

Αισθητήρες Μέτρησης και Ελέγχου

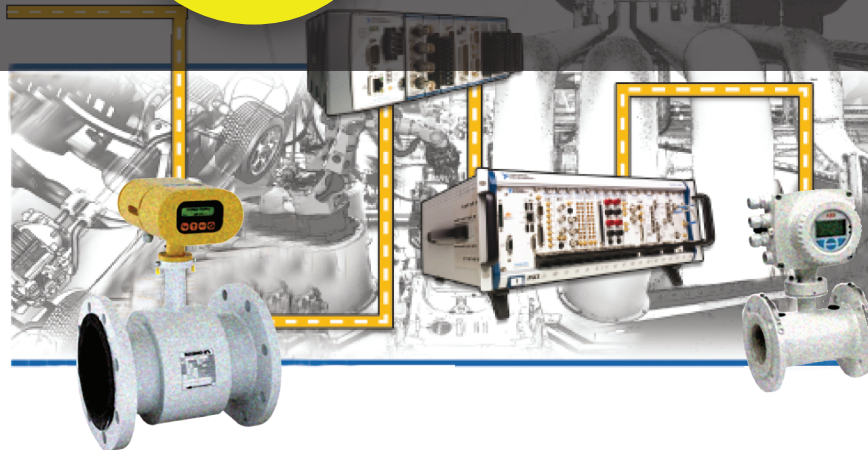


συνδεδεμένο περιεχόμενο

ΤΕΧΝΟΛΟΓΙΑ
ΜΕΤΡΗΣΗΣ

QR
biblio

Σχεδίαση &
Συστημάτων Μετρήσεων
Συστημάτων Συλλογής
Δεδομένων



2η Έκδοση
Νέα βελτιωμένη Έκδοση

Κωνσταντίνος Καλοβρέκτης
Νικόλαος Κατέβας



ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ



Ανάπτυξη εφαρμογών μετρήσεων και ελέγχου με Arduino

36.1 Εισαγωγή

Μέχρι πριν λίγα χρόνια η διαδικασία ανάπτυξης ενσωματωμένων συστημάτων μετρήσεων και ελέγχου απαιτούσε από τον μηχανικό να γνωρίζει σε βάθος τη δομή των μικροελεγκτών, τους καταχωρητές, τον τρόπο λειτουργίας των περιφερειακών, κ.τ.λ.. Η εκπαίδευση πάνω σε τέτοιου είδους συστήματα μετρήσεων απαιτούσε την αγορά εκπαιδευτικού εξοπλισμού με αρκετά υψηλό κόστος και την ενασχόληση με το αντικείμενο για μεγάλα χρονικά διαστήματα ώστε να αποκτηθεί η απαραίτητη εμπειρία.

Η καλή έως άριστη γνώση προγραμματισμού ήταν μέχρι σήμερα απαραίτητο στοιχείο για την ανάπτυξη ενσωματωμένων συστημάτων μετρήσεων. Οι εταιρίες κατασκευής μικροελεγκτών δημιουργούσαν κατά καιρούς εκπαιδευτικές πλατφόρμες οι οποίες παρείχαν ένα μεγάλο σύνολο βιβλιοθηκών για χρήση σε ενσωματωμένα συστήματα που διευκόλυναν τους προγραμματιστές και μείωναν τον χρόνο προγραμματισμού μίας εφαρμογής καθώς και το χρόνο εξαγωγής ενός συστήματος στην αγορά.

Μαζί με τον εξοπλισμό ο μηχανικός έπρεπε να εξοικειωθεί με τις βιβλιοθήκες προγραμματισμού που περιλαμβάνονταν σε αυτόν. Πολλές φορές μία εταιρία παραγωγής εκπαιδευτικού εξοπλισμού για ενσωματωμένα συστήματα δεν ικανοποιούσε τις ανάγκες ενός προγραμματιστή ο οποίος αναγκαζόταν να καταφύγει σε μία άλλη εταιρία που χρησιμοποιούσε συνήθως κάποιον άλλον μεταγλωττιστή (compiler) με τον οποίο και έπρεπε να εξοικειωθεί. Όλα αυτά είναι παράγοντες που συνετέλεσαν στην ανάπτυξη της πλατφόρμας Arduino την οποία περιγράφουμε στη συνέχεια.

36.2 Η πλατφόρμα Arduino

Η πλατφόρμα Arduino είναι μία εκπαιδευτική πλατφόρμα με αντικείμενο τον προγραμματισμό ενσωματωμένων συστημάτων που χαρακτηρίζεται από ένα μεγάλο βαθμό ευκολίας και χρηστικότητας. Αυτή η πλατφόρμα υποστηρίζει ένα σύνολο έτοιμων εκπαιδευτικών μονάδων (πλακέτες) χαμηλού κόστους που καλύπτουν ένα μεγάλο σύνολο εφαρμογών στον κόσμο των ενσωματωμένων συστημάτων. Η πλατφόρμα Arduino χρησιμοποιεί τη γλώσσα προγραμματισμού Wiring C που βασίζεται στην γλώσσα προγραμματισμού C++. Η γλώσσα αυτή είναι κατάλληλα σχεδιασμένη ώστε να παρέχει ευκολία στο χρήστη

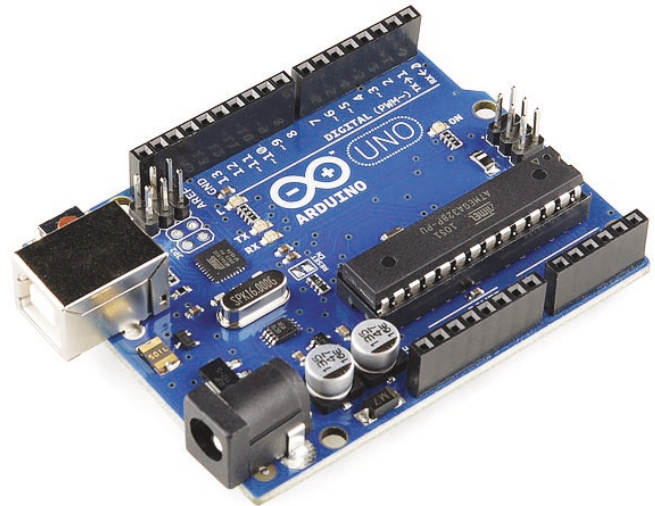
καθώς χρησιμοποιεί έτοιμες βιβλιοθήκες που μειώνουν στο ελάχιστο την διαδικασία οδήγησης περιφερειακών μονάδων σε ένα ενσωματωμένο σύστημα. Ο χρήστης χρειάζεται βασικές γνώσεις προγραμματισμού και λίγο χρόνο να εξοικειωθεί με τις βιβλιοθήκες και τη χρήση τους ώστε να προγραμματίσει ένα ενσωματωμένο σύστημα.

36.3 Ξεκινώντας με την πλατφόρμα Arduino

Για να ξεκινήσει κανείς με την πλατφόρμα Arduino χρειάζεται να αγοράσει μία από τις εκπαιδευτικές μονάδες (πλακέτες) που διατίθενται στο εμπόριο ανάλογα με την εφαρμογή που θέλει να υλοποιήσει. Η πρώτη εκπαιδευτική μονάδα που θα πρέπει να προμηθευτεί κανείς για να ξεκινήσει να εργάζεται με την πλατφόρμα Arduino είναι η εκπαιδευτική μονάδα Arduino UNO που απεικονίζεται στο Σχήμα 36.1.

Η εκπαιδευτική μονάδα Arduino UNO χρησιμοποιεί τον μικροελεγκτή ATmega328 της εταιρίας ATMEL ο οποίος λειτουργεί σε συχνότητα 16MHz. Πάνω στην εκπαιδευτική πλακέτα υπάρχουν όλοι οι ακροδέκτες που υποστηρίζει ο μικροελεγκτής οι οποίοι οδηγούνται κατάλληλα σε θηλυκούς συνδέσμους ώστε να μπορεί εύκολα να έχει πρόσβαση ο χρήστης για διάφορες συνδεσμολογίες.

Εκτός από τους ακροδέκτες εισόδου/εξόδου η πλακέτα Arduino UNO διαθέτει και ακροδέκτες για τον προγραμματισμό του ολοκληρωμένου κυκλώματος μέσω προγραμματίστριας συσκευής καθώς και θύρα USB για την φόρτωση των προγραμμάτων που γράφουμε μέσω του περιβάλλοντος προγραμματισμού Arduino IDE που θα εξετάσουμε στη συνέχεια. Τέλος, στην περίπτωση που δεν χρησιμοποιούμε τροφοδοσία μέσω της θύρας USB, η πλακέτα διαθέτει υποδοχή για την τροφοδοσία της συσκευής από εξωτερικό τροφοδοτικό. Για να ξεκινήσει κανείς με την πλατφόρμα Arduino θα πρέπει να προετοιμάσει κατάλληλα την εκπαιδευτική πλακέτα που έχει επιλέξει και να εκτελέσει τα ακόλουθα βήματα:



Σχήμα 36.1: Η πλατφόρμα Arduino UNO.

Βήμα 1: Κατεβάζουμε από την επίσημη ιστοσελίδα το περιβάλλον προγραμματισμού Arduino IDE (<http://arduino.cc/en/main/software>). Προτείνεται η επιλογή του αρχείου. zip.

Βήμα 2: Χρησιμοποιούμε το καλώδιο USB και συνδέουμε την εκπαιδευτική πλακέτα με τον υπολογιστή μας. Εδώ θα πρέπει να προσέξουμε γιατί ορισμένες πλακέτες απαιτούν ιδιαίτερες ρυθμίσεις στη συνδεσμολογία με τον υπολογιστή όσον αφορά την τροφοδοσία. Κάποιες πλακέτες απαιτούν εξωτερική τροφοδοσία και κάποιες πρέπει να ρυθμιστούν κατάλληλα ώστε να τροφοδοτούνται από την USB. Για περισσότερες πληροφορίες σχετικά με την συνδεσμολογία ανατρέχουμε στην επίσημη ιστοσελίδα της πλατφόρμας Arduino. Η πλακέτα Arduino UNO που μελετάμε δεν απαιτεί κάποια ιδιαίτερη ρύθμιση.

