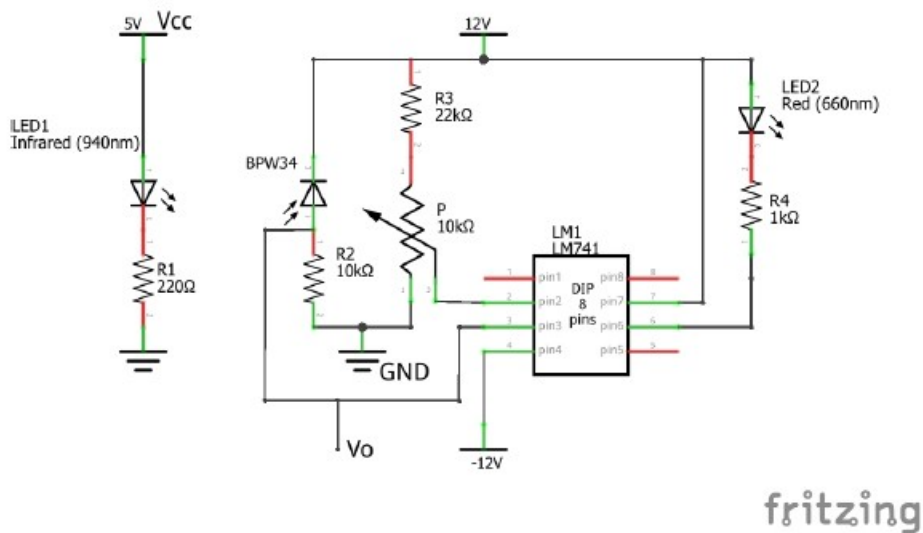


Γιόελ Γιάννη:2019030116  
Ανδρέας Καρόγιαννης:2019030066

### Ηλεκτρικές Μετρήσεις και Αισθητήρες 3η Εργαστηριακή Άσκηση

Στην τρίτη εργαστηριακή άσκηση ασχοληθήκαμε με τον σχεδιασμό αισθητήρα ανίχνευσης αντικειμένου-απόστασης.

1)



Σχ. 5

Για το κύκλωμα που φαίνεται παραπάνω, χρησιμοποιήθηκε ένα ποτενσιόμετρο, 4 αντιστάσεις ένα κανονικό LED, ένα LED υπέρυθρης ακτινοβολίας, μια φωτοδίοδος και ένας τελεστικός ενισχυτής. Η διάταξη δουλεύει ως αισθητήρας ανίχνευσης αντικειμένου-απόστασης και στο εργαστήριο χρειάστηκε να ρυθμιστεί το ποτενσιόμετρο ώστε μόλις που ανάβει το LED όταν η φωτοδίοδος κοιτάει το LED υπέρυθρης ακτινοβολίας.

