

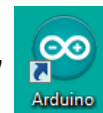
Φύλλο εργασίας

Δραστηριότητα 1^η – Blink LED

Σκοπός: Εξοικείωση με το IDE του Arduino. Μεταγλώττιση και κατέβασμα έτοιμου προγράμματος σε C++ στο Arduino

A. Ακολουθήστε τις παρακάτω οδηγίες

- Ανοίξτε το IDE του Arduino κάνοντας διπλό κλικ στο εικονίδιο στην επιφάνεια εργασίας
- Συνδέστε το Arduino μέσω USB με μια ελεύθερη θύρα του PC (Το πράσινο LED στην πλακέτα του Arduino με την ένδειξη ON θα πρέπει να ανάψει πράσινο)
- Επιλέξτε Εργαλεία->Πλακέτα->Arduino Uno (Την πρώτη φορά ή αν δεν είναι ήδη επιλεγμένο)
- Επιλέξτε Εργαλεία->Σειριακή Θύρα->COM X (Όπου X είναι η θύρα που είναι συνδεδεμένο το Arduino. Αν δεν τη βρίσκετε, αποσυνδέστε το Arduino, δείτε τις θύρες που εμφανίζονται, επανασυνδέστε το Arduino και δείτε ξανά τις θύρες. Η θύρα του Arduino θα εμφανίζεται τη δεύτερη φορά, οπότε και την επιλέγετε)
- Ανοίξτε το sketch (project) Αρχείο->Παραδείγματα->01.Basics->Blink
- Θα εμφανιστεί το παρακάτω παράθυρο





```
Arduino IDE - Blink | Arduino 1.0.5-r2
Αρχείο  Επεξεργασία  Σχέδιο  Εργαλεία  Βοήθεια

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

- Πατήστε το Verify  για μεταγλώττιση του προγράμματος μόνο
- Πατήστε το Upload  για μεταγλώττιση και μεταφόρτωση του προγράμματος, εφόσον δεν υπάρχουν σφάλματα κατά τη μεταγλώττιση, στο Arduino
- Τί παρατηρείτε;
- Αν αναβοσβήνει το ενσωματωμένο LED στην πλακέτα του Arduino όλα έχουν γίνει σωστά.

- Πατήστε το κουμπί RESET. Τί παρατηρείτε;
- Πειραματιστείτε αλλάζοντας την πρώτη εντολή `delay(1000);` σε `delay(500);` και επαναφορτώστε το πρόγραμμα

B. Επεκτάσεις

- Εξοικονομήστε με το περιβάλλον και δοκιμάστε να ανοίξετε και να μεταφορτώσετε και άλλα παραδείγματα. Υπάρχουν πολλά έτοιμα παραδείγματα τα οποία μπορούν να χρησιμοποιηθούν ως βάση για τα δικά σας project.

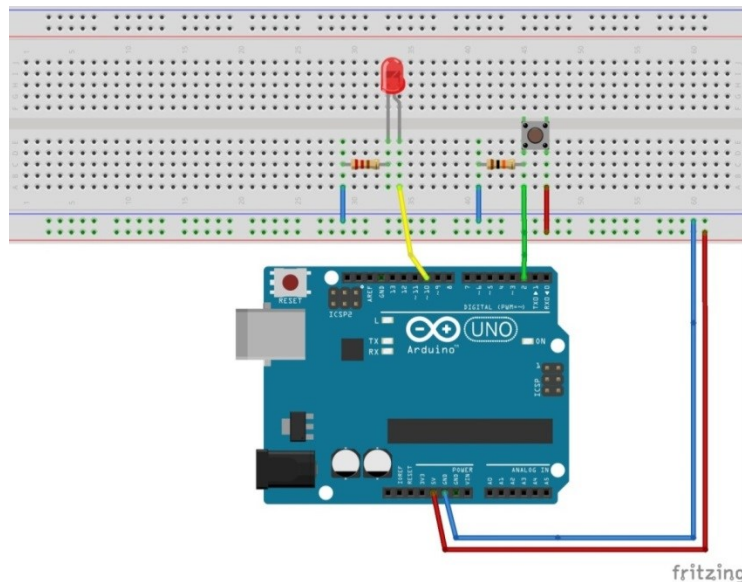
Δραστηριότητα 2^η - Pushbutton + LED (Ψηφιακές Είσοδοι/Εξοδοι)

Σκοπός: Πατώντας τον διακόπτη (pushbutton) να ανάβει το LED και αφήνοντάς τον να σβήνει

A. Υλοποιήστε το παρακάτω κύκλωμα

Θα χρειαστείτε:

- 1 κόκκινο LED
- 1 διακόπτη (pushbutton)
- 1 αντίσταση 220Ω (κόκκινο-κόκκινο-καφέ-χρυσό)
- 1 αντίσταση 10KΩ (καφέ-μαύρο-πορτοκαλί-χρυσό)
- Καλώδιο USB + καλώδια συνδέσεων
- Arduino + Breadboard



TIPS

- Σημειώστε σε ποιες εισόδους/εξόδους συνδέεται το LED και ο διακόπτης καθώς οι ίδιες θα πρέπει να χρησιμοποιηθούν και μέσα στο πρόγραμμα.

