serial-over-USB είναι σωστά εγκατεστημένος στα Windows, το Arduino θα αναγνωρίζεται από τον Device Manager όπως στην εικόνα. Εκεί μπορείτε να δείτε και τον αριθμό της εικονικής σειριακής θύρας που του ανατέθηκε.

Για την σύνδεση θα χρειαστείτε ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών. Για την αναγνώριση από το λειτουργικό θα χρειαστεί να εγκαταστήσετε τον οδηγό του FTDI chip (δηλαδή του ελεγκτή Serial-over-USB) ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE που κατεβάσατε. Την τελευταία έκδοση αυτού του οδηγού μπορείτε επίσης να κατεβάσετε για κάθε λειτουργικό σύστημα από το site της FTDI. Σημειώστε ότι στους τελευταίους πυρήνες του Linux υπάρχει εγγενής υποστήριξη του συγκεκριμένου ελεγκτή.

Αν όλα έγιναν σωστά, το κεντρικό παράθυρο του Arduino IDE θα εμφανιστεί όταν το εκτελέσετε και στο μενού Tools –> Serial Port θα πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyusb## για το Linux). Επιλέξτε αυτή την εικονική θύρα και στην συνέχεια επιλέξτε τον τύπο του Arduino σας (Arduino Duemilanove w/ ATmega328) από το μενού Tools –> Board.

Το Arduino είναι πλέον έτοιμο να δεχτεί τα sketch σας. Αν εμφανίστηκε οποιοδήποτε πρόβλημα διαβάστε τις αναλυτικές οδηγίες εγκατάστασης για κάθε λειτουργικό σύστημα στη διεύθυνση :

http://arduino.cc/en/Guide/HomePage.

2.8 Γλώσσα Προγραμματισμού

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του

Arduino. Οι πιο σημαντικές από αυτές επεξηγούνται στον πίνακα που ακολουθεί:

Όρισμα	**Είδος**	**Τύπος**	**Παράμετροι**	**Περιγραφή**
LOW	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	–	(*pin*, *mode*)	Καθορίζει αν το συγκεκριμένο ψηφιακό *pin* θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο *mode* (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	–	(*pin*, *pinstatus*)	Θέτει την κατάσταση *pinstatus* (HIGH ή LOW) στο συγκεκριμένο ψηφιακό *pin*.
digitalRead	Συνάρτηση	int	(*pin*)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού *pin* (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.
Analog Reference	Εντολή	–	(*type*)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο *type* για να καθορίσει την τάση αναφοράς (V_{ref}) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το pin AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(*pin*)	Επιστρέφει έναν ακέραιο από 0 εώς 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο *pin* αναλογικής εισόδου στην κλίμακα 0 ως V_{ref}.
analogWrite	Εντολή	–	(*pin*, *value*)	Θέτει το συγκεκριμένο ψηφιακό*pin* σε κατάσταση

				ψευδοαναλογικής εξόδου (PWM). Η παράμετρος *value* καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με *value* 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2^32ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
delay	Εντολή	–	(*time*)	Σταματά προσωρινά την ροή του προγράμματος για *time* ms. Η παράμετρος *time* είναι unsigned long (από 0 ως 2^32). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	–	(interrupt, function, triggermode)	Θέτει σε λειτουργία το συγκεκριμένο interrupt, ώστε να ενεργοποιεί την συνάρτηση function, κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο triggermode: • LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW) • RISING (όταν από LOW γίνει HIGH)

				• FALLING (όταν από HIGH γίνει LOW) • CHANGE (όταν αλλάξει κατάσταση γενικά)
Detach/ Interrupt	Εντολή	–	(interrupt)	Απενεργοποιεί το συγκεκριμένο interrupt.
noInterrupts	Εντολή	–	()	Σταματά προσωρινά την λειτουργία όλων των interrupt.
interrupts	Εντολή	–	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή noInterrupts.
Serial.begin	Μέθοδος κλάσης	–	(datarate)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	–	(data)	Διοχετεύει τα δεδομένα data για αποστολή μέσω του σειριακού interface. Η παράμετρος dataμπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες ώστε να έχει την γενική δομή:

```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

void setup()
{
  // ...
}

void loop()
{
  // ...
}

// Υπόλοιπες συναρτήσεις...
```

Η βασική ρουτίνα setup() εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα loop() περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος while (true).

