



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΣΘΗΤΗΡΩΝ

3^η Εργαστηριακή Αναφορά-Εισαγωγή στο Arduino

στο μάθημα «**Τεχνολογία Αισθητήρων και
Μικροσυστημάτων**»

των φοιτητών

Σκόρδα Στεφανία, Α.Μ: 03118852

Ράτσα Ηλίας, Α.Μ: 03118817

Σερλής Εμμανουήλ-Αναστάσιος, Α.Μ: 03118125

8^ο Εξάμηνο

Ιούνιος 2022

Σκοπός της εργασίας:

Σκοπός της άσκησης είναι η υλοποίηση λειτουργίας ενός φάρου με 3 αισθητήρες. Πιο συγκεκριμένα, στη διάθεσή μας έχουμε μία φωτοαντίσταση LDR, έναν αισθητήρα θερμοκρασίας TMP36 και έναν αισθητήρα κίνησης PIR. Η υλοποίηση θα πραγματοποιηθεί με Arduino και ο κώδικας θα πρέπει να δίνει στον φάρο με τους 3 παραπάνω αισθητήρες τις εξής λειτουργίες:

- Κάθε 10 s, γίνεται έλεγχος φωτεινότητας.
- Εάν είναι σκοτεινά, το φως του φάρου (π.χ. ένα LED) ανάβει σταδιακά και μένει αναμμένο για 3 s. Στη συνέχεια σβήνει σταδιακά και μένει σβηστό για 2 s (αυτά τα 5 s μπορούν να θεωρηθούν αμελητέα στους υπόλοιπους χρονικούς υπολογισμούς).
- Κάθε 20 s, μετράται με έναν αισθητήρα θερμοκρασίας (π.χ. TMP36) η θερμοκρασία περιβάλλοντος και εμφανίζεται στο Serial Monitor, στη μορφή: Temp is x C
- Εάν ανιχνευθεί κίνηση κοντά στον φάρο, εμφανίζεται κατευθείαν ειδοποίηση στο Serial Monitor.

