

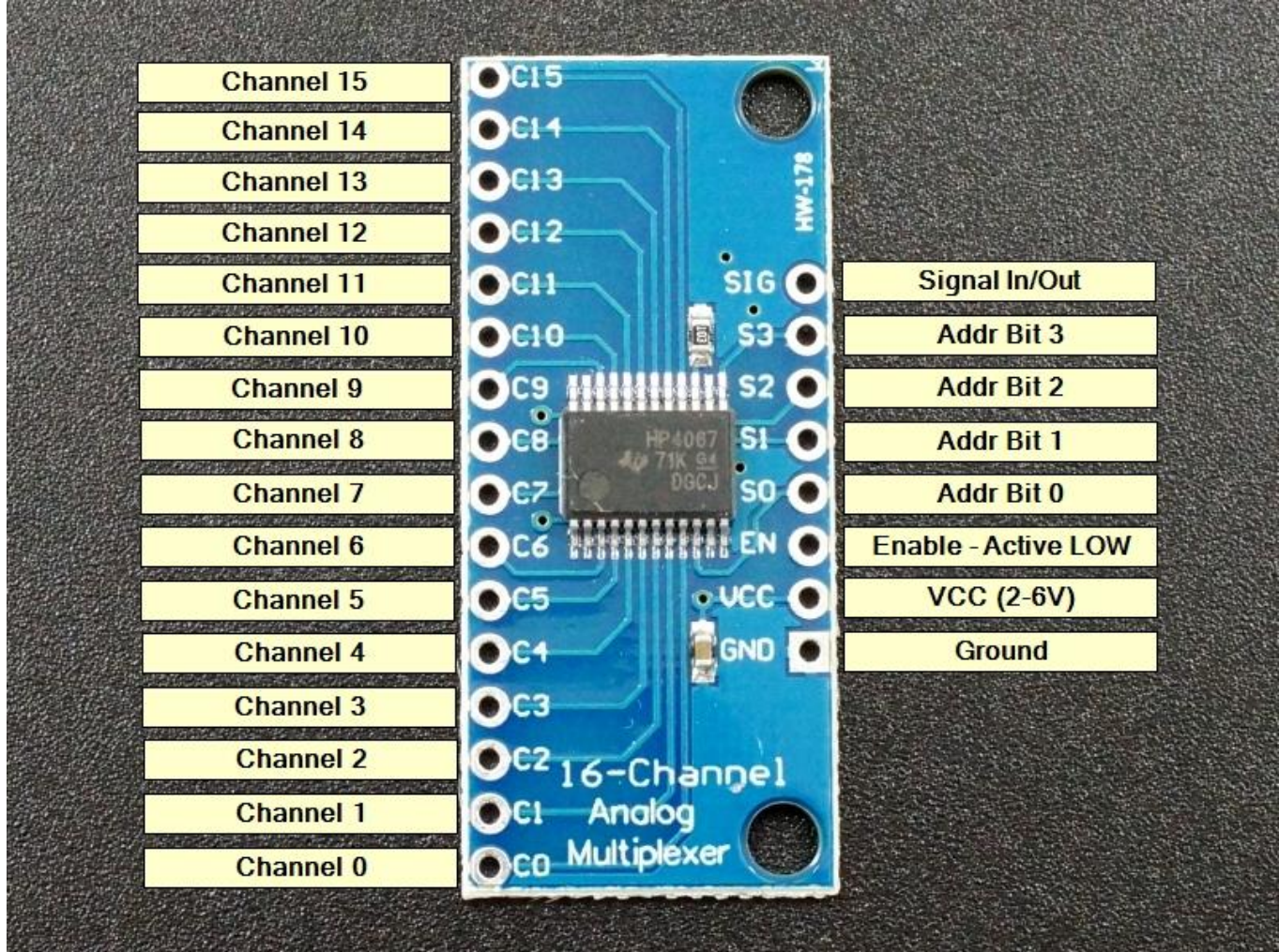
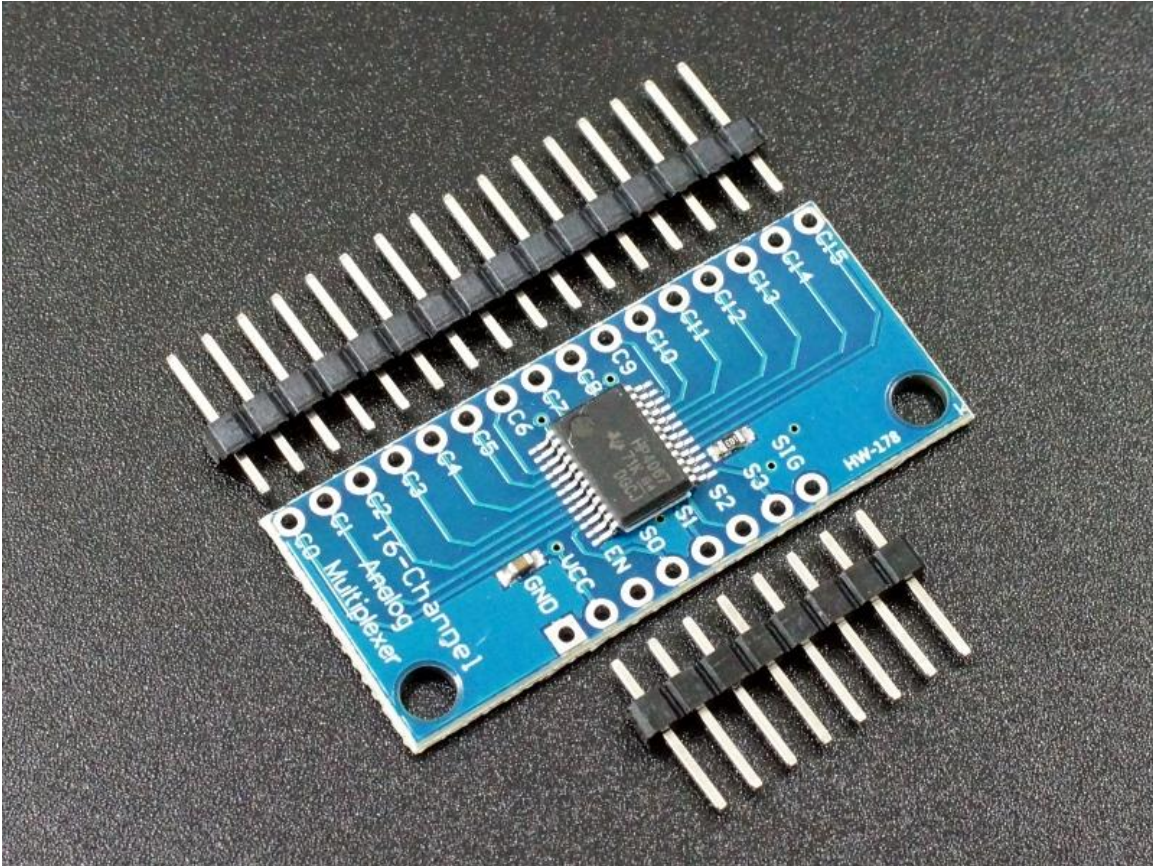
74HC4067 16-Ch Analog /  
Digital Mux Module with  
multiple sensors