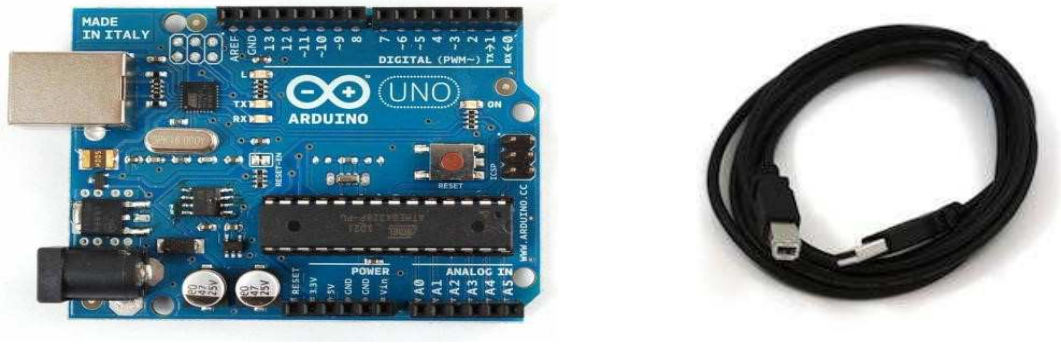
Αν και πρόκειται μόνο για τις πιο βασικές λειτουργίες της γλώσσας του Arduino, με αυτές και με λίγες βασικές γνώσεις C θα μπορέσετε να δημιουργήσετε το sketch ακόμα και για κάποιο αρκετά περίπλοκο project (όπως αυτά που θα ακολουθήσουν σε επόμενα τεύχη). Για το πλήρες reference πάντως, επισκεφτείτε την σχετική σελίδα ενώ ακόμα περισσότερες πληροφορίες μπορείτε να βρείτε στο site της Wiring καθώς και στο εγχειρίδιο της βιβλιοθήκης AVR Libc.

2.9 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.



