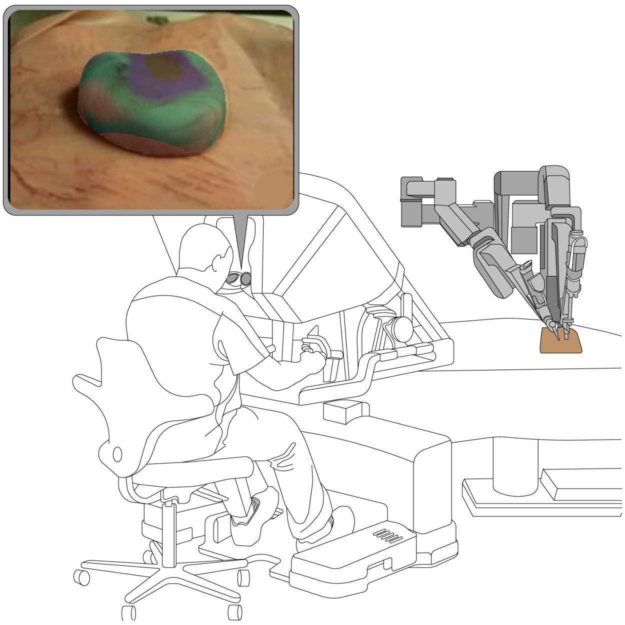
CARNEGIE MELLON UNIVERSITY

MRSD Project 1

February 13, 2019



Individual Lab Report

Augmented Reality for Minimally Invasive Surgery: The Chopsticks Robotic Surgical System

---

Author:
Anjali Pemmaraju

Team A Remaining Members:
Arti Anantharaman
Alex Wu
Cora Zhang
Chang Shi

Contents

1 Individual Progress

1.1 Sensors & Motors Lab

My responsibility for this lab was to create the GUI and help integrate each person's sensor and motor subsystem code into the final overall code. For the GUI, this meant that I designed the entire interface which allows a user to easily control the motors and see sensor states. For the integration, this meant that I helped design the command handling on the Arduino to allow us to change the sensor and motor configuration from the serial monitor. I also helped debug existing code that others had written.

1.1.1 GUI

I decided to implement the GUI in QT. This software is designed to help users create GUI interfaces and is written in C++. Since I have not written a GUI before, I picked this framework since it uses a programming language that I know and included a drag-and-drop GUI creation interface which I thought would be easy to use.