- Το χρώμα των καλωδίων δεν έχει λειτουργικό ρόλο αλλά βοηθητικό στο να καταλαβαίνουμε με μια ματιά τί έχουμε συνδέσει που. Για αυτό καλό είναι να χρησιμοποιούμε:
 - Κόκκινο για 5V / (+)
 - Μπλέ ή Μαύρο για GND / 0V / (-)
 - Πράσινο για ψηφιακές εισόδους
 - Κίτρινο για ψηφιακές εξόδους
 - Πορτοκαλί για εξόδους PWM (αναλογικές εξόδους)
 - Κόκκινο για αναλογικές εισόδους
- Ο διακόπτης (pushbutton) επιτρέπει τη διέλευση του ρεύματος όσο τον πατάμε και διακόπτει τη διέλευση του ρεύματος όταν τον αφήνουμε
- Το LED ανάβει όταν περνάει ρεύμα από την άνοδο (μακρύ ποδαράκι) προς την κάθοδο. Δεν ανάβει όταν δεν περνάει ρεύμα ή όταν το ρεύμα προσπαθεί να περάσει από την κάθοδο προς την άνοδο.
- Η αντίσταση περιορίζει την ποσότητα ρεύματος που περνάει στο κύκλωμα προστατεύοντας τόσο το LED όσο και την ψηφιακή έξοδο του Arduino. Στο διακόπτη χρησιμοποιείται βοηθητικά για να ξεκαθαρίσουν οι καταστάσεις πατημένος/όχι πατημένος διακόπτης.

B. Προγραμματισμός Arduino

1. Με το περιβάλλον προγραμματισμού IDE του Arduino (C++)

- Ανοίξτε το IDE του Arduino
- Ανοίξτε το παράδειγμα Αρχείο->Παραδείγματα-> 01.Basics
-> BareMinimum
- Χρησιμοποιήστε μέσα στη function setup() τις εντολές

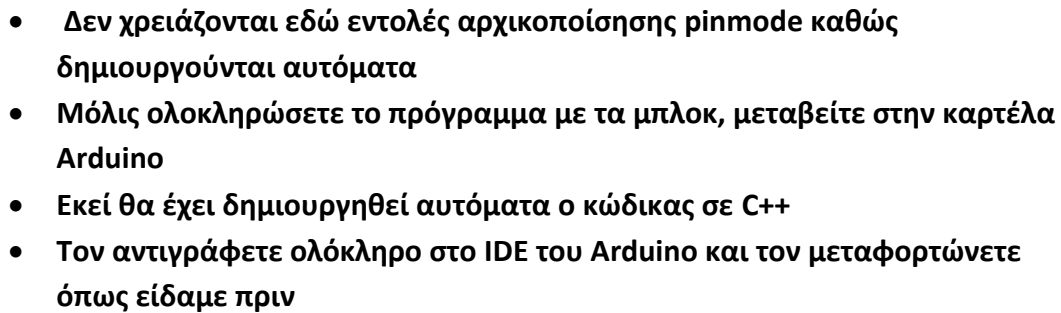
```
pinMode(ledPin, OUTPUT);
pinMode(buttonPin, INPUT);
```
- Χρησιμοποιήστε μέσα στη function loop() τις εντολές

```
digitalRead(buttonPin); // Για να διαβάσετε την κατάσταση του διακόπτη
digitalWrite(ledPin, HIGH); // Για να ενεργοποιήσετε το LED
```

Μπορείτε να χρησιμοποιήσετε μεταβλητές τύπου int και τις σταθερές LOW και HIGH
- Μεταγλωτίστε και μεταφορτώστε

2. Με το περιβάλλον προγραμματισμού Blocklyduino

- Ανοίξτε στο φυλλομετρητή το URL:
<http://www.gasolin.idv.tw/public/blockly/apps/blocklyduino/index.html>
- Το περιβάλλον είναι το γνωστό Blockly τροποποιημένο για προγραμματισμό Arduino
- Χρησιμοποιώντας διάφορα μπλοκ από τις κατηγορίες π.χ. IN\OUT και LOGIC δημιουργούμε το πρόγραμμά μας το οποίο αυτόματα δημιουργεί το αντίστοιχο πρόγραμμα σε C++ στην καρτέλα



- Μπορείτε να ξεκινήσετε να δημιουργείτε το πρόγραμμά σας χρησιμοποιώντας εντολές από την κατηγορία «Κίνηση» όπως:



- Θα πρέπει να επιλέξετε τις αντίστοιχες εισόδους/εξόδους που χρησιμοποιήσατε στο κύκλωμά σας και να τις συνδυάσετε με εντολές από τις υπόλοιπες κατηγορίες για να ολοκληρώσετε το πρόγραμμά σας όπως:
- Όταν ολοκληρώσετε το πρόγραμμα ή θέλετε να το δοκιμάσετε πατήστε την πράσινη σημαία.



Γ. Επεκτάσεις (και για τις 3 μεθόδους προγραμματισμού)

- Το LED να σβήνει αφού περάσουν 5" από τη στιγμή που θα αφήσουμε το διακόπτη (Off Delay)
- Ενα πάτημα του διακόπτη ενεργοποιεί το LED και αυτό μένει αναμμένο. Σβήνει αφού ξαναπατηθεί ο διακόπτης (Latch)

Παρατηρήστε ότι χωρίς να αλλάξουμε καθόλου το υλικό (hardware) αλλάξαμε το πρόγραμμα (software) και έτσι άλλαξε η συμπεριφορά του κυκλώματος.

Δραστηριότητα 3^η – Traffic Lights (Ψηφιακές Είσοδοι/Εξοδοι)

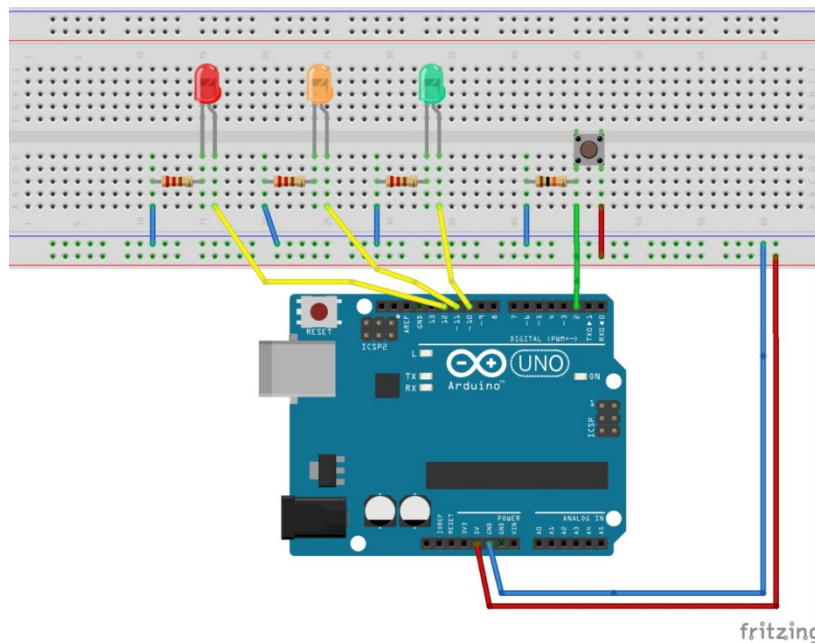
Σκοπός: Ξεκινώντας το πρόγραμμα το κόκκινο LED είναι συνεχώς αναμμένο. Πατώντας στιγμιαία τον διακόπτη (pushbutton) θα πρέπει μετά από 5" να σβήσει το κόκκινο και να ανάψει μόνο το πράσινο LED, το οποίο θα παραμείνει αναμμένο για 10". Έπειτα θα ανάψει μόνο το πορτοκαλί για 3" και τέλος θα ανάψει μόνο το κόκκινο το οποίο και θα παραμείνει αναμμένο μέχρι να ξαναπατηθεί ο διακόπτης και ξεκινήσει ένας νέος κύκλος.

