notebook.community

EDIT AND RUN

Lesson 4: Save Raspberry Pi Sensor Data to MySQL Database

Learning Objectives:

1. Model how IoT device can save sensor data to back end database server for later analyis
2. Learn how to access data from MySQL databases from a Raspberry Pi Python program

Exercise 1: Modify Python script to add code to save data to MySQL database

In [ ]:

```
#!/usr/bin/env python
#
# This project will collect temperature and humidity information using a DHT2
# and send this information to a MySQL database.
#
import Adafruit_DHT
import time
import RPi.GPIO as GPIO
import datetime
import MySQLdb

# General settings
prog_name = "pilogger2.py"

# Settings for database connection
hostname = '172.20.101.81'
username = 'piuser3'
password = 'logger'
database = 'pidata'

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

device = 'pi-003'                                   # Host name of the Pi
```

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

# Routine to insert temperature records into the pidata.temps table:
def insert_record( device, datetime, temp, hum ):
        query = "INSERT INTO temps3 (device,datetime,temp,hum) " \
                "VALUES (%s,%s,%s,%s)"
        args = (device,datetime,temp,hum)

        try:
                conn = MySQLdb.connect( host=hostname, user=username, passwd=p
                cursor = conn.cursor()
                cursor.execute(query, args)
                conn.commit()

        except Exception as error:
                print(error)

        finally:
                cursor.close()
                conn.close()

# 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_sens
                temp = temp * 9/5.0 + 32
                now = datetime.datetime.now()
                date = now.strftime('%Y-%m-%d %H:%M:%S')
                insert_record(device,str(date),format(temp,'.2f'),format(hum,
                time.sleep(180)

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 2: Test the temperature logger program

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

```
$ sudo chmod +x pylogger2.py
$ sudo ./pylogger2.py
```

Now let's check the database`

```
In [5]:

%load_ext sql
```

```
In [7]:

%%sql mysql://piuser3:logger@172.20.101.81/pidata
select * from temps3;
```

```
32 rows affected.

Out[7]:
```

device	datetime	temp	hum
pi-003	2017-07-23 21:10:10	71.6	35.0
pi-003	2017-07-23 21:14:52	71.6	34.0
pi-003	2017-07-23 21:14:54	71.6	39.0
pi-003	2017-07-23 21:14:55	71.6	31.0
pi-003	2017-07-23 21:14:57	71.6	31.0
pi-003	2017-07-23 21:14:58	71.6	31.0
pi-003	2017-07-23 21:15:00	71.6	31.0

pi-003	2017-07-23 21:15:02	71.6	32.0
pi-003	2017-07-23 21:15:03	71.6	31.0
pi-003	2017-07-23 21:15:05	71.6	31.0
pi-003	2017-07-23 21:15:06	71.6	31.0
pi-003	2017-07-23 21:15:08	71.6	31.0
pi-003	2017-07-23 21:15:10	71.6	41.0
pi-003	2017-07-23 21:15:11	71.6	31.0
pi-003	2017-07-23 21:15:13	71.6	31.0
pi-003	2017-07-23 21:15:14	71.6	32.0
pi-003	2017-07-23 21:15:16	73.4	35.0
pi-003	2017-07-23 21:15:17	71.6	31.0
pi-003	2017-07-23 21:15:19	71.6	31.0
pi-003	2017-07-23 21:15:21	71.6	31.0
pi-003	2017-07-23 21:15:22	71.6	31.0
pi-003	2017-07-23 21:15:24	71.6	31.0
pi-003	2017-07-23 21:15:25	71.6	31.0
pi-003	2017-07-23 21:15:27	71.6	31.0
pi-003	2017-07-23 21:15:28	71.6	33.0
pi-003	2017-07-23 21:15:30	71.6	31.0
pi-003	2017-07-23 21:15:32	71.6	31.0
pi-003	2017-07-23 21:15:33	71.6	31.0
pi-003	2017-07-23 21:15:35	71.6	31.0

```
pi-003    2017-07-23 21:15:36    71.6    31.0
pi-003    2017-07-23 21:15:38    71.6    31.0
pi-003    2017-07-23 21:15:39    71.6    31.0
```

Press CTRL+C to cancel the program

You may want to change the sleep time to a larger number to take samples ever 5 minutes (300) seconds for example.

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

Similar notebooks:

- lesson-4-save-raspberry-pi-sensor-data-to-mysql-database
- lesson-1-raspberry_pi_sensor
- programme_in_booklet
- lesson-3-mysql-database-for-raspberry-pi-sensor-ata
- Feedback k domácím projektům
- Input - Temperatuur
- Input - Beweging (PIR)
- Snippets_MySQL
- Python3_tutorial
- Python for MySQL_2jan15