2.9.1 συνδεση Arduino με θσβ καλοδιο

2. Περιβάλλον Arduino

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

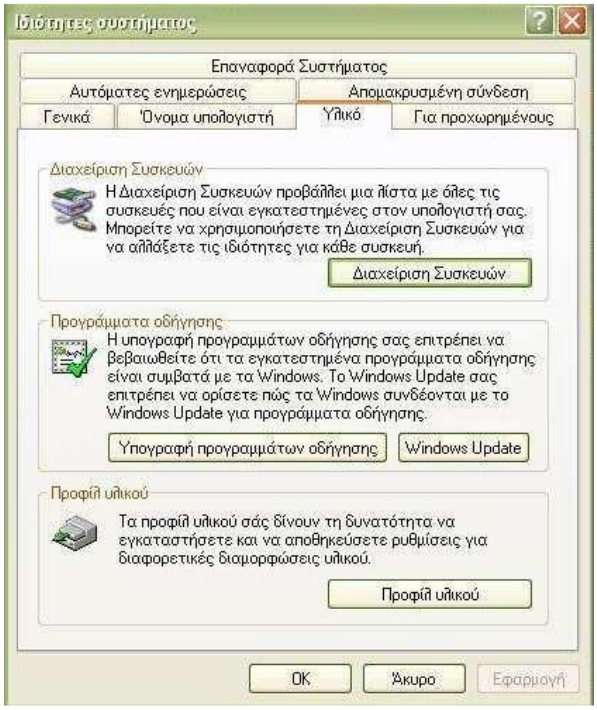
3. Σύνδεση της πλακέτα στον υπολογιστή

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

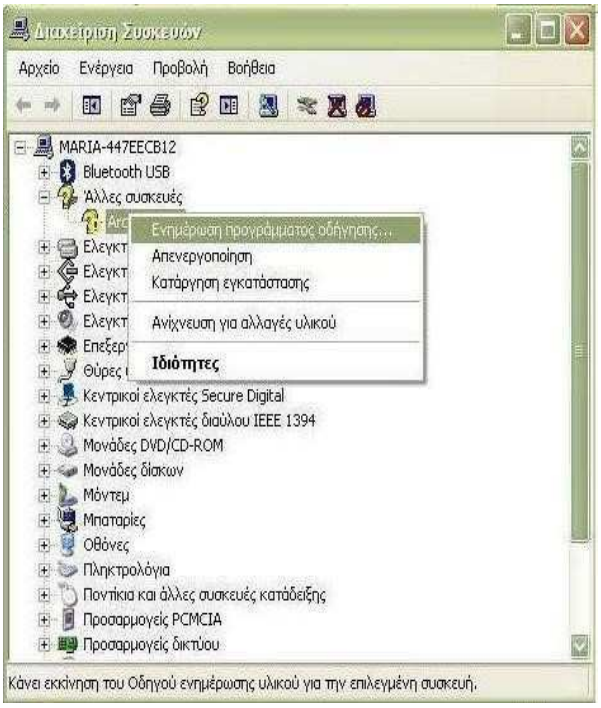
4. Εγκατάσταση του προγράμματος

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

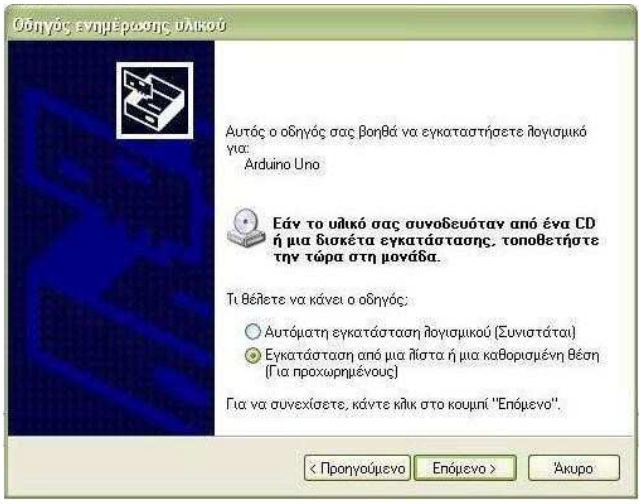
+ Από τον Πίνακα Ελέγχου, μεταβαίνουμε στο «Σύστημα» και ακολούθως «Υλικό» και ανοίγουμε τη διαχείριση συσκευών.



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

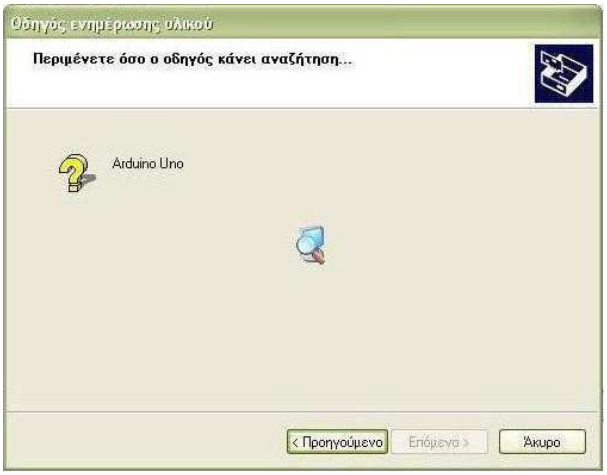


+ Ξεκινάει εγκατάσταση λογισμικού για το Arduino.