Βήμα 3: Πρέπει να εγκαταστήσουμε τους οδηγούς (drivers) της πλακέτας Arduino που έχουμε συνδέσει. Με την σύνδεση στον υπολογιστή μας εμφανίζεται ο οδηγός εγκατάστασης νέων συσκευών. Εδώ πρέπει να προσεχθεί ιδιαίτερα αυτή η διαδικασία ανάλογα με το λειτουργικό σύστημα το οποίο διαθέτουμε. Για περισσότερες πληροφορίες σχετικά με τα λειτουργικά συστήματα και τους οδηγούς συσκευών Arduino μπορούμε να ανατρέξουμε στην επίσημη ιστοσελίδα της πλατφόρμας Arduino. Για την πλακέτα Arduino UNO και για λειτουργικό σύστημα Windows 7 θα πρέπει να περιμένουμε έως ότου ο οδηγός εγκατάστασης νέων συσκευών ολοκληρώσει τη διαδικασία ανίχνευσης. Θα παρατηρήσουμε πως δεν βρίσκει από μόνος του κάποιον συμβατό οδηγό για την συσκευή Arduino. Έπειτα από αυτή τη διαδικασία θα πρέπει να ανοίξουμε τον διαχειριστή συσκευών (Device Manager) των Windows που βρίσκεται στον πίνακα ελέγχου (Control Panel) κάτω από την επιλογή System and Security. Μέσα από τη λίστα με τις συσκευές κάτω από τις θύρες επικοινωνίας (COM & LPT) θα πρέπει να υπάρχει μία ανοιχτή θύρα με όνομα “Arduino Uno (COMxx)”. Στην περίπτωση που κάτω από τις θύρες COM & LPT δεν υπάρχει αυτή η θύρα μπορούμε να κοιτάξουμε κάτω από την επιλογή Other Devices (Άλλες

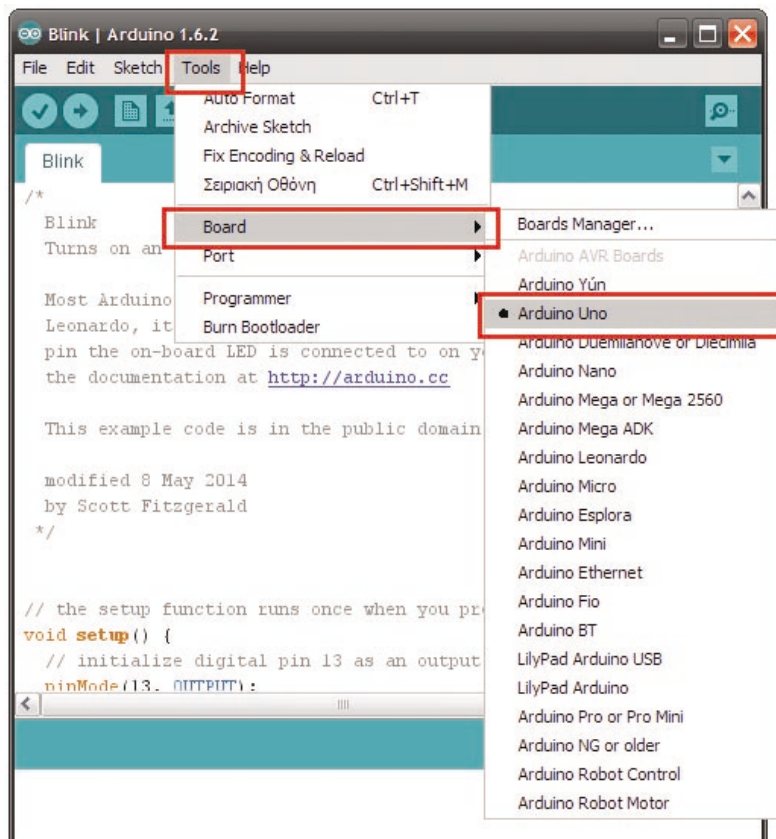
Συσκευές) για μία άγνωστη συσκευή (Unknown Device). Και στις δύο περιπτώσεις κάνουμε δεξί κλικ πάνω σε αυτή την επιλογή και επιλέγουμε “Update Driver Software” (Ενημέρωση οδηγών) από το αναδυόμενο μενού που εμφανίζεται. Έπειτα επιλέγουμε αναζήτηση του υπολογιστή μας (Browse my computer for Driver software) για να βρούμε τον οδηγό της συσκευής. Επιλέγουμε το αρχείο arduino.inf μέσα από το φάκελο Arduino\drivers που βρίσκεται στο φάκελο εγκατάστασης της πλατφόρμας Arduino που κατεβάσαμε στο βήμα 1. Η εγκατάσταση του οδηγού της συσκευής θα ολοκληρωθεί αυτόματα μετά από αυτή την ενέργεια.

Βήμα 4: Επόμενο βήμα είναι η εκτέλεση του περιβάλλοντος Arduino IDE ώστε να αποκτήσουμε μία πρώτη επαφή με την πλατφόρμα Arduino. Ανοίγουμε το περιβάλλον Arduino IDE από το φάκελο εγκατάστασης. Από το μενού επιλέγουμε File > Examples > Basics > Blink. Με αυτή την ενέργεια ανοίγουμε ένα απλό παράδειγμα εφαρμογής η οποία έχει ως λειτουργία να αναβοσβήνει το LED στον ακροδέκτη 13 της πλακέτας Arduino UNO. Στη συνέχεια απεικονίζεται ο κώδικας της εφαρμογής στο περιβάλλον Arduino UNO.

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Βήμα 5: Αφού ανοίξουμε το παράδειγμα πρέπει να επιλέξουμε μέσα από το περιβάλλον προγραμματισμού την πλακέτα Arduino που χρησιμοποιούμε. Μέσα από το μενού επιλογών επιλέγουμε Tools\Board\Arduino Uno για να χρησιμοποιήσουμε την πλακέτα Arduino UNO που έχουμε συνδέσει με τον υπολογιστή μας. Πρέπει να επιλέξουμε και τη σειριακή θύρα που χρησιμοποιούμε για να επικοινωνήσουμε με τον υπολογιστή. Από το μενού Port μπορούμε να επιλέξουμε μία διαθέσιμη σειριακή θύρα COM.



Βήμα 6: Επόμενο βήμα είναι η φόρτωση του προγράμματος στη συσκευή Arduino. Η διαδικασία αυτή είναι πολύ απλή και πραγματοποιείται με το πάτημα του πλήκτρου Upload.



Αν η διαδικασία φόρτωσης του προγράμματος πραγματοποιηθεί με επιτυχία εμφανίζεται ένα μήνυμα “Done Uploading” στη μπάρα κατάστασης.

Βήμα 7: Μετά τη φόρτωση του προγράμματος πραγματοποιείται εκκίνηση της εφαρμογής και εκτελείται από την πλακέτα Arduino. Το LED στον ακροδέκτη 13 της πλακέτας Arduino UNO θα αναβοσβήνει κάθε 1 δευτερόλεπτο.

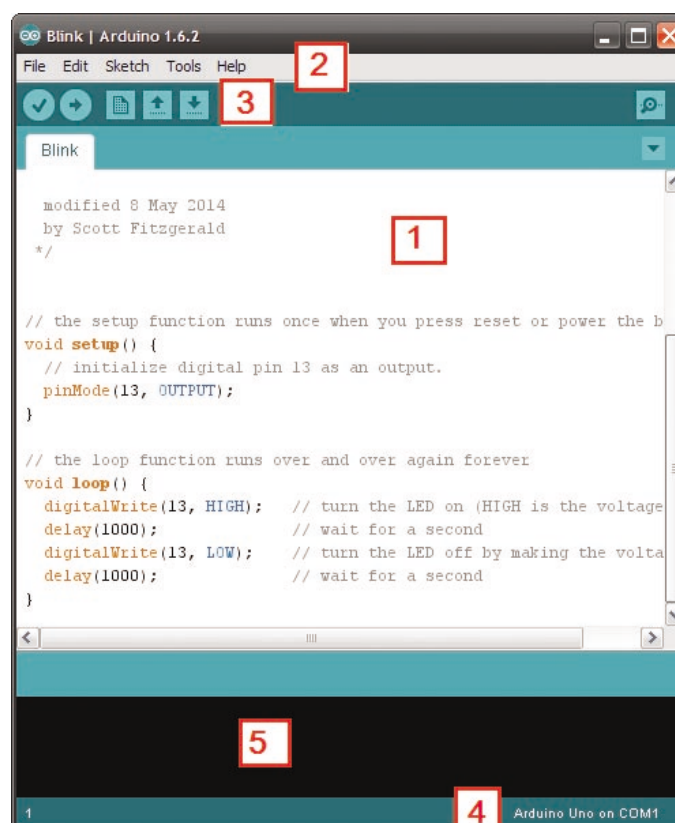
Την διαδικασία που περιγράψαμε πιο πάνω θα πρέπει να ακολουθεί κανείς για να προετοιμάσει μία πλακέτα Arduino και για να φορτώσει ένα πρόγραμμα σε αυτή.

