

Creating a Lighttpd, PHP5, and SQLite3 Web Interface on the Raspberry Pi

Nick Falco

November 2013

Contents

1	Configuration Steps	3
1.1	Installing Lighttpd, PHP5, and SQLite3	3
1.2	Setup SQLite3 Database	4
1.3	Give a fixed IP address to Raspberry Pi	5
2	Accessing Database Using PHP PDO Handler	6
2.1	Using HTML and PHP PDO Handler to Print Data in a Table format	6
3	Arduino Mega Configuration	7
3.1	Testing the Arudino Temperature Circuit	7

1 Configuration Steps

The following section describes the necessary steps to initially configure Lighttpd, PHP5, and SQLite3.

1.1 Installing Lighttpd, PHP5, and SQLite3

Step 1: Install lighttpd

```
apt-get install lighttpd
```

Step 2: Install php5

```
apt-get install php5, php5-common, php5-dev, php5-cli
```

Step 3: Install php5-cgi

```
apt-get install php5-cgi
```

Step 4: Install sqlite3

```
apt-get install sqlite3
```

Step 5:

```
apt-get install php5-sqlite
```

Step 6: Enable the fastcgi module and the php PDO drivers

```
lighttpd-enable-mod fastcgi  
lighttpd-enable-mod fastcgi -php
```

Step 7: Reload lighttpd

```
service lighttpd force-reload
```

NOTE:

At this point php PDO drivers will be enabled. and you must now verify that PHP5 is configured properly.

Step 8: navigate to the directory `/var/www/`

```
cd /var/www/
```

Step 9: Now create a new file `index.php`

```
nano index.php
```

Step 10: Add the following contents to the file:

```
<?php phpinfo(); ?>
```

Step 11: Type the following into the URL on your web browser and verify that PDO drivers are enabled.

```
localhost/index.php
```

1.2 Setup SQLite3 Database

Launch SQLite3 and create new database file SQLite3 ;database name; (e.g. SQLite3 data.db) Create a table with the following command. Count is the sample number, Time is the actual time the sample was taken, Temp is the temperature value measured.

- `CREATE TABLE Temperatures(Count int, Time char(20), Temp real);`

Import a comma separated file `tempData.csv` into table named `Temperatures`.

- `.separated “,”`
- `.import tempData.csv Temperatures`. An example `tempData.csv` is shown below:
1, 3/8/2010 12:00, 22
2, 3/9/2010 12:00, 23
3, 3/10/2010 12:00, 21
4, 3/11/2010 12:00, 27
5, 3/12/2010 12:00, 26
6, 3/13/2010 12:00, 33
7, 3/14/2010 12:00, 32
8, 3/15/2010 12:00, 19
9, 3/16/2010 12:00, 21
10, 3/17/2010 12:00, 19

Verify that the data was successfully imported into the database by running the to following query:

- SQLite3 data.db
- SELECT * FROM Temperatures;

The data you imported should appear below the SQL select statment

1.3 Give a fixed IP address to Raspberry Pi

Navigate to the following directory /etc/network/

- cd /etc/network/

Copy the original interfaces file so you have a backup

- cp interfaces interfaces.o

Edit the interfaces file using nano

- nano interfaces

In the interfaces file make the following changes:

Change the line

iface eth0 inet dhcp

to

iface eth0 inet static

Below this line enter the following using your router settings. (mine are different than yours) Here's an example:

address 192.168.1.118 (your new static ip address)

netmask 255.255.255.0 (netmask from your router)

network 192.168.1.1

broadcast 192.168.100.255

gateway 192.168.100.254 (gateway address from your router)

Now you need to set up your DNS servers. In your router find out what your DNS servers are and append the following onto the file /etc/resolv.conf:

- server <your DNS server number>
- server <your DNS server number>

2 Accessing Database Using PHP PDO Handler

At this point you should have completed the initial configuration steps. The system is now completely setup for data to be read from the database.

2.1 Using HTML and PHP PDO Handler to Print Data in a Table format

In order to access data from the the database through a *.php file you will need to make some changes to your index.php file. If you have the following line of code “<?php phpinfo(); ?>” from Section 1.1 in your index.php file - delete it. Now add the following lines of code to index.php to read data from the database and print it in an html table.