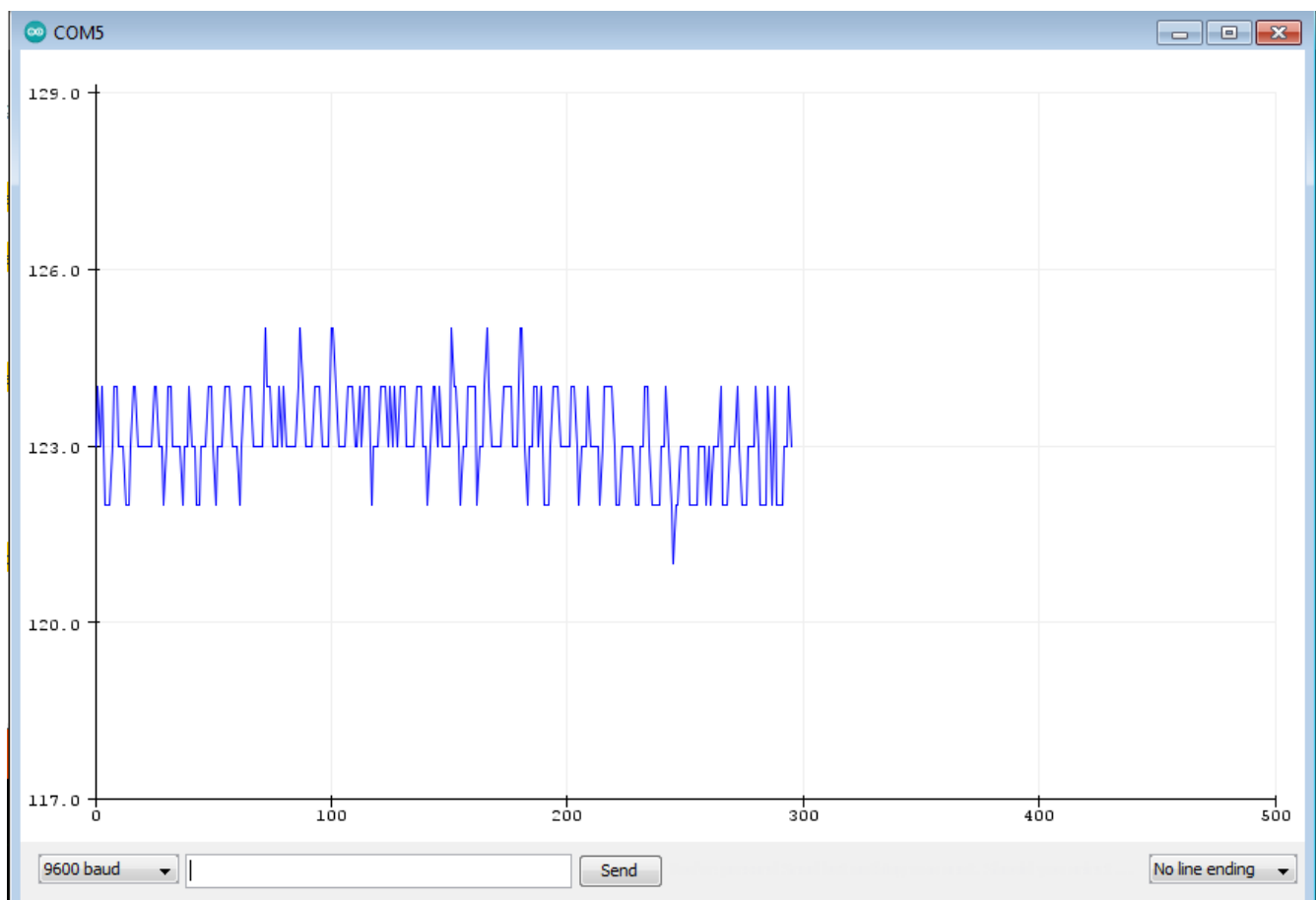
Το ποτενσιόμετρο στο κύκλωμα μπορεί να χρησιμοποιηθεί για τη ρύθμιση της ευαισθησίας του συστήματος. Ρυθμίζοντας την

αντίσταση του ποτενσιόμετρου, μπορούμε να ρυθμίσουμε το κέρδος του λειτουργικού ενισχυτή LM741, ο οποίος θα ενισχύσει το σήμα εξόδου της φωτοδιόδου. Το ποτενσιόμετρο μας έχει ρυθμιστεί σε τιμή χαμηλής αντίστασης ώστε το κέρδος του LM741 να είναι υψηλό και το σήμα εξόδου να είναι μεγάλο. Αυτό σημαίνει ότι το σύστημα θα είναι πιο ευαίσθητο στις αλλαγές στην υπέρυθρη ακτινοβολία που προσπίπτει στη φωτοδίοδο. Επομένως, ρυθμίζοντας το ποτενσιόμετρο σε σχετικά χαμηλή τιμή, μπορούμε να βελτιστοποιήσουμε την ευαισθησία του συστήματος για τη συγκεκριμένη εφαρμογή.