36.4 Το περιβάλλον προγραμματισμού Arduino IDE

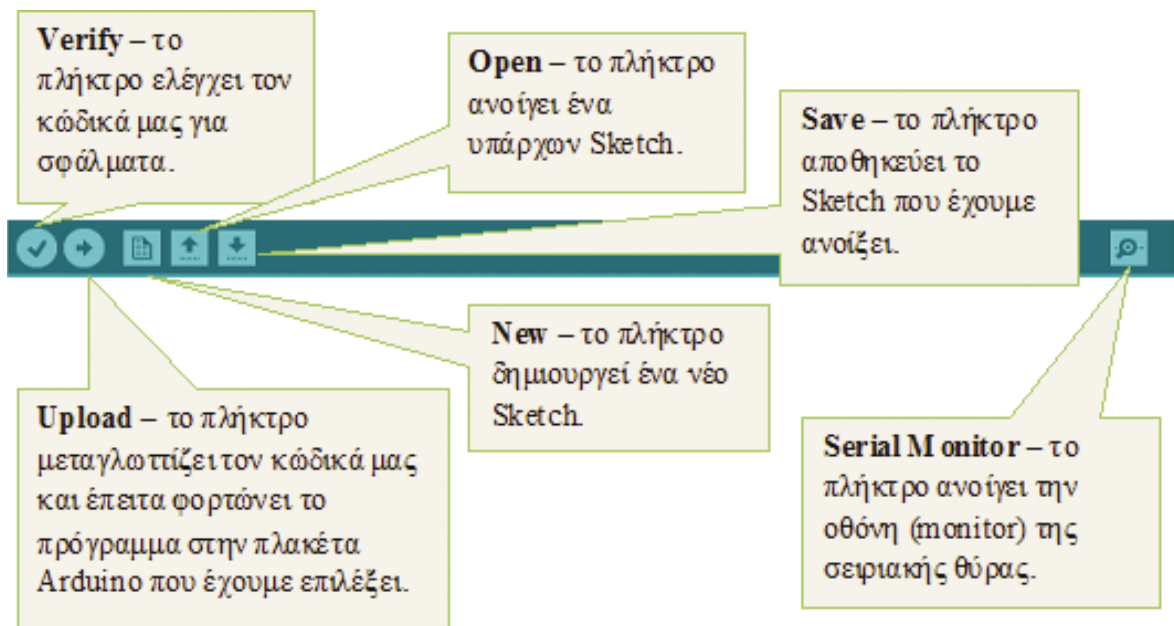
Το περιβάλλον προγραμματισμού Arduino IDE (Σχ. 36.2) διαθέτει:

- έναν επεξεργαστή κειμένου (1)
- ένα μενού επιλογών (2)
- εικονίδια εργαλείων (3)
- την μπάρα κατάστασης (4)
- μία περιοχή εμφάνισης ενημερωτικών μηνυμάτων (5)

Τα προγράμματα που γράφουμε με το περιβάλλον Arduino IDE καλούνται **Sketches** και είναι αρχεία κειμένου τα οποία αποθηκεύονται με κατάληξη *.ino. Το κώδικα τον εισάγουμε ή τον γράφουμε μέσα στον επεξεργαστή κειμένου του Arduino IDE ο οποίος υποστηρίζει λειτουργίες εξασφαλίσωσης του κώδικα. Οποιαδήποτε μηνύματα σχετικά με σφάλματα και διαδικασίες μεταγλώττισης εμφανίζονται στην περιοχή μηνυμάτων του Arduino IDE. Χρησιμοποιώντας την μπάρα με τα εικονίδια συντόμευσης εκτελούμε άμεσα τις σημαντικότερες λειτουργίες του περιβάλλοντος προγραμματισμού Arduino IDE τις οποίες περιγράφουμε στο Σχήμα 36.3.



Σχήμα 36.2: Το περιβάλλον προγραμματισμού Arduino IDE.

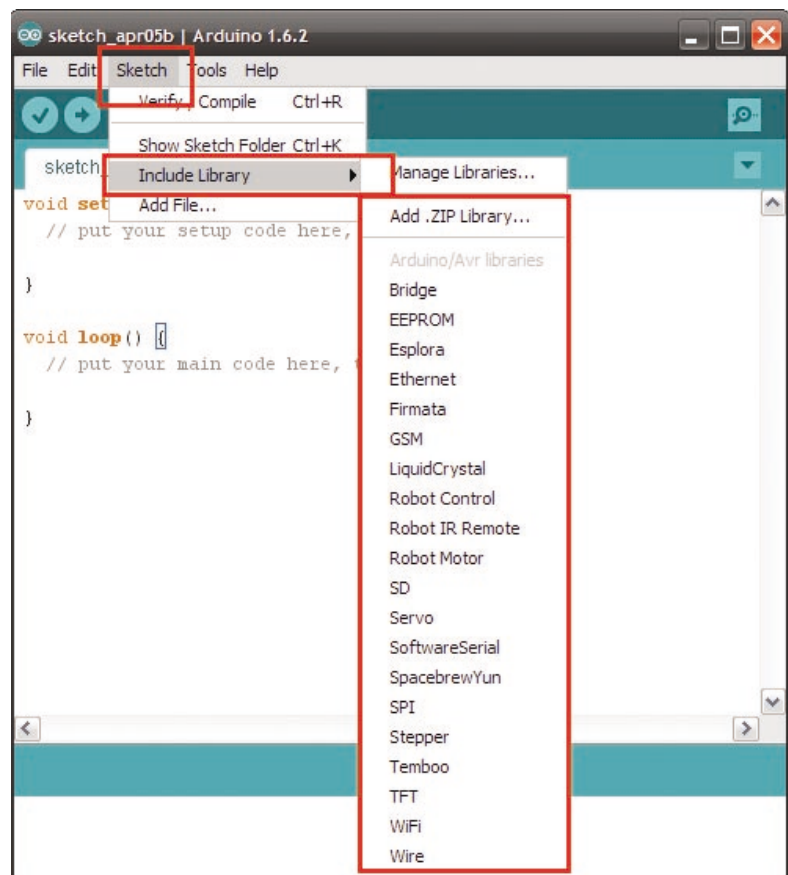


Σχήμα 36.3: Περιγραφή πλήκτρων εργαλείων.

36.5 Βιβλιοθήκες στο περιβάλλον Arduino

Ένα από τα ισχυρότερα εργαλεία του περιβάλλοντος προγραμματισμού Arduino είναι η υποστήριξη βιβλιοθηκών (Libraries). Οι βιβλιοθήκες παρέχουν επιπλέον λειτουργίες για χρήση στα Sketches που γράφουμε με το Arduino IDE. Συμπεριλαμβάνοντας μία βιβλιοθήκη σε ένα Sketch μπορούμε να προσθέσουμε λειτουργίες όπως υποστήριξη νέου υλικού (π.χ. Bluetooth, Wi-Fi, μνήμες, αισθητήρες) καθώς και υποστήριξη επεξεργασίας δεδομένων. Οι κατασκευαστές εκπαιδευτικού και άλλου εξοπλισμού που χρησιμοποιούν την πλατφόρμα Arduino σχεδιάζουν κατάλληλα το υλικό ώστε να προσαρμόζεται στις πλακέτες Arduino παρέχοντας παράλληλα και τις απαραίτητες βιβλιοθήκες για την οδήγησή τους.

Για να συμπεριλάβουμε μία βιβλιοθήκη μέσα στο Sketch που γράφουμε επιλέγουμε από το κεντρικό μενού μία από τις διαθέσιμες βιβλιοθήκες που εμφανίζονται στο Sketch → Import Library... Με την εγκατάσταση του περιβάλλοντος προγραμματισμού Arduino IDE εγκαθίστανται από προεπιλογή ένα σύνολο βιβλιοθηκών που υποστηρίζουν τα πρωτόκολλα USB, WiFi, SPI, GSM κ.α. ώστε να μπορούμε άμεσα, έχοντας προμηθευτεί το κατάλληλο υλικό (hardware), να υλοποιήσουμε εφαρμογές που υποστηρίζουν αυτά τα πρωτόκολλα επικοινωνίας.



Σχήμα 36.4: Κατάλογος βιβλιοθηκών.

Επίσης υπάρχουν εγκατεστημένες από προεπιλογή και βιβλιοθήκες για την οδήγηση οθονών TFT και LCD, βιβλιοθήκες για την οδήγηση σερβοκινητήρων και ρομποτικών συσκευών καθώς και βιβλιοθήκες για την οδήγηση αισθητήρων μέτρησης μέσω αναλογικών σε ψηφιακό μετατροπέων (Σχ. 36.4).

36.6 Μαθαίνοντας να προγραμματίζουμε με τον Arduino

Ο καλύτερος τρόπος να μάθει κανείς να προγραμματίζει στο περιβάλλον Arduino είναι να ακολουθεί τα παραδείγματα και τον οδηγό αναφοράς που υπάρχει στην κεντρική ιστοσελίδα της πλατφόρμας Arduino που απεικονίζεται στο Σχήμα 36.5.

<http://arduino.cc/en/Reference/HomePage>



Σχήμα 36.5: Ιστοσελίδα του Arduino.

Σε αυτή τη σελίδα υπάρχουν όλες οι πληροφορίες και αμέτρητα παραδείγματα που αν τα μελετήσει κανείς θα μπορέσει να αποκτήσει σε σύντομο χρονικό διάστημα πολύ ισχυρές γνώσεις και να υλοποιήσει τις δικές του εφαρμογές σύμφωνα με τις δικές του απαιτήσεις. Ποια είναι όμως η δομή ενός προγράμματος Arduino; Όπως και στις περισσότερες γλώσσες προγραμματισμού έτσι και σε αυτή που χρησιμοποιεί η πλατφόρμα Arduino υπάρχει το τμήμα δηλώσεων και συμπερίληψης βιβλιοθηκών. Το τμήμα αυτό τοποθετείται στην κορυφή του Sketch που γράφουμε κάθε φορά.

```
#include <LiquidCrystal.h>
int sensorPin = A0;
int ledPin = 13;
int sensorValue = 0;
```

Τμήμα δηλώσεων μεταβλητών και συμπερίληψης βιβλιοθηκών.

Αμέσως μετά το τμήμα δηλώσεων μεταβλητών και συμπερίληψης βιβλιοθηκών η πλατφόρμα Arduino χρησιμοποιεί μία συνάρτηση ρύθμισης και αρχικοποίησης των παραμέτρων του συστήματος η οποία καλείται `setup()`. Η συνάρτηση `setup()` καλείται μόνο μία φορά στην αρχή της εκτέλεσης ενός Sketch και μέσα σε αυτή ο προγραμματιστής αρχικοποιεί τις παραμέτρους της εφαρμογής του καθώς και τις παραμέτρους των βιβλιοθηκών που χρησιμοποιεί όπως: αρχικοποιήσεις μεταβλητών, βιβλιοθηκών και δηλώσεις pin εισόδου/εξόδου.

```
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

Η σύνταξη της συνάρτησης `setup()` με εντολή προς εκτέλεση.

Μετά την κλήση της συνάρτησης `setup()` η πλατφόρμα Arduino αντί για την συνάρτηση `main` που χρησιμοποιούν οι περισσότερες γλώσσες προγραμματισμού που βασίζονται στην C, χρησιμοποιεί τη συνάρτηση που καλείται `loop()`. Η συνάρτηση αυτή υλοποιεί έναν βρόχο επανάληψης μέσα στον οποίο επαναλαμβάνεται η διαδικασία εκτέλεσης του προγράμματος που γράφουμε. Ο προγραμματιστής τοποθετεί μέσα σε αυτή τις εντολές ελέγχου της πλακέτας Arduino.

