Power Glove Controller

We used the Nintendo Power Glove and Arduino Nano microcontroller in order to communicate with our Inmoov robotic hand. The following materials were used:

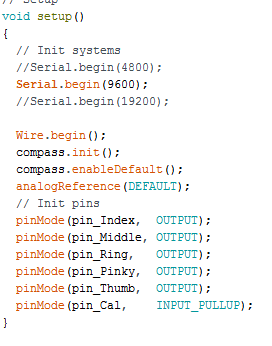
* Nintendo Power Glove
* Arduino Nano and Mega
* LSM303 IMU Compass/Gyroscope sensor
* 2 HC05 Bluetooth modules
* Pololu 18 pin Servo driver
* Extra flex sensor for Power Glove pinky

I cut a hole into the bottom of the Power Glove pinky and fed the flex sensor through the finger. I combined the pinky flex sensors wire with the other wires in the glove. Now, we can get readings from all 5 fingers.

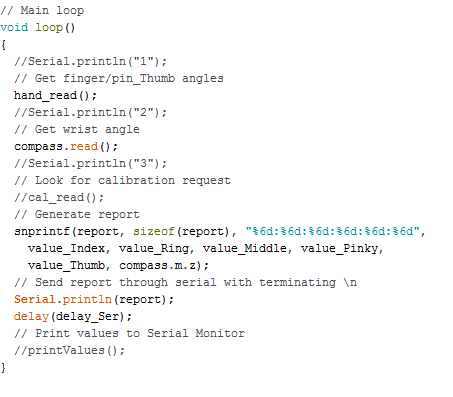
Using the Arduino IDE we came up with the following code. First the Power Glove which we called “Master”.



We had 4 lines from 4 flex sensors (index, middle, ring, thumb) that were connected to a common wire that is then fed to A1. The pinky had its own wire fed to the Nano which was connected to A1. We used a Multiplexor in software to take readings from the common wire from the 4 fingers.



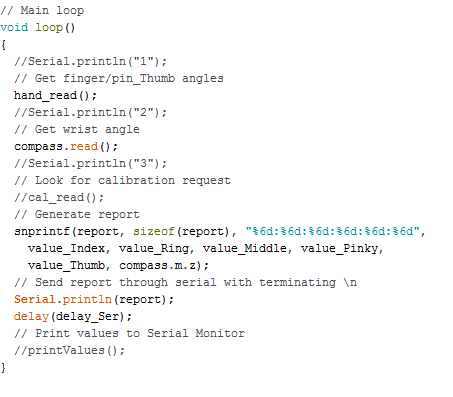
We use a baud rate of 9600 and initialized our output pins and compass.



Our main loop first branches to the hand\_read() function.



