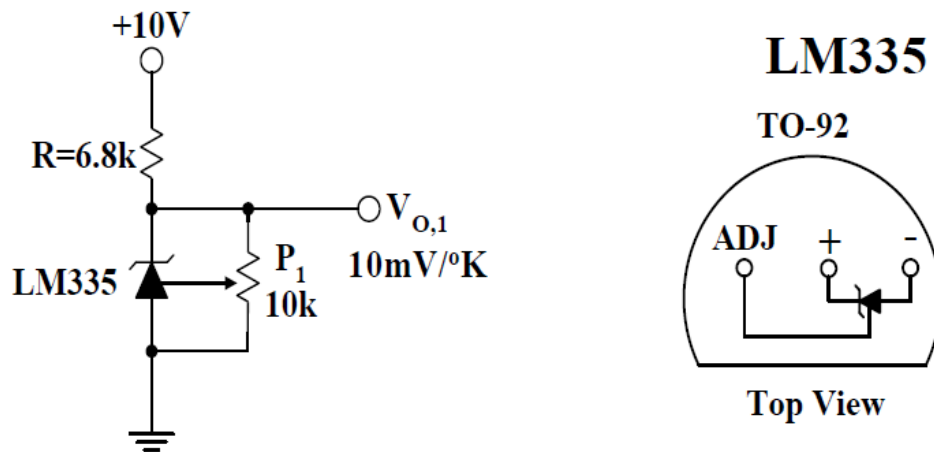


Γιόελ Γιάννη:2019030116
Ανδρέας Καρόγιαννης:2019030066

Ηλεκτρικές Μετρήσεις και Αισθητήρες 4η Εργαστηριακή Άσκηση

Στην τέταρτη εργαστηριακή άσκηση ασχοληθήκαμε με την μελέτη των χαρακτηριστικών λειτουργίας των ολοκληρωμένων αισθητήρων μέτρησης θερμοκρασίας



Σχ. 1.

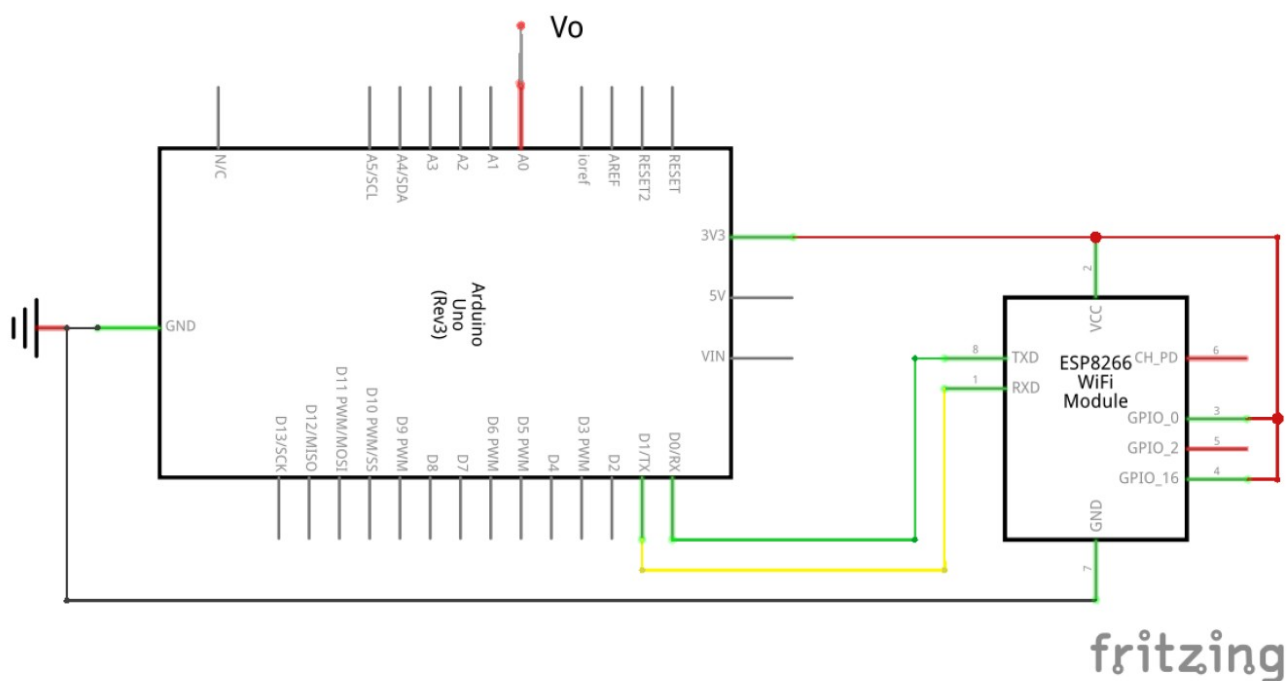
Για το κύκλωμα που φαίνεται παραπάνω, χρησιμοποιήθηκε ένα ποτενσιόμετρο, μια αντίσταση και το ολοκληρωμένο κύκλωμα LM335. Η διάταξη δουλεύει ως αισθητήρας ανίχνευσης θερμοκρασίας και στο εργαστήριο χρειάστηκε να ρυθμιστεί το ποτενσιόμετρο ώστε $V_o = 10mV(273.2 + T_c)$, όπου T_c η θερμοκρασία περιβάλλοντος η οποία μετρήθηκε στο εργαστήριο με το probe του πολύμετρου. Στη συνέχεια θερμάθηκε το LM335 και καταγράφηκαν οι τάσης εξόδου σε διάφορες τιμές θερμοκρασίας οι οποίες βρίσκονται παρακάτω.