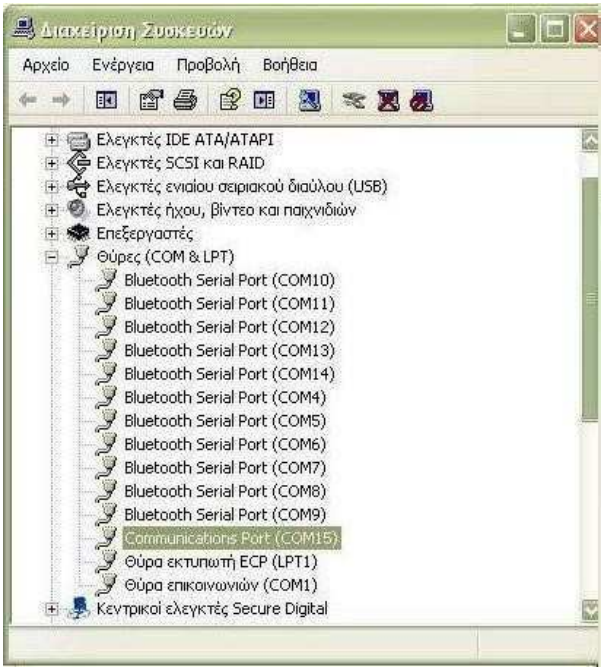


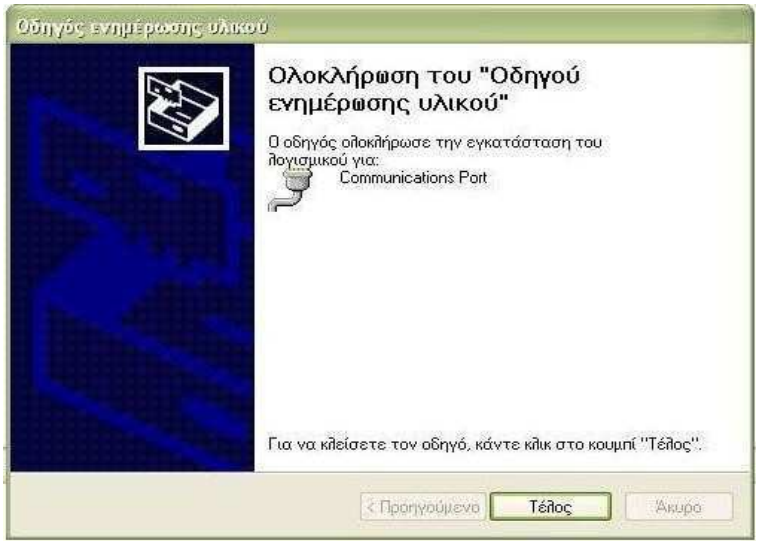
+ Κάνουμε εγκατάσταση τα drivers στον υπολογιστή μας.

+ 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.



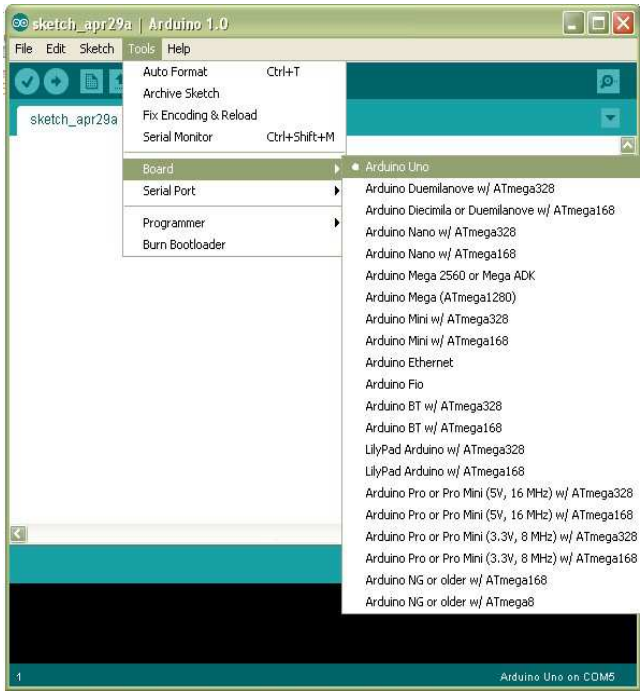


5. Έναρξη της εφαρμογής Arduino

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.

