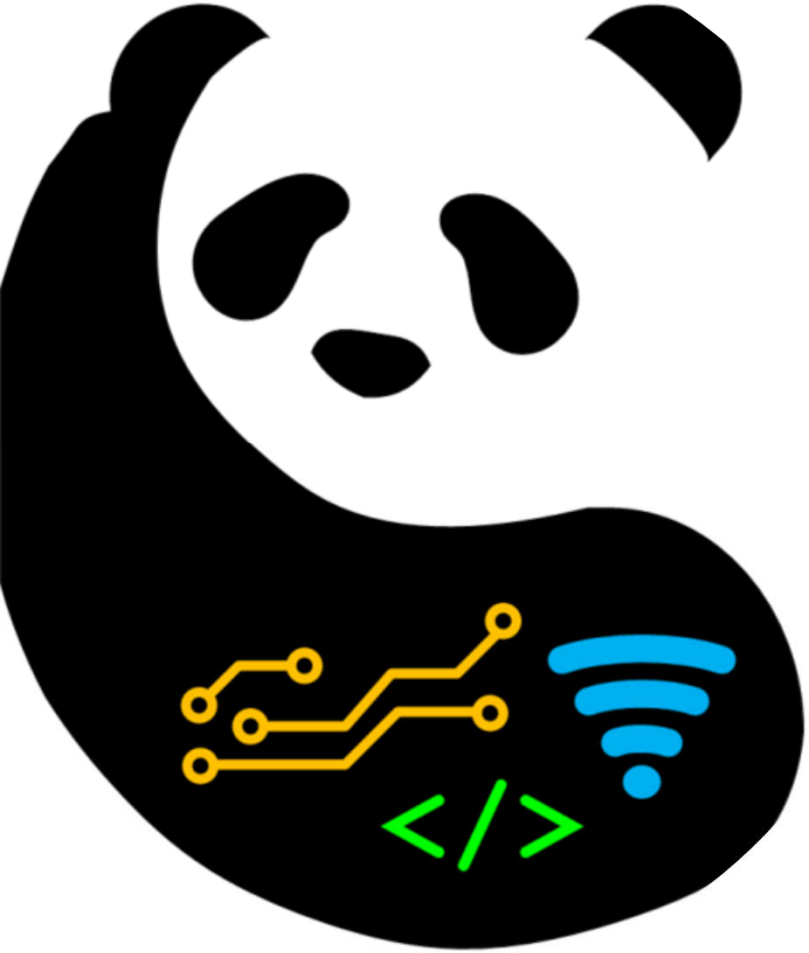
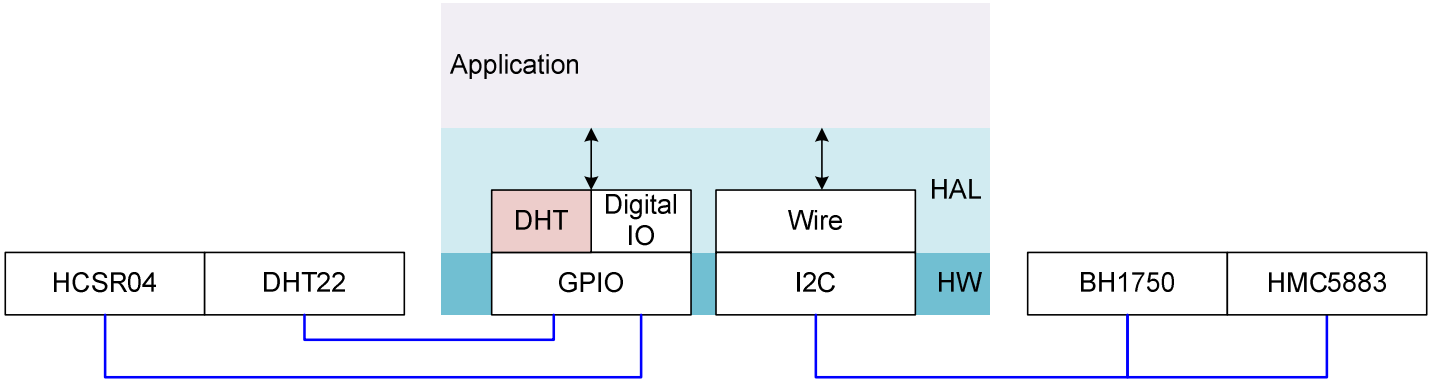


