Constructing a Robot Using Logic Code to Program an IR Board

San José State University,

Charles W. Davidson College of Engineering,

E10 Introduction to Engineering,

Section 5

Group 1

by

Gissela Agurcia,

Emily Espinosa,

Bryan Hernandez Quezada,

Chunli Li,

Dulce Payan,

Table of Contents

Project Summary

The purpose of this project was to design a robot that could effectively perform all of the tasks required in a timely and efficient manner in order to assist people. To do so, one must build a VEX squarebot that weighs no more than 2 kilograms and utilize Easy C to program the VEX cortex to perform tasks. The robot will be equipped with a bumper switch and an ultrasonic sensor to communicate to the robot how to exit the arena once the green beacon is detected. The robot will also have a battery, VEX cortex, a soldered infrared board, and arm which will allow

the robot to perform its tasks. This robot will be tested on how efficiently it can perform the task of locating a red beacon, turning off the red beacon, locating a green beacon, and exiting the arena with it. This can be done by programming code that describes all tasks that must be performed, creating a robot to perform these tasks, utilizing an infrared board to differentiate the beacons, and a cortex to communicate with the robot.

Introduction

To create a robot whose goal is to find the red Beacon, then stop and use its arm to turn off the red signal, then go to the green beacon and carry or push the green lighthouse out of the arena, the project is divided into three parts, machine design, electronic design, and programming design. Each section is responsible for two students, the machine part is responsible for designing the robot theme section which includes wheels and robot arm, the electronic part is responsible for solder circuits boards which is the 'eye' of the robot, the goal is to be able to move the direction of the flag which make sure the robots have the ability to move directions by code. The programming part is responsible for writing code which requires that the code can run and instruct the robot to accomplish the goal.

Results and Analysis

A. Mechanic Design

The robot was designed so it can locate the red beacon. First, a bumper is located in the front of the robot's main body to be accessible to hit the beacon. Then an arm will drop onto the red beacon to turn it off. To do so one must fasten two rails at the base of the robot by the bumper switch so when the green beacon is found and detected by the ultrasonic sensor the robot can carry it and then exit the arena. There must also be an arm created that will be capable of turning off the red beacon's button when the red beacon is created. This can be done using foam attached to a plate that is attached to a rail. The rail will have a shaft going through it with two servo motors on each end to rotate the arm to turn off the button. The reason why servos should be attached is that servos are best at generating torque and are very accurate and have higher precision compared to other motors. The arm is designed to turn off the red beacon, therefore the arm should have a foam to allow the arm to push the button once the arm goes down to push the button. Along with the arm, there should also be a bumper placed on the robot. The bumper should be placed in a manner where it can be pushed when a beacon is present. To do this, the bumper should be placed between the rails where it can be pushed by the beacon to signal that the beacon is present. To distinguish the difference between the green and red beacon, an infrared board should be added which will communicate to the robot what color of the beacon is detected. Once this is done the robot will be told either to turn off the red beacon or exit the arena. To guide the robot out of the arena, the ultrasonic sensor will be added to guide the robot to exit the arena once the green beacon is discovered. A battery is also included in the design of the robot, to maintain balance there should be a battery mount added to counterbalance the weight of the arm as the battery is attached to the back of the robot so the robot does not tip over.