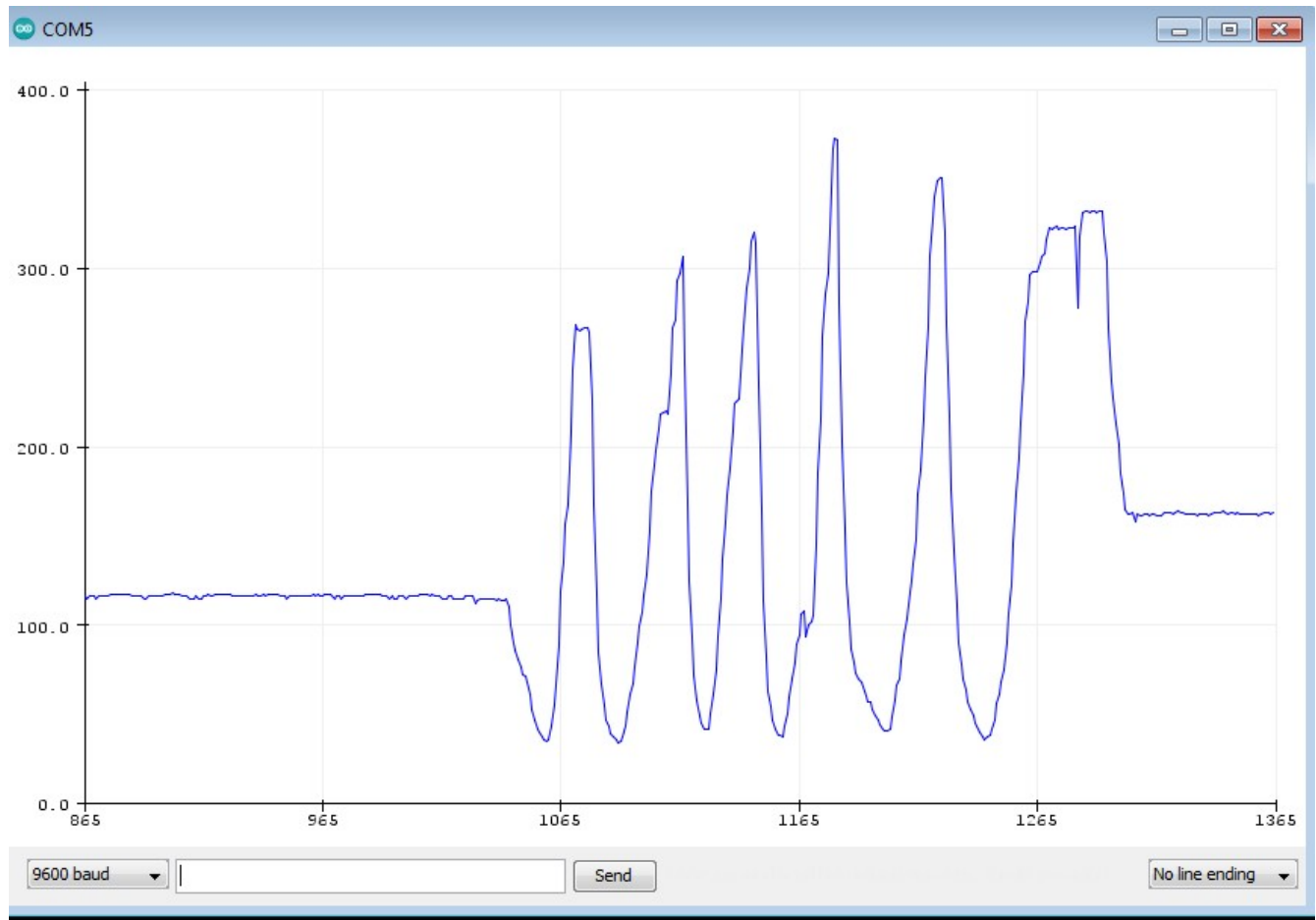
Η τιμή τάσης  $V_o$  που καταγράφηκαν στο εργαστήριο σε αυτή την κατάσταση του κυκλώματος είναι:

$$V_o = 0.645V$$

Μετά συνδέθηκε το παραπάνω κύκλωμα με το arduino που χρησιμοποιούμε στο εργαστήριο και καταγράφηκε η παρακάτω κυματομορφή για την έξοδο  $V_o$  :