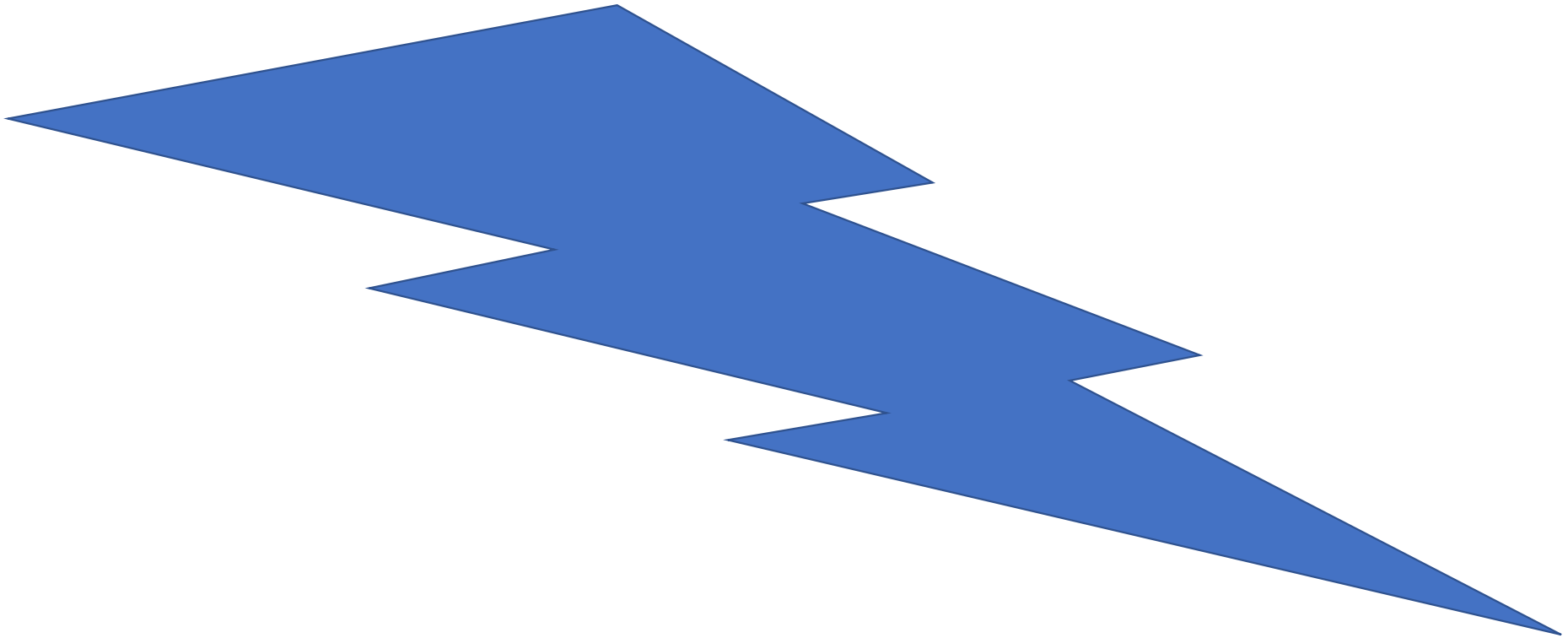
Lab. 4 Sensor Module Interface
- I2C
- One-wire communication
- Pulse width
- Readout data vs processed data
- Study on sensor interface,
    - combine sensor-read function to task scheme

