

Using Python and MySQL

The next step was to add info to a database whenever the sensor was triggered. For my purposes, I would just use three simple entries in the database: date, time, and sensor type. While I am only using one sensor for this project, I have a sensor type entry in the event I wish to expand the project to use more sensors. So, the first thing I would need to do is create a database to connect to, so I log into MySQL and enter:

```
MariaDB [(none)]> CREATE DATABASE sensordb;
Query OK, 1 row affected (0.02 sec)
MariaDB [(none)]> USE sensordb
Database changed
MariaDB [sensordb]> CREATE TABLE sensordat (sdate DATE, stime TIME, senstype
VARCHAR(30));
Query OK, 0 rows affected (0.12 sec)
```

Now to make a PHP script to add data to the database whenever the sensor triggers the 'Input went HIGH' script. Using the test script, I will add one line to add the data to the database when it is triggered. For troubleshooting purposes, I will display the info in the database after every time it is triggered. After the database count reaches 5 entries, I will clear the database.

```
#!/usr/bin/python
from time import sleep
import RPi.GPIO as GPIO
import MySQLdb
GPIO.setmode(GPIO.BOARD)

db = MySQLdb.connect(host="localhost", # your host, usually localhost
                     user="root", # your username
                     passwd="n02293588", # your password
                     db="sensordb") # name of the data base

# you must create a Cursor object. It will let
# you execute all the query you need
cur = db.cursor()

from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)

GPIO.setup(12, GPIO.IN)
x = 0
db_count = 0
while 1:

    if (GPIO.input(12) and x == 1):
        print('Input went HIGH')
        with db:
            cur.execute("INSERT INTO sensordat VALUES
(CURRENT_DATE(), NOW(), 'm')")
            cur.execute("SELECT * FROM sensordat")
            for reading in cur.fetchall():
                print str(reading[0])+" "+str(reading[1])+"
"+str(reading[2])
            x = 0
```

```

        db_count = db_count + 1

    elif (not GPIO.input(12) and x == 0):
        print('Input went LOW')
        x = 1

    if (db_count == 5):
        print("Database reached 5 entries, DB will be cleared")
        with db:
            cur.execute("DELETE FROM sensordat")
            db_count = 0
#         sleep(1);
db.close()

```

I ran into a few problems executing this program. After clearing up some syntax errors, I learned that Arch Linux defaulted to use Python3, while the mysql-python library was for Python2, so instead of running the programs with python, I had to run them with python2. I had to reinstall the RPi-GPIO package in the same way, by running the setup.py file with python2 instead of python.

As a suggestion of a friend, I put all of the database commands into a single if block to make the code a little bit neater. In order to do this, I had to change how the database functioned. If I had used the MySQL functions to gather the DATE and TIME, they wouldn't appear properly because they would all be added to the database at the same time. To fix this, I made a new table that would accept two strings. The first would be the timestamp generated by python, and the second is the sensor type. The python file then read as follows:

```

#!/usr/bin/python
from time import sleep
import RPi.GPIO as GPIO
import MySQLdb
import time
GPIO.setmode(GPIO.BOARD)

db = MySQLdb.connect(host="localhost", # your host, usually localhost
                     user="root", # your username
                     passwd="n02293588", # your password
                     db="sensordb") # name of the data base

# you must create a Cursor object. It will let
# you execute all the query you need
cur = db.cursor()

from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)

GPIO.setup(12, GPIO.IN)
x = 0
db_count = 0

sens = 'm'
senslist = []
biglist = []

```

```

with db:
    cur.execute("DELETE FROM sensordat2")

while 1:

    if (GPIO.input(12) and x == 1):
        print('Input went HIGH')

        localtime = time.asctime( time.localtime(time.time()) )
        senslist = [localtime, sens]
        biglist.append(senslist)

        x = 0
        db_count = db_count + 1

    elif (not GPIO.input(12) and x == 0):
        print('Input went LOW')
        x = 1

    if (db_count == 5):
        print(biglist)

        cur.executemany('INSERT INTO sensordat2 VALUES (%s, %s)',
biglist)

        cur.execute("SELECT * FROM sensordat2")
        for reading in cur.fetchall():
            print str(reading[0])+" "+str(reading[1])

        print("Database reached 5 entries, DB will be cleared")
        with db:
            cur.execute("DELETE FROM sensordat2")
            db_count = 0
            biglist = []

        # sleep(1);
db.close()

```

This code generated this output:

```

Input went LOW
Input went HIGH
Input went LOW
Input went HIGH
Input went LOW
Input went HIGH
Input went LOW
Input went HIGH
Input went LOW
Input went HIGH
[[ 'Wed Nov 20 18:50:34 2013', 'm'], ['Wed Nov 20 18:50:40 2013', 'm'], ['Wed
Nov 20 18:50:46 2013', 'm'], ['Wed Nov 20 18:50:50 2013', 'm'], ['Wed Nov 20
18:50:53 2013', 'm']]
Wed Nov 20 18:50:34 2013      m
Wed Nov 20 18:50:40 2013      m

```

```
Wed Nov 20 18:50:46 2013      m
Wed Nov 20 18:50:50 2013      m
Wed Nov 20 18:50:53 2013      m
Database reached 5 entries, DB will be cleared
```

Now that the database is working, it is time to have it display on a website.

[Previous](#)

[Back to Documents](#)

[Next](#)