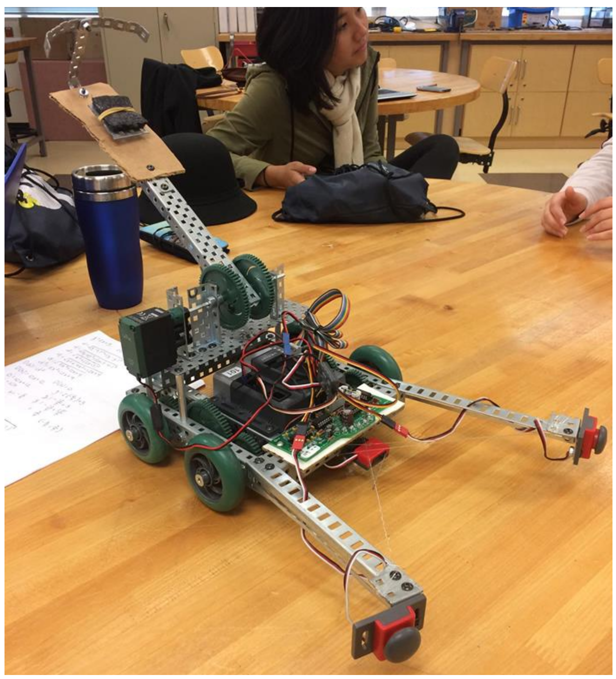
Robot Lab Report

Sean Aochi, Zoe Wakayama, Hung Le, Virginia Cater, and Uyen Nguyen

San Jose State University



Author Note

San José State University. Charles W. Davidson College of Engineering.

E10: Introduction to Engineering. Lab Project.

Section Number 8. Team Number 1. Professor Ken Youssefi. December 9th, 2016

Abstract

The objective of this project was to create a robot that could perform 4 tasks: locate the red beacon that emit a frequency of 10kHz; turn off that red beacon; locate a green beacon that emit a frequency of 1 kHz, and then bring the green beacon out of the arena. The robot was expected to weigh under 2kg, and perform the tasks as quickly as possible. As a group we originally split into teams to tackle the 3 design aspects of the early building process. One team soldered the IR board according to instructions given to us. Another team built the robot using the Vex kits made available to us, according to an agreed upon design from the whole group. Meanwhile, the third team gained familiarity with the software, Easy C, and started the code for the functionality. In the debugging process we weeded out defective parts and methodically coded our way to a fully functioning robot. In the final testing the robot performed all of the tasks within 27 seconds and weighed 2.425 kg.

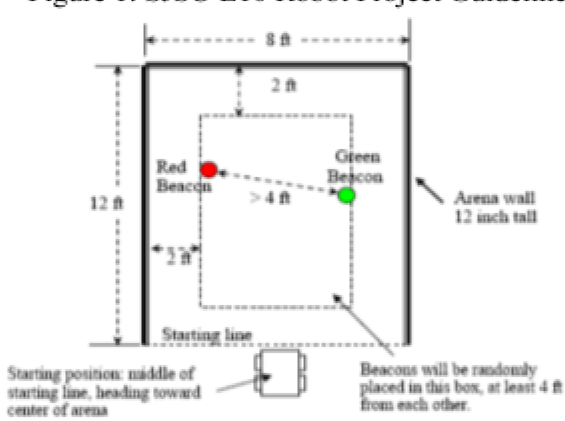
Table of Contents

Introduction

With the technology available today, people are able to design and make robots. Some of these robots are designed to help us with daily tasks and make our lives much easier and more efficient. An example of a robot would be the iRobot Roomba. Vacuuming the carpet floor once every week or month can be a hassle which takes a lot of time and effort. The iRobot automatically vacuums the carpet floor whenever it's dirty. This robot is programmed to efficiently maneuver throughout a person's house, vacuum the floor, and make its way back to it’s charging port.

