One of the advantages of the Raspberry Pi is the ease at which it can be connected to network and by extension the internet. Here’s how we have accessed the GPIO pins of Raspberry Pi using simple PHP scripts and Web Browser of choice.

Step 1: At first we needed to setup a web server. We setup a typical LAMP (Linux, Apache, MySQL, and PHP/Perl) server in our system.Using first the sudo (Super User Do) modifier we use apt-get install/remove to install and uninstall various packages. First we install apache2 which will be our web server that responds to requests from web browsers connecting over port 80. apt-get will ask if we wish to download the resources necessary to install the requested package, we typed Y on the keyboard indicating Yes.

**Code:** pi@raspberrypi ~ $ sudo apt-get install apache2

Step 2: Next we should install a database server, do so by running the command below. You will be asked to enter a MySQL root password, this can be the same as your current password or something different if you want to be more secure about it.

**Code:** pi@raspberrypi ~ $ sudo apt-get install mysql-server

Step 3: Now, We write the PHP code within a file named index.php that must be placed within the folder as such:

**/var/www/html**

And then we need to edit the index.php by running this code in terminal

**Code:** sudoleafpadindex.php

This code opens a graphical editor named **leafpad** for writing the php code within it. We could have used **nano**editor but it is not a graphical editor so it is difficult to edit a webpage here.

Step 4: The code for the PHP page is given here in the repository

In Home Automation part the output is a php page that the client visits once he/she types @localhost within the raspberry pi browser or visits the IP address of the raspberry pi from different device’s browser.

Once visited in your web browser the Web GPIO Utility provides a simple table interface allowing the user to toggle Pin Modes between Input and Output and set Pin States from Low to High. The utility reads the current pin statuses using the command line utility and uses the Wiring-Pi functions to set pin modes and value.



**Figure 12: Table Interface**

The above picture is the page that the client will see once they visit the IP address of the PI itself. This is a interface from where the client can control the GPIO pins of the Raspberry PI.

To show what happens when the client clicks on one of the switches tagged as IN and changes it to OUT lets show the state of the 8 channel relay then all the switches in the PHP page is tagged as IN.



**Figure 13: 8 channel relay that switches in the PHP page tagged as IN**

Now the user clicks all the switches and the state of the PHP page is shown in the next image.



**Figure 14: Table Interface showing the state of php page**

Once the switches are pressed the GPIO pins are activated which in turn activates the Relay modules connected to it.



**Figure 15: Activated Relay Modules**

The above UI was too technical and a layman didn’t actually need all these information to be displayed to them so we have changed the UI as per our intended customer.

We came up with this UI as follows:



Figure 16: PHP UI

Limitations found in the Home Automation i.e. switching on/off devices are as follows:

* If there are many users accessing the php page at the same time then there might be some issues regarding synchronisation of data between different users. As of now to counter this problem the code is ignoring the second command but in future the code needs to be synced properly among different users automatically. This can be come if we use AJAX which will allow web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page i.e. the states on each GPIO pins, without reloading the whole page and thus allowing the second user to know that a task that he/she is attempting to do is previously done by some other user.