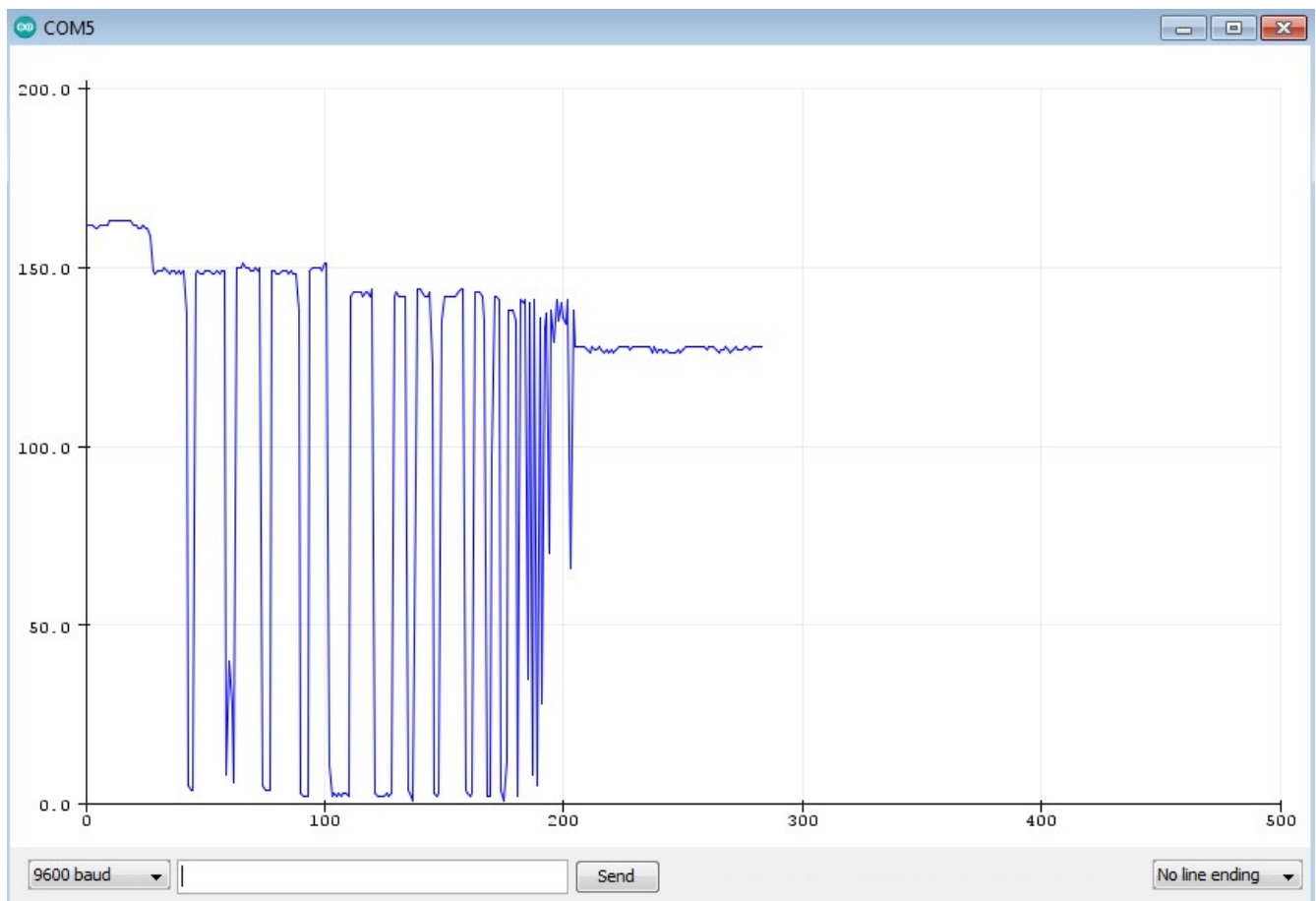


Στη συνέχεια, μεταβλήθηκε η απόσταση ανάμεσα στα δύο κυκλώματα μετατοπίζοντας το κύκλωμα που έχει το LED υπέρυθρης ακτινοβολίας και καταγράφηκε η παρακάτω κυματομορφή για την τάση.



Όσο αυξομειώνεται η απόσταση μεταξύ των κυκλωμάτων παρατηρούνται και αυξομειώσεις στην τάση  $V_o$  του κυκλώματος. Συγκεκριμένα όταν αυξάνεται η απόσταση μειώνεται η τάση και αντίστροφα.

Ακόμα, καταγράφηκε η κυματομορφή της τάσης  $V_o$  όταν τοποθετήθηκε εμπόδιο ανάμεσα στα κυκλώματα.