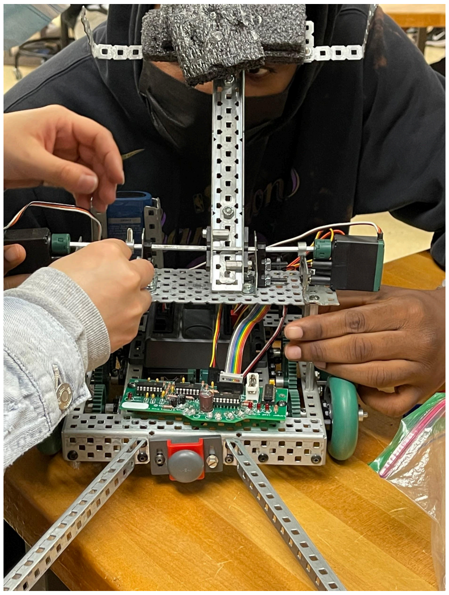


Figure 1: Completed Robot

B. Circuit Design

For Circuit Design, the purpose is to solder things on the circuit board. The Circuit board is the eye of a robot, with eight infrared detectors with almost 100 fields of view to ensure that the program can move to an infrared light source under the right circumstances. Materials are resistors, capacitors, transistors, infrared detectors, LED, and integrated circuits. Resistors use to two-terminal electrical component implements electrical resistance, capacitors are two-terminal electrical components that work with resistors, transistors use to semiconductors that amplify the electrical signals, infrared detectors detect radiation from the infrared region, LEDs is showing connection when robot working, integrated Circuits is a semiconductor wit a set of electronic circuits with small clips. Each applied carefully while being cautious with the amount of solder in hopes to prevent soldering the pipes on the board safety measures were implemented for example the use of the fan so that the harmful chemicals were not inhaled and eye protection too. After applying everything on the board, the ends on the back were trimmed off and the teachers along with her assistants checked the board. Once they verified the quality of the finished board it was tested, the first test didn't execute the desired results. There wasn't enough solder applied to three resistors, some with a delicate hand more solder was added to the resistor. but the rest still couldn't move the flag from one direction to the other direction. The professor and assistants decided that it was possible that the position of the resistor was wrong, but the laboratory did not have a machine that could test the position and energy of every resistor. Finally, A makeup circuit board replaced the failed one. As a memento, the failed one still appeared in the slide show below are two images of the front and back of the completed board (Figure 1 and 2)

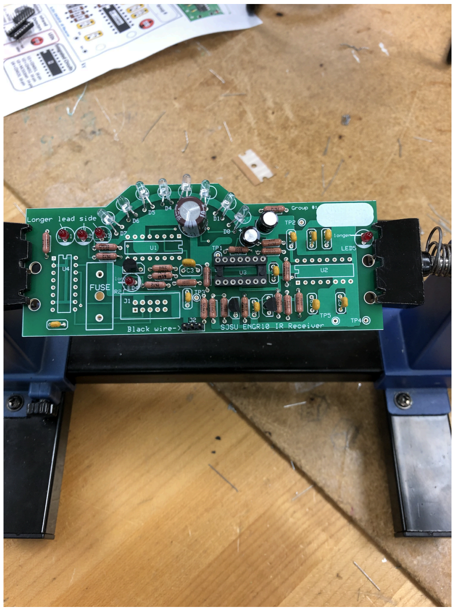


Figure 2: Front of IR board

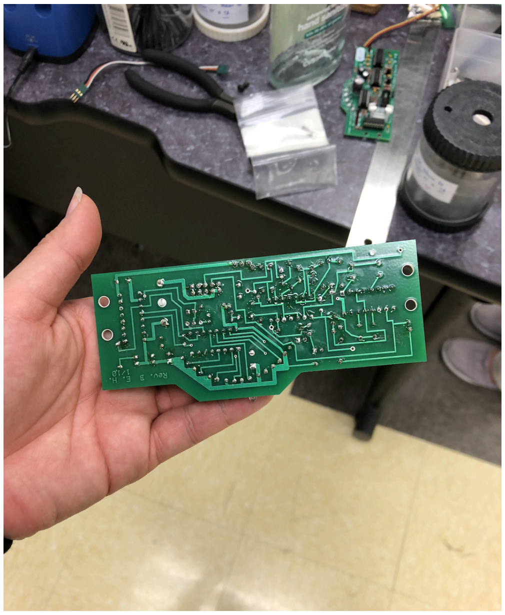


Figure 3: Back of IR board

C. Program Design

This section is going to go over the programming of the robot. As mentioned in project summer. The main goal of the robot is to go to the red beacon, turn off the red beacon, go to the green beacon, capture the green beacon, and leave the arena with the green beacon. The program system that is used is an Easy C Vex V6 Cortex program and lab computers from San Jose State University were used in order to use the programming software. The first step was getting familiar with the programming program used to program the robot. The next step is to make a flowchart to have a base idea of what procedures the robot will be doing.

