notebook.community

EDIT AND RUN

Lesson 1: Configure Raspberry Pi to Collect Data from Temperature Sensor

Learning Objectives:

1. Create IoT device that collects temperature and humidity data from sensor
2. Learn how to wire DHT temperature sensors to GPIO on Raspberry Pi
3. Write Python code to interact with DHT sensor

In this lesson we will be creating a IoT device as a protoype for a sensor collector that collects enviromental data.

We will use this device later to save the data it collects to a back end database so the data can be analysed.

Exercise 1: Create circuit to collect sensor data

Circuit to wire DHT11 to Raspberry Pi:
Reference: http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/

Code to collect data from Raspberry Pi from DHT11 sensor:
Reference: https://github.com/adafruit/Adafruit_Python_DHT

Run the following code on the Raspberry Pi:

```
sudo apt-get update
sudo apt-get install build-essential python-dev
```

Install the DHT sensor libraries:

```
sudo git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

To test:

```
sudo ~/Adafruit_Python_DHT/examples/AdafruitDHT.py 11 4
```

You should get an output similar to:

```
Temp=23.0*  Humidity=35.0%
```

Exercise 2: Write Python code on the Raspberry Pi to collect data from the DHT sensors:

From the command prompt on the Raspberry Pi run:

```
sudo nano pilogger1.py
```

Enter the following code to collect data from DHT sensor:

```
In [ ]:

#!/usr/bin/env python
#
# This project will collect temperature and humidity information using a DHT t
#

import Adafruit_DHT
import time
import RPi.GPIO as GPIO
import datetime

# General settings
prog_name = "pilogger1.py"

# DHT Sensor settings
# Sensor should be set to Adafruit_DHT.DHT11,
# Adafruit_DHT.DHT22, or Adafruit_DHT.AM2302.

dht_sensor_port = 4                     # Connect the DHT sensor to port 4
dht_sensor_type = Adafruit_DHT.DHT11

device = "pi-003"                           # Host name of the data collector

GPIO.setmode(GPIO.BCM)                  # Use the Broadcom pin numbering
GPIO.setup(dht_sensor_port, GPIO.IN)    # DHT sensor port as input
```

```
# Print welcome
print('[{0:s}] starting on {1:s}...'.format(prog_name, datetime.datetime.today

# Main loop
try:
    while True:
        hum, temp = Adafruit_DHT.read_retry(dht_sensor_type, dht_sensor_port)
        temp = temp * 9/5.0 + 32
        now = datetime.datetime.now()
        date = now.strftime('%Y-%m-%d %H:%M:%S')
        print('{0:s},{1:s},{2:0.1f},{3:0.1f}'.format(device,date,temp,hum))
        time.sleep(1)

except (IOError,TypeError) as e:
        print("Exiting...")

except KeyboardInterrupt:
        # here you put any code you want to run before the program
        # exits when you press CTRL+C
        print("Stopping...")

finally:
        print("Cleaning up...")
        GPIO.cleanup() # this ensures a clean exit
```

Exercise 3: Test the temperature logger program

Run the python code on the Raspberry Pi by running the program as follows:

```
$ sudo chmod +x pylogger1.py
$ sudo ./pylogger1.py
```

You should see something like this:

```
rmj@pi223:~ $ sudo ./pilogger1.py
[sudo] password for rmj:
[pilogger.py] starting on 2017-07-23 19:23:19...
pi223,2017-07-23 19:23:20,71.6,36.0
pi223,2017-07-23 19:23:22,71.6,35.0
pi223,2017-07-23 19:23:23,71.6,34.0
pi223,2017-07-23 19:23:25,71.6,33.0
pi223,2017-07-23 19:23:26,71.6,32.0\
```

Press CTRL+C to exit out of the program:

```
^CStopping...
Cleaning up...
```

Content source: richjimenez/mysql-data-raspberry-pi

Similar notebooks:

- lesson-1-raspberry_pi_sensor
- Input - Temperatuur
- Input - Beweging (PIR)
- lesson-4-save-raspberry-pi-sensor-data-to-mysql-database
- 上傳檔案 - MQTT
- Input - Motion (PIR)-checkpoint
- 上傳檔案到 NodeMCU (Upload files to NodeMCU)
- 102 - LEDs - De Raspberry Pi GPIO pinnen aansturen-checkpoint
- 102 - LEDs - De Raspberry Pi GPIO pinnen aansturen
- camera