Task 4 :
Measuring and sending sensor values from one Raspberry device to another

Sensors used :

- SY-HS-220 Humidity Sensor (DHT22/AM2302)
- SHARP GP2Y0A02YK0F – Infrared Distance Sensor
- LM35 Temperature Sensor
- LDR 5mm high precision
- LSM9DS1 (9 DOF)

MEASURING VALUES USING THE SENSORS

1. SY-HS-220 Humidity Sensor (DHT22/AM2302)

This sensor measures the humidity (%) amd temperature of the surrounding. The sensor has 3 pins.

1 - Vcc
2 - GND
3 - OUT

The Vcc pin is connected to the 3v3 pin and GND is connected to any GND pin on Raspberry PI. The OUT pin is the output and is connected to GPIO 4 and specified that in the python program.

This uses Adafruit library to detect the sensor. The library Adafruit_DHT is used to detect the sensor using its name (i.e. AM2302). This library provides a function (read_retry()) to measure humidity and temperature together with the first value being the humidity in percentage and temperature in degree Celcius.

2. SHARP GP2Y0A02YK0F – Infrared Distance Sensor

This sensor measure the distance to the closest object infront of it by sending infrared rays. Its range is from 20cm to 150cm.

This sensor also has 3 pins namely Vcc, GND, OUT.

The Vcc is connected to the 5V pin of the Raspberry Pi.
The GND is connected to any GND pin on the Raspberry Pi.
The OUT is connected to the CH0 of the MCP3008 ADC.

I used an Analog to Digital convertor here since the data given to the RPI is in analog. The MCP3008 is connected as follows :

VDD -> 3.3V
VREF -> 3.3V
AGND -> GND
CLK -> GPIO 18
DOUT -> GPIO 23
DIN -> GPIO 24

CS/SHDN -> GPIO 25
DGND -> GND

Here I have used Adafruit_MCP3008 and Adafruit_GPIO.SPI libraries for python programming.

The distance is calculated as follows :

1. First calculate the voltage using the formula
   v = (mcp.read_adc(0) / 1023.0) * 3.3

2. The distance is calculated using this voltage by the formula
   dist = 16.2537*v^4 – 129.893*v^3 + 382.268*v^2 – 512.611*v + 301.439

The above distance is calculated to be in centimeters.

3. LM35 Temperature Sensor

The LM35 Temperature Sensor also has 3 pins with Vcc, GND and OUT. The Vcc is connected to 3v3, GND is connected to any ground and the OUT is connected to the CH1 of MCP3008

The MCP3008 is connected as follows

VDD -> 3.3V
VREF -> 3.3V
AGND -> GND
CLK -> GPIO 11 (SPI0 SCLK)
DOUT -> GPIO 9 (SPIO MISO)
DIN -> GPIO 10 (SPIO MOSI)
CS/SHDN -> GPIO 8 (SPI0 CS1)
DGND -> GND

I used the Spidev library to detect the analog value using the function spidev.SpiDev.xfer2().

The data is read as follows:

1. Analog data is read using xfer2([1, (8 + 1) << 4, 0])

2. The data is the processed by the formula
   data = ((adc[1] & 3) << 8)+ adc[2]

3. This is then converted to degree Celcius using the formula :
   temp = ((data * 330) / 1023)

4. LDR 5mm high precision

The LDR (Light Dependent Resistor) sensor is used to detect the intensity of the light around it. If the surroundings is dark, it gives a high value and if it is bright, it gives a very low value. Using this we can judge the intensity of light in a particular region.

In the light, this sensor will have a resistance of only a few hundred ohms while in the dark, it can have a resistance of several megaohms.

The one end of the LDR is connected to the CH0 of the MCP3008 and the other end is grounded. A resistor of 10kOhm is connected in series with the LDR and the other end of the resistor is connected to 3v3.

This can be connected along with LM35 Temperature Sensor. We require the MCP3008 here and the connections are done same as LM35 and the light level is calculated using the same function xfer2([1, (8 + 1) << 4, 0]). This data is then printed.

5. LSM9DS1 (9 DOF)

This sensor has an accelerometer, gyrometer, magnetometer and also measures the temperature.
The straight male headers are first soldered in to the sensor and connected on a breadboard.

The connection to the Raspberry Pi can be done in 2 ways.

1. I2C
2. SPI

I used the I2C connection here. The connection is done as follows:

```
VIN -> 3V3
GND -> GND
SCL -> PI SCL (GPIO 3)
SDA -> PI SDA (GPIO 2)
```

This sensor requires adafruit_lsm9ds1 library. After the connections are made the I2C is detected by the command

```
i2cdetect -y 1
```

Ensure that not all the spaces are – in the output of the command before proceeding. The program imports the lsm9ds1 adafruit library, board and busio. The I2C is defined using busio and the sensor is defined using lsm9ds1 library.

The library has 4 functions – acceleration, magnetic, gyro and temperature. These four functions measures the acceleration, magnetic, gyro and temperature respectively. The values are then printed on the terminal

SENDING VALUES TO RASPBERRY PI DEVICES

The values are sent using MQTT protocol. The device uses mosquitto to implement this protocol. The reciever acts as a subscriber and the sender is the publisher.

On reciever’s side, the command is given on the terminal:

```
mosquitto_sub -t datas
```

On sender’s side, a python script is run that connects to the subscriber and then publishes the data to the same topic ‘datas’ everytime a reading is obtained from the sensor. In this way, the data is send to the RPIs.

The following command can also be used to publish a data

```
mosquitto_pub -h “ipaddress” -t datas -m “message_to_send”
```