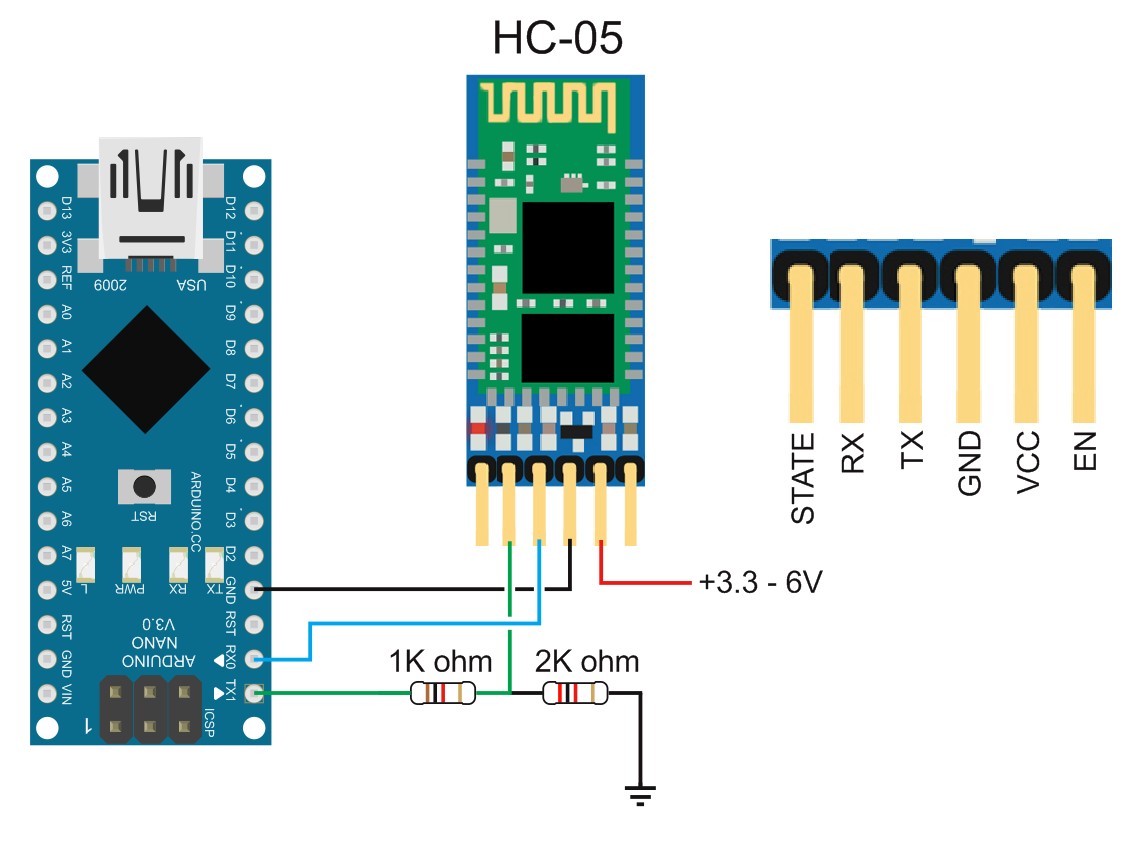
A ***for*** *l*oop counts through all 5 of the fingers. The fingers pin is put HIGH and the value there is put into their associated value\_ (value\_Middle, value\_Thumb, etc.). The pinky is its own wire connected to the Nano so it has its analog input.

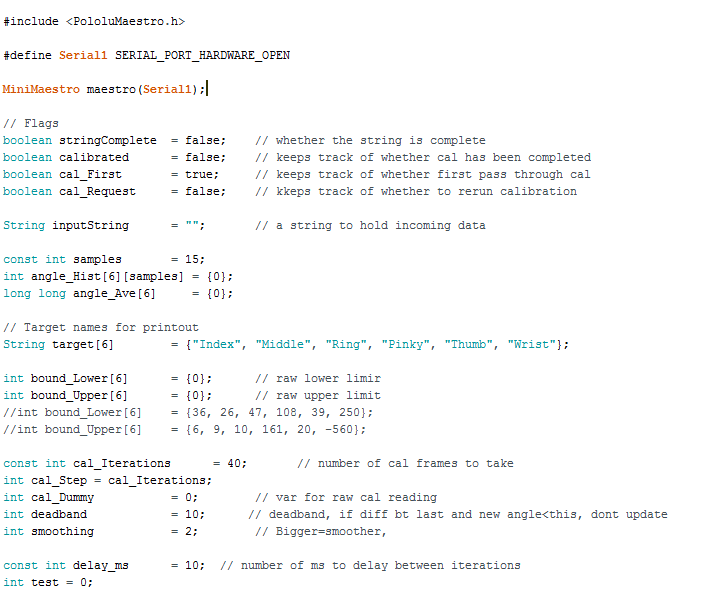


Now, we move to the compass.read(). which gets the value of our roll from the IMU. We only wanted the roll value because we will only be turning the wrist.

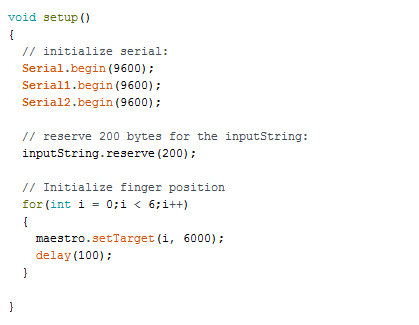
The snprintf function will place 6 values in *report.* The are in order: value\_Index (0), value\_Ring (1), value\_Middle (2), value\_Pinky (3), value\_Thumb (4), and compass.m.z (5).

These will be parsed in the “slave” code and are separated by a ‘:’. Finally, we send this through our HC 05 Bluetooth module which is connected to the Nanos serial port. The Serial.println(report) sends out the data to the other Bluetooth module associated with the Mega.

