

Τεχνολογία Αισθητήρων και  
Μικροσυστημάτων  
Θέμα : «Εισαγωγή στο Arduino»



## Περιεχόμενα

Περιγραφή Προσομοίωσης Arduino.....	3
Ηλεκτρολογικό Σχέδιο.....	6
Κώδικας Wiring.....	7

## Περιγραφή Προσομοίωσης Arduino

Σε αυτή την εργαστηριακή άσκηση θα γίνει η υλοποίηση ενός ηλεκτρονικού κυκλώματος, το οποίο θα προσομοιώνει τη λειτουργία ενός φάρου. Το κύκλωμα αυτό θα έχει τα ακόλουθα χαρακτηριστικά :

- Έλεγχος φωτεινότητας κάθε 10 sec
- Αν είναι σκοτεινά, το φως του φάρου ανάβει σταδιακά και μένει αναμμένο για 2 sec
- Μέτρηση θερμοκρασίας κάθε 20 sec και εμφάνιση του αποτελέσματος στο Serial Monitor, στη μορφή Temp= x C
- Αν ανιχνευθεί κίνηση κοντά στον φάρο, εμφανίζεται κατευθείαν ειδοποίηση στο Serial Monitor

Για την κυκλωματική υλοποίηση αυτού του φάρου θα χρησιμοποιηθούν τα ακόλουθα εξαρτήματα :

- Μικροελεγκτής : Arduino Uno
- Φως Φάρου : LED + αντίσταση  $R = 220\Omega$
- Φωτεινότητα : Φωτοαντίσταση + αντίσταση  $R = 10\text{ k}\Omega$
- Αισθητήρας Θερμοκρασίας : TMP36
- Αισθητήρας θέσης : PIR

Αυτά τα εξαρτήματα επιλέχθηκαν ως τα πιο ευρέως χρησιμοποιούμενα από την αγορά, μιας και δεν υπήρχαν περιορισμοί στο μέγεθος ή στις απαιτήσεις των αισθητήρων, που να επηρεάζουν την επιλογή αυτή των εξαρτημάτων.

Για το LEDακι επιλέχθηκε αντίσταση 220 Ω , έτσι ώστε να επιτυγχάνονται οι ακόλουθες τιμές στο κύκλωμα:

Μέγιστη ένταση ρεύματος στο LED :

$$I_{\max} = 20\text{mA}$$

Χρήση της θύρας του Arduino, που δίνει τάσης τροφοδοσίας :

$$V_s = 5\text{ V}$$

Πτώση τάσης στο LED :

$$V_{\text{Drop}} = 1.8\text{ V}$$

Επομένως, θα πρέπει να επιλεγεί μια τιμή αντίστασης, που να επαληθεύει τη σχέση :

$$R_{\min} = \frac{V_s - V_{\text{drop}}}{I_{\max}} \Leftrightarrow$$

$$R_{\min} = \frac{5 - 1.8}{0.02} \Leftrightarrow$$

$$R_{\min} = 160\ \Omega$$

Συνεπώς, θα πρέπει να επιλεγεί μια αντίσταση εμπορίου όσο πιο κοντά σε αυτή την αντίσταση γίνεται. Για αυτό το λόγο και επιλέγεται η τιμή  $R = 220\ \Omega$  . Με αυτή την επιλογή καθίσταται δυνατό το σταδιακό άναμμα και σβήσιμο του LED με τη βοήθεια της εντολής `analogWrite` και ενός `counter`, ο οποίος αυξάνεται από 0 έως 255 (ή μειώνεται από 255 σε 0) μεταβάλλοντας το duty cycle από 0 σε 100%.

Όσον αφορά τον υπολογισμό της θερμοκρασίας χρησιμοποιείται ο αισθητήρας TMP36 με τα ακόλουθα χαρακτηριστικά :