Background knowledge



- + knowledge from Lab. 1, 3

L04p01

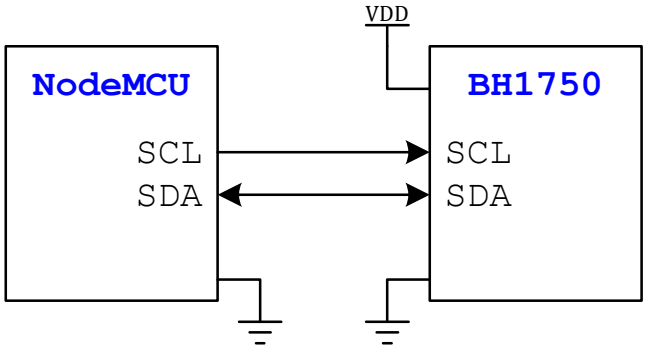


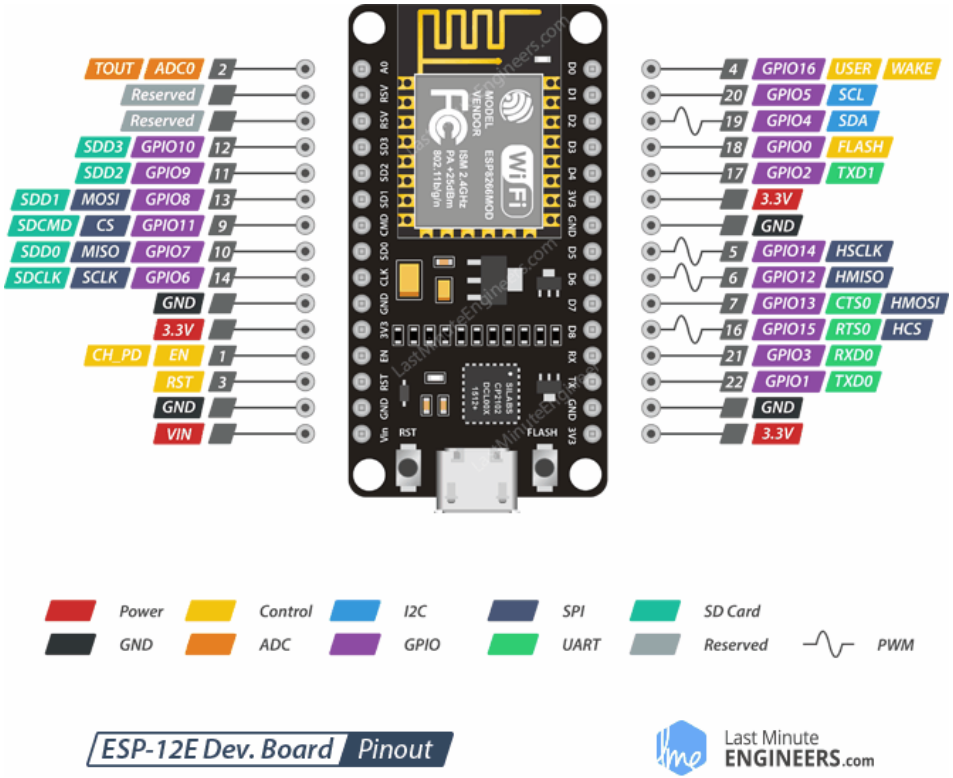
I2C Write | BH1750

```
#include <Wire.h>
#define BH1750address  0x23
```

Write Data to BH1750, (setup)

```
Wire.begin();
Wire.beginTransmission(BH1750address);
Wire.write(0x10);
Wire.endTransmission();
```





I2C Read | BH1750

Read Data from
BH1750

16-bit unsigned number of
Light intensity

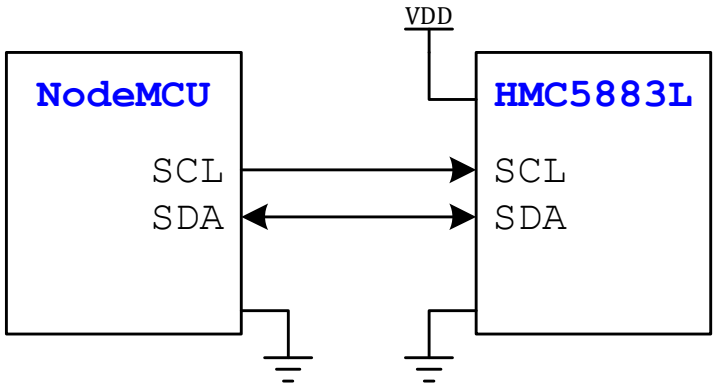
1 step readout-data → ? Lux (SI unit)

```
void readBH1750() {
    byte i, buff[2];
    unsigned int rodata;
    Serial.print("Task: read data from BH1750, ");
    Wire.beginTransmission(BH1750address);
    Wire.requestFrom(BH1750address, 2);
    i = 0;
    while(Wire.available()) //
    {
      buff[i] = Wire.read();  // receive one byte
      i++;
    }
    Wire.endTransmission();
    if (i == 2) {
      rodata = buff[0]*256 + buff[1];
      Serial.print(" Readout = ");
      Serial.print(i, HEX);
      Serial.print("\n\r");
    }
    else {
      Serial.println("!reading fails! \n\r");
    }
}
```