Dinesh Kumar

ISRO Satellite Center

Bangalore



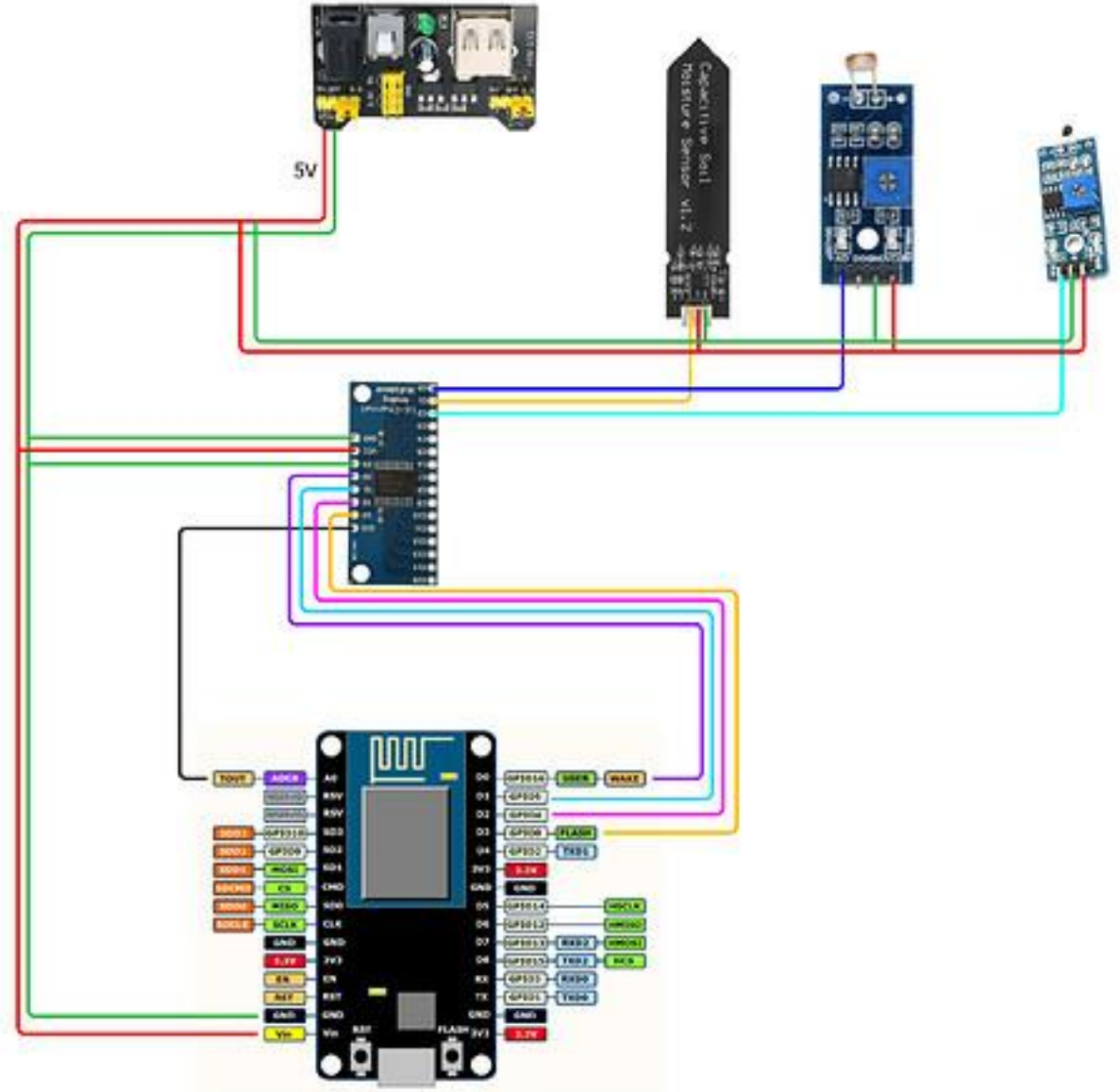


[74HC4067 16-Ch Analog / Digital Mux Module](#)



**Hardware:**  
NodeMCU (ESP8266)  
Multiplexer 16ch CD74HC4067

**Sensors:**  
LDR Module  
Capacitive Soil Moisture Sensor v1.2  
Thermistor Module 10K



```

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <Thermistor.h>

#define S0 D0
#define S1 D1
#define S2 D2
#define S3 D3
#define SIG A0

int sensor0;
int sensor1;
int sensor2;

char auth[] = "xxx";
char ssid[] = "xxx";
char pass[] = "xxx";

BlynkTimer timer;

void setup()
{
  /* Debug console */
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  Serial.println("");

  /* Timers */
  timer.setInterval(1000L, getSensor0); /* LDR */
  timer.setInterval(2500L, getSensor1); /* Soil */
  timer.setInterval(5000L, getSensor2); /* Thermistor */

  /* Multiplexer */
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(SIG, INPUT);
}