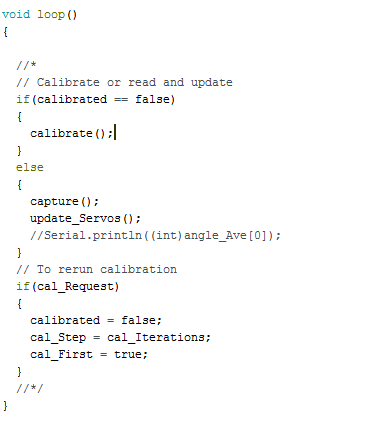




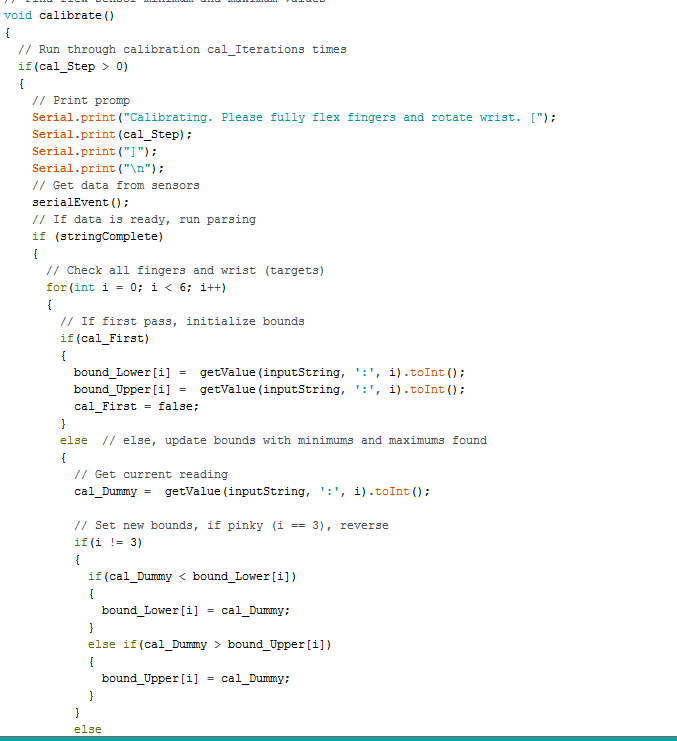
The “slave” code is using an Arduino Mega with HC05 Bluetooth and a Polulu Maestro. These integers will be explained further in the code.



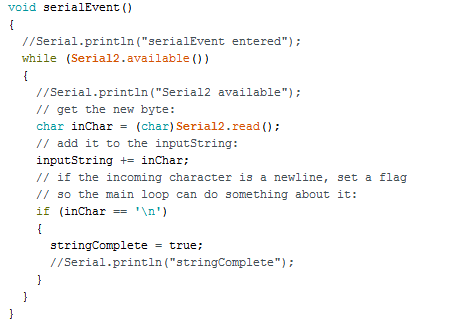
The setup has 3 serial ports initialized one for incoming Bluetooth data, Polulu Maestro data, and for printing to the screen. Then we initialize all 5 finger positions.



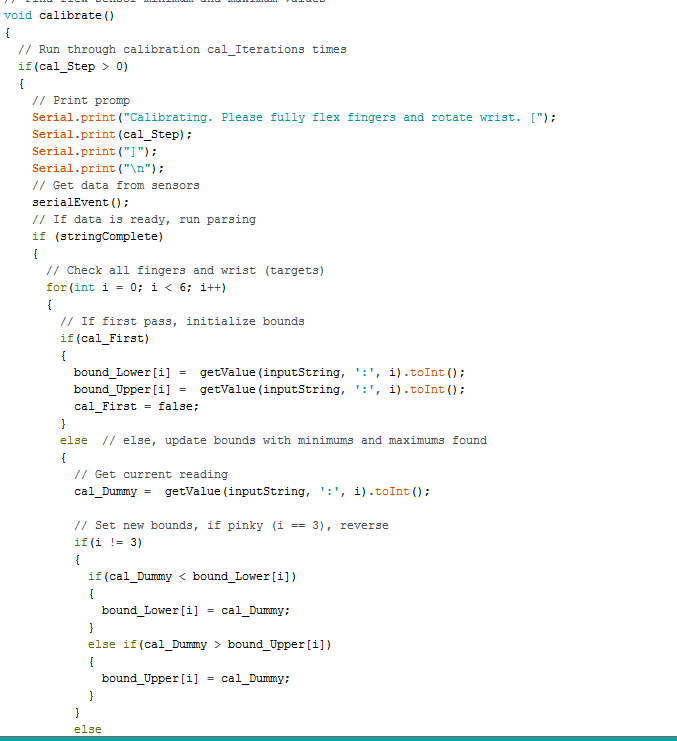
Our loop first goes into calibrate(). Due to the flex sensors having different values depending on the users fingers size, room temperature, etc., we had to use a calibration to better read the flex sensor values.



