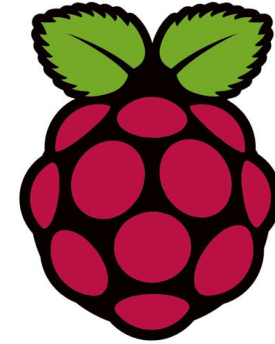


The Saga Continues

LEAP
M O T I O N



New to the Project

Auto-login

WiringPi

Start-up Script

Code to control external programs using Leap Motion

Future Works

Writing PWM code

Integrate SSH functionality

Finish Leap code.

Auto-Login and wiringPi

Auto – login:

- For the project to be truly “headless” I wanted to be able to log in to the command line of the Pi without entering a username/password combination.
- I found a tutorial that showed how to modify the inittab file in the /etc directory that allows for auto-login
- This means that the Pi will require no input to begin working and my project can be fully headless.

WiringPi

- WiringPi is a module for the Raspberry Pi that allows users to easily control the board's GPIO functionality.
- It came with a host of sample programs covering the capabilities of the GPIO including control over LEDs and software pulse width modulation.
- WiringPi is especially helpful for this project because it does not require root access to control LEDs.

Startup Script

- My project will be controlled directly from the command line of the Pi. I will use scripts run from the command line in order to control the LED's various functions.
- I realized that this meant I would have to have some way of knowing when the Pi was ready to accept inputs to the command line.
- Without a monitor, I would be unaware of when the Pi was ready for inputs.
- I needed to create a start-up script that would notify me(using the LED) when the pi was able to receive commands.
- Using the sample programs included with the wiringPi library, I wrote and compiled a simple program that would blink the LED 4 times.
- That's all well and good, but I needed to have this program run automatically when the Pi had fully booted up.

Startup Script Continued

- I did some searching and found a website that detailed how to run a bash script upon startup. All you need to do is edit the `.bashrc` file to have a reference to the script and it will automatically run on startup.
- I took this information and wrote a very simple script that simply pointed the command line to the directory containing the blink program. The script then runs the program, causing the LED to blink 4 times.
- This serves as a simple way of notifying the user(without a monitor) that the pi is ready to receive inputs, inputs that will eventually come through ssh.

Code to control external Programs, etc.

- Another discovery I made that will help my project is the ability to execute and control external applications using java.
- I plan to use a java program in conjunction with the leap motion controller to ssh into my pi and send various commands to it in order to control an LED connected to the Pi's GPIO outputs.
- This therefore necessitates the use of an external program like Putty.
- My java program controlled by the Leap now contains runtime and process methods that allow for the execution and manipulation of Putty.exe.
- I'll soon be able to send commands to the Raspberry Pi using only the Leap Motion and putty ssh client.

Future Works

- WiringPi includes tools for what is called “software pulse width modulation” which can control both the level of brightness and (in the case of an RGB LED) the color of the LED.
- As previously mentioned, I have figured out how to run putty using the leap but still need to set it up to automatically connect to my Pi.
- I have made quite a bit of progress with the java code controlling the functions of the Leap Motion but have yet to fully integrate the operations it is to perform. This will require more programming on both the desktop and Raspberry Pi.