```

```

void getSensor0() {
  /* Channel 0 (C0 pin - binary output 0,0,0,0) */
  digitalWrite(S0, LOW); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, LOW);
  sensor0 = analogRead(SIG);

  Blynk.virtualWrite(V0, sensor0);
}

void getSensor1() {
  /* Channel 1 (C1 pin - binary output 1,0,0,0) */
  digitalWrite(S0, HIGH); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, LOW);
  sensor1 = analogRead(SIG);

  Blynk.virtualWrite(V1, sensor1);
}

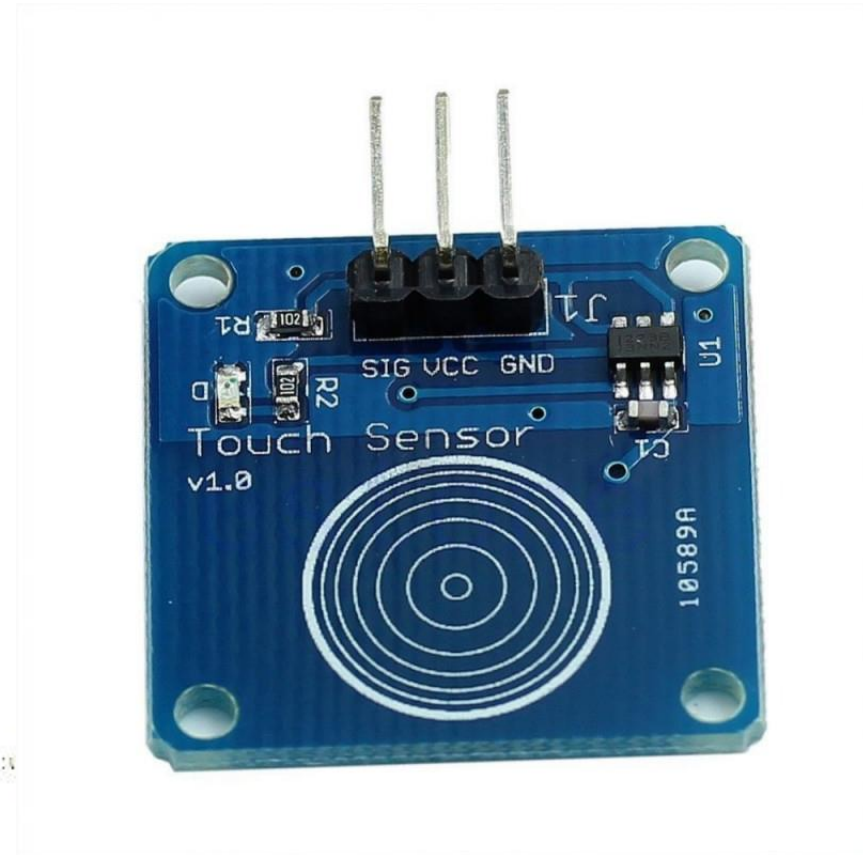
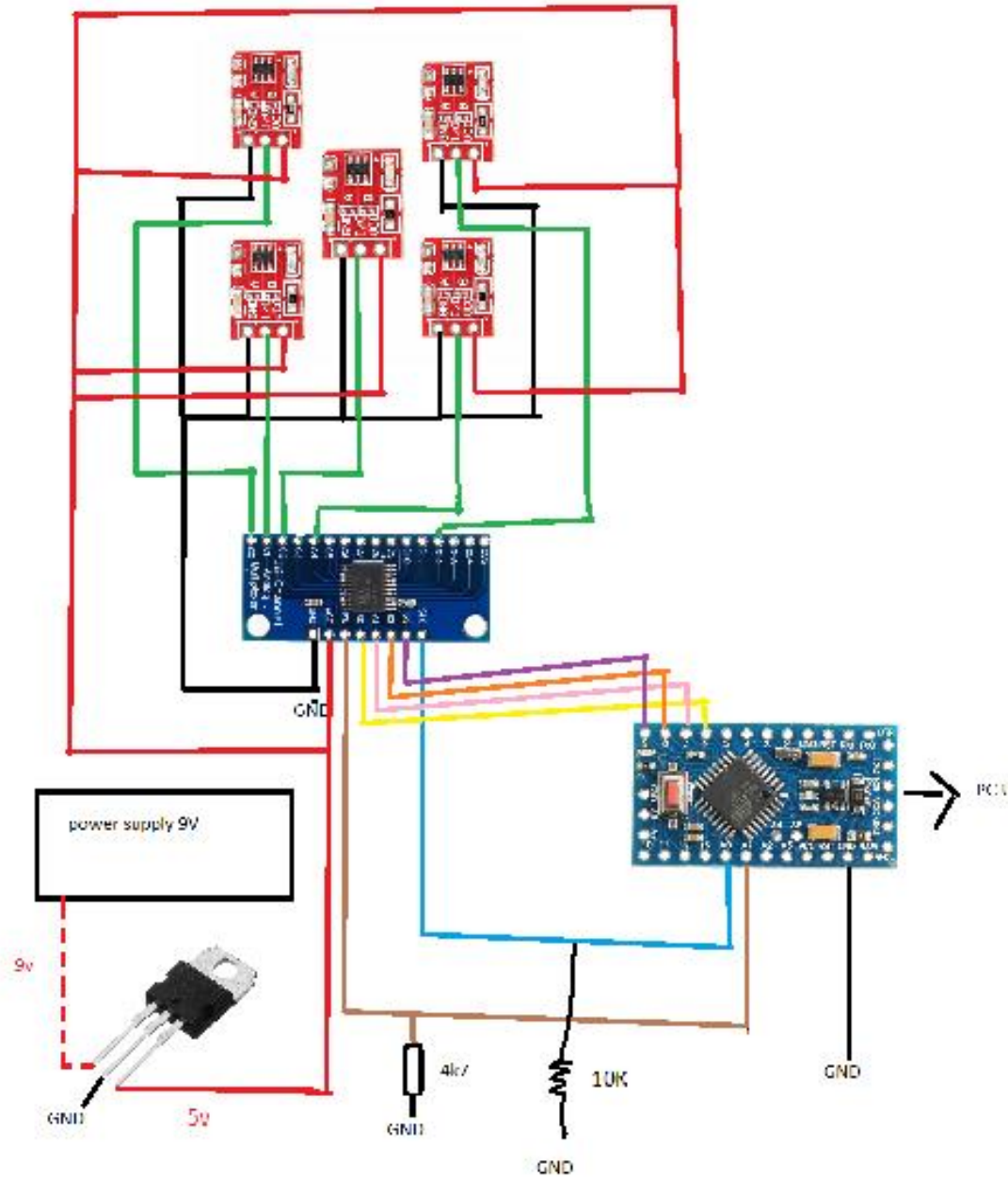
void getSensor2() {
  /* Channel 2 (C2 pin - binary output 0,1,0,0) */
  digitalWrite(S0, LOW); digitalWrite(S1, HIGH); digitalWrite(S2, LOW); digitalWrite(S3, LOW);
  sensor2 = analogRead(SIG);

  Blynk.virtualWrite(V2, sensor2);
}

void loop()
{
  Blynk.run();
  timer.run();
}