Η τάση εξόδου, V_O , σε θερμοκρασία $T^{\circ}\text{K}$ δίνεται από τη σχέση:

$$V_O = \frac{T}{T_{O,ref}} V_{O,ref}$$

όπου $V_{O,ref}$ είναι η τάση εξόδου στη θερμοκρασία αναφοράς $T_{O,ref}$ (π.χ. 298.2°K ή 25°C).

Στη συνέχεια συνδέσαμε την έξοδο V_O του σχήματος 1 παραπάνω, στην αναλογική είσοδο A_0 του Arduino. Όπως και προηγουμένως, ρυθμίστηκε το ποτενσιόμετρο ώστε η έξοδος V_O να έχει τη τιμή 2.982V στους 25°C . Το πρόγραμμα του Arduino συνδέεται με την πλατφόρμα <https://thingspeak.com/> και στέλνει την θερμοκρασία και την τάση.

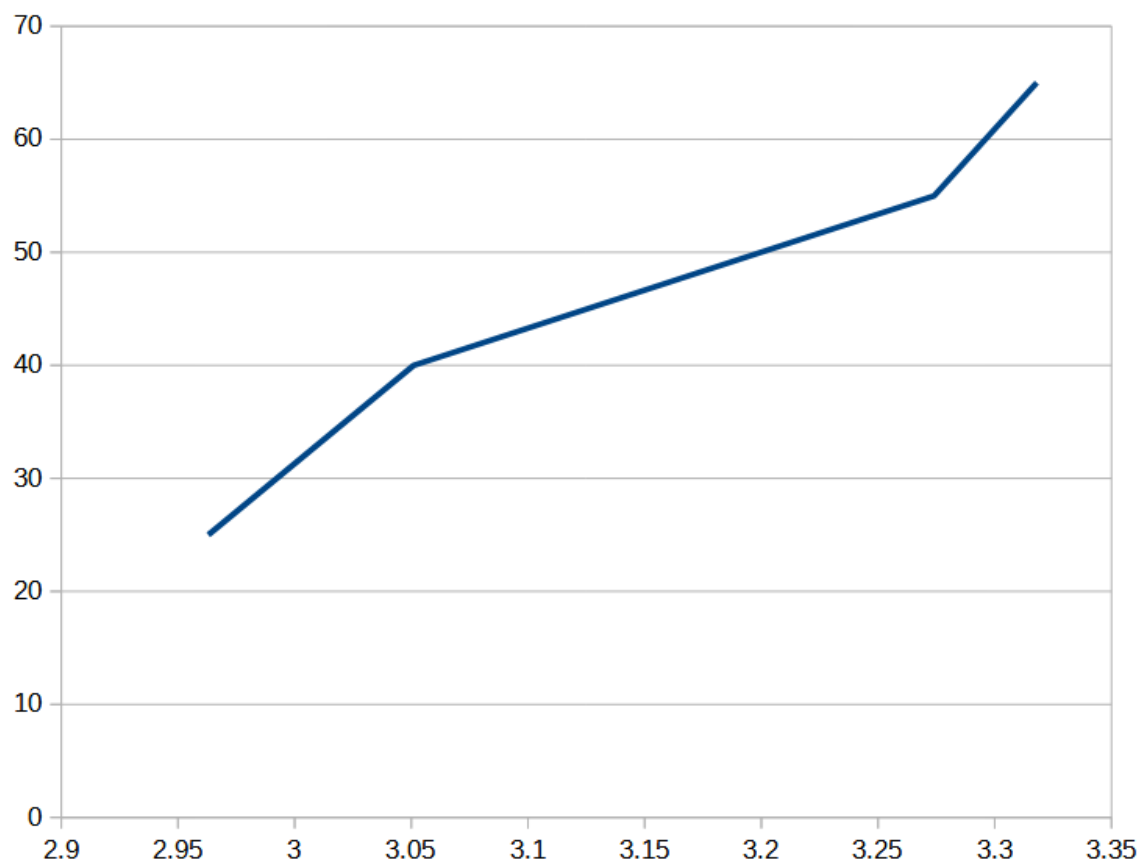


Σχ. 2.