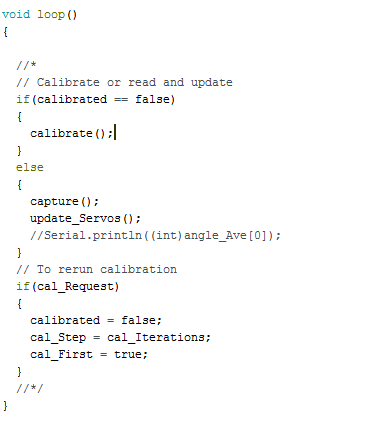
The first part of this will print text to the screen using serial port 0/ USB port. Then we branch to serialEvent().



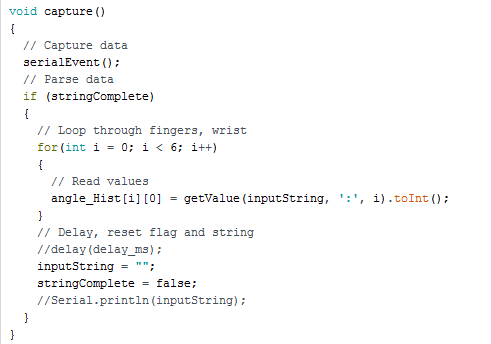
serialEvent() will read from serial port 2. If data is available we get the characters and put them into inChar. We exit after setting stringComplete to true.



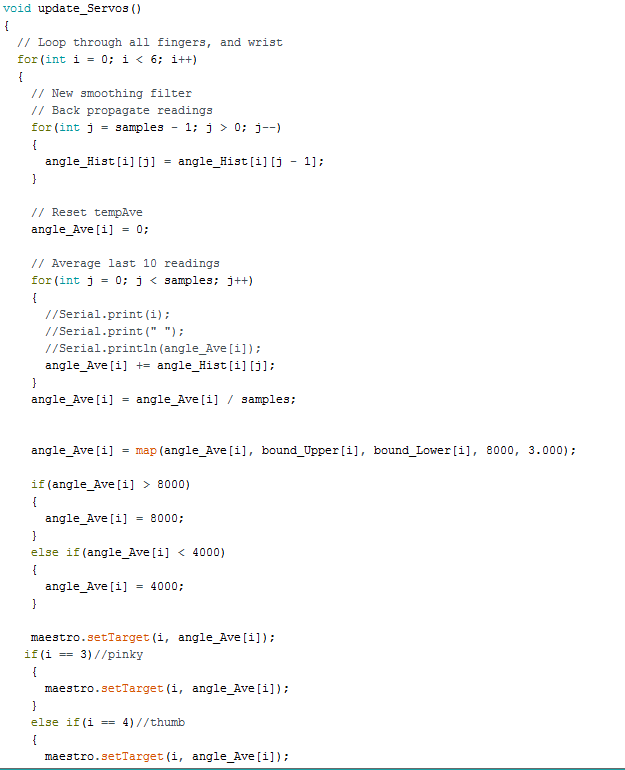
Back to calibrate(), we proceed to the first ***if*** statement where stringComplete will be true. The ***for*** loop will go through 5 fingers and the wrist. The upper and lower bounds are determined by first reading from the flex sensor values from the Power Glove. They are compared with a cal\_Dummy which is just a another reading that is taken. This is done 40 times or whatever calIterations is set to.



Back to the loop, we move on to capture(). which takes the incoming parsed data and

puts in into angle\_Hist which is just a 2D array. One with data and the other is set to 0. 

After that we go to update\_Servos().



We used a filter to get a smoother response to the servos for the arm. The upper and lower bound values are mapped to angle\_Ave. The angle\_Ave is mapped to the associated servo on the arm.