A. Η δραστηριότητα αποτελεί συνέχεια τις προηγούμενης επομένως μπορείτε να συμπληρώσετε το κύκλωμα αντί να το φτιάξετε από την αρχή:

Θα χρειαστείτε:

- 1 κόκκινο LED
- 1 πορτοκαλί LED
- 1 πράσινο LED
- 1 διακόπτη (pushbutton)
- 3 αντιστάσεις 220Ω (κόκκινο-κόκκινο-καφέ-χρυσό)

- 1 αντίσταση 10KΩ (καφέ-μαύρο-πορτοκαλί-χρυσό)
- Καλώδιο USB + καλώδια συνδέσεων
- Arduino + Breadboard



Β. Προγραμματισμός Arduino

- Για τον προγραμματισμό του Arduino μπορείτε να χρησιμοποιήσετε οποιοδήποτε από τα 3 περιβάλλοντα θέλετε
- Οι εντολές που θα χρειαστείτε είναι οι ίδιες με αυτές της 2ης δραστηριότητας

Γ. Επεκτάσεις (και για τις 3 μεθόδους προγραμματισμού)

- Δοκιμάστε να προσθέσετε 2 LED (κόκκινο και πράσινο) για τους πεζούς, τα οποία θα πρέπει να ανάβουν σε συγχρονισμό με τα αρχικά 3 LED (δηλαδή όταν οι πεζοί έχουν πράσινο, για τα αυτοκίνητα θα είναι κόκκινο και αντιστρόφως)

Δραστηριότητα 4^η - Pushbutton + LED + Piezoelectric (Αναλογική έξοδος PWM)

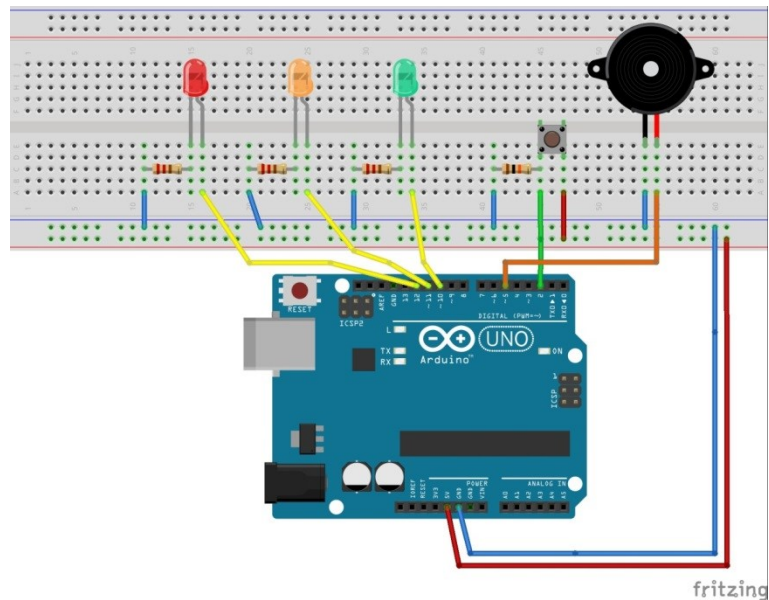
Σκοπός: Πατώντας στιγμιαία τον διακόπτη (pushbutton) θα πρέπει να αρχίσει να ηχεί το πιεζοηλεκτρικό στοιχείο με 2 συχνότητες για 300 msec την κάθε μία (σαν ασθενοφόρο) και να αναβοσβήνει ταυτόχρονα το κόκκινο LED 5 φορές.