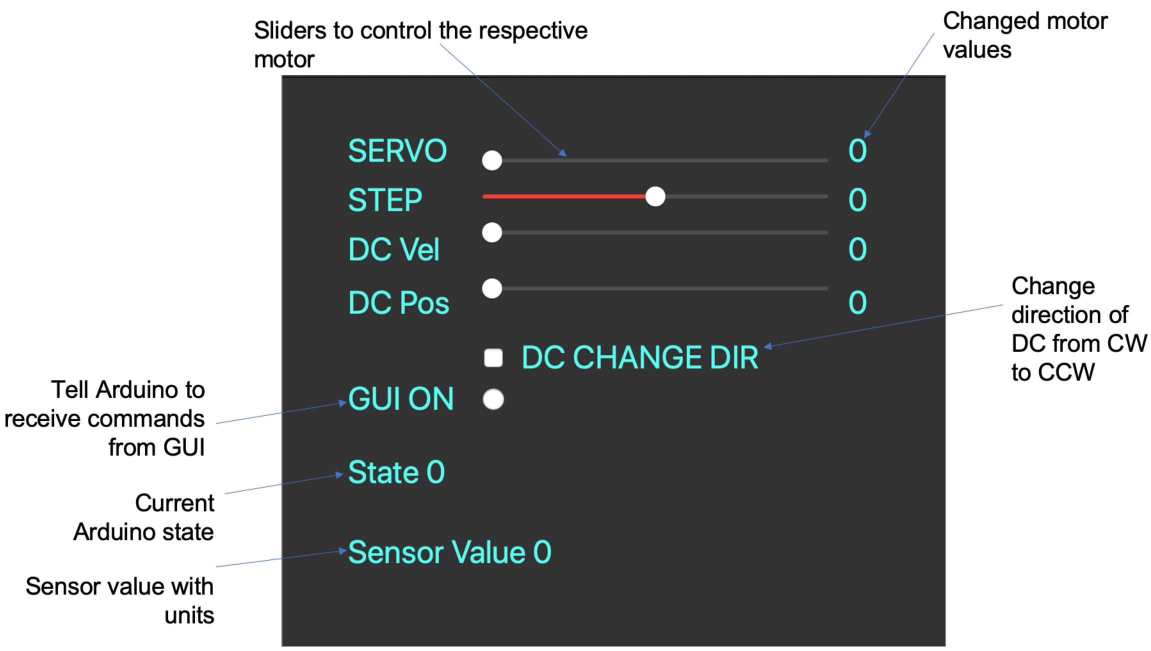


Figure 1: Final GUI Interface

The final GUI that I created is shown in 1. When any of the components of the GUI are interacted with, functions in the main function are called. These parse the value of the component and accordingly send a command through the serial port to the Arduino. In order to use the GUI to interface with the Arduino, I added a radio button that had to be clicked for the GUI to send commands to the Arduino. This ensured that both the GUI and the sensor values were not trying to change the state of the motors concurrently. The most intuitive interface for changing the motor states to me was a slider. There are 3 sliders which control each of the motors and displays the value of the motor at that time. The servo motor slider's range is from 0 to 180 degrees. The stepper motor slider's range is from -360 to 360. The DC motor slider's range is from 0 to 55 with a checkbox to reverse the direction of the motor. There is also a "State" label that shows the state of the Arduino while the GUI is not controlling it. Each of the states corresponds to a different sensor and motor configuration. There are 4 states total, representing each of the 4 sensors. When the Arduino is reading data from the sensors to control the motors, the sensor state also displays in a graphical form. This better communicates the changing state of the sensor. To see this code, refer to the files on canvas.

1.1.2 Arduino Code

I helped write the command handling code within the Arduino. This was part of my responsibility because I had to send commands from the GUI to the Arduino to let the user control the Arduino easily. Since I was the one who knew the way the GUI interface worked with the Arduino, I contributed this knowledge and influenced how the Arduino control was written to make the command processing and handling as easy as possible on both ends. We decided that the easiest command format was a letter followed by a number. This format was able to communicate all the functionality we needed. For example, to switch to sensor mode from the GUI, the command "S1" was sent. To move the step motor to a certain position, we mapped the negative range, from -360 to -1, to the values, 361 to 720 and mapped the positive range to their values. This meant that we never needed to process negative numbers. The conversion back to the actual range of -360 to 360 rather than from 0 to 720 was done by the Arduino after receiving the command. In a similar manner, we mapped out the user commands to parseable commands for the Arduino. Additionally, in order to read the sensor output from the Arduino over the serial port and into the GUI a similar message format was used. For example, when the ultrasound sensor was being used, the Arduino would report it's value to the serial port as "U" followed by the value of the sensor reading. The GUI would then read in this information from the serial port and accordingly change the sensor value in text and on the graph. The final Arduino with integrated sensors and motors is shown in Figure 2. This picture was taken by Arti.

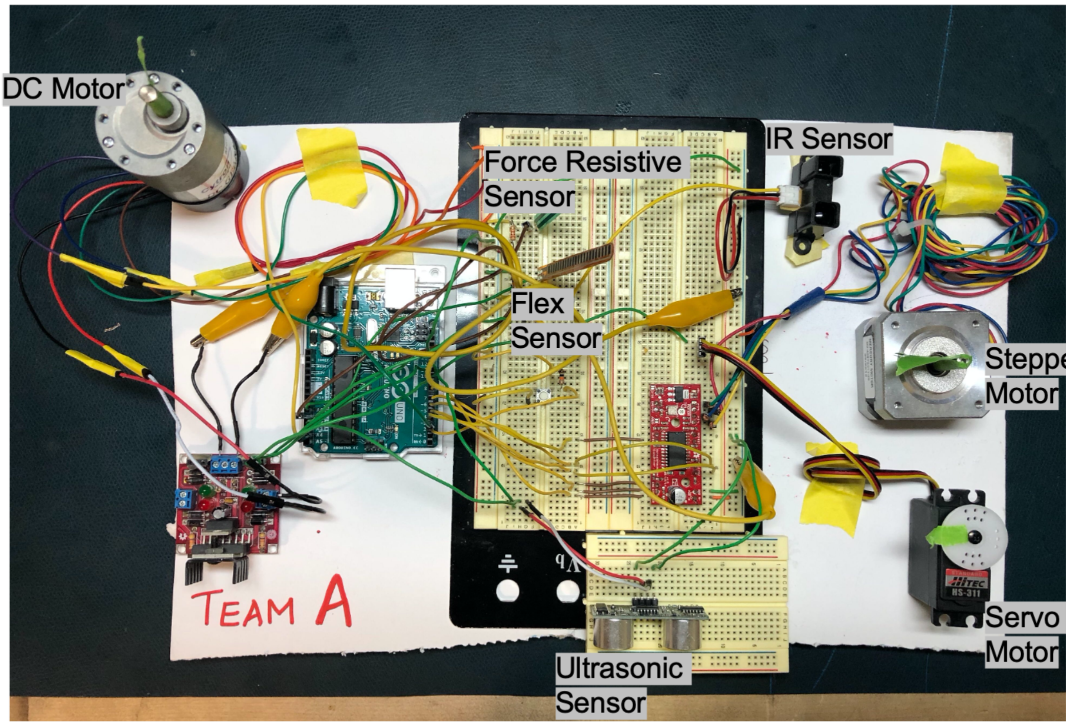


Figure 2: Final Integrated Circuit

1.2 MRSD Project

Since January, my main responsibilities have been to manage our project and to start looking at how to use our sensors with the DVRK setup. As the team project manager, I am in charge of ensuring that we stay on schedule. Because this past month was the first month that we were working on our project, I worked hard to ensure that everyone started off on a smooth foot. This included scheduling meetings and trying to get all the resources we needed from the members of the Biorobotics lab.

