**Πείραμα με αισθητήρα θερμοκρασίας(Temperature module)**

1. **Περιγραφή:**

Στο συγκεκριμένο πείραμα χρησιμοποιούμε module θερμοκρασίας ,συνδεόμενο σε Raspberry Pi να εμφανίζει στο terminal την θερμοκρασία που έχει ο χώρος .

1. **Υλικά:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **1\* Raspberry pi** | **1\* GPIO Extension board** | **1\* 40 pin colorful jumper wires** |

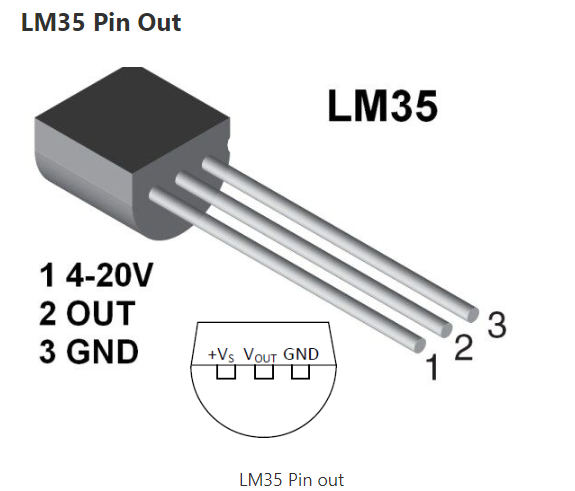
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **1\* Breadboard** | **Jumper Wires** | **1\* Temperature Module** | **1\*analog to digital signal converter and vice versa** |

1. **Γνώση υλικού:**

Στο module χρησιμοποιούμε τον αισθητήρα θερμοκρασίας LM35.

Ο LM35 είναι ένας αισθητήρας θερμοκρασίας που εξάγει ένα αναλογικό σήμα το οποίο είναι η απεικόνιση της στιγμιαίας θερμοκρασίας. Η τάση εξόδου μπορεί εύκολα να ερμηνευθεί για να ληφθεί μια ένδειξη θερμοκρασίας σε βαθμούς Κελσίου. Το πλεονέκτημα του LM35 έναντι άλλων αισθητήρων θερμοκρασίας είναι ότι δεν απαιτεί εξωτερική βαθμονόμηση. Η επίστρωση το προστατεύει επίσης από την αυτοθέρμανση.

Το LM35 μπορεί να μετρήσει από τους -55 βαθμούς Κελσίου έως τους 150 βαθμούς Κελσίου. Το επίπεδο ακρίβειας είναι πολύ υψηλό εάν λειτουργεί σε βέλτιστα επίπεδα θερμοκρασίας και υγρασίας. Η μετατροπή της τάσης εξόδου σε βαθμούς Κελσίου είναι επίσης εύκολη και απλή.Η τάση εισόδου στο LM35 μπορεί να είναι από 4 V έως 20 V.



**Πειραματική συσκευή:**

* Πίνακας ελέγχου Raspberry Pi
* Μονάδα αισθητήρα θερμοκρασίας
* Γραμμές σύνδεσης
* A/D converter

**Πειραματική σύνδεση:**

|  |  |
| --- | --- |
| The temperature module | Raspberry Pi |
| Vcc | 5V |
| Gd | GND |
| OUT | Ain0 |

|  |  |
| --- | --- |
| A/D converter | Raspberry Pi |
| SCL | SCL |
| SDA | SDA |
| Vcc | 5 V |
| Gnd | Gnd |

1. **Πειραματικό συμπέρασμα:**

Με τον παρακάτω κώδικα διαβάζουμε την αναλογική τάση από τον αισθητήρα LM35 και υπολογίζουμε τη θερμοκρασία σε βαθμούς Κελσίου με τη βοήθεια του A/D converter module που μετατρέπει την ψηφιακή πληροφορία σε αναλογική . Βεβαιωθείτε ότι έχετε προσαρμόσει τον αριθμό ακροδέκτη GPIO και τον συντελεστή μετατροπής θερμοκρασίας ανάλογα με τη ρύθμιση που έχετε κάνει. Οι πυκνωτές χρησιμοποιούνται για τη σταθεροποίηση της τροφοδοσίας του αισθητήρα και τη βελτίωση της ακρίβειας.

Πρέπει να επισημάνουμε ότι για το LM35 το Threshold value=10.24 για να πάρω τη θερμοκρασία σε βαθμούς Κελσίου.