Α. Υλοποιήστε το παρακάτω κύκλωμα (Αν έχετε υλοποιήσει την προηγούμενη δραστηριότητα απλά συνδέστε τον πιεζοηλεκτρικό αισθητήρα)

Θα χρειαστείτε:

- 1 κόκκινο LED
- 1 πορτοκαλί LED

- 1 πράσινο LED
- 1 διακόπτη (pushbutton)
- 3 αντιστάσεις 220Ω (κόκκινο-κόκκινο-καφέ-χρυσό)
- 1 αντίσταση 10KΩ (καφέ-μαύρο-πορτοκαλί-χρυσό)
- 1 πιεζοηλεκτρικό αισθητήρα/ηχείο
- Καλώδιο USB + καλώδια συνδέσεων
- Arduino + Breadboard



B. Προγραμματισμός Arduino

Χρησιμοποιήστε τις τιμές 50 και 250 για τους 2 τόνους του ήχου.
Η τιμές που δέχεται η παράμετρος `outputValue` είναι 0-255.

1. Με το περιβάλλον προγραμματισμού IDE του Arduino (C++)

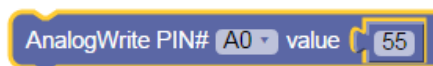
- Χρησιμοποιήστε την εντολή

```
analogWrite(analogOutPin,outputValue);
```

 // Ενεργοποιεί μια συγκεκριμένη έξοδο με μία παλμοσειρά μεταβλητού παλμού (PWM) για να ηχήσει το πιεζοηλεκτρικό ηχείο

2. Με το περιβάλλον προγραμματισμού Blocklyduino

- Το Blocklyduino έχει την εντολή



Η οποία δεν είναι συμβατή με το Arduino Uno που διαθέτουμε αλλά μπορείτε εντούτοις να τη χρησιμοποιήσετε αφαιρώντας στον παραγόμενο κώδικα το A από την διεύθυνση της εξόδου που χρησιμοποιείτε π.χ. το A0 θα πρέπει να γίνει 0.

3. Με το περιβάλλον προγραμματισμού Scratch for Arduino

- Δεν ξεχνούμε να μεταφορτώσουμε στο Arduino το scetch S4AFirmware16.ino με το IDE του Arduino όπως κάναμε στην 1η δραστηριότητα και να συνδέσουμε το το Arduino σε μια θύρα USB
- Η εντολή που θα χρειαστείτε εδώ είναι η:



Γ. Επεκτάσεις (μόνο για τη C++)

- Δοκιμάστε να χρησιμοποιήσετε τη συνάρτηση tone αντί της analogWrite
`tone(pin, frequency, duration);`
 Παράμετροι:
 pin: Η έξοδος που θέλουμε να ενεργοποιήσουμε
 frequency: η συχνότητα σε Hz
 duration: η διάρκεια σε milliseconds

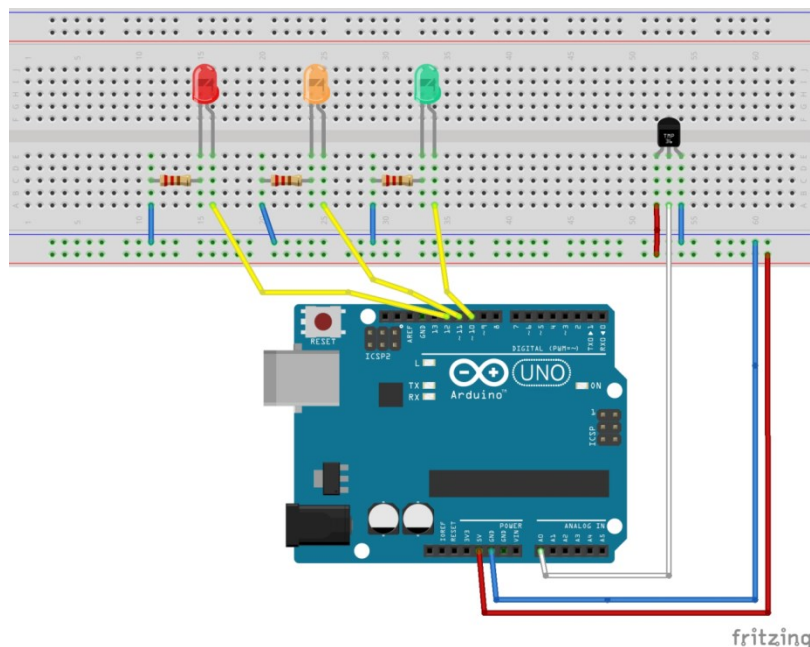
Δραστηριότητα 5^η – Temperature sensor + LED (Analog Input + Serial port)

Σκοπός: Να μετρήσουμε την τιμή που μας δίνει ο αισθητήρας θερμοκρασίας και να στείλουμε τη μέτρηση μέσω της θύρας USB στον υπολογιστή.