L04p02

HMC5883L connection

```
#include <Wire.h>
#define HMC5883Laddress  0x0D
```



I2C Write | HMC5883L

Write Data to HMC5883L, (setup)

```
Wire.beginTransmission(HMC5883Laddress);
Wire.write(0x0B);
Wire.write(0x01);
Wire.endTransmission();
delay(10);
Wire.beginTransmission(HMC5883Laddress);
Wire.write(0x09);
Wire.write(0x1D);
Wire.endTransmission();
```

Register address of HMC5883L

Data for the address

I2C Read | HMC5883L

Write 0x00 to HMC5883L
in order to set register address to 0x00

Read 6 bytes from HMC5883L
ranging from 0x00 to 0x05

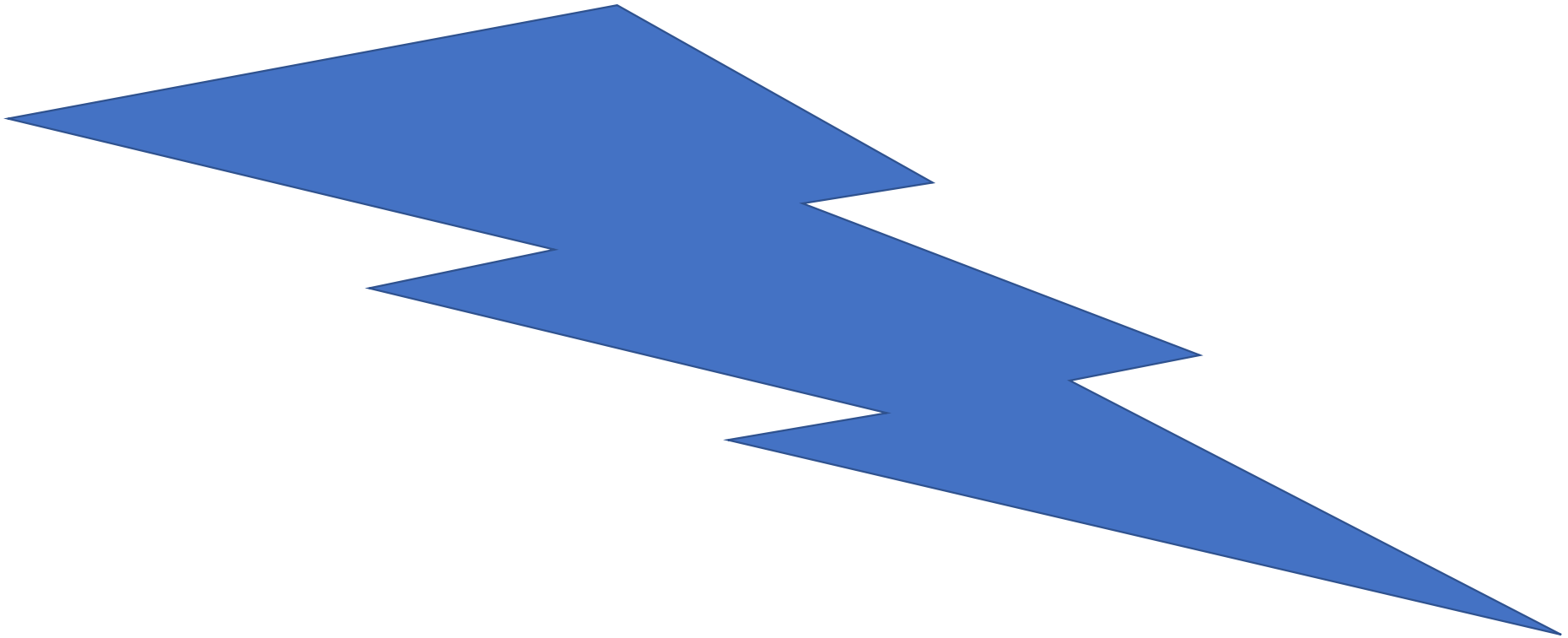
```
Wire.beginTransmission(HMC5883Laddress);
Wire.write(0x00);
Wire.endTransmission();
delay(1);
Wire.requestFrom(HMC5883Laddress, 6);
while(Wire.available()) //
{
  buff[i] = Wire.read();  // receive one byte
  i++;
}
Wire.endTransmission();
```