In our lab, our main objective is to design and build a robot which has specific tasks to execute along with certain constraints. Other objectives involve understanding computer compramming(Easy C), build and program a circuit board, and working as a team. The robot has to be designed and programmed to enter a 12ft by 8ft arena (Figure 1). In the arena, there are two beacons, red and green. The robot has to be able to detect the red beacon, turn it off, locate the green beacon, and bring it back across the starting line. The beacons are randomly placed in the arena and have to be at least 4 ft from each other. Each beacon in the arena is a battery or electric powered infrared emitter which allows the robot to detect them. They can be turned off by pressing down on the button located at the top of the beacon. The robot’s starting position must be behind the starting line.

Figure 1: SJSU E10 Robot Project Guidelines



Design Description

<u>IRB assembly process:</u>

Figure 2

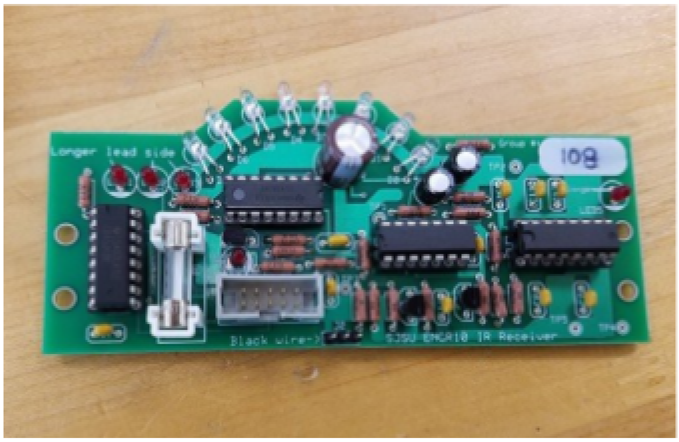
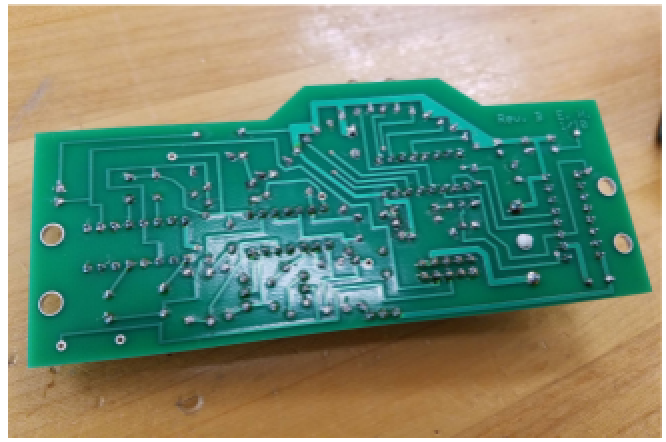


Figure 3



The circuit board required soldering to get the components onto the IRB board. There were a total of 20 resistors, 17 capacitors, 3 transistors, 8 infrared detectors, 5 LEDs, and and 4 integrated circuits (Figure 2,3).

Soldering the capacitors were crucial to the circuit board's operation. A capacitor stores electrical energy and then releases it later. This component sometimes is used with a resistor to produce a timer or it can also be used to smooth out current in a circuit. When the circuit board has power, the capacitor charges up. But when the power is off, the capacitor slowly discharges the electrical energy. Some of capacitors provided to solder onto the board are polarized, meaning they have a positive and negative lead. The longer lead of the capacitor is positive while the shorter lead is negative. On the circuit board, the square pad is for the positive lead while the circle pad is for the negative lead. It's important to place the positive and negative lead of the capacitor onto the circuit board according to the polarity that is shown on the circuit board. If placed in the wrongly onto the board and then soldered, the circuit board would most likely not function.

The second most important IRB component would be the infrared detectors. Without the infrared detectors, the robot would not be able to detect the beacons. There is a total of 8 infrared detectors and soldering them all onto the circuit board correctly is crucial to the success of the robot finding the beacons. The infrared detectors look similar to an LED. While soldering the infrared detectors, we had to make sure to not place them all the way into the circuit board. We had to leave some room where the infrared detectors can bend forward since they have a focal direction. Bending the detectors forward at a 90 degree angle allows the controller to detect where the infrared light is coming from. But before soldering the detectors into the circuit board, we had to make sure that we align the longer pin with the longer pin marking on the board.

