

Sensors and Actuators harjoitus

Tässä tulee ohjeita harjoitustyön suorittamiseksi.

Sensorit eli anturit on toki Automaatiotekniikassa ja IoT – järjestelmissä aivan oleellisia komponentteja. Tässä harjoituksessa perehdytään etenkin analogisten sensorien ongelmaan eli kohinaan ja sen suodattamiseen.

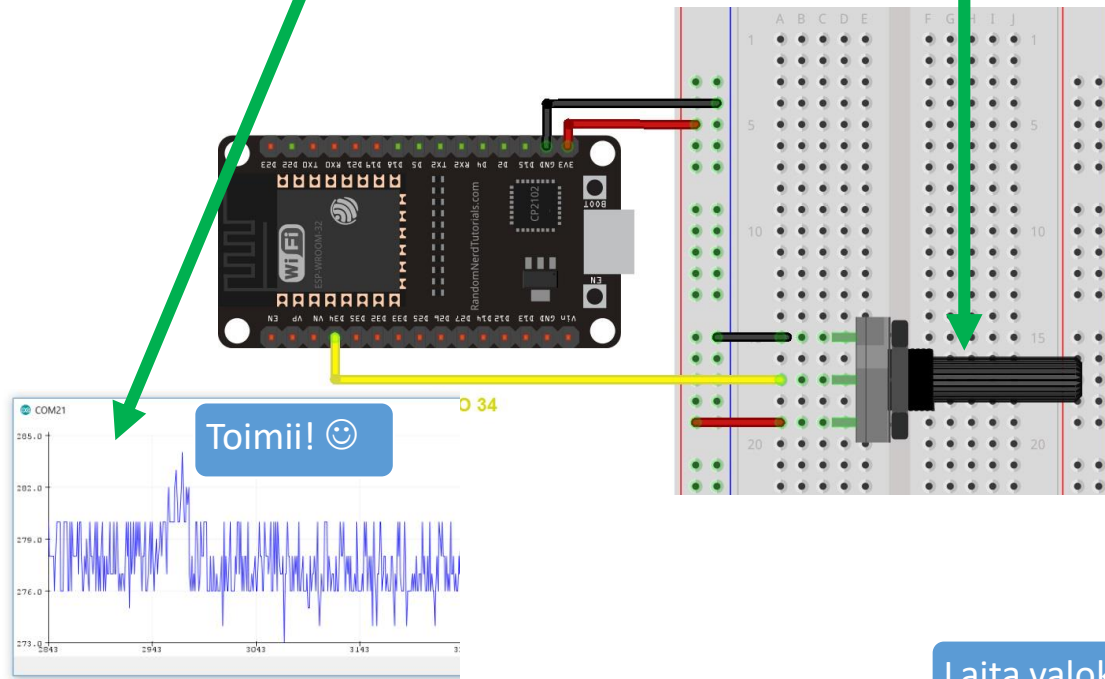
2. Sensors and Actuators Exercise

ESP32 kytkentä

Katsoppa tämä www-sivu! <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>

1. Tee kytkentä! Uploada koodi ja testaa että toimii. Ota valokuva kytkennästä ja snapshot siitä, että koodi toimii!

Tämä on potentiometri



Open your Arduino IDE and copy the following code.

```
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value
int potValue = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);
}

void loop() {
  // Reading potentiometer value
  potValue = analogRead(potPin);
  Serial.println(potValue);
  delay(500);
}
```

Laita valokuva ja snapshot raporttiin. 2p tienattu!

ESP32 kytcentä ja kohinaaaaaaaa!

```
/*
 * Noise_Test_1.ino
 * 27th Sept. by Mr T
 */

int sensorPin = A0; // select the input pin for the potentiometer
int sensorValue = 0; // variable to store the sensor value

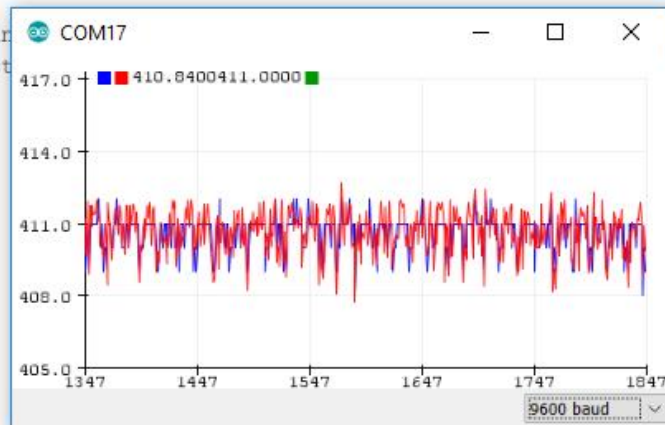
//Serial output refresh time
const long SERIAL_REFRESH_TIME = 10;
long refresh_time;

void setup() {
  Serial.begin(9600);
  analogReference(EXTERNAL);
}

void loop() {
  // read the value from the sensor (Potentiometer)
  // float sensorValue = (analogRead(sensorPin)/1024*100)+random(-100,100)/100.0;

  float sensorValue = analogRead(sensorPin);
  float sensorValueNoise = sensorValue + random(-100,100)/100.0;

  if (millis() > refresh_time) {
    Serial.print(sensorValue,4);
    Serial.print(",");
    Serial.print(sensorValueNoise,4);
    Serial.print(",");
    Serial.println();
    refresh_time = millis() + SERIAL_REFRESH_TIME;
  }
}
```



2. Vieressä on Arduino, ei ESP32-koodi, jossa Tane on lisännyt pseudosatunnaista kohinaa mitattuun signaaliin. Pölli Tanen idea ja lisää omaan koodiisi vastaavalla tavalla kohinaa!

Ota snapshot koodista ja kohinaisesta signaalista ja liitä ne raporttiin. 4p LISÄÄ tienattu!

ESP32 ja kohinan poistoa!

3. Itse menit lisäämään kohinaa signaaliin ja nyt se sitten pitäisi poistaa keskiarvoistamalla!



```
const int numReadings = 10;

int readings[numReadings];
int readIndex = 0;
int total = 0;
int average = 0;
int inputPin = A0;

void setup() {
  Serial.begin(9600);
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}

void loop() {
  total = total - readings[readIndex];
  readings[readIndex] = analogRead(inputPin);
  total = total + readings[readIndex];
  readIndex = readIndex + 1;

  if (readIndex >= numReadings) {

    readIndex = 0;
  }

  average = total / numReadings;
  Serial.println(average);
  delay(1);
}
```

Tutustu huolella tämän linkin tarinaan!
<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Smoothing>

Käytä smoothing tekniikkaa ja poista kohina ESP32sta!

Tulosta ALKUPERÄINEN ja KESKIARVOISTETTU SIGNAALI samana näkymään!

Ota snapshot koodista ja signaaleista ja liitä ne raporttiin. 4p LISÄÄ tienattu!