Readout 6 bytes are 3 sets of 16-bit signed number

1 step readout → ? (SI unit)

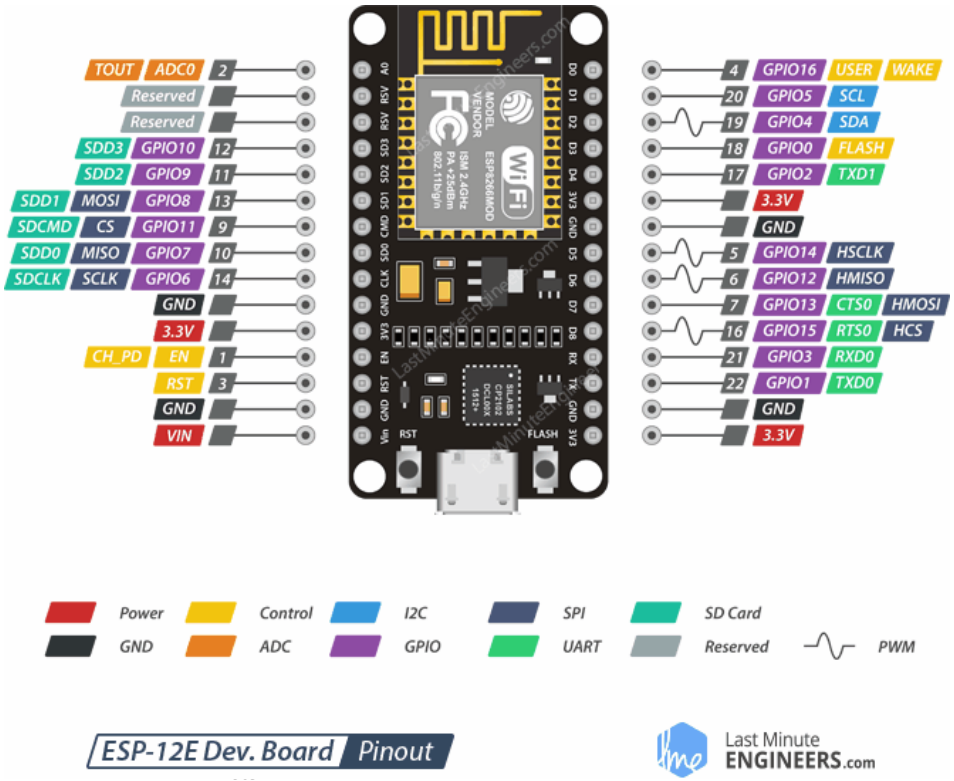
```
if (i == 6) {
  Serial.print(" X = ");
  rodata = buff[0]*256 + buff[1];
  Serial.print(rodata, HEX);
  Serial.print(", Z = ");
  rodata = buff[2]*256 + buff[3];
  Serial.print(rodata, HEX);
  Serial.print(", Y = ");
  rodata = buff[4]*256 + buff[5];
  Serial.print(rodata, DEC);
  Serial.print("\n\r");
}
else {
  Serial.println("!reading fails! \n\r");
}
```