The third most important IRB component during the soldering process would be the connectors. Soldering the connectors is probably the most important component to the IRB. There are two connectors that should be soldered onto the board, J1 and J2. J1 is the 10 pin ribbon cable connector and J2 is the 3 stick connector. The notched side(shorter pin side) of J1 should be placed into the circuit board while J2's placement doesn't really matter but recommended to also place on shorter pin side. The connectors allow the IRB and the VEX controller to connect via wires. Without the connectors, the IRB board would be useless and the robot would not be able to detect the beacons.

<u>Robot assembly process:</u>

We used 3 motors in total in designing the robot. 2 of which were used in the base of the robot to control the movement of the wheels of the robot (Figure 4) and the third one was located on the upper level of the robot and used to rotate the arm which was used to turn the beacon off and hold it in place (Figure 5).

Figure 4

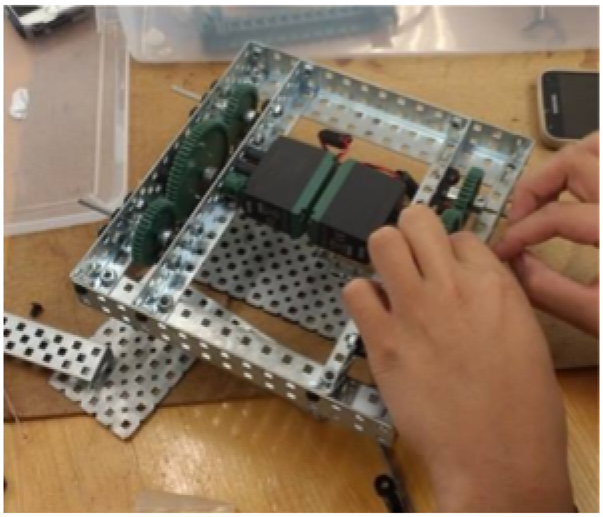
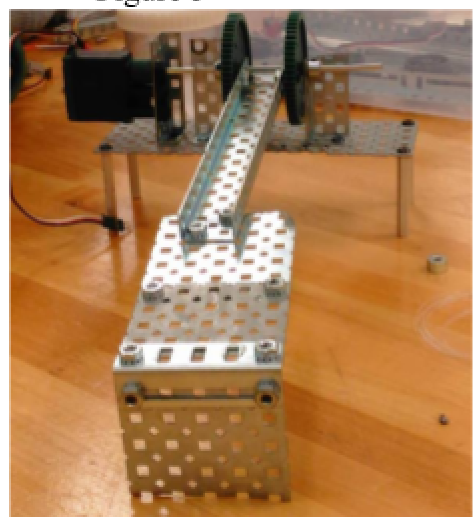
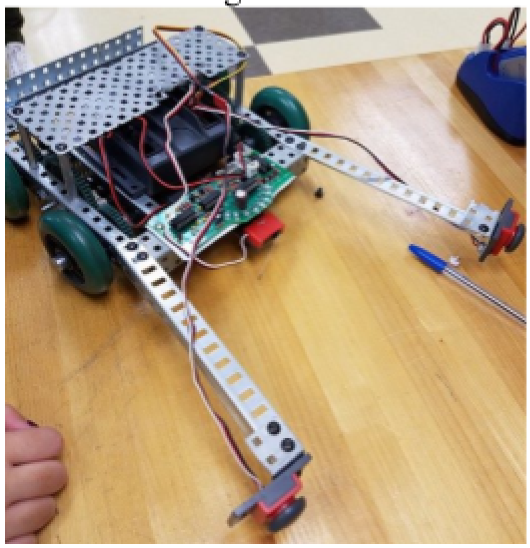