```
void loop() {
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
}
```

Η σύνταξη της συνάρτησης `loop()` με ένα παράδειγμα εντολών προς εκτέλεση.

```
digitalWrite(ledPin, LOW);
delay(sensorValue);
}
```

Ένα ολοκληρωμένο παράδειγμα προγράμματος Arduino είναι το παράδειγμα Blink που φορτώσαμε σε προηγούμενη παράγραφο.

```
void setup() {
  pinMode(13, OUTPUT); // Αρχικοποίηση του ακροδέκτη 13 ως έξοδο.
}
void loop() {
  digitalWrite(13, HIGH); // LED ON.
  delay(1000);           // Καθυστέρηση 1 δευτερόλεπτο.
  digitalWrite(13, LOW); // LED OFF.
  delay(1000);           // Καθυστέρηση 1 δευτερόλεπτο.
}
```

36.7 Παραδείγματα μετρήσεων με το περιβάλλον Arduino

Στον τομέα των μετρήσεων είναι πλέον δεδομένη η χρήση ενσωματωμένων συστημάτων. Η πλατφόρμα Arduino έχει καλύψει ένα μεγάλο μέρος των εφαρμογών αισθητήρων μέτρησης και ελέγχου και παρέχει ένα σύνολο βιβλιοθηκών για τη χρήση τους σε εφαρμογές. Το πιο βασικό χαρακτηριστικό των μετρήσεων με τη χρήση ενσωματωμένων συστημάτων είναι η μετατροπή των αναλογικών σημάτων των αισθητήρων σε ψηφιακά δεδομένα. Με τη χρήση απλών συναρτήσεων μέσα σε ένα Sketch όπως η `analogRead()` μπορούμε να μετατρέψουμε ένα αναλογικό σήμα από το εξωτερικό περιβάλλον σε ψηφιακό και να το επεξεργαστούμε. Στη συνέχεια παρουσιάζουμε παράδειγμα για την ανάγνωση ενός αναλογικού σήματος και τη μετατροπή του σε ψηφιακό για την πλακέτα Arduino UNO. Το παράδειγμα βρίσκεται στη διαδρομή του μενού `File`→`Examples`→`03.Analog`→`AnalogInput`.

```
int sensorPin = A0; // Ορίζουμε ως ακροδέκτη σύνδεσης του αισθητήρα τον ακροδέκτη A0
int ledPin = 13;
int sensorValue = 0; // Αρχική τιμή της μεταβλητής που θα διαβάσουμε από τον αισθητήρα.

void setup() {
  pinMode(ledPin, OUTPUT); // Ορίζουμε τον ακροδέκτη του 13 ως έξοδο για το LED.
}

void loop() {
  sensorValue = analogRead(sensorPin); // Διαβάζουμε τη στάθμη του αναλογικού σήματος.
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue); // Εκτελούμε καθυστέρηση σε ms ανάλογα με το πλάτος /του σήματος.
}
```

Το παραπάνω παράδειγμα διαβάζει το πλάτος ενός αναλογικού σήματος στον ακροδέκτη A0 της πλακέτας Arduino UNO και ανάλογα με το μέγεθος του οδηγεί κατάλληλα ένα LED στον ακροδέκτη 13 ώστε να αναβοσβήνει. Όταν το πλάτος του αναλογικού σήματος είναι χαμηλό τότε το LED αναβοσβήνει γρήγορα, ενώ εάν είναι υψηλό, το LED αναβοσβήνει πιο αργά. Η τιμή του πλάτους του σήματος που επιστρέφεται από τη συνάρτηση `analogRead()` κυμαίνεται από την ακεραία τιμή 0 έως 1023 και αντιστοιχεί σε πεδίο τιμών τάσης από 0V έως την τάση τροφοδοσίας του Arduino UNO (5V). Χρησιμοποιώντας την συνάρτηση `delay()` ορίζουμε μία τιμή καθυστέρησης από 0ms έως 1023ms (περίπου 1 δευτερόλεπτο) για να παρατηρούμε το LED να αναβοσβήνει σύμφωνα με το πλάτος του αναλογικού σήματος.

36.7.1 Μέτρηση θερμοκρασίας με το LM35 και την πλατφόρμα Arduino

Ένας πολύ εύκολος τρόπος να μετρήσουμε τη θερμοκρασία είναι το ολοκληρωμένο κύκλωμα της Texas Instruments LM35. Το ολοκληρωμένο φέρει 3 ακροδέκτες συνδεσμολογίας και μπορεί εύκολα να συνδεθεί σε ένα αναλογικό κανάλι εισόδου μίας πλακέτας Arduino όπως στον ακροδέκτη A0 του Arduino UNO. Στο Σχήμα 36.6 απεικονίζεται η συνδεσμολογία.

Ο αισθητήρας μετρά θερμοκρασία η τιμή της οποίας εκφράζεται σε αναλογικό σήμα τάσης σύμφωνα με την ακόλουθη εξίσωση:

$$Temp = \frac{5 \cdot \text{analogRead}(\text{tempPin}) \cdot 100}{1024}$$

Όπου tempPin ο ακροδέκτης που θα συνδεθεί το αναλογικό σήμα του αισθητήρα (A0) και analogRead() είναι η συνάρτηση ανάγνωσης του αναλογικού σήματος που έχουμε γνωρίσει. Η τιμή που επιστρέφει η συνάρτηση είναι μία τιμή από 0 έως 1023 και χρησιμοποιώντας την πιο πάνω συνάρτηση υπολογίζουμε τη θερμοκρασία σε βαθμούς °C. Η συνάρτηση αυτή όμως δεν αποδίδει μεγάλη ανάλυση στη μέτρηση γιατί ο αισθητήρας LM35 μας δίνει τάση από 0 έως 1V ενώ ο μετατροπέας ADC του Arduino έχει μέγιστη τιμή τάσης μέτρησης τα 5V. Αυτό σημαίνει πως δεν εκμεταλλευόμαστε το 80% του εύρους μέτρησης. Για να πετύχουμε το μέγιστο εύρος μέτρησης θα πρέπει να αλλάξουμε την τάση αναφοράς του μετατροπέα ADC σε 1,1V. Δεδομένου πως ο αισθητήρας μετρά 10mV ανά βαθμό °C και με τάση αναφοράς στον μετατροπέα 1,1V η εξίσωση για τον υπολογισμό της θερμοκρασίας θα είναι:

$$\text{calculationTemp} = \frac{0,010}{\frac{1,1}{1024}} = 9,31$$

Επομένως η εξίσωση γίνεται:

$$Temp = \frac{\text{analogRead}(\text{tempPin})}{9,31}$$

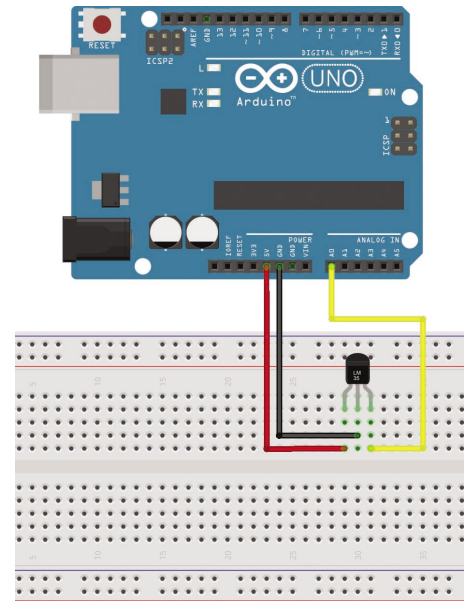
Στη συνέχεια απεικονίζεται το Sketch στο περιβάλλον Arduino IDE που υλοποιεί την διαδικασία που μόλις περιγράψαμε.

```
float tempC;
int reading;
int tempPin = 0;

void setup(){
  analogReference(INTERNAL); // Ρυθμίζουμε τάση αναφοράς του μετατροπέα σε 1,1V.
}
void loop(){
  reading = analogRead(tempPin);
  tempC = reading / 9.31; // Υπολογίζουμε τη θερμοκρασία.
}
```

36.7.2 Απεικόνιση δεδομένων με την πλατφόρμα Arduino

Ένα σύστημα μέτρησης συνήθως διαθέτει συσκευές απεικόνισης των δεδομένων σε οθόνες LCD/TFT ή οθόνες 7 τμημάτων. Η πλατφόρμα Arduino μέσω βιβλιοθηκών υποστηρίζει ένα σύνολο από διαφορετικές συσκευές απεικόνισης καθώς και μέθοδο απεικόνισης μέσω της οθόνης (monitor) της σειριακής



Σχήμα 36.6: Συνδεσμολογία αισθητήρα LM35 με την πλατφόρμα Arduino.

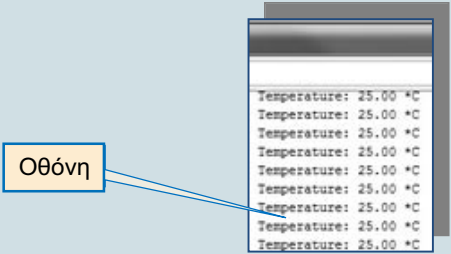
θύρας στο περιβάλλον Arduino IDE. Η οθόνη (monitor) της σειριακής θύρας μας επιτρέπει να εμφανίζουμε στην οθόνη του υπολογιστή μας διάφορα μηνύματα και διάφορες τιμές μέτρησης που έρχονται μέσω της σειριακής διασύνδεσης RS-232 που διαθέτουν οι πλακέτες Arduino.

Συνδέοντας την πλακέτα Arduino με USB στον υπολογιστή μας δίνεται αυτή η δυνατότητα μέσω του Arduino IDE. Αυτή τη μέθοδο τη χρησιμοποιούμε κατά τη διάρκεια της ανάπτυξης και του ελέγχου της εφαρμογής που υλοποιούμε καθώς αποτελεί μία άμεση μέθοδο απεικόνισης χωρίς τη χρήση ιδιαίτερων συνδεσμολογιών στο υλικό. Για να κάνουμε χρήση αυτής της μεθόδου πρέπει να ρυθμίσουμε κατάλληλα τη σειριακή θύρα της πλακέτας που χρησιμοποιούμε μέσα στη συνάρτηση `setup()` ενός Sketch και έπειτα να χρησιμοποιήσουμε τις συναρτήσεις που διατίθενται για την οδήγηση της σειριακής θύρας. Παρακάτω παρουσιάζεται μία νέα έκδοση του προγράμματος μέτρησης θερμοκρασίας με τον αισθητήρα LM35 όπου αυτή τη φορά απεικονίζουμε στο Monitor της σειριακής θύρας τα δεδομένα της θερμοκρασίας.