Α. Υλοποιήστε το παρακάτω κύκλωμα (Αν έχετε υλοποιήσει την προηγούμενη δραστηριότητα απλά αποσυνδέστε τον πιεζοηλεκτρικό αισθητήρα και συνδέστε τον αισθητήρα θερμοκρασίας)

Θα χρειαστείτε:

- 1 κόκκινο LED
- 1 πορτοκαλί LED
- 1 πράσινο LED
- 1 διακόπτη (pushbutton)
- 3 αντιστάσεις 220Ω (κόκκινο-κόκκινο-καφέ-χρυσό)
- 1 αντίσταση 10KΩ (καφέ-μάυρο-πορτοκαλί-χρυσό)
- 1 αισθητήρα θερμοκρασίας TMP36
- Καλώδιο USB + καλώδια συνδέσεων
- Arduino + Breadboard



B. Προγραμματισμός Arduino

1. Με το περιβάλλον προγραμματισμού IDE του Arduino (C++)

- Χρησιμοποιήστε την εντολή

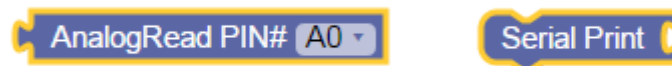
```
analogRead(analogInputPin); // Μετατρέπει το επίπεδο τάσης που υπάρχει στην συγκεκριμένη αναλογική είσοδο (0-5V) μέσω ενός Analog to Digital Converter (ADC) σε μία τιμή 0-1023 (10bit) την οποία μπορούμε να χρησιμοποιήσουμε μέσα στο πρόγραμμά μας.
```
- Για να στείλουμε την τιμή μέσω της θύρας USB στον Η/Υ θα πρέπει:
 - Μέσα στη συνάρτηση `setup()` να καλέσουμε τη μέθοδο:

```
Serial.begin(9600); // αρχικοποίηση σειριακής επικοινωνίας
```
 - Να χρησιμοποιήσουμε τη μέθοδο

```
Serial.println(sensorValue); // στέλνει την τιμή sensorValue μέσω της θύρας USB στον ΗΥ
```
- Για να δείτε τα δεδομένα που μεταδίδονται στη σειριακή θύρα επιλέξτε **Tools->Serial Monitor**

2. Με το περιβάλλον προγραμματισμού Blocklyduino

- Το Blocklyduino διαθέτει τις εντολές:



3. Με το περιβάλλον προγραμματισμού Scratch for Arduino

- Δεν ξεχνάμε να μεταφορτώσουμε στο Arduino το `scetch S4AFirmware16.ino` με το IDE του Arduino όπως κάναμε στην 1η δραστηριότητα και να συνδέσουμε το το Arduino σε μια θύρα USB
- Η εντολή που θα χρειαστείτε εδώ είναι η:

- Εδώ δεν χρειάζεται να στείλετε σειριακά τα δεδομένα για να τα δείτε καθώς εμφανίζονται συνέχεια στον πίνακα εισόδων όπως είδαμε στην δραστηριότητα 2.