Figure 5



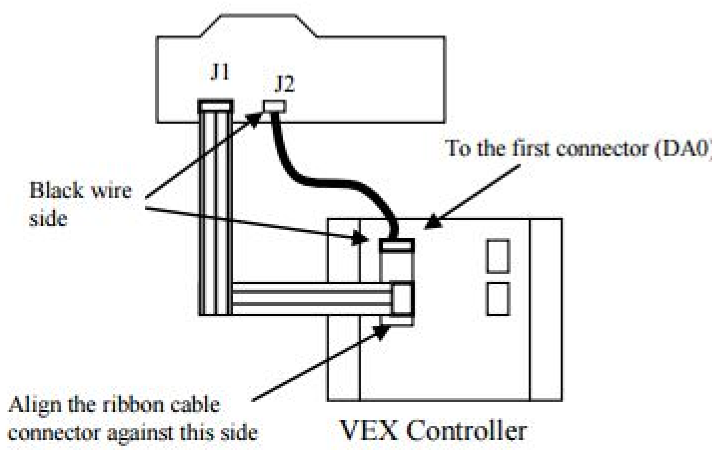
For the sensors we used 3 in total: 1 limit switch directly in the front of the robot and 2 bumper switches which were positioned so that both arms had a bumper switch (Figure 6). The purpose of the limit switch was for it to allow the robot to be able to sense when it made contact with one of the beacons. This was important because without it the robot would not be able to know when to move its arm. The purpose of the 2 bumper switches was to help the robot navigate out of the arena. These were important because once the robot found and captured the green beacon it then moved in a straight line until it hit a wall and one of the bumper switches were activated. When this happened it moved back and slightly changed directions. From this design the robot would eventually be able to find its way out of the arena.

Figure 6



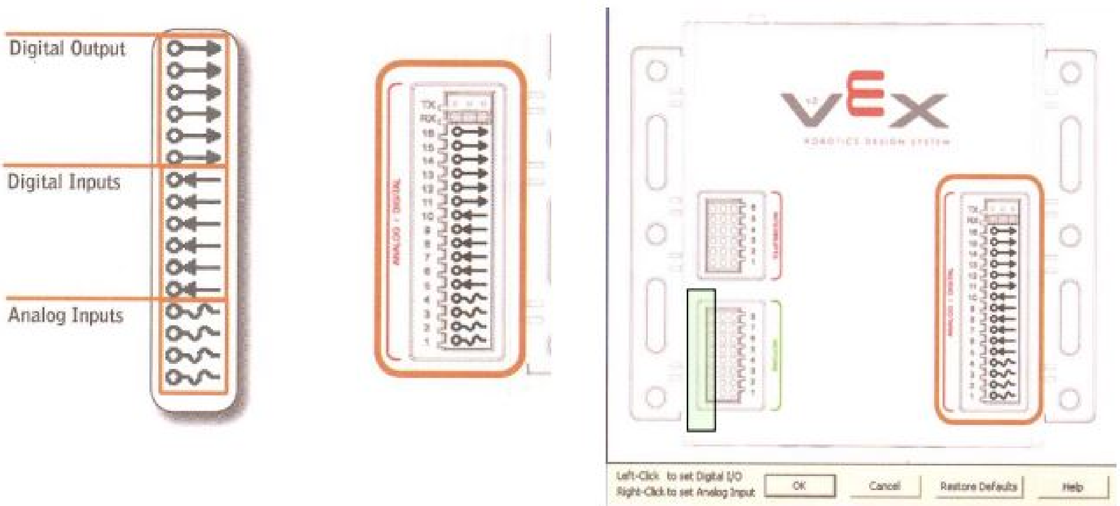
3.For the wiring of the robot we used a VEX controller, layout seen below(Figure 8), to connect all of the different components. To connect the IR board to the controller we used a ribbon wire and a connector wire, as seen below (Figure 7).