```
float tempC;
int reading;
int tempPin = 0;

void setup() {
  analogReference(INTERNAL); // Ρυθμίζουμε τάση αναφοράς του μετατροπέα σε 1,1V.
  Serial.begin(9600); // Ρυθμίζουμε τη σειριακή θύρα με Baud Rate 9600.
}

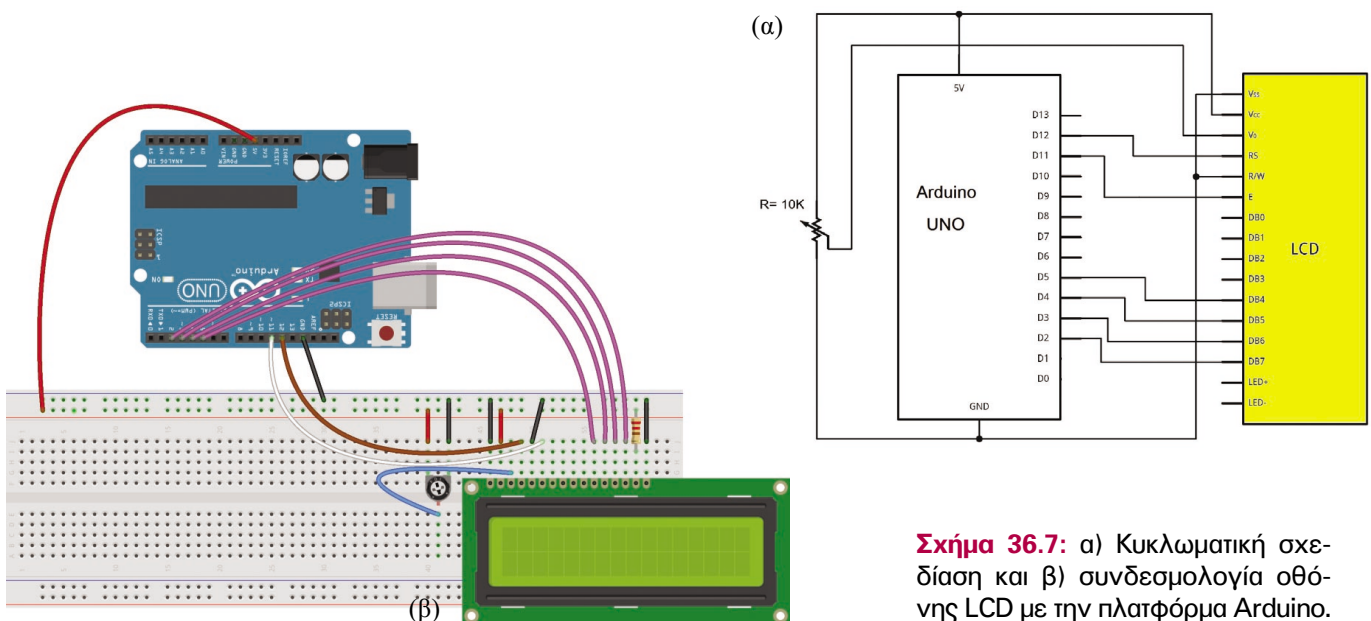
void loop() {
  reading = analogRead(tempPin);
  tempC = reading / 9.31; // Υπολογίζουμε τη θερμοκρασία.
  Serial.print("TEMPRATURE = ");
  Serial.print(tempC);
  Serial.print("°C");
  Serial.println();
  delay(500);
}
```



Οθόνη

Χρησιμοποιούμε τη συνάρτηση `print` της βιβλιοθήκης `Serial` για να εκτυπώσουμε μία συμβολοσειρά με τη μορφή 'Temperature = XX°C', όπου XX η τιμή της θερμοκρασίας.

Μία άλλη μέθοδος απεικόνισης είναι οι οθόνες υγρών κρυστάλλων LCD οι οποίες μπορούν να οδηγηθούν εύκολα με την πλατφόρμα Arduino κάνοντας απλά την απαραίτητη συνδεσμολογία με τις πλακέτες Arduino. Οι περισσότερες οθόνες LCD που κυκλοφορούν στο εμπόριο έχουν τον ίδιο ελεγκτή και συνδέονται με παρόμοιο τρόπο. Στο Σχήμα 36.7 απεικονίζεται η συνδεσμολογία μίας απλής οθόνης LCD με την πλακέτα Arduino UNO.



Σχήμα 36.7: α) Κυκλωματική σχεδίαση και β) συνδεσμολογία οθόνης LCD με την πλατφόρμα Arduino.

Οι οθόνες αυτές οδηγούνται είτε με 8 γραμμές δεδομένων και 3 γραμμές ελέγχου είτε με 4 γραμμές δεδομένων και 3 γραμμές ελέγχου για μείωση των αγωγών πάνω σε πλακέτες PCB. Διασυνδέοντας την οθόνη με τον τρόπο που απεικονίζεται στη συνέχεια το μόνο που έχουμε να κάνουμε είναι να χρησιμοποιήσουμε τις έτοιμες βιβλιοθήκες που μας παρέχονται από την πλατφόρμα Arduino για να απεικονίσουμε μηνύματα και διάφορες τιμές πάνω σε αυτές. Η βιβλιοθήκη που μας παρέχεται για τις LCD οθόνες καλείται LiquidCrystal και μπορούμε να την συμπεριλάβουμε μέσα στο Sketch που γράφουμε από το μενού Sketch→Import_Library→LiquidCrystal. Οι συνηθέστερα χρησιμοποιούμενες συναρτήσεις για την οδήγηση της LCD οθόνης είναι οι LCD.begin() και LCD.print(). Η LCD.begin() αρχικοποιεί την LCD οθόνη που έχουμε συνδέσει και τοποθετείται μέσα στη συνάρτηση setup() ενός Sketch. Η LCD.print() χρησιμοποιείται απλά για να τυπώσει μηνύματα στην οθόνη.

Στη συνέχεια παρουσιάζεται παράδειγμα Sketch που χρησιμοποιεί την βιβλιοθήκη LiquidCrystal για να οδηγήσει μία οθόνη LCD.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Ορισμός των ακροδεκτών οδήγησης της LCD

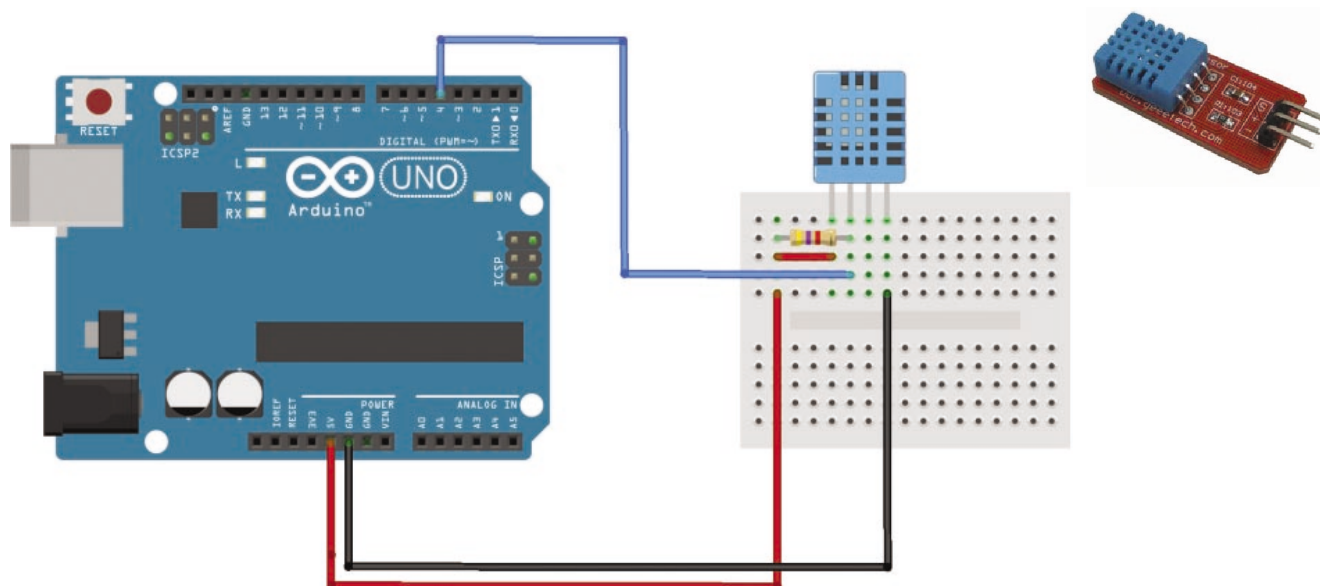
void setup() {
  lcd.begin(16, 2); // Αρχικοποίηση της LCD σε 16x2 γραμμές
  lcd.print("hello, world!"); // Εμφάνιση του μηνύματος hello, world
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis()/500);
}
```

Επανάληψη διαδικασίας ενεργοποίησης και απενεργοποίησης της οθόνης (Blink).