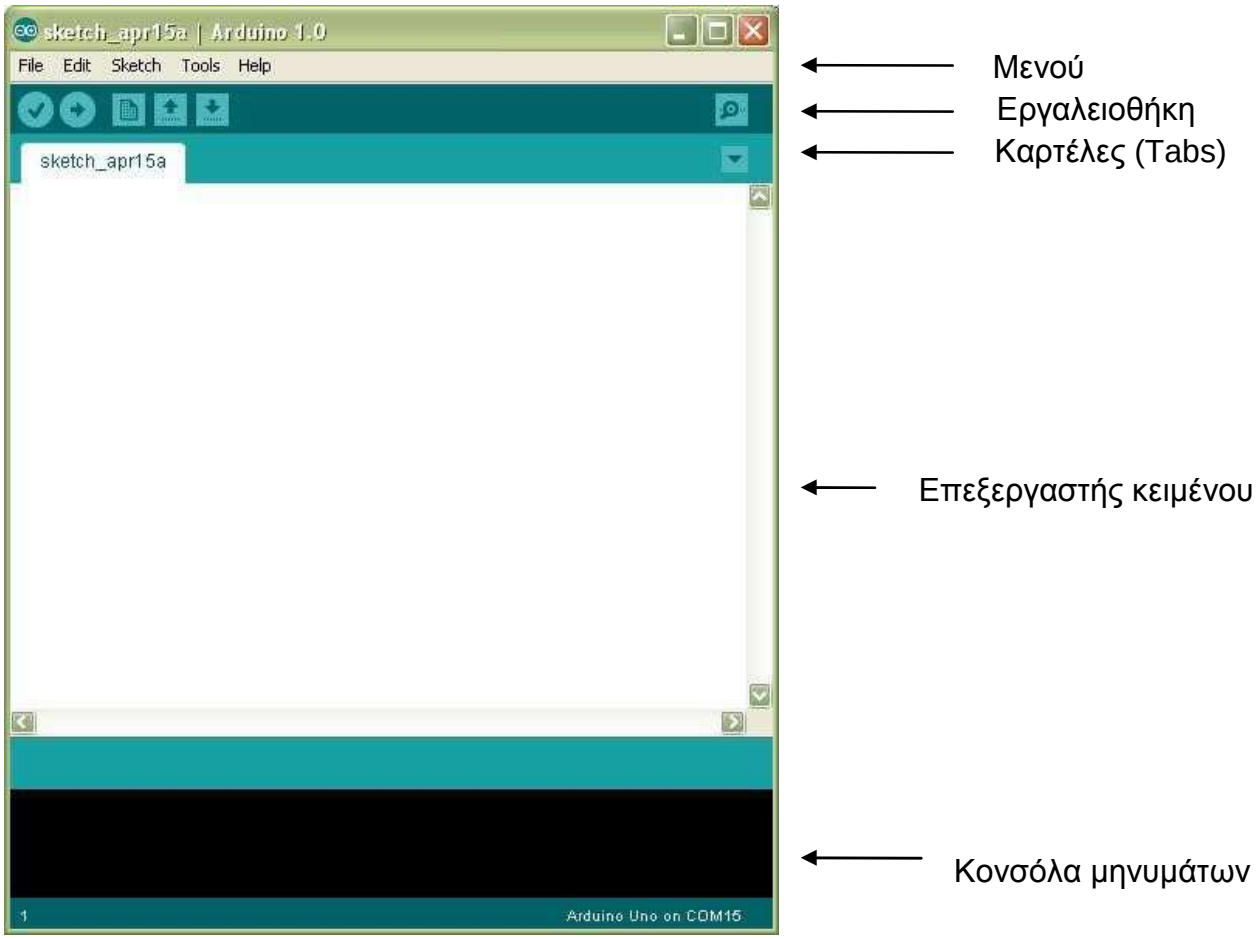


2.10 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino

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 2.10.1 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

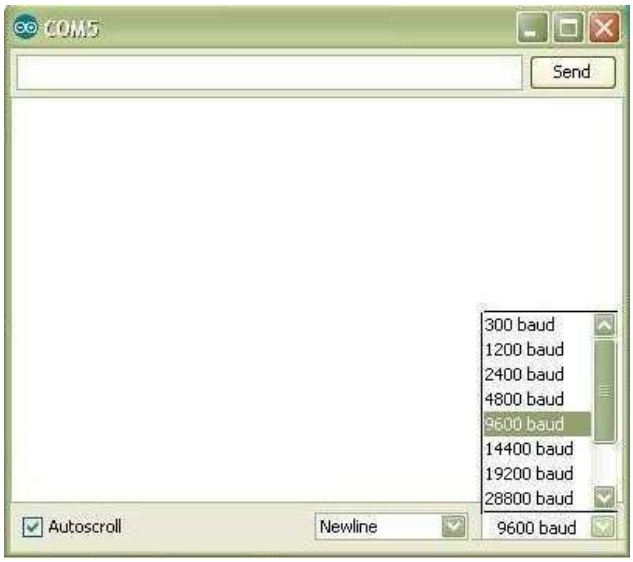
 Open (Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα sketch, κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο.

 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 επιλεχθεί στο προγραμματισμό του Arduino με το Serial.begin().



Εικόνα 2.4: Serial Monitor

2.12 Η Δομή το 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 εξόδου)
- digitalRead() (διαβάζει από μία ψηφιακή επαφή εισόδου)

Συναρτήσεις αναλογικής εισόδου και εξόδου:

- analogReference() (ορίζει την τάση αναλογικής αναφοράς)
- 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:

- 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 υπερχείλιση ή υποχείλιση ορίων)
- map() (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- 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 μίας 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, απλά 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).

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 μέθοδο για την απλοποίηση δηλώσεων εκτύπωσης.

Για την συγγραφή των παραπάνω υποκεφαλαίων πολύ χρήσιμες ήταν οι ιστοσελίδες [1], [2], [3], [4] και [5].

ΚΕΦΑΛΑΙΟ 3

Πρακτικό Μέρος

- Επίσκεψη και περιήγηση στην Ιστοσελίδα www.arduino.cc .
- Κατέβασμα και εγκατάσταση εφαρμογής ARDUINO IDE.
- Επικοινωνία εφαρμογής ARDUINO IDE και της πλακέτας ARDUINO UNO.