Γ. Επεκτάσεις

- Τροποποιήστε το πρόγραμμα ώστε το πράσινο LED να ανάβει όταν η τιμή της αναλογικής εισόδου είναι από 0-100, το πορτοκαλί από 101-150 και το κόκκινο από 151 και πάνω

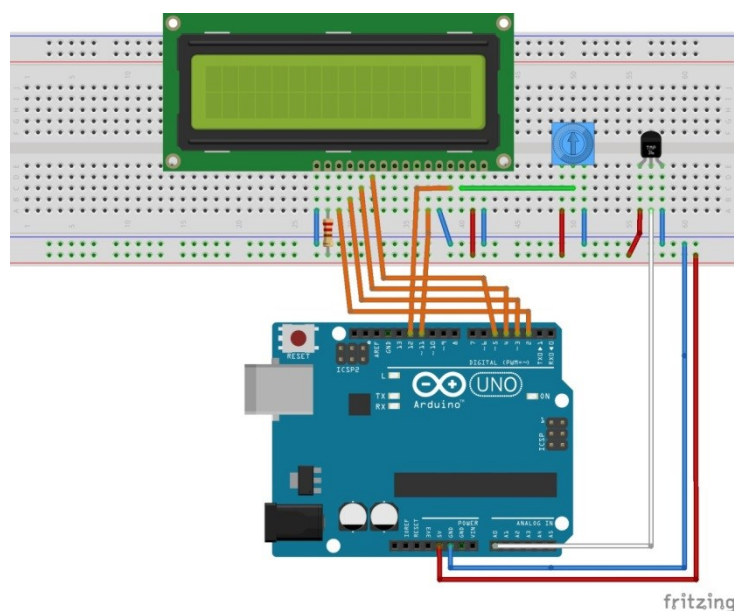
Δραστηριότητα 6^η – Temperature sensor + LED + LCD (Analog Input + LCD Display + Libraries + Scaling)

Σκοπός: Να μετρήσουμε τη θερμοκρασία του δωματίου με τον αισθητήρα θερμοκρασίας και να απεικονίσουμε την ένδειξη “Room Temperature” καθώς και τη μέτρηση στην οθόνη LCD σε βαθμούς Κελσίου.

A. Υλοποιήστε το παρακάτω κύκλωμα (Διατηρείστε μόνο τον αισθητήρα θερμοκρασίας από την προηγούμενη δραστηριότητα)

Θα χρειαστείτε:

- Την οθόνη LCD (2X16 χαρακτήρων)
- 1 ποτενσιόμετρο 10KΩ
- 1 αντίσταση 220Ω (κόκκινο-κόκκινο-καφέ-χρυσό)
- 1 αισθητήρα θερμοκρασίας TMP36
- Καλώδιο USB + καλώδια συνδέσεων
- Arduino + Breadboard



B. Προγραμματισμός Arduino

Η συγκεκριμένη δραστηριότητα μπορεί να υλοποιηθεί μόνο με το περιβάλλον προγραμματισμού IDE του Arduino (C++)

- Θα πρέπει να συμπεριλάβουμε τη βιβλιοθήκη LiquidCrystal για το σκοπό αυτό θα πρέπει:
 - Να επιλέξουμε από το μενού Sketch->Import Library->LiquidCrystal
Θα πρέπει να εμφανιστεί στην πρώτη γραμμή του προγράμματος η εντολή: #include <LiquidCrystal.h>
 - Να γράψουμε αμέσως μετά την εντολή
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
Για να δημιουργήσουμε ένα αντικείμενο με όνομα lcd τύπου LiquidCrystal. Οι παράμετροι είναι οι ψηφιακές E/E που έχουμε συνδέσει το LCD display.