1)

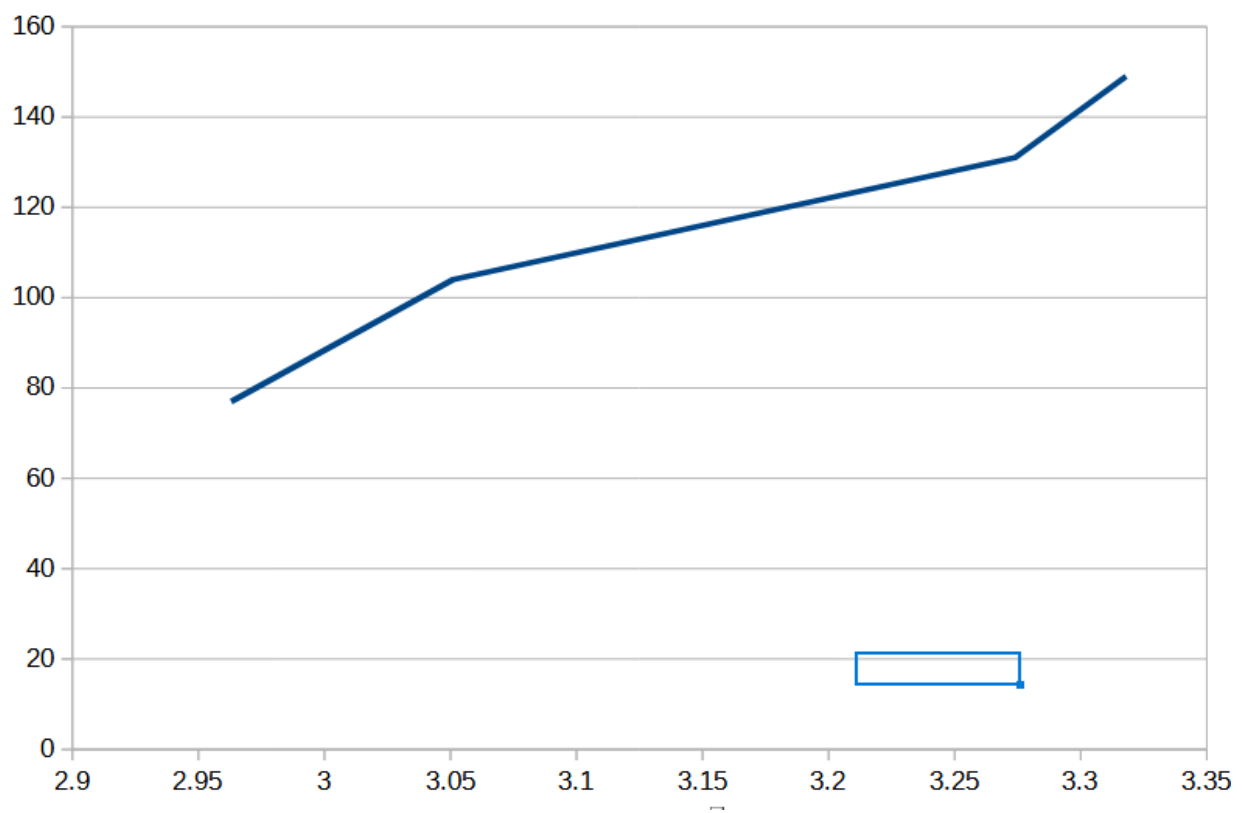
Τάση εξόδου σε σχέση με τη θερμοκρασία
σε βαθμούς °C