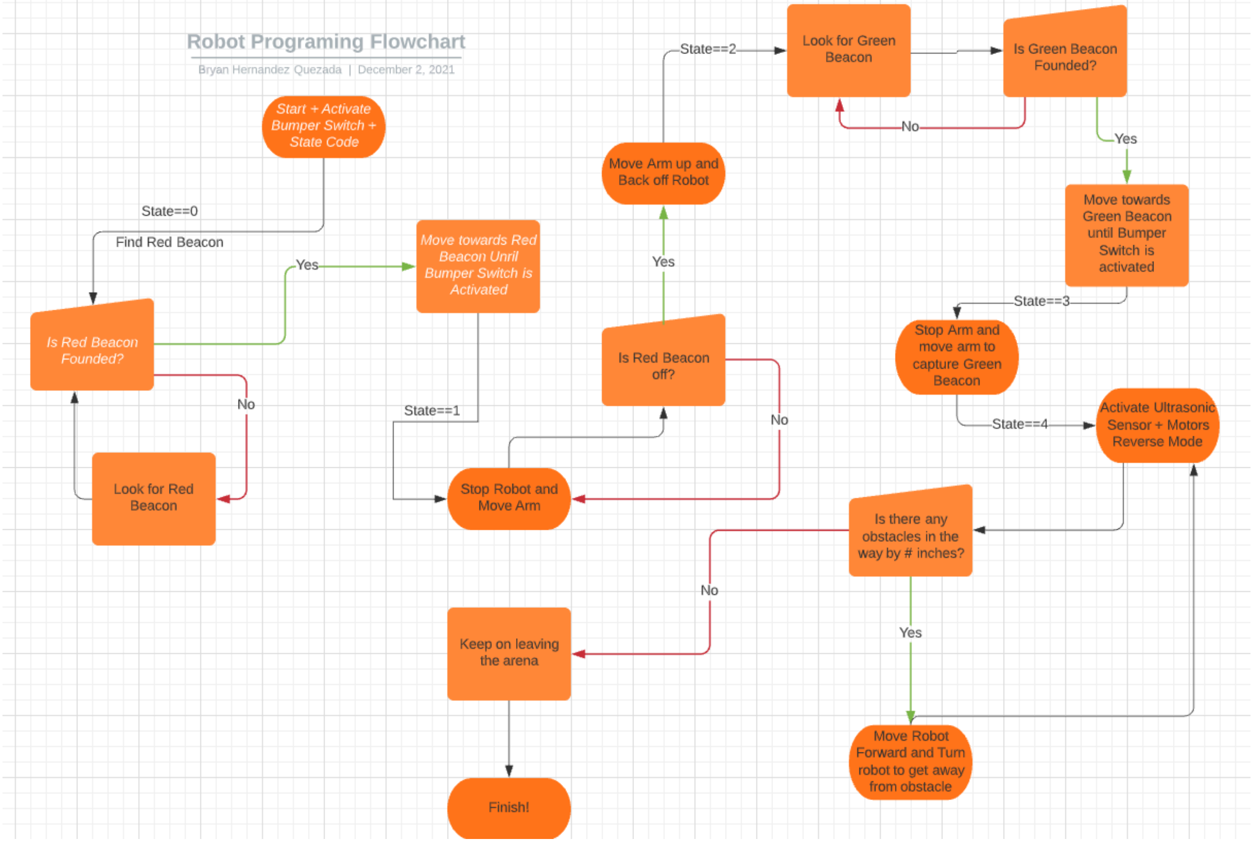
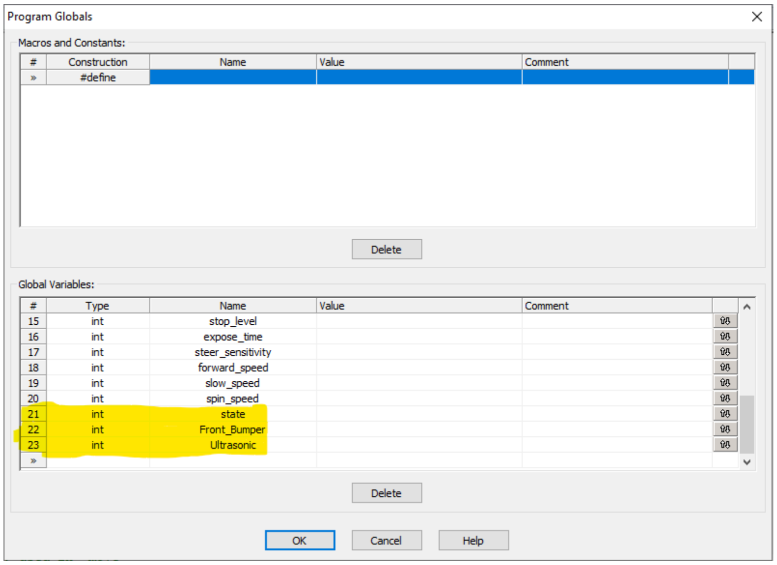