Για την ανάπτυξη κώδικα χρησιμοποιήθηκε ο editor του Arduino ενώ για την δοκιμή και προσομοίωση της λειτουργικότητας του φάρου χρησιμοποιήθηκε ο 3D Simulator του TinkerCad. Παρακάτω παρατίθεται ο υλοποιημένος κώδικας από το περιβάλλον του Visual Studio Code:

```
int LedOut = 13; //ορισμός pin 14 ελέγχου του φάρου
int light_pin = A0; //pin για μέτρηση στάθμης φωτός
int temp_pin = A1; //pin για μέτρηση θερμοκρασίας
int move_pin = 8; //pin για ανίχνευση κίνησης
float brightness=-227.00;
const float night=0;
float analog_value;
unsigned long previous1=0;
unsigned long previous2=0;
float step=10;

void temperature_level(){
    int temp_value = analogRead(temp_pin);
    float temp = temp_value*5000.0/1024.0;
    float temp_C = (temp-500)/10.0; //turn to celcius
    Serial.print("Temp = ");
    Serial.print(temp_C);
    Serial.println("C");
}
```

```

void movement() {
    bool movement = digitalRead(move_pin);
    //Serial.println(movement);
    if(movement == HIGH){
        Serial.println("Movement detected");
    }
}

void setup() {
    pinMode(move_pin, INPUT);
    pinMode(LedOut, OUTPUT);
    Serial.begin(9600);
}

void loop(){
    if(millis() >= previous1){
        float value=analogRead(light_pin);
        analog_value=map(value,512,1023,0,255); //-228 έως 230
        Serial.print("Brightness value is = ");
        Serial.println(analog_value);
        previous1 = millis() + 1000; //κάθε 10 sec μέτρηση
    }
    if(analog_value<night){ //when we have insufficient light i.e when
analog_value <-150
        analogWrite(LedOut,brightness); //move previous brightness value
to led
        if(brightness<=-228.0){ //add delay when we reach minimum led
value
            delay(2000);
        }
        if(brightness>=230.0){ //delay for max led value
            delay(3000);
        }
        brightness = brightness + step;
        Serial.println(brightness);
        if (brightness <= -228.0 || brightness >= 230.0){ //change mode
from increasing brightness to decreasing(and vice versa)
            step = -step;
        }
    }

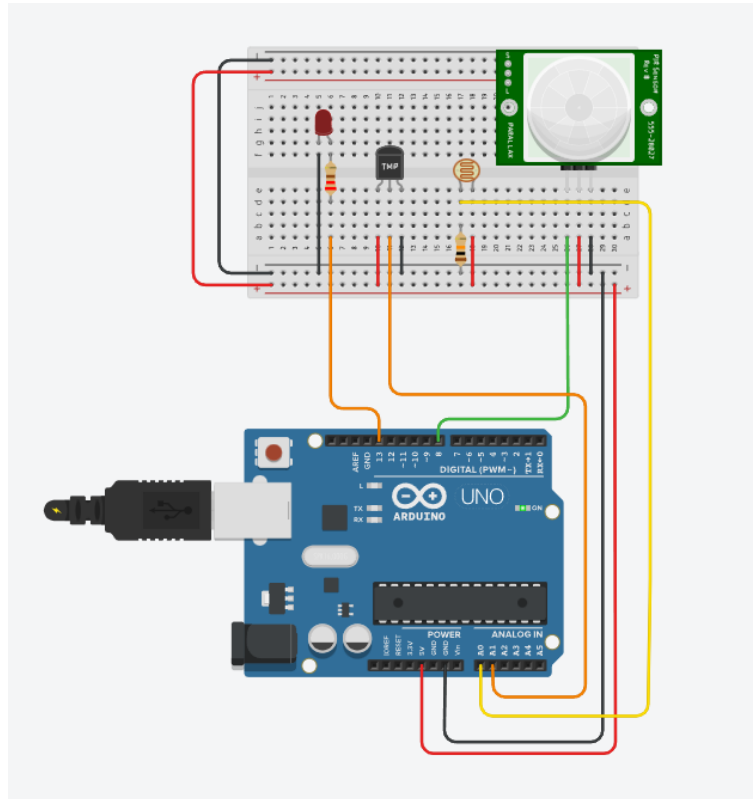
    if(millis() >= previous2){
        temperature_level();
        previous2 = millis()+2000; //kathe 20 sec metrisi
    }

    movement();
}

```

}

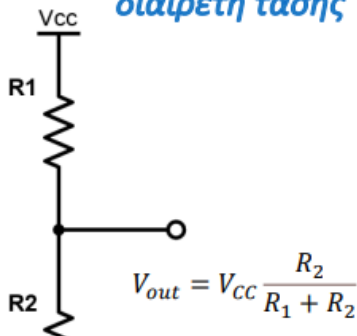
Το κύκλωμα που σχεδιάστηκε στον 3D Simulator του TinkerCad σύμφωνα με τα pins εισόδου και εξόδου που ορίσαμε παραπάνω μέσω του κώδικα. Το σχηματικό παρατίθεται παρακάτω:



Περιγραφή Λειτουργίας Κυκλώματος

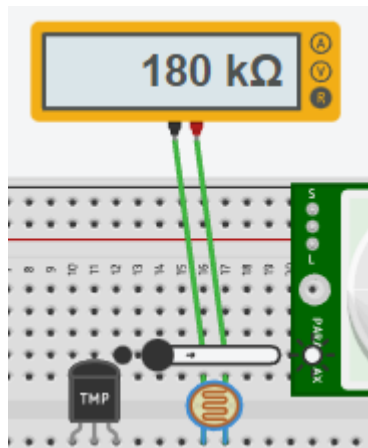
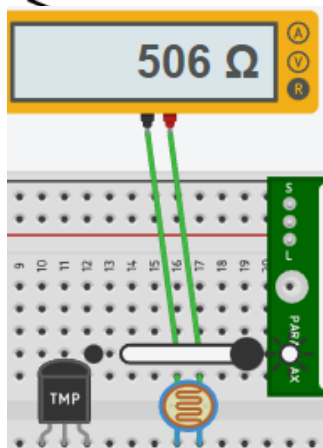
I. Ανιχνευτής Φωτός