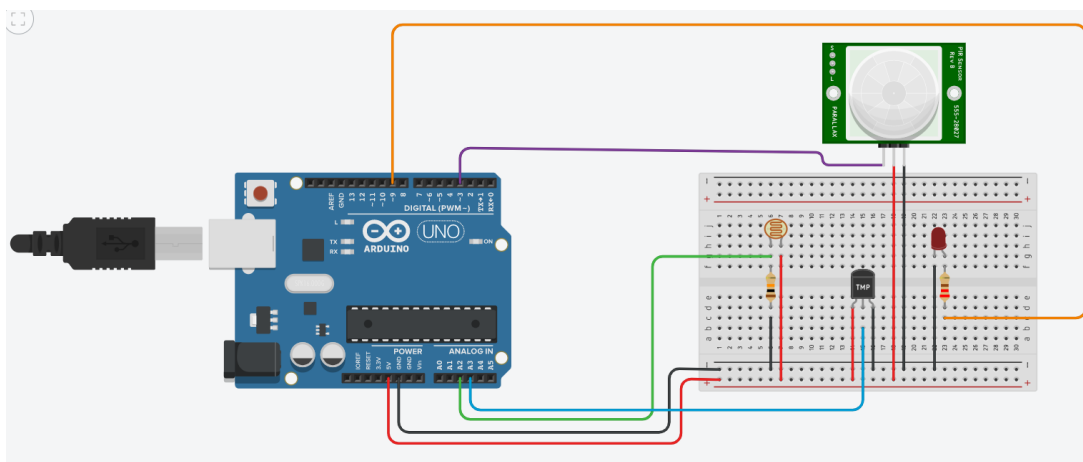
- $V_{offset} = 0.5 \text{ V}$
- Μεταβολή 10mV αντιστοιχεί σε μεταβολή 1 °C
- Ισχύει η σχέση :  $V_{out} = \frac{V_{in} * V_{op}}{1024}$

Αξιοποιώντας, λοιπόν, αυτά τα χαρακτηριστικά επιτυγχάνεται η εμφάνιση της θερμοκρασίας σε βαθμούς Κελσίου σύμφωνα με την ακόλουθη σχέση:

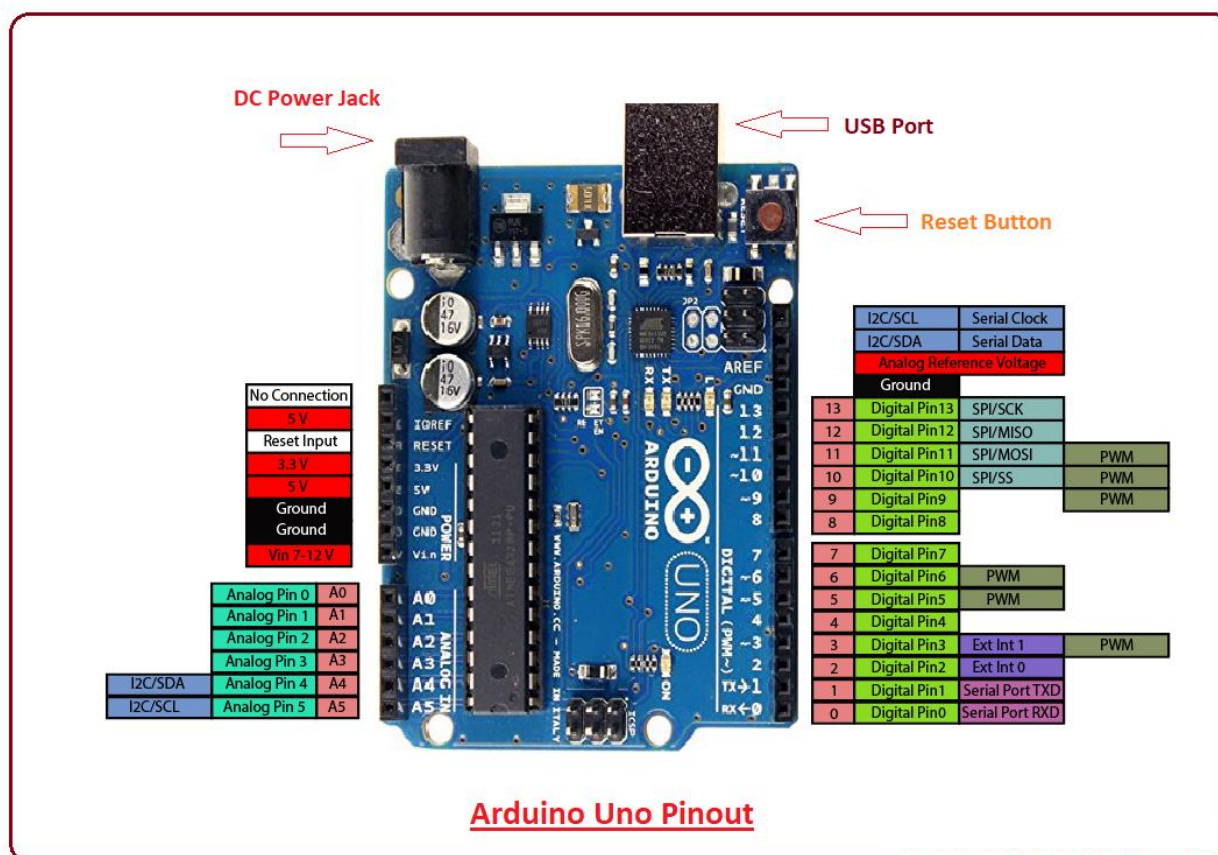
$$\text{Temperature} = \frac{V_{out} - V_{offset}}{0.1} \quad (^\circ\text{C})$$

## Ηλεκτρολογικό Σχέδιο

Αξιοποιώντας, λοιπόν, τα εξαρτήματα αυτά και υλοποιώντας τις κατάλληλες συνδέσεις προκύπτει το ακόλουθο κύκλωμα στο περιβάλλον κυκλωματικών υλοποιήσεων της Tinkercad.