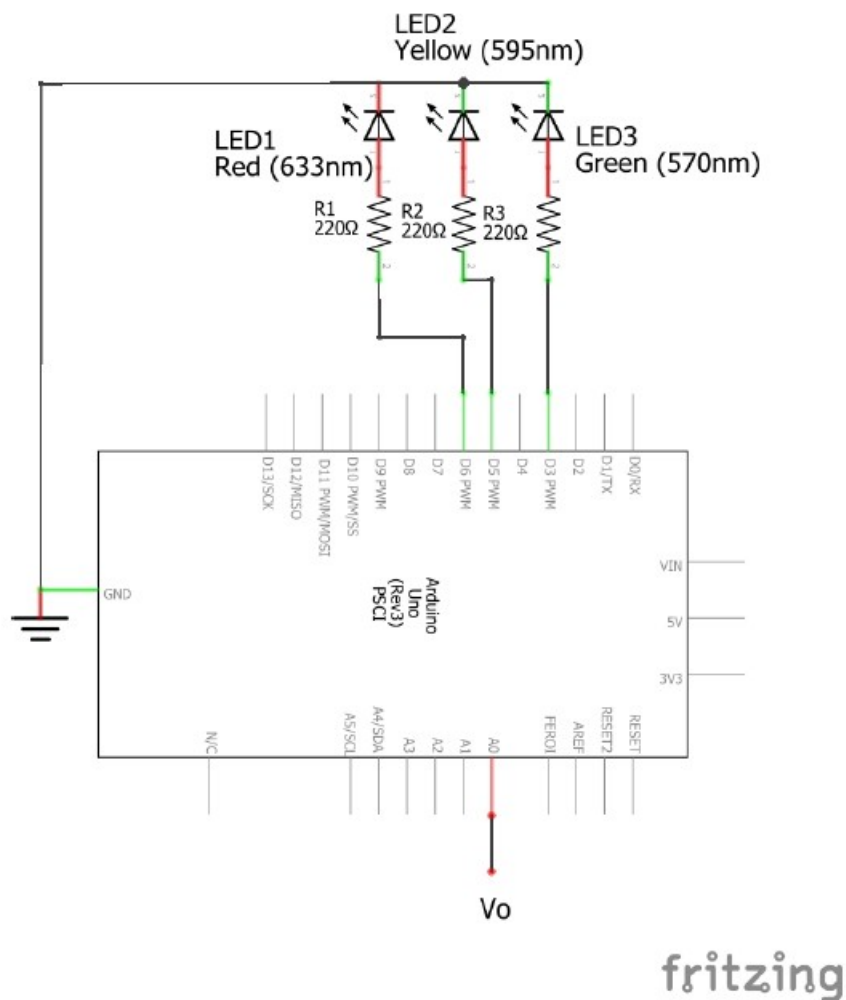


Κάθε φορά που τοποθετήθηκε το εμπόδιο ανάμεσα στα δύο κυκλώματα η τάση  $V_o$  φαίνεται ότι σχεδόν μηδενίζεται, το οποίο είναι λογικό καθώς δεν λαμβάνει την ακτινοβολία από το LED.

Τέλος, μετρήθηκε η τάση  $V_o$  για τιμές τάσης εισόδου  $V_{cc} = 5, 3.3, 2.5V$  χωρίς εμπόδια και την απόσταση μεταξύ των κυκλωμάτων σταθερή.