Συνδεσμολογία διαιρέτη τάσης



Για την μέτρηση της στάθμης φωτός επιλέχθηκε μία φωτοαντίσταση LDR και μία συνδεσμολογία διαιρέτη τάσης σε σειρά με αντιστάσεις. Αυτή η συνδεσμολογία επιλέχθηκε έτσι ώστε σε συνθήκες υψηλού φωτισμού η τιμή της αντίστασης LDR.

$$V_{out} = \frac{R_s}{R_s + R_L} V_{in} \text{ όπου } V_{in} = 5V \text{ και } R_s = 10k\Omega$$



Ακραίες Τιμές LDR:

Έντονο Φως : $R_L = 506\Omega$

Απόλυτο Σκοτάδι : $R_L = 180k\Omega$

II. Αισθητήρας Θερμοκρασίας TMP36



Για την μέτρηση της θερμοκρασίας έγινε χρήση του αισθητήρα θερμοκρασίας TMP36. Η μετατροπή της αναλογικής μέτρησης σε θερμοκρασία Κελσίου γίνεται με τη χρήση της σχέσης $temp_c = \frac{V_{out} - 500}{10}$, αφού από το datasheet η εξάρτηση τάσης-θερμοκρασίας είναι $10mV/^{\circ}C$ με ένα offset $500mV$ και η ανάλυση του ADC του Arduino UNO είναι 1024 δείγματα.

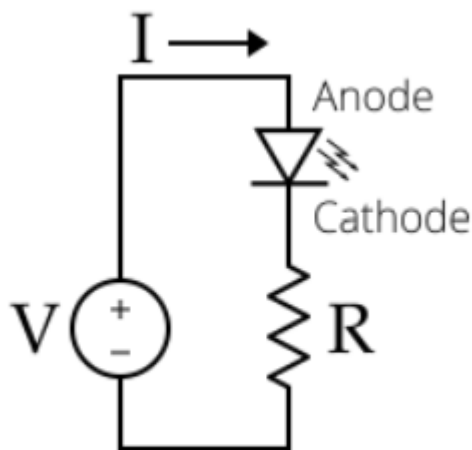
III. Αισθητήρας Κίνησης



Για την ανίχνευση κίνησης έγινε χρήση του αισθητήρα κίνησης PIR (Passive Infrared Sensor). Ο συγκεκριμένος αισθητήρας όταν ανιχνεύσει κίνηση μπροστά του στέλνει για ένα μικρό χρονικό διάστημα παλμό με τάση ίση με 5V, όσο είναι και η τροφοδοσία του. Μας επιτρέπεται έτσι να χρησιμοποιήσουμε το ψηφιακό pin και να μετρήσουμε την τάση του PIR για να μπορέσει να γίνει η ανίχνευση κίνησης.

IV. Επιλογή αντίστασης για το LED

Προκειμένου να αποφευχθεί η ανάπτυξη πολύ υψηλών ρευμάτων κατά μήκος του LED, συνδέουμε μία αντίσταση σε σειρά με το LED όπως φαίνεται στο παρακάτω σχήμα:



Επιθυμούμε να έχουμε ρεύμα $I_{LED} = I_R = 20\text{mA}$ και πτώση τάσης στα άκρα της αντίστασης ίση με $V_R = V - V_{LED} = 5 - 0.7 = 4.3\text{V}$. Έτσι, καταλήγουμε σε τιμή αντίστασης ίση με:

$R = V_R / I_R = 215\Omega$, με την πιο κοντινή τιμή αντίστασης σε αυτήν να είναι των **220Ω**