3.1 Απαραίτητα Υλικά

Τα απαραίτητα υλικά που θα χρειαστούμε στη συνέχεια για την ολοκλήρωση του πρακτικού μέρους είναι τα παρακάτω:

3.1.1 Arduino Uno x1

3.1.2. BreadBroad x1

3.13. USB καλώδια πλατφόρμων Arduino

3.1.4. Led 5mm κόκκινο x2

3.1.5. Arduino rip sensor x 1

3.1.6. Arduino speaker x2

3.1.7. Αντιστάσεις x2

3.1.8 door sensor x4

3.2 Συνδεσμολογία και μέρη της κατασκευής

Αρχικά βρήκαμε μια μακέτα, ένα χάρτινο κουτί το οποίο βάφτηκε κατάλληλα. πάνω σε αυτό σχημάτιστηκαν 4 εσοχές για να γίνουν οι πόρτες. Πάνω σε αυτές της πόρτες τοποθετήθηκαν 4 αισθητήρες η αλλιώς παγίδες για να μπορούμε να ελέγχουμε την πιθανή παραβίαση της πόρτας. Οι αισθητήρες τοποθετήθηκαν σε σειρά και οδηγήθηκαν στο bread board και από εκεί στην θύρα 4 του Arduino.

Έπειτα τοποθέτησα των ανιχνευτή κινήσεις τον στην κορυφή της κατασκευής και με ελάχιστη κλήση για να ανίχνευση οποιαδήποτε κίνηση. Οδηγούμε τα καλώδια του στο bread board και από εκεί στο Arduino στην θύρα.

3.3 Εφαρμογή

Σύστημα συναγερμού σε αυτοκίνητο

Η εφαρμογή θα προσομοιώνει τη λειτουργία ενός αυτοκινήτου με 4 πόρτες και αντίστοιχα 4 door sensors , 1 ανιχνευτή κινήσεις δυο LED (κόκκινα) και δυο ηχεία. Όταν ανιχνευτή οποιαδήποτε κίνηση είτε άνοιγμα της πόρτας είτε ανιχνευτή κάποια κινήσει τα LED αρχίζουν να ανάβουν και τα ηχεία να παίζουν.

3.4 Η εφαρμογή με μια ματιά

```
int ledPin2 = 12;       // διαλέγουμε είσοδο για το LED για το αισθητήρα στις 4 πόρτες

int ledPin = 13;        // διαλέγουμε είσοδο για το LED για το αισθητήρα τον ανιχνευτή κινήσεις

int inputPin = 2;       // διαλέγουμε είσοδο για το  για το PIR sensor

int pirState = LOW;     // ξεκινάμε, υποθέτοντας ότι δεν ανιχνευθεί κίνηση

int val = 0;            // μεταβλητή για την ανάγνωση της κατάστασης pin

int speaker = 10;       // διαλέγουμε είσοδο για το ήχο

int switchPin = 4;      // διαλέγουμε είσοδο για το  door sensor

int speaker2 = 11;      // διαλέγουμε είσοδο για το ηχείο  για τον door sensor


void setup() {

pinMode(ledPin, OUTPUT);            // Κηρύσσω LED ως έξοδο

pinMode(inputPin, INPUT);           // δηλώνουν είσοδο

pinMode(speaker, OUTPUT);           // Κηρύσσω το ηχείο  ως έξοδο

pinMode(speaker2, OUTPUT);          // Κηρύσσω ηχείο ως έξοδο

Serial.begin(9600);                  // δηλώνουν είσοδο
pinMode(switchPin, INPUT);           // δηλώνουν είσοδο
digitalWrite(switchPin, HIGH);      // Δίνω τι μέγιστη τιμή στην είσοδο του door sensor


}

void loop(){

if(digitalRead(switchPin) == LOW){          // Ελέγξτε αν η είσοδος είναι LOW

digitalWrite(speaker2, LOW);                // σβήσε το ηχείο
```

```
digitalWrite(ledPin2, LOW);                    //σβήσε το led
}
else{                                          // αλλιώς
digitalWrite(speaker2, HIGH);                  // άναψε το ηχείο
digitalWrite(ledPin2, HIGH) );                 // άναψε το led
Serial.println("door open!");                  // να μας εμφανίσει ότι η πορτα είναι ανιχτη
}
val = digitalRead(inputPin);    // διάβασε την τιμή του input
if (val == HIGH) {                             // ελέγξτε αν η είσοδος είναι HIGH
digitalWrite(ledPin, HIGH);     // άναψε το LED
digitalWrite(speaker, HIGH)  ; // παίζει  τό ηχείο
delay(150);
if (pirState == LOW) {                         // ελέγξτε αν η είσοδος είναι LOW
                                               //έχει μόλις τεθεί σε λειτουργία
Serial.println("Motion detected!");// // να μας εμφανίσει ότι η κίνηση ανιχνευτικέ
pirState = HIGH;
}
  } else {
digitalWrite(ledPin, LOW);      // σβήσε το LED
 digitalWrite(speaker, LOW );  // σβήσε το ηχείο
delay(300);
if (pirState == HIGH){                  //  έχει μόλις απενεργοποιηθεί
Serial.println("Motion ended!"); // να μας εμφανίσει ότι η κίνηση σταμάτησε
pirState = LOW;
   }
```

