Using Raspberry Pi's GPIO Pins

Finally getting a motion sensor and some wires, I began to work on setting up data collection. My goal of this portion is to use Raspberry Pi's Input/Output pins to add a database entry when the sensor is triggered.
I began by figuring out how to use the GPIO pins. I found a great youtube video here.
Embedded for easier use:

The about section in this video provided links for RPi.GPIO, a tutorial they used, and a link to their .zipped project files.
Since he was not using Arch Linux, I had to do some internet searching on how to install RPi.GPIO. I was brought to the Arch Linux forums, and found a post that told me to do this:

```
pacman -Sy file base-devel abs git
wget https://aur.archlinux.org/packages/ra/raspberry-gpio-python/raspberry-gpio-python.tar.gz
tar xf raspberry-gpio-python.tar.gz
cd raspberry-gpio-python
makepkg -Acs --asroot
pacman -U raspberry-gpio-python
```

But a few posts later showed that the RPi.GPIO package was no longer hosted there, so I had gotted a direct link from the website. When I tried the code above after successfully downloading the package, after running the makepkg command, I ran into an error.

```
ERROR:PKGBUILD does not exist
```

Looking around a bit, I found that the new package no longer is installed in that manner. Instead, after extracting the package files, there is a python script used to set up the package, so I moved into the directory where I extracted it and ran

```
python setup.py install
```

After installing the program and watching through that youtube video, I figured using the examples from the RPi.GPIO homepage would be better learning material, so I could start off small.... and so I did. I made a python script called test2.py to see how the GPIO pins work.

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

GPIO.setup(12, GPIO.IN)
while 1:

        if GPIO.input(12):
                print('Input went HIGH')

        else:
                print('Input went LOW')
```

This program worked, but I was running into a problem. While the sensor was on (it hangs at 1 for a while, as long as there is motion in it's range), the console was spammed with Input went HIGH, and when it was off, the console was spammed with Input went LOW. I had to find out how to make this happen only once. The closest I got to that was when I replaced GPIO.input(12) with GPIO.wait_for_edge(12, GPIO.RISING). Then, only the Input went LOW was appearing once when it went low. I changed RISING to FALLING and the console showed Input went LOW despite whether or not the motion sensor was turning on or off. Finally, after

troubleshooting with my friends and classmates, and trying a few different methods, my friend/classmate Kevin suggested I add an if else statement instead of just and else, after we decided to try using variables to count how often it went off. We came up with this code below, which succesfully printed Input went HIGH once when it went high, and Input went LOW when it went low.

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

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

        if (GPIO.input(12) and x == 1):
                print('Input went HIGH')
                x = 0
        elif (not GPIO.input(12) and  x == 0):
                print('Input went LOW')
                x = 1
#       sleep(1);
```

My next goal is to put this into a database.

Thus concludes the installation of the LEMP server

Next

Previous

Back to Documents