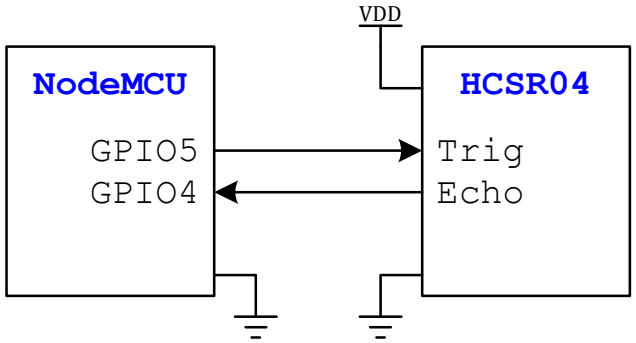
L04p03



HCSR04 connection

```
const int pingPin = 5;
const int inPin = 4;
```





Supply – 5 V

HCSR04: Ultrasonic distance sensor

```
const int pingPin = 5;
const int inPin = 4;
```

```
pinMode(inPin, INPUT);
pinMode(pingPin, OUTPUT);
```

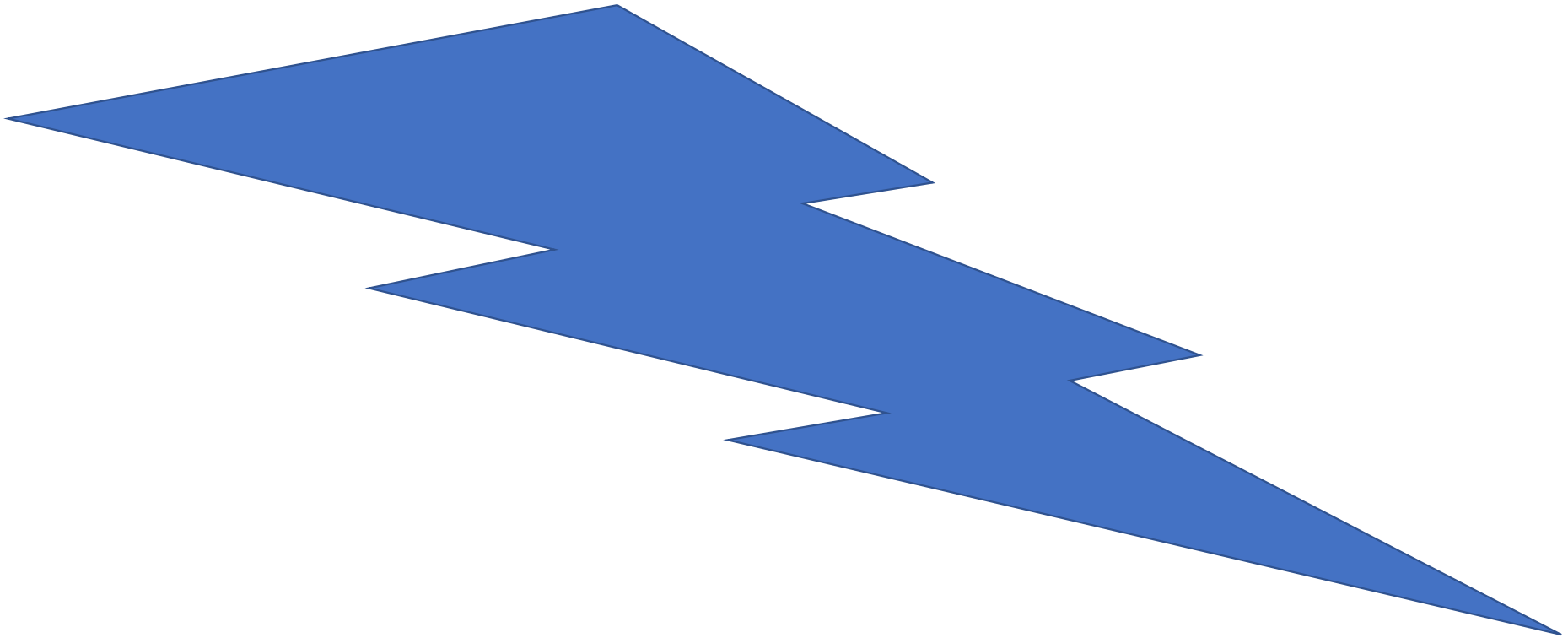
Send a high pulse to trig the module

Read pulse width of echo signal

```
void readHCSR04() {
    int i = 0;
    Serial.print("Task: read data from HCSR04, ");
    unsigned long duration;
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    duration = pulseIn(inPin, HIGH);
    Serial.print(duration, HEX);
    Serial.print("\n\r");
}
```