3.5 Ανάλυση κυρίων σημείων του παραπάνω προγράμματος

int ledPin2 = 12; // δηλώνουμε την έξοδο του σήματος που θα καταλήξει στο led εάν έχει γίνει παραβιάσει μιας από της 4 πόρτες.

int ledPin = 13; // δηλώνουμε την έξοδο του σήματος που θα καταλήξει στο led εάν έχει γίνει παραβιάσει στο χώρο δηλαδή ανιχνευτή κάποια κίνηση .

int inputPin = 2; // Πρέπει να δηλώσουμε μια είσοδο πάνω στο Arduino για να μπορεί να λαμβάνει το σήμα .Το σήμα αυτό έρχεται απευθείας από την μια άκρη του ανιχνευτεί κινήσεις που βρίσκεται στην κορυφή της κατασκευής. Ο ανιχνευτής κινήσεις η RIP έχει τρεις ακροδέκτες ο ένας είναι το σήμα που όπως είπαμε καταλήγει στην είσοδο input 2 η άλλη είναι η τροφοδοσία 5 v ενώ η τελευταία καταλήγει στη γείωση

int pirState = LOW; // Ξεκινάμε, υποθέτοντας ότι δεν ανιχνευθεί κίνηση γι' αυτό δηλώνομε την είσοδο στον ανιχνευτή κινήσεις ως LOW

int val = 0; // Δηλώνουμε ότι val θα έχει 0 τιμή γιατί θα μας χρησιμεύσει στη καταμετρήσει παρακάτω.

int speaker = 10; // Δηλώνουμε είσοδο για το σήμα του ιχιου εάν έχει γίνει παραβιάσει μιας από της 4 πόρτες.

int switchPin = 4; // Πρέπει να δηλώσουμε μια είσοδο πάνω στο Arduino για να μπορεί να λαμβάνει το σήμα .Το σήμα αυτό έρχεται απευθείας από την μια άκρη των μαγνητικών επαφών που βρίσκονται πάνω στις 4 πόρτες ενώ η άλλη άκρη καταλήγει στην γείωση.

int speaker2 = 11; // Δηλώνουμε είσοδο για το σήμα του ιχιου εάν έχει γίνει παραβιάσει στο χώρο δηλαδή ανιχνευτή κάποια κίνηση

void setup() {

ΣΕ αυτό το σημιο δηλονουμε της εισοδους και της εξοδους του πρωγραματος

```
pinMode(ledPin, OUTPUT);            // Κηρύσσω LED ως έξοδο

pinMode(inputPin, INPUT);           // δηλώνουν είσοδο

pinMode(speaker, OUTPUT);           // Κηρύσσω το ηχείο  ως έξοδο

pinMode(speaker2, OUTPUT);          // Κηρύσσω ηχείο ως έξοδο

Serial.begin(9600);
pinMode(switchPin, INPUT);             // δηλώνουν είσοδο
digitalWrite(switchPin, HIGH);      // Δίνω τι μέγιστη τιμή στην είσοδο του door
sensor


}
```

```
void loop(){

if(digitalRead(switchPin) == LOW){          // Ελέγξτε αν η είσοδος είναι LOW

digitalWrite(speaker2, LOW);                // σβήσε το ηχείο

digitalWrite(ledPin2, LOW);                 //σβήσε το led
```

Σε αυτό το κώματι το πρόγραμμα ελέγχει την τιμή του switchPin όσο αυτή παραμένει στο Low τότε δεν έχουμε παραβίαση της πόρτα αρά δίνουμε εντολή στο πρόγραμμα να μην δώσει σήμα ούτε στο ηχείο αλλά ούτε και στο led

```
}


else{                                       // αλλιώς
```

```
digitalWrite(speaker2, HIGH);                // άναψε το ηχείο

digitalWrite(ledPin2, HIGH) );                // άναψε το led

Serial.println("door open!");                //μας γραφεί στην οθόνη ότι η πόρτα είναι
ανοιχτή
```