36.7.3 Μέτρηση υγρασίας με την πλατφόρμα Arduino

Εκτός από τους αισθητήρες μέτρησης που παράγουν αναλογικά σήματα υπάρχουν και αισθητήρες που οδηγούνται με ψηφιακές μεθόδους. Οι αισθητήρες αυτοί χρησιμοποιηθούν ψηφιακό πρωτόκολλο μετάδοσης δεδομένων. Σε τέτοιες περιπτώσεις δεν απαιτείται η χρήση του μετατροπέα αναλογικού σήματος σε ψηφιακό αλλά η χρήση ψηφιακών ακροδεκτών εισόδου/εξόδου οι οποίοι θα πρέπει να προγραμματίζονται ώστε να λειτουργούν σύμφωνα με το πρωτόκολλο επικοινωνίας του αισθητήρα. Ένας τέτοιος αισθητήρας είναι ο DHT11 που μετρά την υγρασία και χρησιμοποιεί το πρωτόκολλο επικοινωνίας I²C. Η συνδεσμολογία αυτού του αισθητήρα με την πλακέτα Arduino UNO απεικονίζεται στο Σχήμα 36.8.

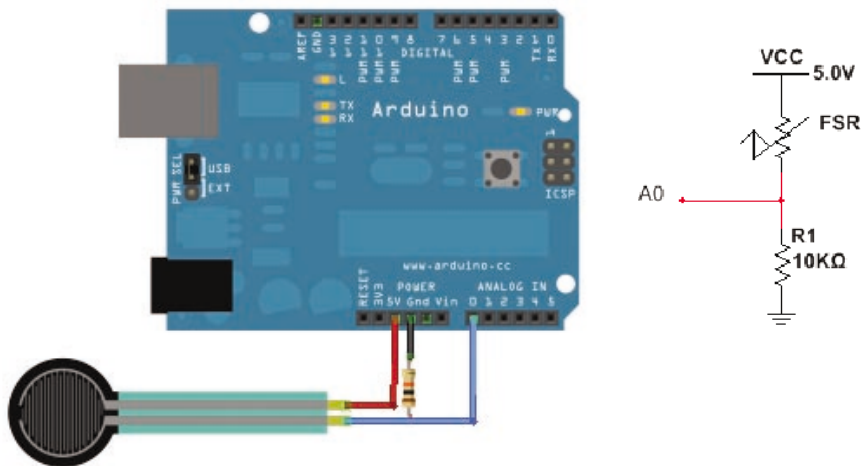


Σχήμα 36.8: Συνδεσμολογία του αισθητήρα DHT11 με την πλατφόρμα Arduino.

Συμπεριλαμβάνοντας την βιβλιοθήκη οδήγησης του αισθητήρα DHT11 που μπορεί να την βρει κανείς στον ιστότοπο του Arduino (<http://playground.arduino.cc/Main/DHTLib>) μπορούμε να μετράμε την υγρασία χρησιμοποιώντας τις παρεχόμενες συναρτήσεις.

36.7.4 Μέτρηση δύναμης με στοιχείο FSR και την πλατφόρμα Arduino

Χρησιμοποιώντας την πλατφόρμα Arduino και τις συναρτήσεις του αναλογικού σε ψηφιακό μετατροπέα μπορούμε να μετράμε δύναμη μέσω αισθητήρων εύκαμπτων δυναμικών αντιστάσεων FSR. Οι αισθητήρες αυτοί διασυνδέονται σε αναλογικό κανάλι εισόδου της πλακέτας Arduino μέσω διαιρέτη τάσης όπως απεικονίζεται στο Σχήμα 36.9.



Σχήμα 36.9: Συνδεσμολογία του αισθητήρα FSR με την πλατφόρμα Arduino.

Στη συνέχεια παρουσιάζουμε έναν απλό κώδικα μέτρησης δύναμης με αισθητήρα δυναμικής αντίστασης FSR στο αναλογικό κανάλι 0 της πλακέτας Arduino UNO. Το ακόλουθο παράδειγμα χρησιμοποιεί τη σειριακή θύρα για να απεικονίζει τις τιμές αυτές στην οθόνη (monitor). Η μέτρηση εκτελείται κάθε 50ms και η τιμή του αισθητήρα εμφανίζεται ως ακέραιος αριθμός από 0 έως 1023.

```
int resfsrAnalogPin = 0;
int resfsrReading;

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  resfsrReading = analogRead(resfsrAnalogPin);
  Serial.print("Analog reading = ");
  Serial.println(resfsrReading);
  delay(50);
}
```

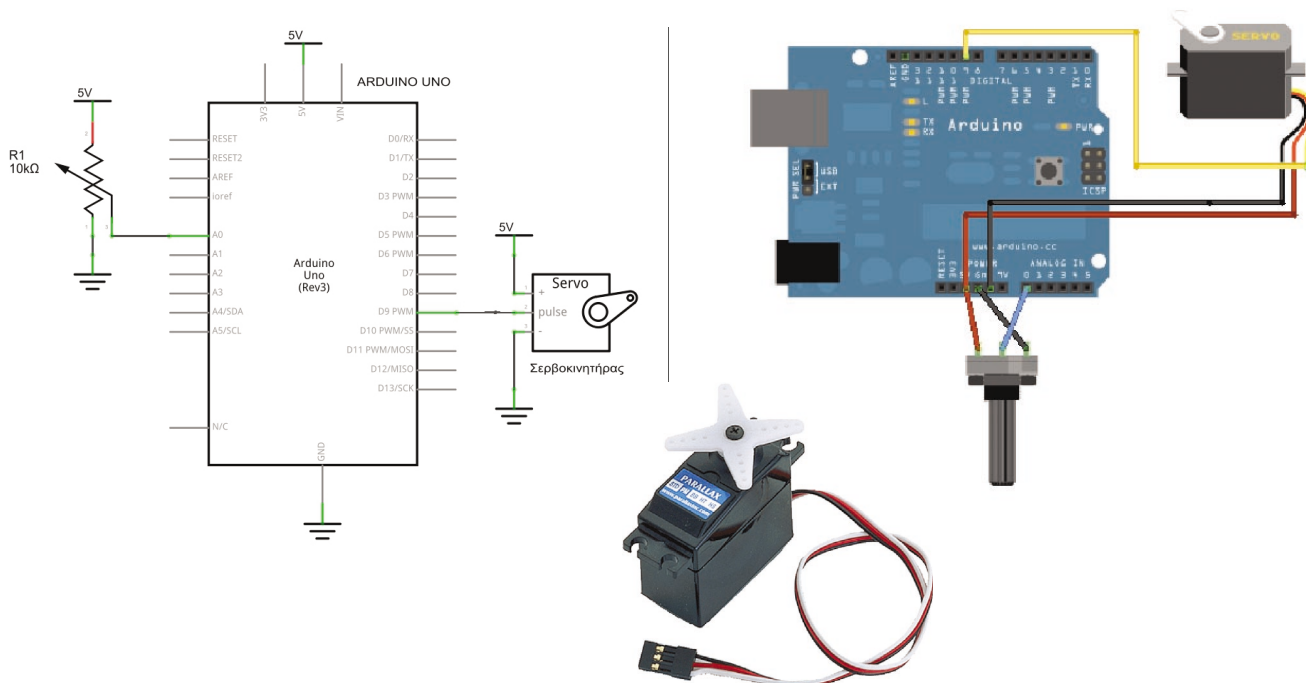
36.7.5 Αισθητήρες προσέγγισης με Arduino

Η πλατφόρμα Arduino υποστηρίζει και αισθητήρες προσέγγισης με υπέρηχους χρησιμοποιώντας συγκεκριμένο υλικό κατάλληλα σχεδιασμένο για αυτό το σκοπό. Για να εργαστεί κανείς με τέτοιου είδους αισθητήρες αρκεί να προμηθευτεί τον κατάλληλο εξοπλισμό και να τον διασυνδέσει με μία πλακέτα Arduino. Ο αισθητήρας υπέρηχων δεν απαιτεί κάποια ιδιαίτερη βιβλιοθήκη και μπορεί να οδηγηθεί με απλές συναρτήσεις της πλατφόρμας Arduino. Η μέτρηση της απόστασης γίνεται με την μέτρηση των παλμών των υπέρηχων που ανακλώνται σύμφωνα με την ταχύτητα του ήχου και υπολογίζεται από την ακόλουθη σχέση:

$$\text{distance} = \frac{\text{duration}}{58,2}$$

36.7.6 Οδήγηση σερβοκινητήρων με το Arduino

Κάνοντας χρήση της βιβλιοθήκης Servo που συμπεριλαμβάνεται μέσα στο λογισμικό Arduino IDE και υλοποιώντας την ακόλουθη συνδεσμολογία σε μία πλακέτα Arduino, μπορούμε να οδηγήσουμε με ευκολία ένα σερβοκινητήρα (Σχ. 36.10).



Σχήμα 36.10: Συνδεσμολογία σερβοκινητήρα με την πλατφόρμα Arduino.

Στη συνέχεια παρουσιάζεται ο κώδικας από το παράδειγμα knob.ino του περιβάλλοντος του Arduino File > Examples > Servo > Knob.

