**2x Slide-off potentiometer**

1. **Περιγραφή:**

Στο παρακάτω πείραμα έχω συρόμενο ποτενσιόμετρο: και τα δύο ρυθμιζόμενα ποτενσιόμετρα με το ηλεκτρικό σύμβολο όπως φαίνεται στο σχήμα.

Chart

Description automatically generated

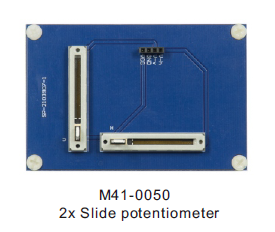
Δύο ακροδέκτες των ποτενσιόμετρων Slide-off είναι σταθεροί και ο άλλος είναι ακροδέκτες μεταβλητής αντίστασης. Η τιμή αντίστασης που αναγράφεται στο ποτενσιόμετρο ονομάζεται ονομαστική τιμή αντίστασης, η οποία ισούται με την τιμή αντίστασης μεταξύ των δύο σταθερών ακροδεκτών του ποτενσιόμετρου, η σειρά των οποίων είναι παρόμοια με τη σειρά αντίστασης. Η αντίσταση του ποτενσιόμετρου μεταβάλλεται όταν κινείται η ράβδος ολίσθησης.

1. **Υλικά:**

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **1\* Breadboard** | **Jumper Wires** | **1\*analog to digital signal converter and vice versa** | **1\* 2 slide potentiometer** |

1. **Γνώση υλικού :**



Number of resistors: single  
Installation direction: Vertical  
Stroke: 75mm  
Voltage: 3.3v/5v  
Shape of operating part: T-Bar  
Length of operating part: 15mm  
Resistance value: 10k Ω ± 10%  
Temperature range: - 10 ℃ to+80 ℃

**Πειραματική συσκευή:**

* Πίνακας ελέγχου Raspberry Pi : 1
* Μονάδα ποτενσιόμετρου ολίσθησης: 1
* Γραμμές σύνδεσης
* A/D converter module

**Πειραματική σύνδεση:**

|  |  |
| --- | --- |
| 2\*slide resistor module | A/D converter module , Raspberry Pi |
| VCC | 5V (Raspberry Pi) |
| GND | GND (Raspberry Pi) |
| Vh | Ain0 (A/D converter) |
| Vv | Ain1 (A/D converter) |

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

Σημαντική επισήμανση για αναλογικές τιμές εισόδου στο Raspberry Pi:

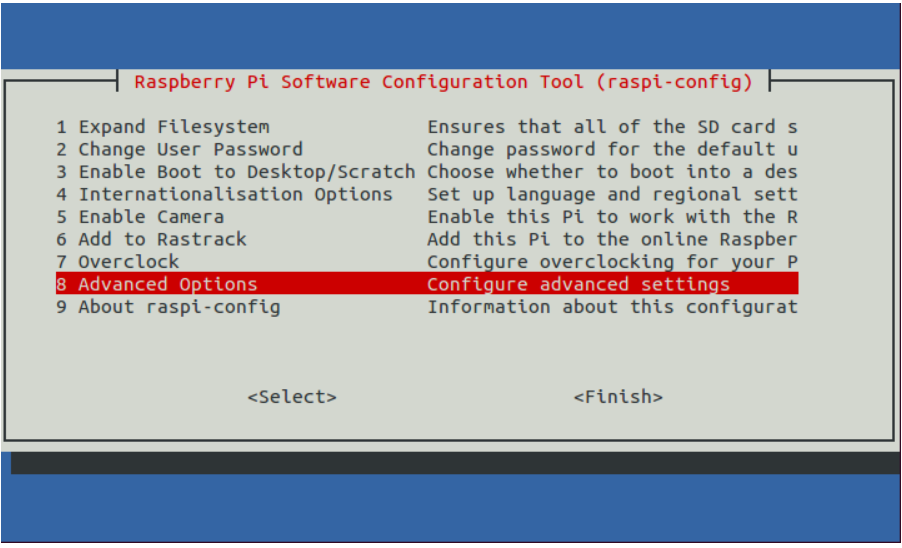
**Ενεργοποίηση Ι2C θύρας:**

**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. **Συμπέρασμα πειράματος:**

Αφού συνδέσουμε το ποτενσιόμετρο στο A/D converter module και αυτό στο Raspberry Pi, τρέξουμε τον κώδικα σε Python, μετακινώ τα ποτενσιόμετρα και παίρνω τις τιμές σε τάση και αντίσταση.

**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**

1. **Κώδικας Python:**

import smbus

import time

address = 0x48  # Default address of PCF8591

bus = smbus.SMBus(1)  # Create an instance of smbus

cmd = 0x40  # Command

def analogRead(chn):  # Read ADC value, chn: 0, 1, 2, 3

    value = bus.read\_byte\_data(address, cmd + chn)

    return value

def analogWrite(value):  # Write DAC value

    bus.write\_byte\_data(address, cmd, value)

def loop():

    while True:

        value1 = analogRead(0)  # Read the ADC value of channel 0

        value2 = analogRead(1)  # Read the ADC value of channel 1

        voltage1 = value1 / 255.0 \* 3.3  # Calculate the voltage value for channel 0

        voltage2 = value2 / 255.0 \* 3.3  # Calculate the voltage value for channel 1

        print(f'ADC Value (Channel 0): {value1}, Voltage: {voltage1:.2f}V')

        print(f'ADC Value (Channel 1): {value2}, Voltage: {voltage2:.2f}V')

        time.sleep(0.01)

def destroy():

    bus.close()

if \_\_name\_\_ == '\_\_main\_\_':

    print('Program is starting ... ')

    try:

        loop()

    except KeyboardInterrupt:

        destroy()