```

keypad with multiple ttp123 break-boards



```

// interfacing mux with tp223
//Mux control pins
const int s0 = 6;
const int s1 = 7;
const int s2 = 8;
const int s3 = 9;
//Mux in "SIG" pin
int SIG_pin = A0;
int EN_pin = 10;

void setup(){
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(SIG_pin, INPUT);
  pinMode(EN_pin, OUTPUT);
  digitalWrite(s0, LOW);
  digitalWrite(s1, LOW);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  digitalWrite(EN_pin, LOW);
  Serial.begin(115200);
  Serial.println("keypad interfacing_CD74HC4067 starting...");
}

void loop(){
  //Loop through and read all 16 values
  for(int i = 0; i < 16; i ++){
    int val = readMux(i);

    if (val > 0) {
      Serial.print(i);
      Serial.print(" pressed ");
      Serial.print(val);
      Serial.println();
    }
  }
}

```

```

float readMux(int channel){
  int controlPin[] = {s0, s1, s2, s3};

  int muxChannel[16][4]={
    {0,0,0,0}, //channel 0
    {1,0,0,0}, //channel 1
    {0,1,0,0}, //channel 2
    {1,1,0,0}, //channel 3
    {0,0,1,0}, //channel 4
    {1,0,1,0}, //channel 5
    {0,1,1,0}, //channel 6
    {1,1,1,0}, //channel 7
    {0,0,0,1}, //channel 8
    {1,0,0,1}, //channel 9
    {0,1,0,1}, //channel 10
    {1,1,0,1}, //channel 11
    {0,0,1,1}, //channel 12
    {1,0,1,1}, //channel 13
    {0,1,1,1}, //channel 14
    {1,1,1,1} //channel 15
  };

  //loop through the 4 sig
  for(int j = 0; j < 4; j ++){
    digitalWrite(controlPin[j], muxChannel[channel][j]);
  }
  //read the value at the SIG pin
  int val = digitalRead(SIG_pin);
  //return the value
  return val;
}

```