V_o (V)	T_c (°C)
2.963	25
3.051	40
3.274	55
3.318	65



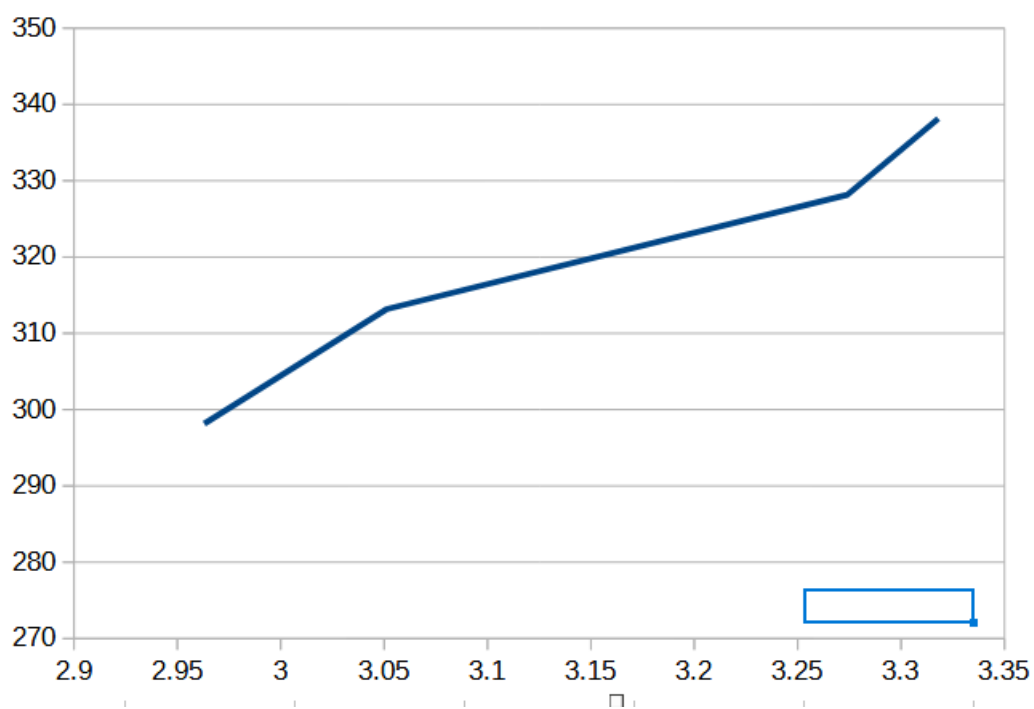
Τάση εξόδου σε σχέση με τη θερμοκρασία
σε βαθμούς °F

V_o (V)	T_F (°F)
2.963	77
3.051	104
3.274	131
3.318	149



Τάση εξόδου σε σχέση με τη θερμοκρασία
σε βαθμούς °K

V_o (V)	T_K (°K)
2.963	298.15
3.051	313.15
3.274	328.15
3.318	338.15



2)

Το κύκλωμα μπορεί να χρησιμοποιηθεί για τη μέτρηση της θερμοκρασίας σε ένα περιβάλλον, όπως ένα δωμάτιο, ένα θερμοκήπιο ή ένα εργαστήριο. Με τη συνεχή παρακολούθηση της θερμοκρασίας, μπορεί να εφαρμόστούν συστήματα ελέγχου για θέρμανση, ψύξη ή εξαερισμό με βάση συγκεκριμένα όρια θερμοκρασίας.

Η ενσωμάτωση του κυκλώματος αισθητήρα θερμοκρασίας σε ένα σύστημα οικιακού αυτοματισμού επιτρέπει τον έλεγχο σε διάφορες συσκευές με βάση τις συνθήκες θερμοκρασίας. Για παράδειγμα, μπορεί να αυτοματοποιηθεί η λειτουργία των ανεμιστήρων, των κλιματιστικών ή των θερμαντικών συσκευών για τη διατήρηση μιας άνετης θερμοκρασίας σε ένα δωμάτιο.

Σε βιομηχανικές εγκαταστάσεις, είναι ζωτικής σημασίας η παρακολούθηση των διακυμάνσεων της θερμοκρασίας για διεργασίες, εξοπλισμό ή χώρους αποθήκευσης. Αυτό το κύκλωμα μπορεί να ενσωματωθεί σε βιομηχανικά συστήματα ελέγχου για να διασφαλίσει τις βέλτιστες συνθήκες θερμοκρασίας και να αποτρέψει την υπερθέρμανση ή το αντίθετο.

Το κύκλωμα μπορεί να χρησιμοποιηθεί σε γεωργικές εφαρμογές, όπως η παρακολούθηση θερμοκηπίων ή η γεωργία ακριβείας. Με την παρακολούθηση και τον έλεγχο των συνθηκών θερμοκρασίας, μπορεί να βελτιστοποιηθεί η ανάπτυξη των καλλιεργειών, να αποτραπούν ζημιές από τον παγετό και να βελτιωθεί η συνολική παραγωγικότητα.