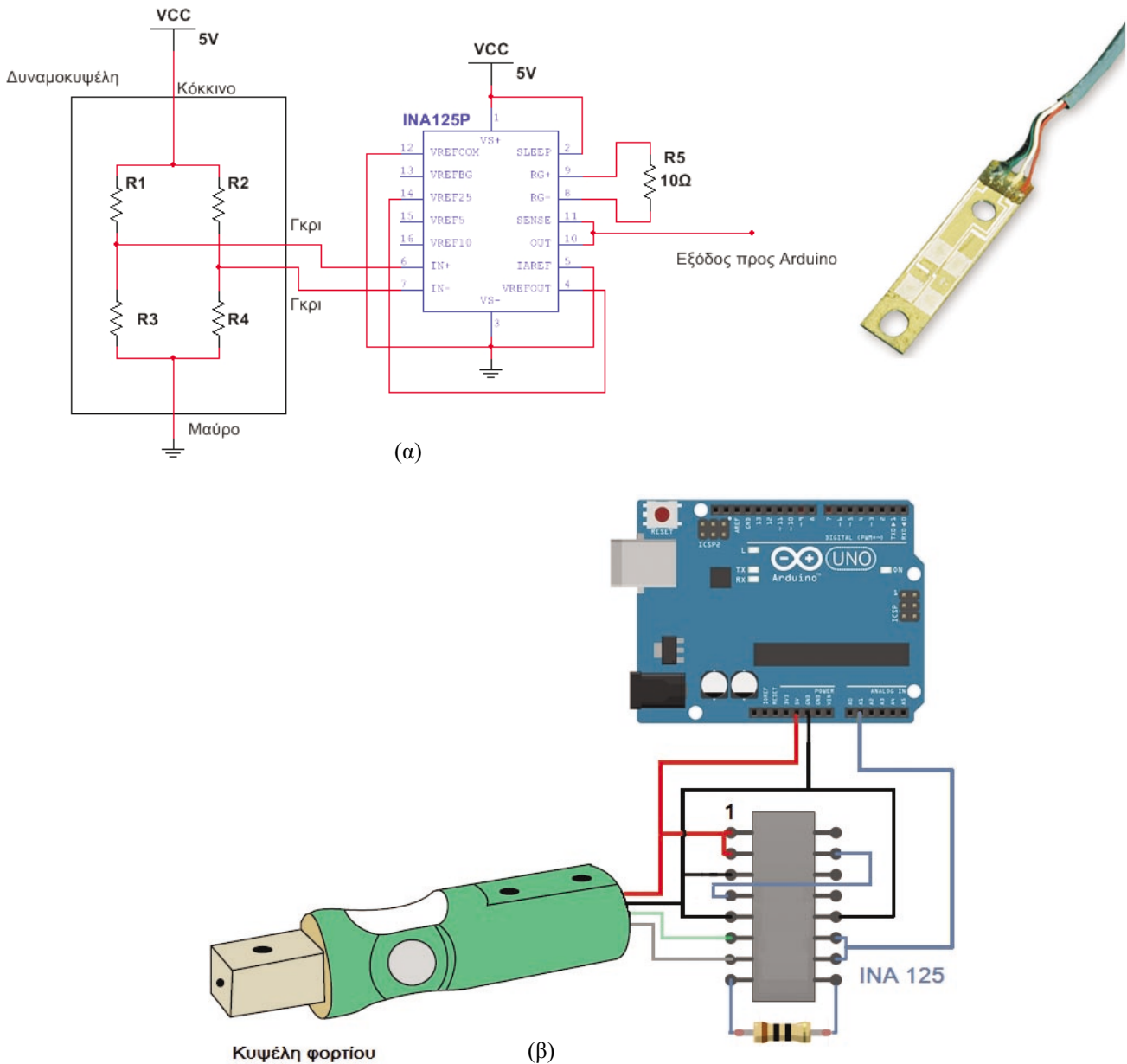
```
#include <Servo.h>
Servo myservo; // Διαδικασία ελέγχου σέρβο.
int potpin = 0;
int val;

void setup() {
  myservo.attach(9); // Έλεγχος σέρβο από ακροδέκτη 9.
}

void loop() {
  val = analogRead(potpin); // Ανάγνωση θέσης από 0 έως 1023.
  val = map(val, 0, 1023, 0, 180); // Ορισμός δεδομένων εισόδου και γωνία περιστροφής.
  myservo.write(val); // Εγγραφή για κίνηση του σέρβο.
  delay(15);
}
```

36.7.7 Μέτρηση φορτίου με δυναμοκυψέλη και την πλατφόρμα Arduino

Για τη μέτρηση φορτίου μέσω δυναμοκυψέλης θα πρέπει να χρησιμοποιήσουμε το ολοκληρωμένο INA 125 το οποίο είναι ένας ενισχυτής οργανολογίας με καθορισμό κέρδους ώστε το σήμα εξόδου της δυναμοκυψέλης να ενισχυθεί ως προς το πλάτος του πριν την είσοδο του στο αναλογικό κανάλι του Arduino (Σχ. 36.11). Για την ακρίβεια της μέτρησης θα πρέπει ο κώδικας να αποδίδει την τιμή ως μέσο όρο τουλάχιστον 10 μετρήσεων.



Σχήμα 36.11: α) Κυκλωματική συνδεσμολογία ενισχυτή οργανολογίας με δυναμοκυψέλη, και β) συνδεσμολογία δυναμοκυψέλης πλήρους γέφυρας με την πλατφόρμα Arduino.

Στη συνέχεια παρουσιάζεται ο κώδικας για τη μέτρηση της δύναμης και απεικόνιση στην σειριακή οθόνη.

```

int loadCell = A1; // Χρησιμοποιούμε το αναλογικό κανάλι A1.
int analogValue = 0; // Αρχικοποίηση της τιμής του αναλογικού καναλιού.
const int numReadings = 10; // Όρισε τον αριθμό μετρήσεων για τον μέσο όρο.
int readings[numReadings]; // Όρισε έναν πίνακα που θα αποθηκεύει τις μετρήσεις.
int index = 0; // Μεταβλητή ως δείκτης του πίνακα μετρήσεων.
int total = 0; // Μεταβλητή που θα κρατά το σύνολο των μετρήσεων που έχουν γίνει.
int measurementAverage = 0; // Μεταβλητή που θα κρατά το μέσο όρο των μετρήσεων.

void setup()
{
  Serial.begin(9600);

  // Μηδένισε όλα τα στοιχεία του πίνακα μετρήσεων.
  for (int thisReading = 0; thisReading < numReadings; thisReading++)
  
```

```

readings[thisReading] = 0;
}

void loop(){
analogValue = analogRead(loadCell); // Διάβασε την τιμή της δυναμοκουψέλης.
measurementRead(); // Εκτέλεσε τη διαδικασία μετρήσεων.
Serial.print(analogValue); // Εμφάνισε την τιμή του αισθητήρα.
Serial.print(", ");
Serial.println(measurementAverage);
delay(500);
}

// Διαδικασία μετρήσεων

void measurementRead () {
total= total - readings[index]; // Αφαίρεσε την τελευταία μέτρηση.
readings[index] = analogRead(loadCell); // Κάνε ανάγνωση από τον αισθητήρα.
total= total + readings[index]; // Πρόσθεσε την τιμή που διάβασες στο σύνολο.
index = index + 1; // Πήγαινε στην επόμενη θέση μέσα στον πίνακα.
if (index >= numReadings) // Αν βρισκόμαστε στο τέλος του πίνακα πήγαινε στην αρχή.
index = 0; // Μηδένισε τον δείκτη.
measurementAverage = total / numReadings;
delay(1); // Καθυστέρησε μεταξύ των αναγνώσεων για σταθεροποίηση.
}

```

36.8 Διασύνδεση LabVIEW με πλατφόρμα Arduino

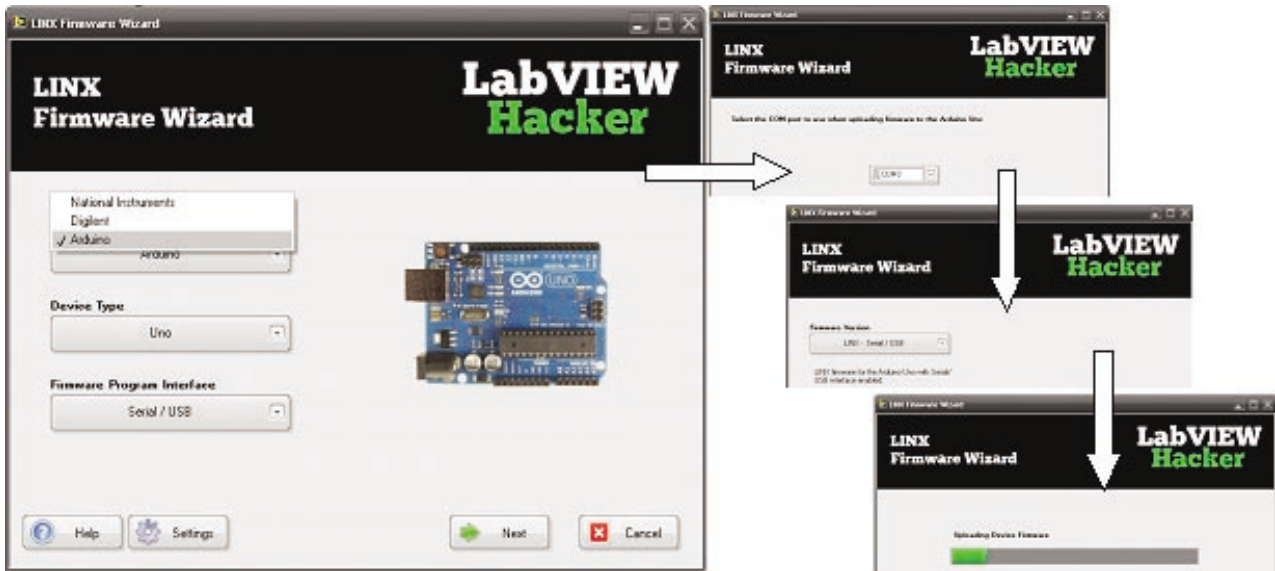
Για τη διασύνδεση του LabVIEW με την πλατφόρμα Arduino απαιτούνται τα παρακάτω εργαλεία:

- [1] LabVIEW 2011 ή νεότερη έκδοση.
- [2] Ο οδηγός NI VISA Driver για τη σειριακή θύρα (RS-232). Μπορούμε να λάβουμε ελεύθερα τον οδηγό από τη διεύθυνση: <http://www.ni.com/visa/>
- [3] Το περιβάλλον προγραμματισμού Arduino IDE που μπορούμε να το λάβουμε ελεύθερα από τη διεύθυνση: <http://arduino.cc/en/Main/Software>
- [4] Τον διαχειριστή εργαλείων JKI VI Package Manager (VIPM) για την εγκατάσταση βιβλιοθηκών.
- [5] Εγκατάσταση του εργαλείου (LINX) για τη διασύνδεση του LabVIEW με την πλατφόρμα Arduino. Το εργαλείο LINX εγκαθίσταται μέσω του λογισμικού JKI VI Package Manager (VIPM). Μπορούμε να κατεβάσουμε αυτό το εργαλείο από την ακόλουθη διεύθυνση: https://labviewhacker.com/doku.php?id=learn:libraries:linx:tutorials:install_linx