$V_{cc}=5V$	$V_o=0.633V$
$V_{cc}=3.3V$	$V_o=0.343V$
$V_{cc}=2.5V$	$V_o=0.213V$

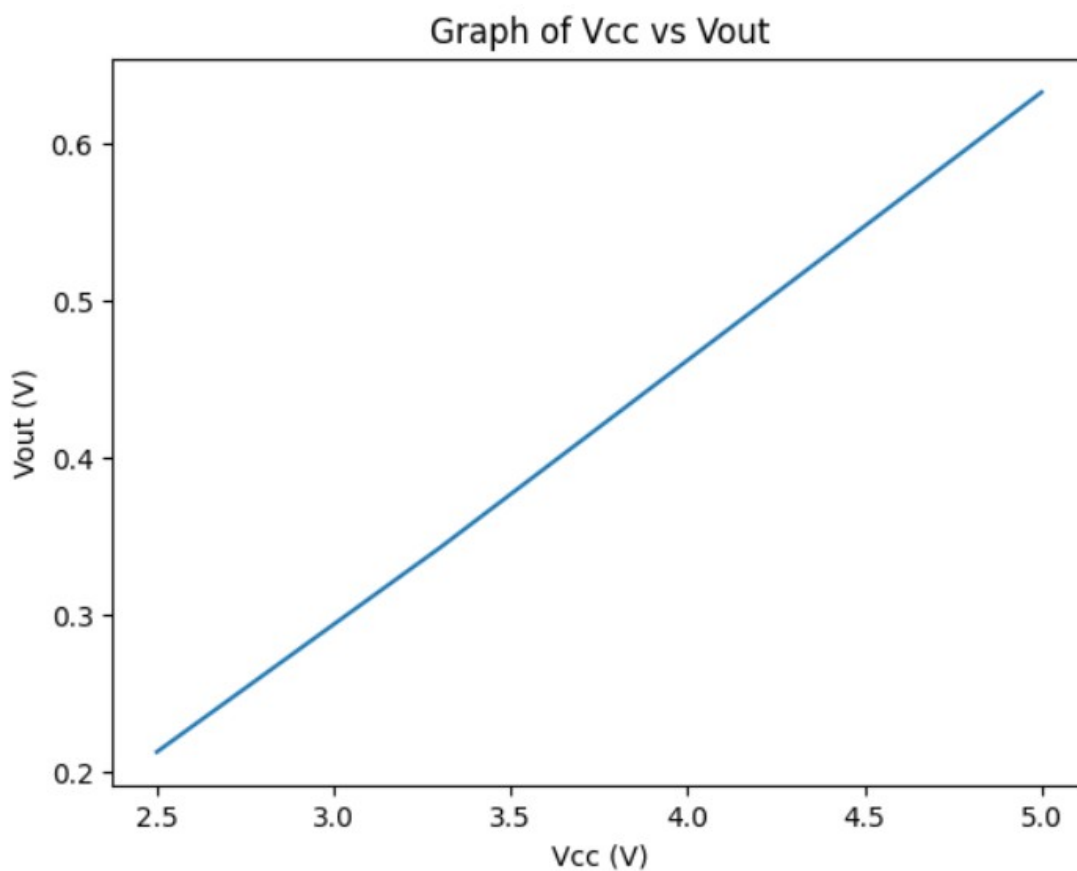
Αφού ολοκληρώθηκε αυτό το μέρος του εργαστηρίου, συνδέσαμε το παραπάνω κύκλωμα με το arduino μας όπως φαίνεται παρακάτω για να προσομοιώσουμε έναν φωτεινό σηματοδότη 3 καταστάσεων όπως στην πραγματική ζωή.



Σχ. 6

Ο κώδικας που χρησιμοποιήθηκε για την προσομοίωση βρίσκεται στο τέλος της αναφοράς.

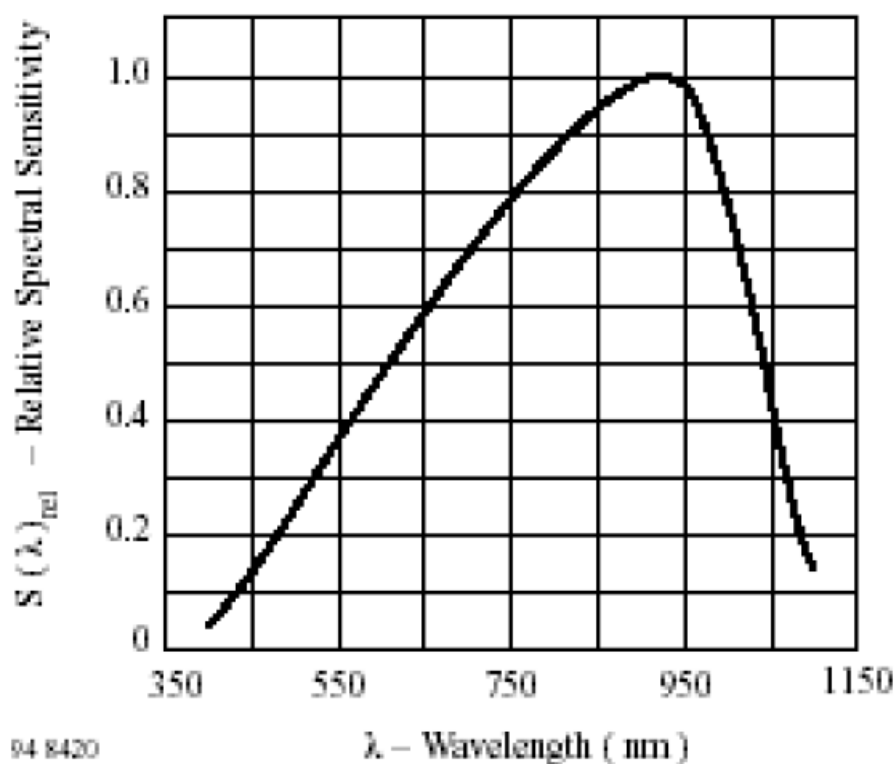
2)