Robot Programing Flowchart
Bryan Hernandez Quezada | December 2, 2021
Start + Activate Bumper Switch + State Code
State==0
Find Red Beacon
Is Red Beacon Founded?
Yes
No
Look for Red Beacon
Move towards Red Beacon Unril Bumper Switch is Activated
State==1
Stop Robot and Move Arm
Is Red Beacon off?
Yes
No
Move Arm up and Back off Robot
State==2
Look for Green Beacon
Is Green Beacon Founded?
No
Yes
Move towards Green Beacon until Bumper Switch is activated
State==3
Stop Arm and move arm to capture Green Beacon
State==4
Activate Ultrasonic Sensor + Motors Reverse Mode
Is there any obstacles in the way by # inches?
No
Yes
Keep on leaving the arena
Move Robot Forward and Turn robot to get away from obstacle
Finish!

Figure 4: Code logic flowchart

Following the flowchart, the robot will first turn on. The robot will utilize user codes and global variables to learn about its surroundings and also what it has at the moment. The robot will also use three systems; a bumper switch, an IR board, and an ultrasonic sensor Then the robot will start at state==0. The state is another way of saying what stage the robot is in. In this case, state=0 is the state of finding the red beacon by using an IR board. The first question in state=0 is if the red beacon is founded. If the red beacon is not found, then the robot will be instructed to keep on finding the red beacon. If the robot finds the red beacon, then the robot will move towards the red beacon until a bumper switch is activated. Once the bumper switch is activated, the robot will go into state=1, which is the state of stopping the robot and using the arm in order to turn off the red beacon. The robot will then move the arm to the down position and then the up position. After moving the arm, the robot will calculate its PD Sum and compare it to the ambient level in order to determine if the red beacon is on or off. If the red beacon is on, then the robot will continue to move its arm up and down. If the red beacon is turned off, then the robot will move its arm up and use the motors on the wheels to back the robot up. The robot will then go into state=2 which is the state of finding the green beacon. The robot will use the IR board in order to detect the green beacon by changing the frequency of the user code. When comparing state=2 and state=0, although both states use very similar codes, the main difference is the frequency used in the user code in order to read either red beacon or green beacon. Once the robot goes to the green beacon and the bumper switch is activated, state=3 will be active which stops the robot and moves the arm to the down position in order to capture the green beacon. The last state of the robot is state=4, which is the state of leaving the arena with the green beacon. In state=4, the robot uses the ultrasonic sensor and moves the motors in reverse in order to leave the arena. The ultrasonic sensor is used to read any obstacles that could be in the way of the direction that the robot is heading. If the ultrasonic sensor detects that there is an obstacle in the way of the path that the robot is heading by 20 inches, then the robot will go forward and perform a motor vectoring in order to steer the robot away from the obstacle. Once the obstacle is away from the robot, the robot will resume backing up the robot until it leaves the arena. After the robot leaves the arena, the robot is finished with completing all of its tasks.