Figure 7: Lab report guideline

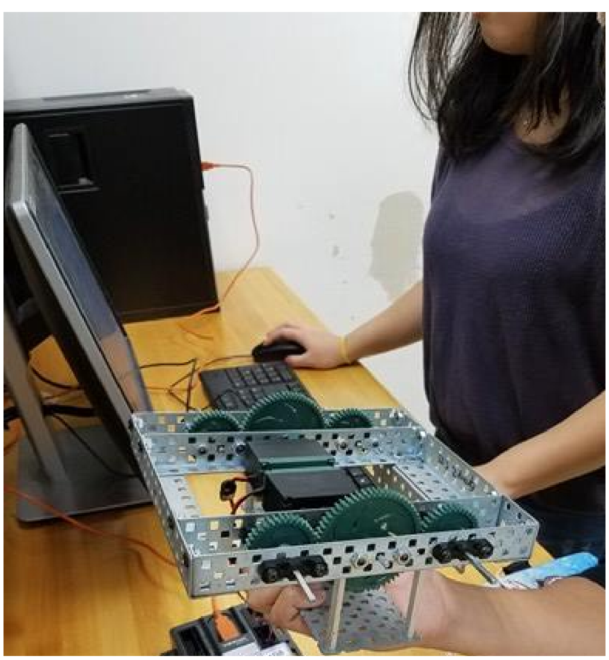


Along with the IR board the various motors and sensors were connected to the controller. The right motor was connected to the motor port 10, the left motor was connected to the motor port 1, and the motor which was used to control the arm was plugged into motor port 2. As for the sensors the left bumper switch was put into digital port 1, the right bumper switch was put into digital port2 and the limit switch was put in digital port 3

Figure 8: Project report Guideline



Software:



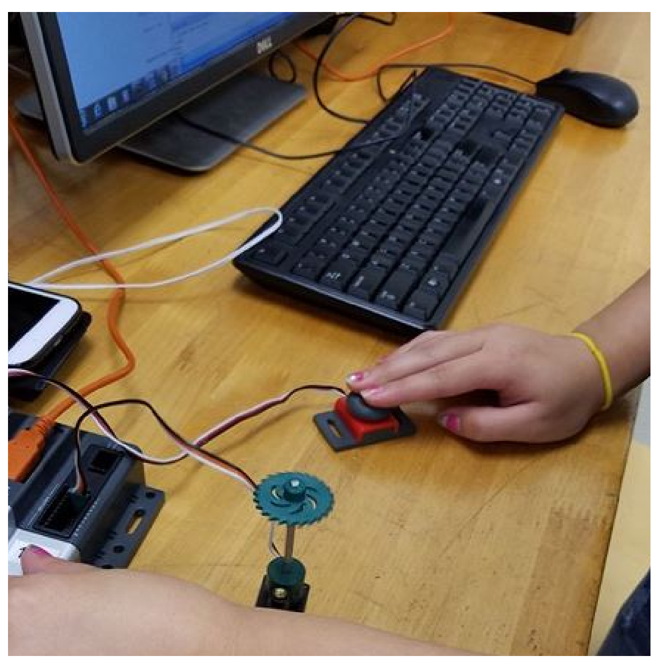


Figure 9 and 10: verifying that components are functioning well enough to be used.

The ultimate goal of the program is to make the robot do its required tasks. From the GOTO_Beacon_Cortex file, we had a basic pre-made code that programs the robot to go to the red beacon and then go to the green beacon. The code that was put in helped the robot turn off the red and green beacons, deliver the green one outside the arena, and navigate any obstacles the robot might encounter. We coded the program based on the tasks it needs to complete in order. We first programmed the robot to be notified when the red beacon is in reach. A limit switch is used to sense whether the beacon is close enough to be turned off. Once the beacon is near enough, the arm comes down and turns off the beacon via pressing a button. To verify that it is turned off after that swing, we programmed a while loop to check that the button is off. The loop also makes the robot turn off the beacon again if it did not succeed at first. Once off, the robot pulls its arm up and drives back and proceeds to find the green beacon. The code to notify when the green beacon is near and turn off the green beacon is the same as the code for the red beacon. However, because the final task is to bring the green beacon outside the arena, we do not make the arm come up. Instead, the arm is used to hold the green beacon while it leaves the arena. Bumper switches are then used to help navigate the arena until it finally exits. When any bumper switch is pressed, the robot turns right and will continue moving until it is pressed again.

The program described above are placed after the functions Read_PD, find_max, and move. We changed the ambient level to 70 so that it will be easier for the robot to find the red beacon. The motor ports in the move function are also altered so that it will match the ports on the robot. Programs are executed by state via an if statement so that it will be easier for us to program it all together. Only when the robot verifies that the red beacon is turned off and after the robot obtains the green beacon do we use a while loop. Throughout the lab time, the program was tweaked to improve the robot's performance.