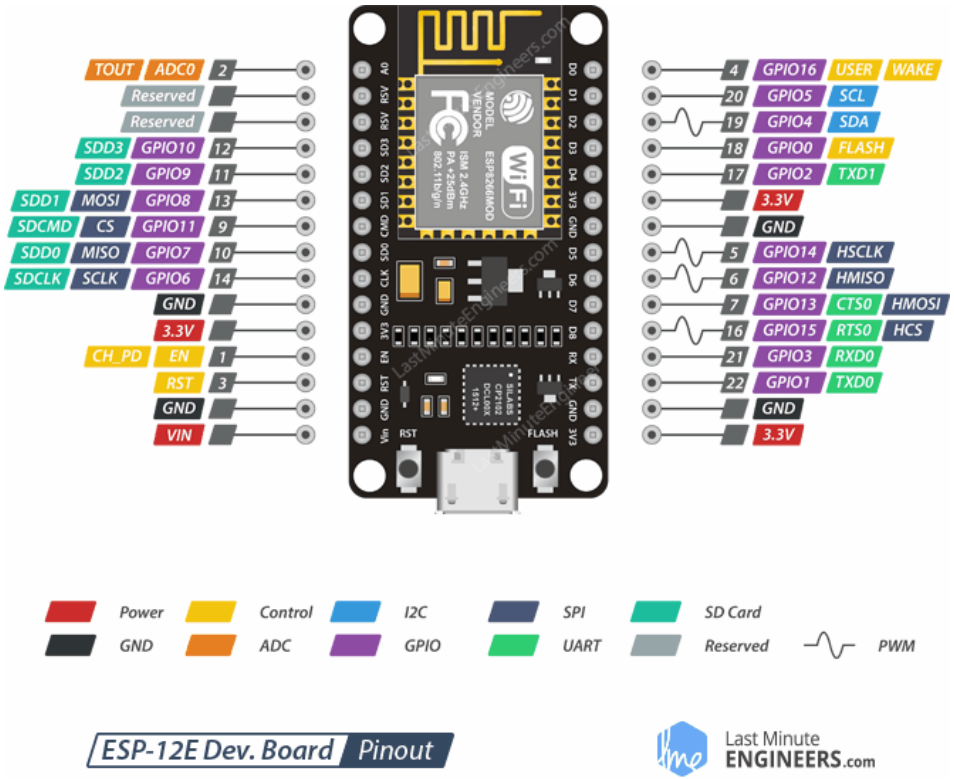
convert pulse width to distance [cm],
Calibrate & Calculation

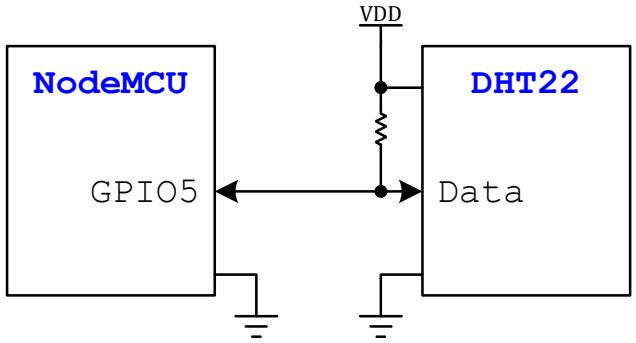
L04p04



DHT22 connection

```
#include <DHT.h>
const int DHTPIN = 5;
```





Supply – 5 V

DHT22: One-wire communication

```
#include <DHT.h>
const int DHTPIN = 5;
```

Read data from DHT22
The library read and convert data in SI unit.

```
void readDHT22() {
    int i = 0;
    float hum, temp;
    Serial.print("Task: read data from DHT22, ");
    hum = dht.readHumidity();
    temp = dht.readTemperature();
    if (isnan(hum) || isnan(temp)) {
      Serial.println("Failed to read from DHT sensor!");
      return;
    }
    Serial.print("Humidity: ");
    Serial.print(hum);
    Serial.print(" %, Temp: ");
    Serial.print(temp);
    Serial.println(" Celsius");
}
```