When creating the robots' programming, while loops and if statements were most used in order to create the state= of the robots' programming. Also in order to get the sensors working, global variables were used in order to tell the robot what sensors it has, and then user code was used in order to tell the robot in what state to use the sensors. In order to determine the beacon's proximity and whether the beacon is on or off, PD sum was used in the process which is measured by the IR board attached to the robot. PD sum was then compared to the ambient level of the surrounding area of the robot. If the PD sum is less than the ambient level of the robot, then the beacon is most likely turned off. If the PD sum is more than the ambient level of the robot's surroundings, then the beacon is most likely turned on. From a performance standpoint, the robot was able to complete all of its tasks and the behavior of the robot was expected.



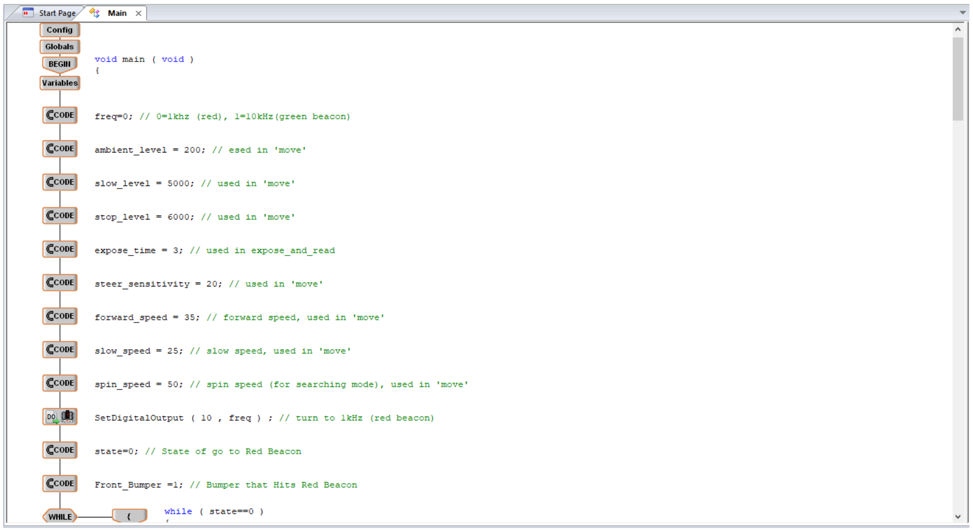
Macros and Constants:

#	Construction	Name	Value	Comment
»	#define			

Global Variables:

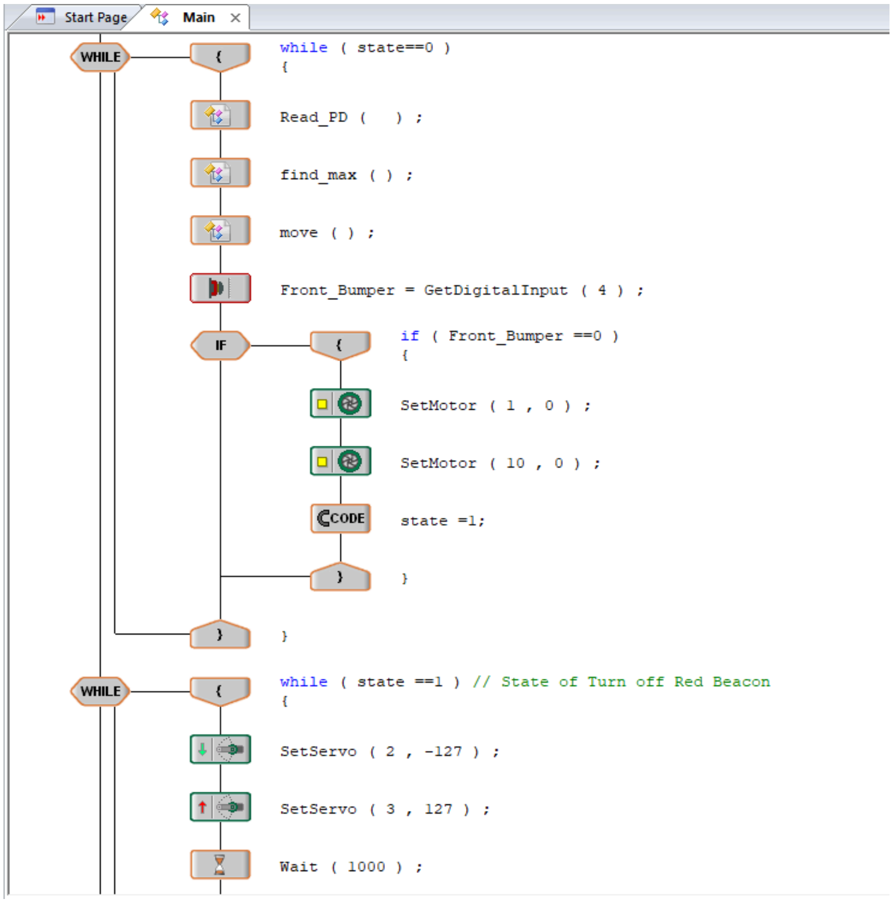
#	Type	Name	Value	Comment
15	int	stop_level		
16	int	expose_time		
17	int	steer_sensitivity		
18	int	forward_speed		
19	int	slow_speed		
20	int	spin_speed		
21	int	state		
22	int	Front_Bumper		
23	int	Ultrasonic		
»				