Η κυκλωματική αυτή διάταξη προέκυψε σύμφωνα με τον pinout του Arduino UNO, όπως φαίνεται ακολούθως.



## Κώδικας Wiring

Ακολουθως φαίνεται ο κώδικας που χρησιμοποιήθηκε στο περιβάλλον προσομοίωσης της Tinkercad, καθώς και στο προγραμματιστικό περιβάλλον του Arduino

```
/*
Konstantinos Spathis AM : 03117048
*/

/*DECLARE ALL THE VARIABLES*/

//pin declaration
int LEDpin = 9 ; //led pin PWM
int LRpin = A2 ; //lightresistance pin
int TMPpin = A3 ; //temp sensor pin
int PIRpin = 3 ; //PIR pin PWM

//helper variables declaration
int LEDstate ;
int LRvalue;
int pirValue;
int tempValue;
int fos;
int kin = 0;
int count = 0;

unsigned long tora = 0;
unsigned long prin = 0;
unsigned long prinON = 0;
unsigned long prinOFF = 0;

void BrightnessChecker() {
    //check if it is day or night
    int LRhelp = analogRead(LRpin);
    if (LRhelp < 512)
        fos=0; //night
    else
        fos=1; //day
}
```

```

void stadiakoSbisimo() {
    //turn off led step by step
    for(int ldof=255; ldof>=0; ldof-=15) {
        analogWrite(LEDpin,ldof);
        delay(9); //just a little bit of delay to see the difference
    }
    LEDstate=0;
    digitalWrite(LEDpin,LOW);
}

void stadiakoAnama() {
    //turn on led step by step
    for(int ldon=0; ldon<=255; ldon+=15) {
        analogWrite(LEDpin,ldon);
        delay(9); //just a little bit of delay to see the difference
    }
    LEDstate=1;
    digitalWrite(LEDpin,HIGH);
}

void LEDkatastasi() {
    // declare state of LED
    if (LEDstate == 1)
        Serial.println(" LED is ON ");
    else
        Serial.println(" LED is OFF ");
}

float convert2Kelsiou (int tempVAL){

    // Value from TMP36 sensor (type int) --> temperature in Celsius.
    float tempHelp = tempVAL * 5000.0 / 1024.0;
    float tempKels = (tempHelp - 500) / 10;
    return tempKels;
}

```

---



```

void ThermoPrinter(int tempVAL){
    // print temperature to 2 decimal points
    float tempKels = convert2Kelsiou(tempVAL);
    Serial.print("Temp [C]: ");
    Serial.print(tempKels, 2);
    Serial.print("\n");
}

void AnixneusiKinisis(int pirVAL){
    //detect movement
    if (pirVAL == 1 & kin==0) {
        Serial.println(" Movement Detected ");
        kin=1;
    }
    if (pirVAL == 0){
        kin = 0;
    }
}

void setup()
{
    Serial.begin(9600); //set serial communication
    //set input-output modes for the pins
    pinMode(LEDpin, OUTPUT); //set output for the LED
    pinMode(LRpin, INPUT); //collect data from the lightresistance
    pinMode(TMPpin, INPUT); //collect data from the temperature sensor
    pinMode(PIRpin, INPUT); //collect data from the PIR sensor
}

```

```

void loop() {
    unsigned int help=1000,helpON=2000,helpOFF=1000;

    //check if it is day or night
    BrightnessChecker();

    //store the current time
    tora = millis();

    //check if it is night & 2 sec pasted --> slowly turn off LED
    if (tora > prinON + helpON && fos==0 && LEDstate==1) {
        prinOFF=tora;
        stadiakoSbisimo();
    }

    //check if it is night & 1 sec pasted --> slowly turn on LED
    if (tora > prinOFF + helpOFF && fos==0 && LEDstate==0) {
        prinON=tora;
        stadiakoAnama();
    }

    //if 20 sec pasted get the temp
    if (tora > prin + 10*help) {
        prin=tora;
        count++;
        tempValue = analogRead(TMPpin);
        ThermoPrinter(tempValue);
        if (count%3==0) {
            //print led state
            LEDkatastasi();
        }
    }

    pirValue = digitalRead(PIRpin); //diavasma ejodou pir sensor
    AnixneusiKinisis(pirValue);
}

```

---