---

Παρατηρούμε ότι η σχέση μεταξύ αυτών των 3 τιμών είναι σχεδόν γραμμική όπως και θα περιμέναμε.

**3)** Η συμπεριφορά του συστήματος μπορεί να εξηγηθεί με βάση τη σχετική φασματική ευαισθησία της φωτοδιόδου BPW34. Από το γράφημα, μπορεί να παρατηρηθεί ότι η φωτοδίοδος έχει μια κορυφαία ευαισθησία στα 900 nm περίπου, και σταδιακά μειώνεται καθώς το μήκος κύματος απομακρύνεται από αυτήν την κορυφή. Στο συγκεκριμένο σύστημα, το υπέρυθρο LED εκπέμπει φως με μήκος κύματος 950 nm, το οποίο είναι κοντά στην ευαισθησία κορυφής της φωτοδιόδου. Ως αποτέλεσμα, η φωτοδίοδος θα είναι πιο ευαίσθητη στο υπέρυθρο φως που εκπέμπεται από το LED και το ρεύμα εξόδου της θα αυξηθεί. Επομένως αφού η τιμή είναι πολύ κοντά στο Peak η φωτοδίοδος είναι ευαίσθητη ακόμα και στις πιο μικρές αλλαγές και υπάρχει μεγάλη ακρίβεια.



**Σχ. 2**

4)

$V_{out}$ (V)	$I_{ph}$ ( $\mu A$ )
0.633V	63.3 $\mu A$
0.343V	34.3 $\mu A$
0.213V	21.3 $\mu A$

5)

Μια εφαρμογή αυτού του κυκλώματος είναι στους αισθητήρες απόστασης. Το υπέρυθρο LED εκπέμπει ακτινοβολία που ανακλάται πίσω από ένα αντικείμενο και ανιχνεύεται από τη φωτοδίοδο. Με βάση την ένταση της ανακλώμενης ακτινοβολίας, το κύκλωμα μπορεί να καθορίσει την απόσταση μεταξύ του αντικειμένου και του αισθητήρα. Μια άλλη εφαρμογή είναι στους συναγερμούς. Το υπέρυθρο LED μπορεί να τοποθετηθεί στην είσοδο ενός κτιρίου και η φωτοδίοδος μπορεί να τοποθετηθεί στην άλλη πλευρά. Όταν ένα εμπόδιο, όπως για παράδειγμα ένας άνθρωπος ή κάποιο ζώο, διακόψει την υπέρυθρη ακτινοβολία, το κύκλωμα μπορεί να ανιχνεύσει την αλλαγή και να ενεργοποιήσει συναγερμό. Το ποτενσιόμετρο και το εξάρτημα LM741 μπορούν να χρησιμοποιηθούν για τη ρύθμιση της ευαισθησίας και του κατωφλίου του κυκλώματος, καθιστώντας το περισσότερο ή λιγότερο ευαίσθητο στις αλλαγές στην ακτινοβολία που ανιχνεύεται από τη φωτοδίοδο. Το κανονικό κόκκινο φως μπορεί να χρησιμοποιηθεί ως ένδειξη κατάστασης, παρέχοντας οπτική ανατροφοδότηση για την κατάσταση του κυκλώματος, όπως όταν ενεργοποιείται ο συναγερμός. Τέλος θα μπορούσε να χρησιμοποιηθεί σε ένα αυτοματοποιημένο σύστημα φωτισμού που προσαρμόζει τη φωτεινότητα ενός δωματίου με βάση την ποσότητα φυσικού φωτός.



```

// Pin Definitions
const int led1Pin = 3;
const int led2Pin = 5;
const int led3Pin = 6;
const int photodiodePin = A0;

// State Definitions
const int NORMAL = 0;
const int OBJECT_DETECTED = 1;

// Timing Constants
const unsigned long LED1_DURATION = 15000; // 15 seconds
const unsigned long LED2_DURATION = 5000; // 5 seconds
const unsigned long LED3_DURATION = 20000; // 20 seconds

// Variables
int currentState = NORMAL;
unsigned long stateStartTime = 0;
bool objectFound = false;

void setup() {
    pinMode(led1Pin, OUTPUT);
    pinMode(led2Pin, OUTPUT);
    pinMode(led3Pin, OUTPUT);

    // Initial state: Normal traffic light cycle
    currentState = NORMAL;
    stateStartTime = millis();
}

void loop() {
    // Read photodiode input
    float voltage = analogRead(photodiodePin) * 5.0 / 1023.0;
    if (voltage > 0.3) {
        objectFound = true;
    } else {
        objectFound = false;
    }

    // Handle state transitions and behavior
    switch (currentState) {