Figure 5: Global Variables of code program



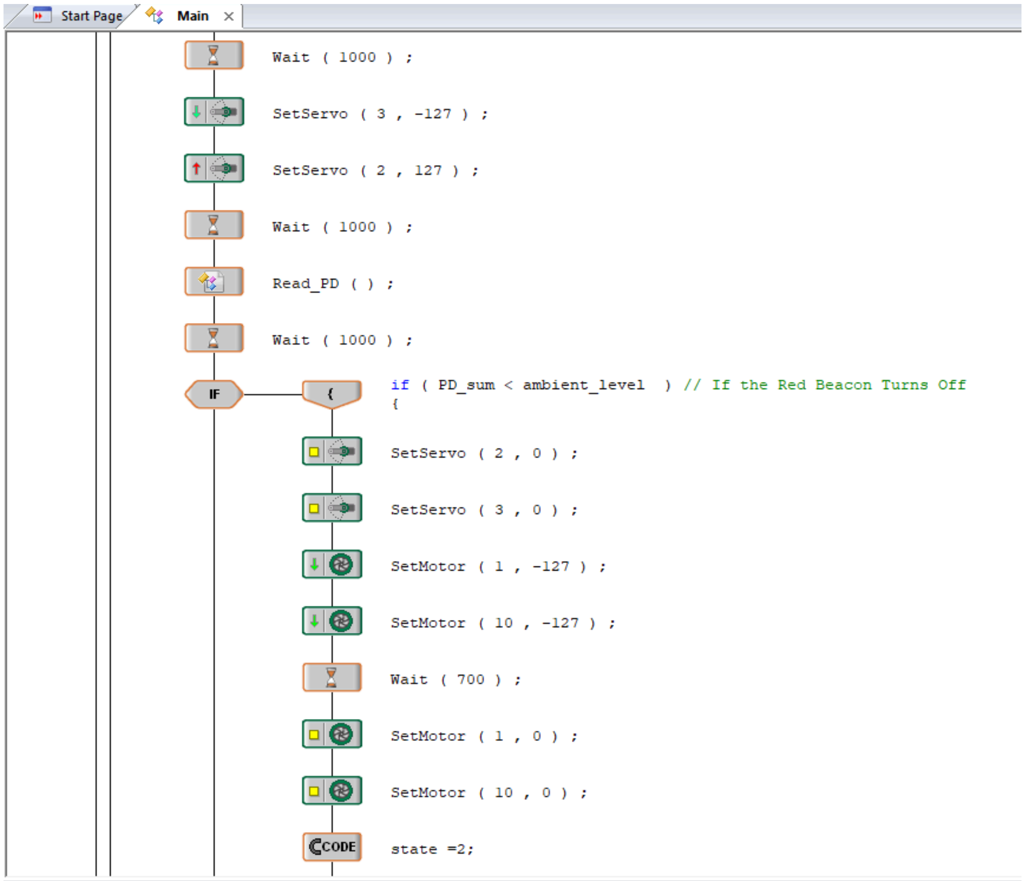
```
void main ( void )
{
freq=0; // 0=1khz (red), 1=10kHz(green beacon)
ambient_level = 200; // esed in 'move'
slow_level = 5000; // used in 'move'
stop_level = 6000; // used in 'move'
expose_time = 3; // used in expose_and_read
steer_sensitivity = 20; // used in 'move'
forward_speed = 35; // forward speed, used in 'move'
slow_speed = 25; // slow speed, used in 'move'
spin_speed = 50; // spin speed (for searching mode), used in 'move'
SetDigitalOutput ( 10 , freq ) ; // turn to 1kHz (red beacon)
state=0; // State of go to Red Beacon
Front_Bumper =1; // Bumper that Hits Red Beacon
while ( state==0 )
```

Figure 6: Programing user codes before state==0



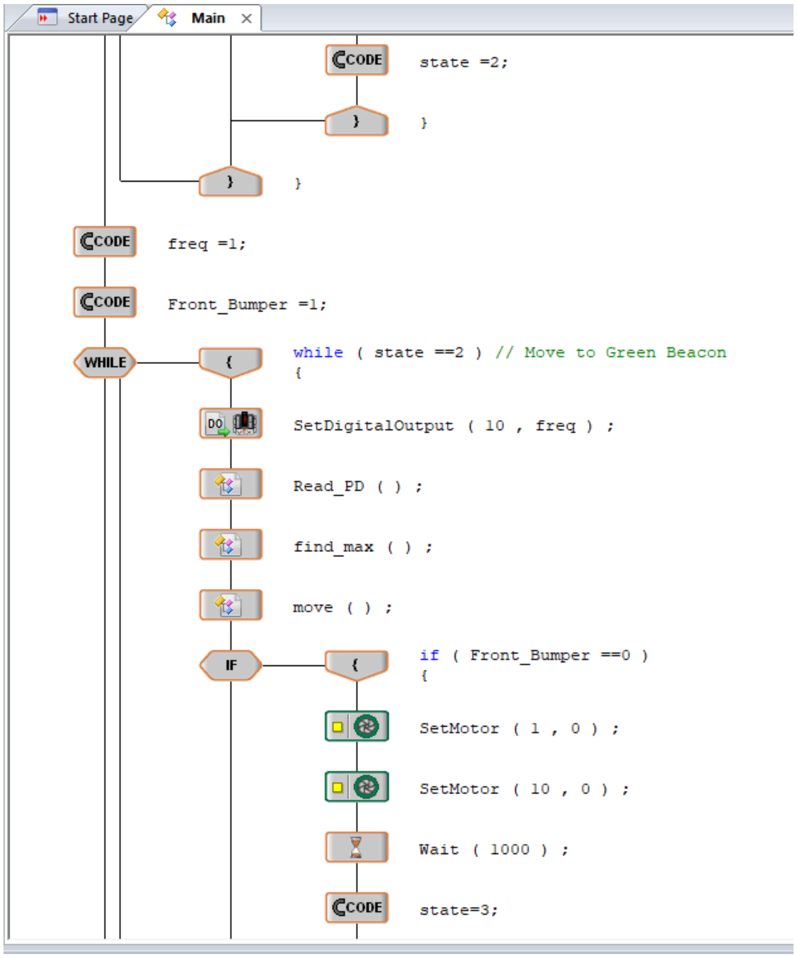
```
while ( state==0 )
{
Read_PD (   ) ;
find_max ( ) ;
move ( ) ;
Front_Bumper = GetDigitalInput ( 4 ) ;
    if ( Front_Bumper ==0 )
    {
    SetMotor ( 1 , 0 ) ;
    SetMotor ( 10 , 0 ) ;
    state =1;
    }
}
while ( state ==1 ) // State of Turn off Red Beacon
{
SetServo ( 2 , -127 ) ;
SetServo ( 3 , 127 ) ;
Wait ( 1000 ) ;
```