Ειδάλλως στην περίπτωση που switchPin πάρει τιμή high τότε έχουμε παραβίαση της πόρτας . Αρά δίνουμε εντολή στης εξόδους speaker2 και ledPin2 να πάρουν την τιμή HIGH .Δηλαδή να ενεργοποιηθούν και το ηχείο και το led. Επιπλέον αν ανοίξουμε το παράθυρο Tools > Serial Monitor θα μπορούμε και από εκεί να ενημερωθούμε αν παραβιάστηκε η πόρτα.

```
}

val = digitalRead(inputPin);    // διάβασε την τιμή του input

if (val == HIGH) {              // ελέγξτε αν η είσοδος είναι HIGH

digitalWrite(ledPin, HIGH);     // άναψε το LED

digitalWrite(speaker, HIGH)  ; // παίζει  τό ηχείο

if (pirState == LOW) {          // ελέγξτε αν η είσοδος είναι LOW

                                //έχει μόλις τεθεί σε λειτουργία

Serial.println("Motion detected!");// // να μας εμφανίσει ότι η κίνηση ανιχνευτικέ

pirState = HIGH;
```

Σε αυτήν την φάση ελέγχουμε την είσοδο από τον ανιχνευτή κίνησης .Αρχικά κάθε φορά στην αρχή της επανάληψης θα διαβάζει την τιμή val οπού εμείς στην αρχή την είχαμε δηλώσει μηδέν .Εάν λιπών έχουμε ανίχνευση κάμπιας κίνησης τότε δίνουμε εντολή στο led να ανοίξει και στο ηχείο να ξεκινήσει να εκπέμπει. Επίσης μπορούμε να ανοίγοντας το παράθυρο Tools > Serial Monitor να αναγράφεται ποτέ ανιχνεύεται κίνηση

```
}


  } else {
digitalWrite(ledPin, LOW);      // σβήσε το LED
 digitalWrite(speaker, LOW );  // σβήσε το ηχείο
delay(300);
if (pirState == HIGH){                  //  έχει μόλις απενεργοποιηθεί
Serial.println("Motion ended!"); // να μας εμφανίσει ότι η κίνηση σταμάτησε
pirState = LOW;
    }
  }
}
```

Στην τελευταία φάση του προγράμματος βλέπουμε την περίπτωση να μην ανιχνευτή κάποια κίνηση αρά να μην πάρει κάποια τιμή το val οπότε το ηχιο και το led δεν θα ενεργοποιηθούν και το παράθυρο Tools > Serial Monitor θα μας γράψει ότι δεν ανιχνευτικέ κίνηση. Τελος ξαναδηλονουμε την μεταβλητη prostate σαν low

3.6 ΕΠΕΚΤΑΣΗ

Στην καθημερινότητα μπορεί η συγκεκριμένη εργασία να βρει πολλές εφαρμογές
Το σύστημα της ασκήσεις θα μπορέσουμε να το χρησιμοποιήσουμε με ελάχιστες τροποποίησης σε οτιδήποτε χώρο θέλουμε . Είτε σε κάποιο αμάξι με τροφοδότηση του Arduino και του στησίματος με μια τάση 5v. Η οποία θα γίνει μεσώ μετασχηματιστή. Είτε σε κάποιο σπίτι σαν σύστημα συναγερμού .Τοποθετώντας τις 4 η παραπάνω μαγνητικές επαφές σε πόρτες και παράθυρα έτσι ώστε όταν παραβιαστεί κάποιο απτής παραπάνω επαφές να ενεργοποιήσει τον συναγερμό . Επιπλέον ο ανιχνευτής κινήσεις μπορεί να τοποθετηθεί σε περίοπτή θέση μέσα στο σπίτι έτσι ώστε όταν ανιχνευθεί κίνηση να θέσει σε λειτουργία τον συναγερμό .
Επίσης τα LED στο ardunino μπορούν να αντικατασταθούν με μεγαλύτερης ισχύεις και φωτεινότητάς LED.Όπως επίσης και το ειχιο από κάποιο άλλο μεγαλύτερης ισχύεις για την ασφάλεια ενός σπιτιού.

Βιβλιογραφία – Πηγές πληροφοριών

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.

[C] Παναγιώτης Παπάζογλου. Ανάπτυξη Εφαρμογών με το Arduino ένας πλήρης οδηγός για αρχάριους και προχωρημένους.

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