```

```

case NORMAL:
    // Normal traffic light behavior

    // Check for object detection
    if (objectFound) {
        currentState = OBJECT_DETECTED;
        stateStartTime = millis();
    } else {
        // Normal traffic light sequence
        // Implement the desired behavior for each state of
the traffic light

        // Green light
        digitalWrite(led3Pin, HIGH);
        delay(LED3_DURATION);

        // Yellow light
        digitalWrite(led3Pin, LOW);
        digitalWrite(led2Pin, HIGH);
        delay(LED2_DURATION);

        // Red light
        digitalWrite(led2Pin, LOW);
        digitalWrite(led1Pin, HIGH);
        delay(LED1_DURATION);

        // Switch back to the normal state (loop back to
the beginning of the cycle)
        currentState = NORMAL;
        stateStartTime = millis();

        // Turn off all lights
        digitalWrite(led1Pin, LOW);
        digitalWrite(led2Pin, LOW);
        digitalWrite(led3Pin, LOW);
    }
    break;

case OBJECT_DETECTED:
    // Object detected, take appropriate action

    if (currentState == LED2_ON) {

```

```

        // LED2 ON state: Continue the cycle from Yellow to
Red
        delay(5000); // Wait for 5 seconds before
transitioning to Red
        currentState = LED1_ON;
        stateStartTime = millis();
    } else if (currentState == LED1_ON) {
        // LED1 ON state: Restart from the beginning
        currentState = NORMAL;
        stateStartTime = millis();
    } else if (currentState == LED3_ON) {
        // LED3 ON state: Handle different sub-cases
        unsigned long elapsedTime = millis() -
stateStartTime;

        if (elapsedTime >= 3000) {
            // Sub-case γ1: Continue in the same state for 3
seconds
            // No additional action required
            delay(3000);
        } else if (elapsedTime >= 1000) {
            // Sub-case γ2: LED2 blinks for 8 seconds (1
second ON, 1 second OFF)
            if ((elapsedTime % 2000) < 1000) {
                digitalWrite(led2Pin, HIGH);
            } else {
                digitalWrite(led2Pin, LOW);
            }
        }

        // Check if object is still detected
        if (objectFound) {
            // Sub-case γ3: LED1 remains ON until there is no
optical contact
            digitalWrite(led1Pin, HIGH);
        }

        // Check for change in object detection status
        float newVoltage = analogRead(photodiodePin) *
5.0 / 1023.0;

```

```

        if (newVoltage > 0.3) {
            objectFound = true;
        } else {
            objectFound = false;
        }

        // Transition back to normal state after object
detection
        if (!objectFound) {
            currentState = NORMAL;
            stateStartTime = millis();
            digitalWrite(led1Pin, LOW);
            digitalWrite(led2Pin, LOW);
            digitalWrite(led3Pin, LOW);
        }
    }

    break;
}
}

```