Το κύκλωμα μπορεί να ενεργοποιεί συναγερμούς ή μέτρα ασφαλείας όταν υπερβαίνονται ή δεν πληρούνται τα όρια θερμοκρασίας. Για παράδειγμα, σε αίθουσες διακομιστών ή ηλεκτρικούς πίνακες, εάν η θερμοκρασία αυξηθεί πάνω από ένα ασφαλές επίπεδο, το κύκλωμα μπορεί να στείλει ειδοποιήσεις ή να ενεργοποιήσει συστήματα ψύξης για την αποφυγή ζημιών στον εξοπλισμό.

```

#include <SoftwareSerial.h>
#define      RX_PIN      2
#define      TX_PIN      3
SoftwareSerial AT(RX_PIN,TX_PIN);

String wifiSSID      = "SimplePSK"; //Wifi SSID
String wifiPass = "PSKS1mpl3!@#"; //Wifi Password

//Requests
String requestUri1 = "GET /update?
api_key=PMU2GPMWIZGXGAMD&field1=";
String tempA = "&field2=";
String requestUri2 = "GET /update?
api_key=MVZ6ZSXYVRZ5X7GK&field2=";

String host = "api.thingspeak.com"; // API host name
int AT_c_time;
boolean AT_c_result = false;

int sensorValue = 0;
float voltage1 = 0.0;
float temp = 0.0;
float temp_ref = 298.2;
float v_ref = 2.982;

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

    Serial.println("***** Program Start : Arduino and
ESP8266 connected to service *****");
    AT.begin(115200);

```

```

    Serial.println("--> Starting AT communicatin with
ESP8266...");
    sendATcmd("AT",5,"OK");
    sendATcmd("AT+CWMODE=1",5,"OK");

    //Connect to wireless network
    Serial.print("--> Connecting to WiFi: ");
    Serial.println(wifiSSID );
    sendATcmd("AT+CWJAP=\""+ wifiSSID +"\", \""+ wifiPass
+"\"\"",20,"OK");

    delay(2000);
    openTCP();
}

void loop() {
    sensorValue = analogRead(A0);

    voltage1 = sensorValue * (5.0 / 1023.0);
    temp = (voltage1 * temp_ref)/v_ref;
    temp = temp - 273.2;
    Serial.print(voltage1);
    Serial.print(";");
    Serial.print(temp);
    Serial.print(";");

    sendData(requestUri1, voltage1, temp);
    delay(5000);
    closeTCP();

    openTCP();
    shareData(requestUri2, temp);
    delay(5000);
    closeTCP();

    delay(2000);

```



```

    openTCP();

    Serial.println("\n\nNew  measurements\n");
}

void sendData(String request_url, float voltage, float
temp){
    request_url += String(voltage,2) + tempA +
String(temp,2);
    Serial.print("\n--> Request URI: ");
    Serial.println(request_url);
    sendATcmd("AT+CIPSEND=0," + String(request_url.length()
+ 4), 10, ">");
    AT.println(request_url);
    delay(3000);
}
void shareData(String request_url,float temp){
    request_url += String(temp,2);
    Serial.print("\n--> Request URI: ");
    Serial.println(request_url);
    sendATcmd("AT+CIPSEND=0," + String(request_url.length()
+ 4), 10, ">");
    AT.println(request_url);
    delay(3000);
}

void openTCP(){
    Serial.println("***** Open TCP connection to Thingspeak
REST API *****");
    sendATcmd("AT+CIPMUX=1", 10, "OK");
    sendATcmd("AT+CIPSTART=0,\"TCP\", \"" + host + "\", " +
"80", 20, "OK");
}

void closeTCP(){
    delay(2000);
}

```

```
Serial.println("--> Closing TCP connection...");
sendATcmd("AT+CIPCLOSE=0", 10, "OK");
Serial.println("***** TCP Connection
terminated *****");
```

```
void sendATcmd(String AT_c, int AT_c_maxTime, char readR[])
{
    Serial.print("--> AT command to send: ");
    Serial.println(AT_c);

    while(AT_c_time < (AT_c_maxTime)) {
        AT.println(AT_c);
        if(AT.find(readR)) {
            AT_c_result = true;
            break;
        }
        AT_c_time++;
    }

    Serial.print("...Result:");

    if(AT_c_result == true) {
        Serial.println("SUCCESS");
        AT_c_time = 0;
    }

    if(AT_c_result == false) {
        Serial.println("FAILURE");
        AT_c_time = 0;
    }
    AT_c_result = false;
}
```