**Βεβαιωθείτε ότι έχετε εγκαταστήσει τη βιβλιοθήκη RPi.GPIO στο Raspberry Pi σας, αν δεν είναι ήδη εγκατεστημένη. Μπορείτε να το κάνετε αυτό με την ακόλουθη εντολή στη γραμμή εντολών(command line):**

**pip install RPi.GPIO**

Σημαντική επισήμανση για αναλογικές τιμές εισόδου στο Raspberry Pi:

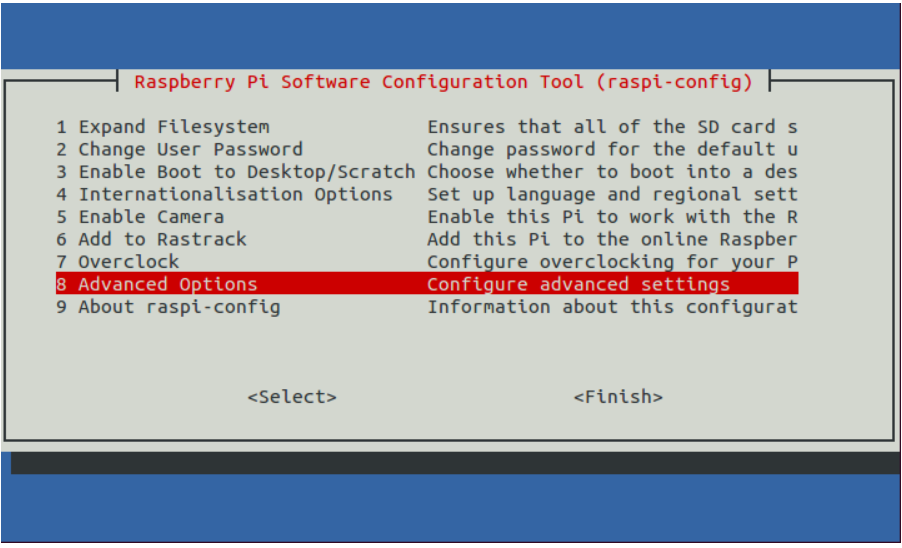
**Ενεργοποίηση Ι2C θύρας:**

**Βήμα 1**:

Ενεργοποιήστε τη θύρα I2C του Raspberry Pi σας (Αν έχετε κάνει την ενεργοποίηση, παραλείψτε αυτό. Αλλά αν δεν είστε σίγουροι αν την έχετε ενεργοποιήσει ή όχι, συνεχίστε):

sudo raspi-config

**Advanced options**

****

**A screenshot of a computer

Description automatically generated**

**Βήμα 2:**

**Ελέγξτε ότι οι μονάδες i2c είναι φορτωμένες και ενεργές:**

lsmod **|** grep i2c

**Τότε θα εμφανιστεί ο ακόλουθος κώδικας (ο αριθμός μπορεί να είναι διαφορετικός)**

i2c\_dev                     6276    0

i2c\_bcm2708                 4121    0

**Βήμα 3:**

**Εγκαταστήστε τα i2c-tools**

sudo apt-get install i2c-tools

**Βήμα 4:**

**Ελέγξτε τη διεύθυνση της συσκευής I2C:**

**A screenshot of a computer

Description automatically generated**

**Αν είναι συνδεδεμένη μια συσκευή I2C, τα αποτελέσματα θα είναι παρόμοια με τα παραπάνω - αφού η διεύθυνση της συσκευής είναι 0x48, εκτυπώνεται η τιμή 48.**

**Βήμα 5:**

sudo apt-get install python-smbus

1. **Κώδικας Python:**

import smbus

import time

# Define the I2C bus and address

bus = smbus.SMBus(1)  # Use bus number 1 for Raspberry Pi 3 and later

pcf8591\_address = 0x48  # PCF8591 default I2C address

# Function to read ADC channel

def read\_adc(channel):

    # PCF8591 has 4 analog input channels (Ain0 to Ain3)

    command = 0x40 | (channel & 0x03)

    bus.write\_byte(pcf8591\_address, command)

    time.sleep(0.01)  # Allow some time for the conversion to complete

    adc\_value = bus.read\_byte(pcf8591\_address)

    return adc\_value

# Define the ADC channel for the LM35 temperature sensor

lm35\_channel = 0

try:

    while True:

        # Read analog value from the LM35 temperature sensor

        analog\_value = read\_adc(lm35\_channel)

        # Calculate the temperature in degrees Celsius

        temperature = (analog\_value ) / 10.24

        # Print the temperature to the console

        print(f"Temperature: {temperature:.2f} degrees Celsius")

        # Sleep for a short time to avoid printing too quickly

        time.sleep(0.1)

except KeyboardInterrupt:

    # Clean up on Ctrl+C

    pass

finally:

    # Clean up on normal program exit

    print("Cleaning up...")

    bus.close()