On a more technical level, I helped to begin integrating our laser sensor, the Blaser, with the DVRK. We initially thought that we would have a Blaser for our use only at the beginning of January but the lab later informed us that they did not want us to use the Blaser because they thought we would break it. As a result, we spent a significant portion of time trying to use the Blaser in a simulation software called VREP Coppelia Sim. This simulation software ended up being very difficult to use because the lab had not set up the Blaser and DVRK robots well. After facing significant difficulty in using the Blaser in simulation and seeing that it was not calibrated and did not give good laser returns, we decided that it would be best to start writing code for the real hardware instead and just use some dummy values while we were waiting for the real Blaser. Me and Cora looked at writing a transform broadcaster and listener for the Blaser using ROS. We used a dummy value for the Blaser transform from the DVRK end effector for now and are planning on changing this once we are able to calibrate the Blaser and have designed the holder that will attach the Blaser to the DVRK.

I also initiated talks with members of the lab to see how feasible our current technical thought process was. At the end of last semester we thought that a deep learning approach would work for segmenting the point clouds of the liver and the surrounding organs. However, after a few conversations we were informed that trying to do segmentation on partial point clouds for this use case may not be possible because there are not many medical-focused datasets that we could use to train this sort of network. Members of the Biorobotics lab are currently working on this problem but have not yet achieved good performance. As a result, we decided that a better idea would be to perform the segmentation by using the stereo camera data and then creating some sort of mask to segment out the pixels in the point cloud space. I have been looking into some more classical computer vision methods for segmentation that may be more suited towards our application.

2 Challenges

2.1 Sensors & Motors

The most significant challenges we faced were with the integration of the Sensors & Motors lab. The integration was difficult because people seemed to miss the part of the assignment where the Arduino state is controllable. They assumed that I would be able to just control the Arduino from the GUI, but did not realize that I needed them to write command handling code for the Arduino. This meant that during integration, all of the decisions about command handling had to be made really quickly and this was stressful. I handled this problem by trying to create simple commands that would be easy to parse from the Arduino and from the GUI.

2.2 MRSD Project

Additionally, the VREP Coppelia SIM software gave me significant troubles because it would continuously crash and was not intuitive to debug. The laser scanner in simulation would also not produce points in a straight line since it was not calibrated. We were not able to fix these problems on our own and had to rely on members of the Biorobotics lab to show us how their simulation worked and how to tune the parameters within the simulation. By asking them for help, I was able to solve the problem and we are ready to work with the hardware. We've also recently learned that this sensor may not be appropriate for our use case becauser the laser emitted is red and our organs are red which would give poor readings. I, along with Cora, am looking into new sensors that may be suitable for this use case.

3 Teamwork

We tried to split the work on the Sensors & Motors lab evenly. Each person worked with one of the sensors to control a motor and I wrote the GUI. For the project, we have tried to divide the tasks based on expertise to ensure we stay on schedule. A more explicit breakdown is shown below.

- Arti Anantharaman - Arti worked on helping to monitor the 3D printing of the organ molds. She also worked on writing code to move the DVRK smoothly in simulation. For the Sensors & Motors lab she worked with the force sensor and servo motor.
- Cora Zhang - Cora worked on the VREP simulation of the Blaser as well as on writing the transform broadcaster and listener for the Blaser from the DVRK end effector. She also worked on creating the organ molds and casting the silicone organs. For the Sensors & Motors lab she worked on integrating the step motor with the ultrasonic sensor.
- Alex Wu - Alex worked on designing the organ molds and casting the silicone organs. He also worked on assembling the movement simulation platform and designing the blaser holder cad model. For the Sensors & Motors lab he helped integrate the entire circuit and worked with the flex sensor and servo motor.
- Chang Shi - Chang worked on setting up the DVRK software environment and writing code to move the DVRK smoothly in simulation and hardware. For the Sensors & Motors lab she worked with the IR sensor and DC motor, including controlling the DC motor position and velocity.

4 Plans

4.1 Sensors & Motors

Unfortunately, the work we did in the Sensors & Motors lab is not really pertinent to our project because we are not designing any sensor and motor interface. The DVRK already has it's own motors and the Blaser scanner and camera that we're using were not able to be used in this lab.

4.2 MRSD Project

The team plans to print the mold for the silicone organs and recast the organs using colored silicone. Additionally, we will try to get the DVRK to execute a smooth scanning motion. We also are going to 3D print the blaser holder. I plan to use the blaser holder that is designed to attach the blaser to the DVRK arm. Then I want to calculate the transform from the Blaser frame to the end effector and add this transform to the DVRK transform tree. Afterwards, I will try to get a preliminary 3D scan of a known object to see if the point cloud generation is working correctly. I also plan to look into more classical techniques for segmentation using stereo camera to see if any are suitable for our use case.