Design Performance

Robot	Weight (kg)	Trial 1 (min)	Trial 2 (min)	Trial 3 (min)
Lucas	2.425kg	1:17	0:27	1:22

The robot weighed 2.456 kg total along with the battery.

This section addresses the outcome of the project. In both trials, the robot was able to do all its required tasks. The robot was able to find the red beacon, turn it off, turn off the green beacon, and finally carry the same beacon out of the arena. These tasks were done in varying lengths of time. In the first trial, the robot attempted to turn the red beacon off three times before running into a corner and proceeded to complete its remaining tasks. In the second trial, the robot attempted two times for the red beacon and immediately completed the remaining task. In the final trial, the robot immediately turned off the red beacon but was unable to reach the green beacon due to its arms unintentionally getting in its way before it finally completed the tasks. Overall, the fastest time was done in trial two by 27 seconds.

Conclusion

The robot was able to complete all its required tasks within 27 seconds. When building the robot, we divided into three groups and each group focused on the three main components of the project. We were able to solder the IRB board, build the robot from scratch, and program it to do its tasks together within the time period we were given. We had to deal with a few problems with each component of the robot. We accidentally shortened the IRB board and had to replace it in order for the robot to perform. For mechanical, we had to slightly redesign the arm and change the servo into a motor because it was not strong enough to turn off the button. Software also ran in some problems where the robot would perform literal actions and the program would have to be changed in order to make the robot do exactly as it needs to do instead. Despite this, we managed to solve these problems and learned how to improve our robot based on what we encountered. We learned that not only the IRB board needed a thicker material to be placed between it and the robot, but also the robot arm needs to be light and supported in order for the motor to lift and drop it; we also learned to carefully program the code so that what we wanted the robot to do will go as planned. In conclusion, we were able to build a robot that completed its tasks in a short amount of time.

Recommendation:

When we first began the project, we used lined paper between the IRB board and the robot. We did not realize that it caused the board to shorten and we had to replace the IRB board. We eventually found out that we needed to use a thicker material, like cardboard, instead of paper to avoid this from happening again. Our robot overall was heavy and exceeded two kilograms. Most of our robot, especially the arm, was built out of several pieces of metal. This was especially a problem for the arm because our servo initially was not strong enough to lift the arm up. Although cardboard slightly improved this problem, we might want to build a lighter robot by either using lighter materials or minimizing the amount of metal we had to use to build the robot. The bolt were also a problem because they often came out when we tested and the gears were usually loose. Since it might also affect the speed of the robot and its sustainability, we would reinforce the gears and bolts more so that it will not fall apart on a longer term. When it comes to the robot's program, we only programmed what we were required to do. Because of this, the robot had a lot of blind spots and we had to rely on the chance that those blind spots would not be revealed. The bumpers were also programmed to only activate after the first two tasks of turning off the beacons were complete. If we were to do the program again, we would redesign the arms of the robot to decrease the blind spots and program the bumpers to be used throughout the code.

References

Athanasiou, J. *Signals, Circuits and Computers Part B*. 2010. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/CKTs_and_Signals_Part_B_JA.ppt

Du, W. *Signals, Circuits and Computers Part A.* 2012. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/Signals-Circuit_Part_A_F12.ppt

Hsu, P. Robot Project Guidelines. 2008.

Youseffi, K. *Robot Project Overview*. 2016. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/11-Robot-Project-Overview-NC.pptx

Youseffi, K. and Ping Hsu. *Programming Concepts (Part A)*. 2016. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/12-Programming-Concepts-part-A-NC.pptx

Youseffi, K. and Ping Hsu. *Programming Concepts (Part B)*. 2016. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/13-Programming-Concepts-B-NC.ppt

Youseffi, K. and Ping Hsu. *Programming Concepts (Part C)*. 2016. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/14-Programming-Concepts-C-Go_To_Beacon_S13-NC.pptx

