

CSCI 373-02 Raspberry Pi GPIO Homework

Option 1:

In class we mentioned the need for PWM in developing a servo controller. This is the approach taken in the Servoblaster code package. Look at the Servoblaster repository on GitHub and read the README.txt file. As described in the README file, Mr Ghrist uses DMA to implement PWM. In your own works, explain DMA and how it might be used to implement PWM.

Submit your answer to Moodle before midnight on Oct 10 for full credit.

Option 2:

If you select this option, you will develop a better solution to controlling servos with the Raspberry Pi. The developer of WiringPi suggests using ServoBlaster by Richard Ghirst, but in order to use ServoBlaster you have to re-install your operating system. That's right, ServoBlaster does not work properly on Occidentalis so you will need to install Raspbian on your SD card before you can run Servoblaster. Fortunately, this is an easy job and you can get a new SD card if you don't want to lose your existing setup. To install Raspbian, follow the instructions provided [here](#). (It's really the same as installing Occidentalis, just a different image.)

Once you've installed Raspbian, you can install ServoBlaster. The code and documentation is available on GitHub at <https://github.com/richardghirst/PiBits/tree/master/ServoBlaster>.

You can clone the repository to your Raspberry Pi with the following command:

```
git clone git://github.com/richardghirst/PiBits.git
```

After obtaining the repository follow the instructions in ServoBlaster's README.txt to compile and install the code. Note that there are 2 implementations of ServoBlaster: a user space daemon, and a kernel space mod. I suggest using the user space daemon; I have not tried the kernel mod.

Develop a shell script or C program to drive a servo on GPIO 17 using ServoBlaster. You can produce any movement desired including the same movement produced using WiringPi in class. Submit your program to Moodle prior to midnight on Oct 10 for full credit.