- Μέσα στη setup() να γράψουμε την μέθοδο:
lcd.begin(16, 2); // Το display έχει 16 χαρακτήρες ανά γραμμή και μπορεί να απεικονίσει ταυτόχρονα 2 γραμμές
- Τέλος μπορούμε να χρησιμοποιήσουμε τις εξής μεθόδους του αντικειμένου lcd για να απεικονίσουμε μηνύματα στο display:
 - *lcd.clear();* // Καθαρίζει το display
 - *lcd.setCursor(0, 1);* // Θέτει τη θέση που θα τυπωθεί ο επόμενος χαρακτήρας (πρώτη θέση (0) , δεύτερη γραμμή (1))
 - *lcd.print("Temperature");* // Τυπώνει την παράμετρο σε μορφή κειμένου ξεκινώντας από τη θέση που ορίσαμε με την setCursor
- Το πρόβλημα που έχουμε τώρα είναι ότι δεν ξέρουμε ακόμη την πραγματική θερμοκρασία που μετράει ο αισθητήρας αλλά μία μέτρηση (0-1023) χωρίς μέγεθος. Η διαδικασία που πρέπει να ακολουθήσουμε ονομάζεται scaling (μετατροπή δηλαδή μιας κλίμακας σε μια άλλη)
- Στο Arduino υπάρχει η συνάρτηση map
int outVal = map(inVal, 0, 1023, 0, 30);
η οποία δουλεύει όμως μόνο με ακέραιους, οπότε για να πετύχουμε μεγαλύτερη ακρίβεια στις μετρήσεις μας, δεν θα χρησιμοποιήσουμε τη map() αλλά μαθηματικές πράξεις με πραγματικούς αριθμούς (float)
- Πρώτα θα πρέπει να μετατρέψουμε την μέτρηση σε τάση (0-5V) και κατόπιν σε βαθμούς κελσίου.
 - Για το λόγο αυτό θα χρησιμοποιήσουμε μία μεταβλητή voltage και μία temperature και οι δύο τύπου float
 - *float voltage, temperature;*
 - Θα μετατρέψουμε το πηγαίο νούμερο της αναλογικής μας εισόδου σε τάση (0-5V) ως εξής:
 - *voltage = (analogRead(0) / 1024.0) * 5.0;*

- Αν θέλουμε να δούμε ότι είναι σωστή η πράξη στέλνουμε την voltage στη σειριακή θύρα όπως είδαμε πριν ή μπορούμε πλέον να χρησιμοποιήσουμε την `lcd.print(voltage);` και να την απεικονίσουμε στο display.
- Από το datasheet (εγχειρίδιο) του αισθητήρα θερμοκρασίας εξάγουμε ότι ο τύπος για τον υπολογισμό της θερμοκρασίας από την τάση δίνεται είναι ο εξής:

$$temperature = (voltage - 0.5) * 100.0;$$

- Τέλος εμφανίζουμε στην 1η γραμμή του display το μήνυμα "Temperature:" και στη 2η γραμμή την τιμή της μεταβλητής temperature

Γ. Επεκτάσεις

- Κρατήστε τη μέγιστη και την ελάχιστη θερμοκρασία και απεικονίστε τις στη 2^η γραμμή του display. Η πρώτη γραμμή να γίνει "TEMP MIN MAX"
- Δοκιμάστε να συνδέσετε ένα διακόπτη σε μια ελεύθερη ψηφιακή είσοδο και κάθε φορά που πατιέται ο διακόπτης να εμφανίζεται η θερμοκρασία εναλλάξ σε βαθμούς Κελσίου και Φαρενάιτ. Ο τύπος είναι $F = C * 1,8 + 32$. Θα πρέπει δίπλα από την τιμή της θερμοκρασίας να εμφανίζεται το σύμβολο "C" για Κελσίου και "F" για Φαρενάιτ.
- Δοκιμάστε να κάνετε scrolling το κείμενο "Arduino" πρώτα στην 1^η γραμμή του LCD και αφού εξαφανιστεί από την 1^η γραμμή να εμφανίζεται στην 2^η.



Αυτό το υλικό διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/>). Η αναφορά σε αυτό θα πρέπει να γίνεται ως εξής:

«Κατασκευές και Προγραμματισμός Arduino με C++, Blocklyduino και Scratch», Παπαπαρασκευάς Πάρης ΠΕ20, ΚΕΠΛΗΝΕΤ Χίου, ΣΕΠ Χίου
Μάρτιος 2015