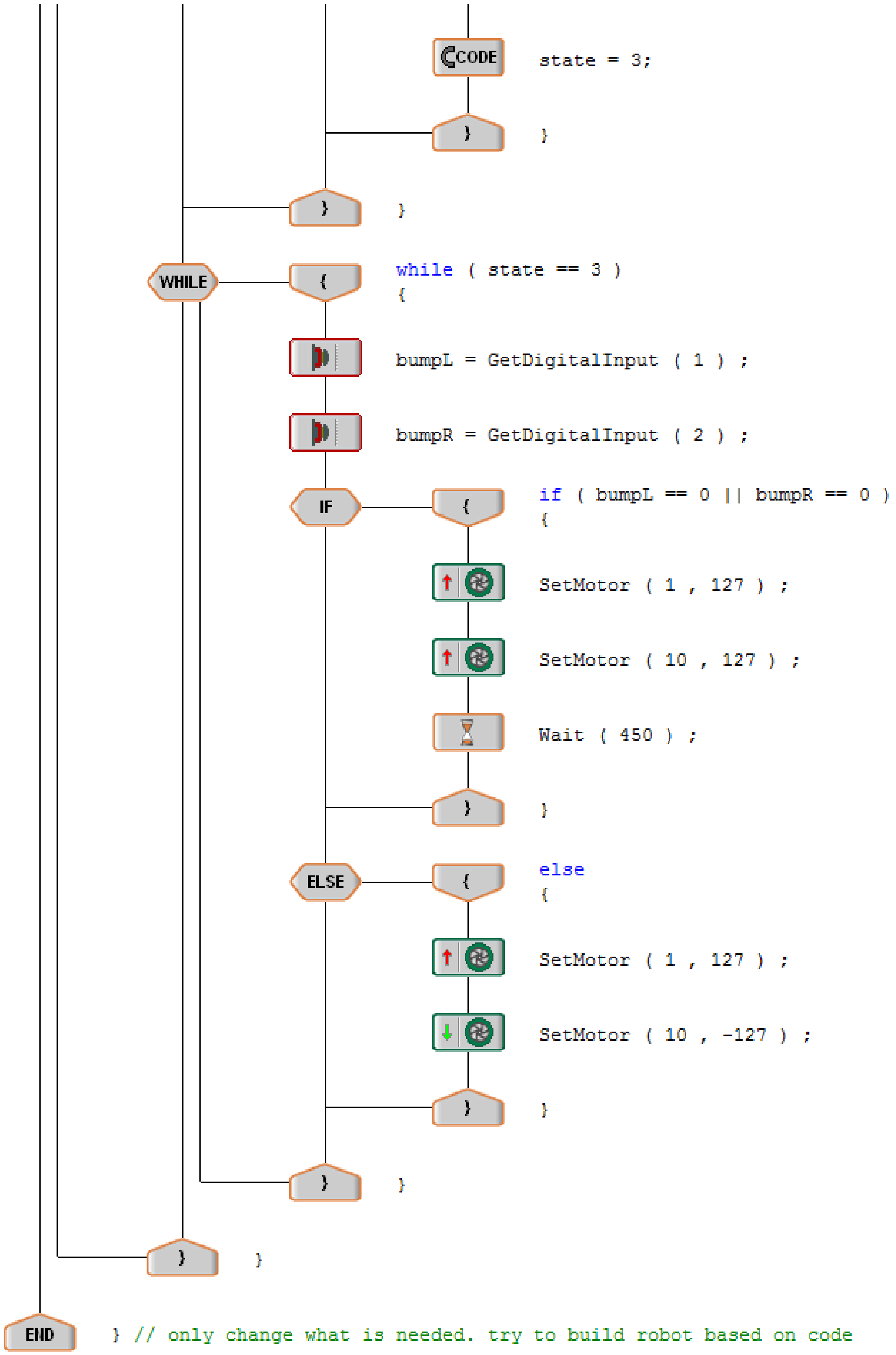
Youseffi, K, Winncy Du, and Ping Hsu. *Programming Concepts (Part D)*. 2016. Presentation. Retrieved from http://engineering.sjsu.edu/e10/wp-content/uploads/15-Sensors-Actuators-Signals-and-Computers-Part-D-F14-NC.pptx

