# Overview

Two different python programs were developed. One is used for setting up the eyeball mechanism. The second program nicely demonstrates the eyeball performance.

To run either one of them, you must login to the PI and type '*startx*' which starts a windows GUI. From this GUI, you start a terminal screen. It will put you into the correct directory. If you do an '*ls*' command, you will see the two programs listed. To run either program, you need to type *'python filename.py'* where filename is either of the two program names. Both of these launch a new window. The remaining details for using each window are detailed below.

# Details

## EyeMouser.py

I developed this to demonstrate the eyeball. You will see a blank screen. This screen represents the eyeball direction. The screen center represents the eyeball looking straight out. The lower right corner represents an eyeball looking down in that direction..

**CLICKING**: Mouse clicks on a spot will cause the eyeball to look in that direction. Try several clicks and watch the eyeball point in the corresponding direction.

**STREAMING**: There is another mode of operation called streaming. In this mode, the cursor location is continuously linked to the eyeball servos. Any and every cursor motion causes the eyeballs to track along. Streaming is the **best** way to quickly demonstrate the eyeballs' capabilities.

To turn streaming ON, hit any key. Move the cursor around. Watch the eyeball try to keep up. It does a pretty good job! Write delays of 5 (5msec) were used and in this situation, that value worked extremely well. To turn streaming off, hit any key.

## EyeTestRig.py

I developed this to setup, adjust and test the eyeball. It has various sliders and a few buttons.

The test software allows you to move each servo with a slider. This was useful for discovering what each servo's min and max servo settings were. In the intial setup phase, every slider was set to go from 100msec to 250 msec. These values were dangerous as they would cause parts to crash into other parts. Next, each slider was manipulated to discover what were safe min and max servo values. These min/max values kept the parts motion within its allowable mechanical limits. The slider software was then modified to stay within these safe min/max values.

NOTE: You, the user, must exercise caution NOT TO FORCE the upper eyelid into the lower eyelid.

An additional caution is in order. Each buttons behavior locks up the system until it is complete. Thus, you cannot exit the GUI nor can you change the Write Delay until the last clicked button has finished its job. For example, if you start a large number of test cycles, you will have to wait until they are done so set the Write Delay speed correctly before you begin. Experienced users can crash the program and start over if they wish.

### SLIDERS

There are five sliders.

Four sliders are used to control the servos for the **pan**, **tilt**, **upper** **eyelid**, and **lower** **eyelid**. The slider numeric values are the PWM times sent to the servo motor controller.

Move the eyelid servos in small amounts until you understand which direction the eyelid will move in. This prevents mechanical collisions between the upper and lower eyelid. It is common to leave both eyelids WIDE open for most work.

A fifth slider, **'Write Delay'**, adjusts the amount of a pause after each servo command regardless of which servo written to.

This was implemented since it is believed that writing too many commands too fast prevents the servos from actually having enough time to complete each action. On the other hand, too large a delay after every servo command makes every move too slowly. The actual position should be adjusted to your immediate needs. A value of 5 (msec) is the default. This is too fast for some commands and the eyeball just wiggles around without moving completely to each position. It is rather amusing.

A value of 30 (msec) is a compromise. Each command seems to be completed and the speed is fairly snappy. Finally, certain commands, such as the home command, will override the slider setting to automatically change to a much slower servo write speed to ensure the servos do actually have enough time to reach their home position.

### BUTTONS

The buttons, HOME, CYCLE, BLINK, and DART initiate the following functions.

**HOME:** Sets all servos to the associated parts neutral position. Eyeball straight ahead and both eyelids closed.

**BLINK**: Just snaps the eyelids closed and then reopens them. At 5msec write speed this does not work. At higher delays, you can see the sequential performance. The upper eyelid closes, then the lower. The blink behavior is NOT very human-like.

**CYCLE**: This performs several mechanical tests. I ran these tests on the eyeball for one solid hour. The eyeball held up nicely despite the very heavy workout that it was getting. The tests were as explained below.

The entire test suite cycles several times, thus the button name, CYCLE. The current cycle number is displayed on the terminal screen. During each cycle, several individual tests are displayed on the terminal screen as they perform. The tests cover a range of behaviors. Setting the Write Delay slider to 100 (msec) gives each movement plenty of time to successfully complete. Setting the slider to 1000(1sec) gives you a chance to see the individual steps, should you want to have to investigate what is going wrong.

The tests are simple. *LIDS* test the eyelids' ability to open and close. The servos on the side look like they are pumping iron. The *ROLL* tests the combined pan/tilt motions needed to ROLL the eyes around the outer limits of the eyeball range of motion. It is interesting to watch. *CROSS* has the eyeball move in an X pattern across the central position. It is dull. The PAN and TILT tests just do a lot of their jobs. I forget if there were any others.

**DART**: This simulates the darting behavior of the human eye to dart about at all times. The eyeball repeatedly points in random directions with a random length pause between movements. It works rather well. The upper eyelid is coordinated to ride along always positioned on the upper pupil's edge just like in a real human. The lower eyelid is set to do nothing but stay out of the way by remaining fully open at its lowest position. Lower eyelids are pretty dull. They do hardly anything.

**ADJUSTMENTS**:

During the development of the hardware and software, certain software constants were routinely changed. The below behaviors are adjustable through software modification. You may wish to modify them if you know what you are doing. You will have to read the code to find the associated constants to change.

* NUMBER OF MAJOR TEST CYCLES
* NUMBER OF TIMES TO REPEAT EACH INDIVIDUAL TEST
* SERVO MIN/MAX/HOME POSITIONS
* AMOUNT OF DARTING PER BUTTON CLICK