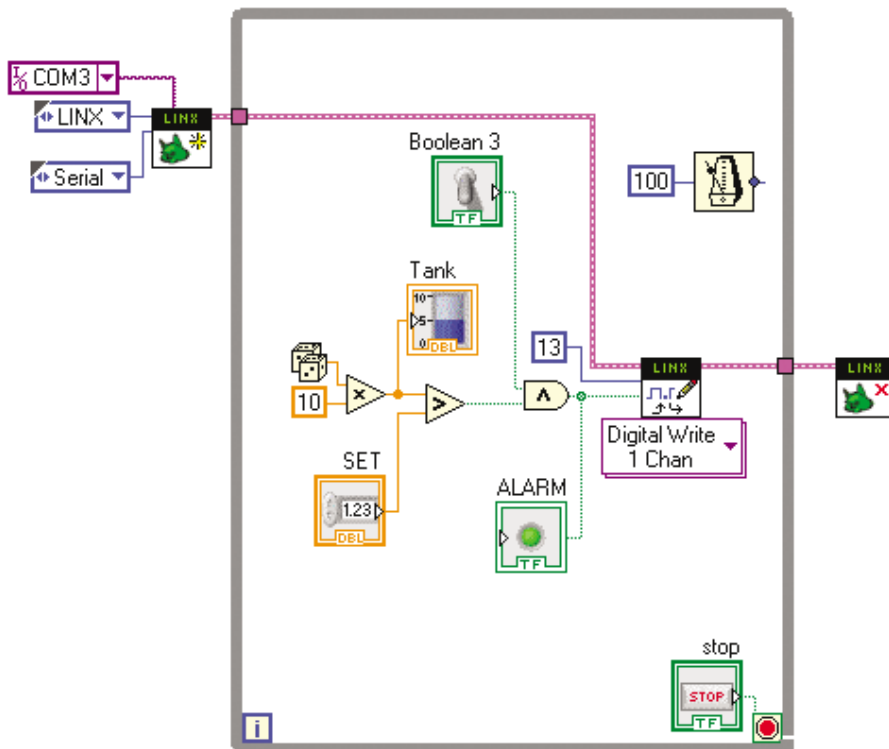


Για την επικοινωνία του LabVIEW με το Arduino επιλέγουμε από το μενού Tools→LabVIEW Hacker→LINX→LINX Firmware Wizard και ακολουθούμε τις οδηγίες που εμφανίζονται στο παράθυρο σχετικά με τον τύπο πλατφόρμας Arduino και τη θύρα που θα χρησιμοποιήσουμε. Στο τέλος ο οδηγός φορτώνει το Firmware στην πλατφόρμα του Arduino (Σχ. 36.12).

Μέσω της βιβλιοθήκης που έχει εισάγει το εργαλείο LINX χρησιμοποιούμε τη λειτουργία Initialize.vi για να ορίσουμε τη θύρα επικοινωνίας (έξω από το δομή While Loop) και τη λειτουργία Digital Write.vi μέσω της οποίας μπορούμε στο απλό παράδειγμα που παρουσιάζεται στο Σχήμα 36.13 να ελέγξουμε την ψηφιακή έξοδο D13 του Arduino UNO με χρήση κώδικα VI. Για τον τερματισμό της επικοινωνίας χρησιμοποιούμε τη λειτουργία Close.vi έξω από τη δομή While Loop.



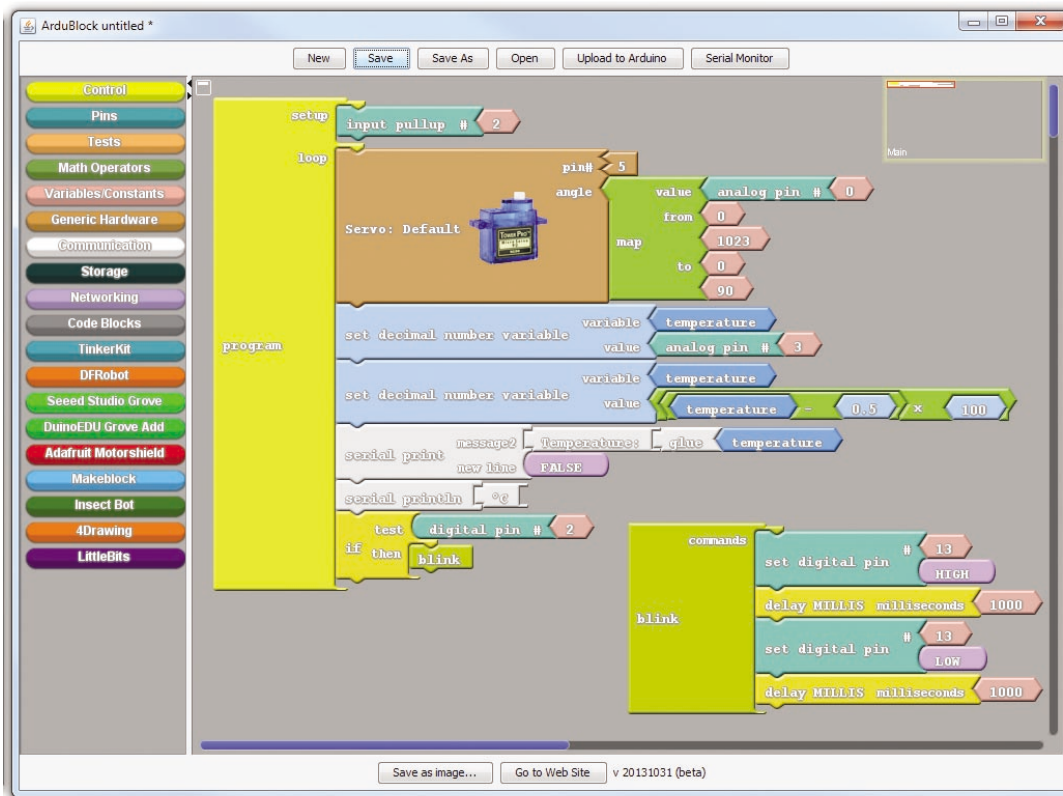
Σχήμα 36.12: Παράθυρα του εργαλείου LINX Firmware Wizard.



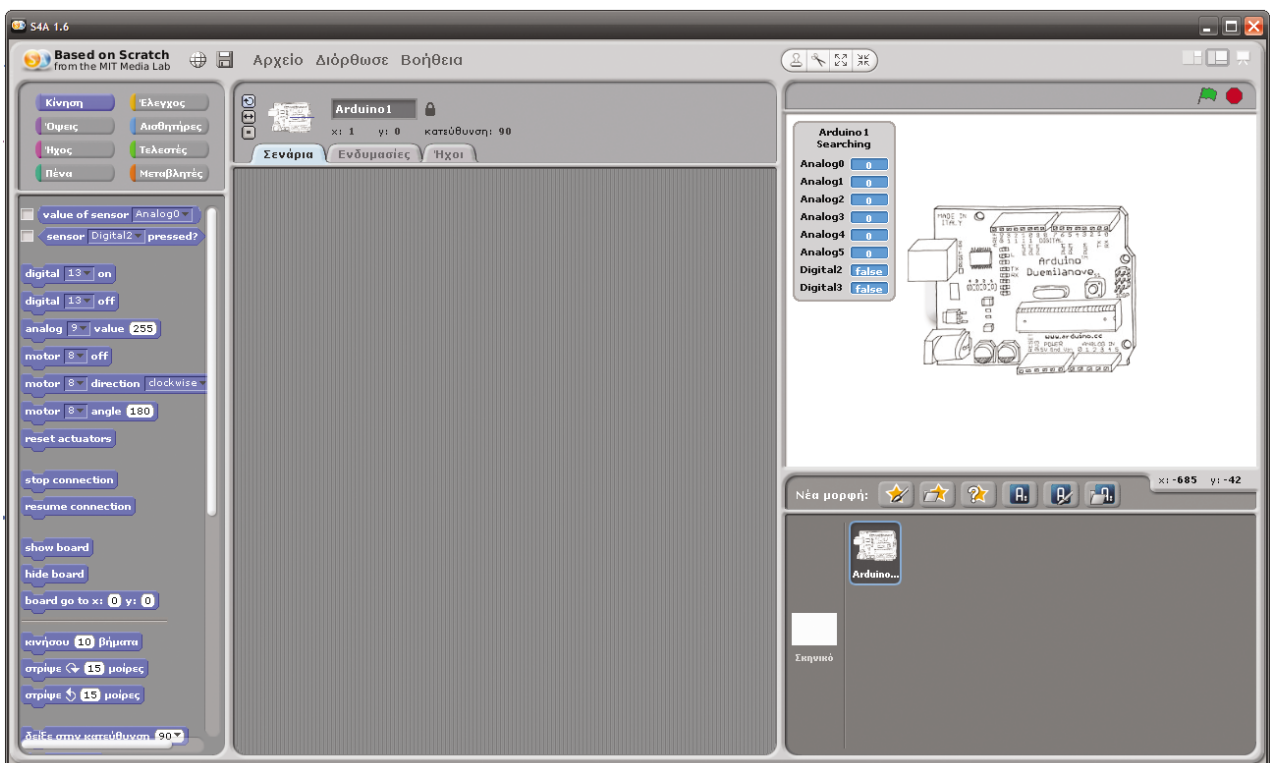
Σχήμα 36.13: Παράδειγμα VI ελέγχου της ψηφιακής εξόδου D13 της πλατφόρμας Arduino UNO.

36.9 Άλλα εργαλεία γραφικού προγραμματισμού Arduino

Μέσω του προγράμματος ArduBlock (<http://blog.ardublock.com/>) (<http://sourceforge.net/projects/ardublock/>) μπορούμε να προγραμματίσουμε ένα Arduino με χρήση πλακιδίων όπως απεικονίζεται στο Σχήμα 36.14, ενώ με το εργαλείο S4A (<http://s4a.cat>) (Σχ. 36.15) το οποίο είναι βασισμένο στον προγραμματισμό του Scratch μπορούμε να διασυνδέσουμε το Arduino για ανάπτυξη απλών εφαρμογών στις οποίες χειριζόμαστε το Arduino ως κάρτα συλλογής δεδομένων σε μόνιμη σύνδεση του Arduino με τον υπολογιστή για πολύ απλές εφαρμογές.



Σχήμα 36.14: Περιβάλλον του εργαλείου ArduBlock.



Σχήμα 36.15: Περιβάλλον του εργαλείου S4A.