Appendix A

```
void main ( void )
{
freq=0; // 0=1khz (red), 1=10kHz(green beacon)
ambient_level = 70; // esed in 'move'. change value if robot can't identify sensor
slow_level =4500; // used in 'move'
stop_level = 6000; // used in 'move'
expose_time = 3; // used in expose_and_read. do not alter
steer_sensitivity = 20; // used in 'move'
forward_speed = 35; // forward speed, used in 'move'
slow_speed = 25; // slow speed, used in 'move'
spin_speed = 25; // spin speed (for searching mode), used in 'move'
SetDigitalOutput ( 10 , freq ) ; // turn to 1kHz (red beacon)
int state = 1;
SetServo ( 2 , -60 ) ; // lift arm up
Flag_1 = 0;
while ( 1==1 )
{
Read_PD ( ) ;
find_max ( ) ;
```

```
move ( ) ; // robot movement
limit = GetDigitalInput ( 3 ) ;
if ( state ==1 )
{
    if ( limit == 0 ) // arm program. turn off red. see if red is turned off
    {
        SetMotor ( 1 , 0 ) ;
        SetMotor ( 10 , 0 ) ;
        Wait ( 1000 ) ;
        while ( PD_sum >1500 )
        {
            SetServo ( 2 , 60 ) ; // 60 degree drop?
            Wait ( 2000 ) ;
            SetServo ( 2 , -60 ) ; // lift arm up
            Wait ( 2000 ) ;
            Read_PD ( ) ;
        }
        SetServo ( 2 , -60 ) ;
        state = 2;
        SetMotor ( 1 , -127 ) ;
        SetMotor ( 10 , 127 ) ;
        Wait ( 1000 ) ;
```

```
freq = 1;
SetDigitalOutput ( 10 , freq ) ; // turn to 1kHz (red beacon)
ambient_level = 30; // esed in 'move'. change value if robot can't identify sensor
SetServo ( 2 , -60 ) ;
state = 2;
}
}
if ( state == 2 )
{
limit = GetDigitalInput ( 3 ) ;
SetServo ( 2 , -60 ) ;
if ( limit == 0 ) // arm program. turn off red. see if red is turned off
{
SetMotor ( 1 , 0 ) ;
SetMotor ( 10 , 0 ) ;
Wait ( 3000 ) ;
SetServo ( 2 , 60 ) ; // 60 degree drop?
Wait ( 2000 ) ;
SetMotor ( 1 , -127 ) ;
SetMotor ( 10 , 127 ) ;
Wait ( 1000 ) ;
```


CODE
state = 3;
}
}
WHILE
while (state == 3)
{
bumpL = GetDigitalInput (1) ;
bumpR = GetDigitalInput (2) ;
IF
if (bumpL == 0 || bumpR == 0)
{
SetMotor (1 , 127) ;
SetMotor (10 , 127) ;
Wait (450) ;
}
ELSE
else
{
SetMotor (1 , 127) ;
SetMotor (10 , -127) ;
}
}
}
END
} // only change what is needed. try to build robot based on code

Appendix B

Answer the following questions:

i. Who worked on each aspect of the project:

Hung and Virginia worked on the electrical component of the robot. Zoe, Sean, and Hung worked on the mechanical components and Virginia and Uyen worked on the software component.

ii. Who contributed to each section of the report:

Hung wrote the introduction and IRB assembly. Uyen wrote the software and recommendations. Virginia wrote the abstract and helped insert photos. Sean and Zoe collaborated on the Mechanical write up.

iii. Describe one challenge the team faced, how the team resolved it, and a lesson the team learned from that experience:

One challenge that we experienced as a team was what to do when our robot suddenly was unable to find the red beacon when it had been able to prior. We resolved this problem by troubleshooting every part of the robot and eventually concluded that it was a problem with the IR board which we ended up needing to replace. From this experience we assumed that the board no longer worked because it was shorted out and learned that when applying a board to a robot something thicker than a piece of paper should be used as a buffer between the two.

iv. Use the table below to rank (on a scale of 5 to 1) your team performance in the following six areas (5= we all did this all of the time to 1= did not do this at all)

Teamwork Skill	Performance Level
1) open and honest communication among members	5
2) each individual carried his/her own weight	5
3) collaboration in decision making	5
4) team set goals and milestones	5
5) people listened to each other	5
6) leadership was shared among the members	5