First access your index.php file using nano.

- sudo nano index.php

Now, add the following php code to your index.php file.

```
<?php
try {
$dbh = new PDO('sqlite:/var/www/realtime_data.db');
foreach($dbh->query('select * from Temperatures') as $row)
{
$result_temp[] = $row['Temp'];
$result_count[] = $row['Count'];
}
}
catch(PDOException $e) {
echo $e->getMessage();
}
? >
```

3 Arduino Mega Configuration

The following section discusses how to use the Arduino Mega to read the LM35 temperature sensor and transmit the temperature value serially to the Raspberry Pi. It is best to use an engineering approach in collecting data from the Arduino and sending data to the Raspberry Pi. This involves first verifying the temperature circuit works properly and then sending data to the Raspberry Pi.

3.1 Testing the Arudino Temperature Circuit

This section discusses how to verify the temperature recording circuit on the Arduino by reading in a temperature value and then sending it through the serial port to a PC.

First

Download the Arduino IDE at <http://arduino.cc/en/Main/Software> and upload the following code onto the Arduino Mega board.

```
double temp;
int tempPin = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  temp = analogRead(tempPin);
  temp = ( 5.0 * temp * 100.0) / 1024.0;
  Serial.print("TEMPRATURE = ");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

Second

The next step is to download a serial port monitoring tool on your PC and see if you are getting accurate data from the Arduino. I choose to use the open source program “Tera Term” as my serial monitoring tool. Tera Term may be downloaded at the following address:

<http://ttssh2.sourceforge.jp/index.html.en>

Configure Tera Term to the COM Port that the RS232 Cable is connected. The baud rate should also be set to 9600. If all is wired and configured properly you should begin seeing temperature data on the screen being sent from the Arduino as shown in Figure 1.

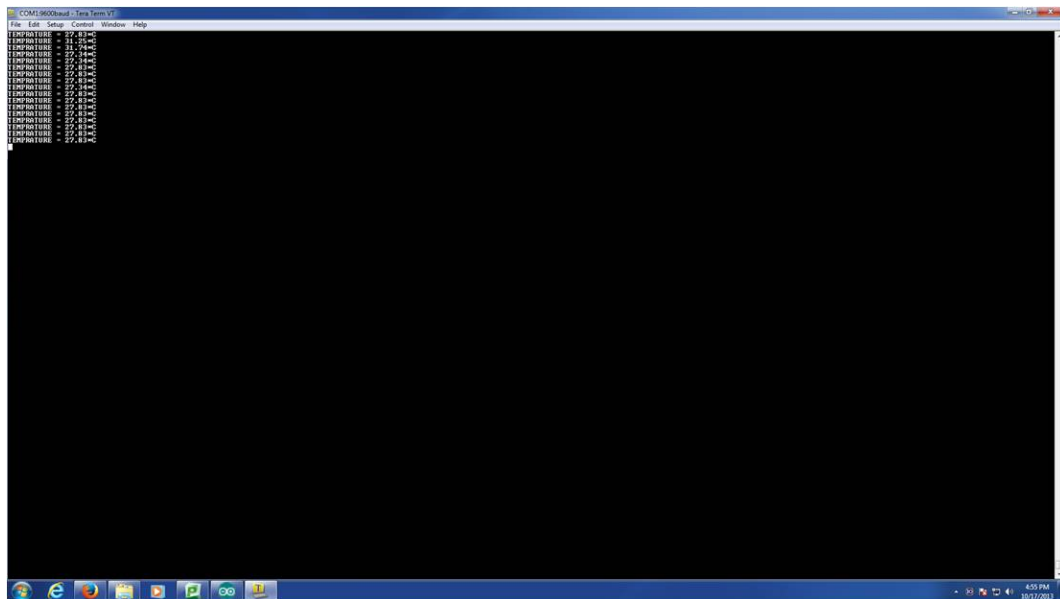


Figure 1: Tera Term After Recieving Several Temperature Values