Figure 7: State=0 of Code (Find Red Beacon)



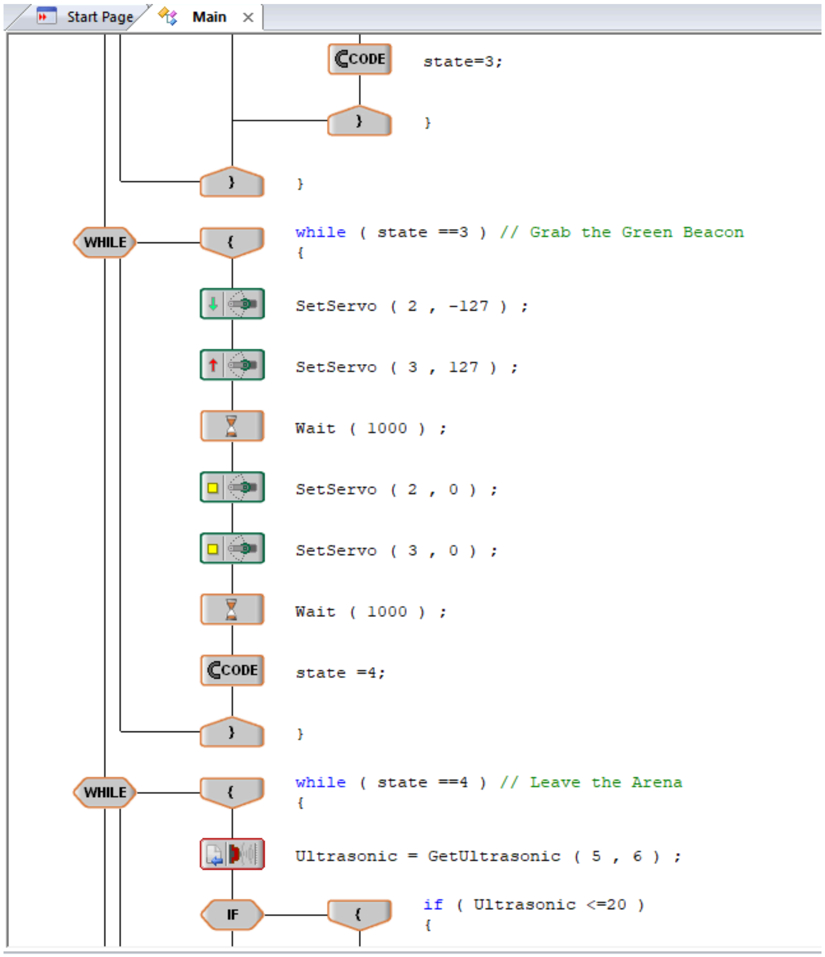
```
Wait ( 1000 ) ;
SetServo ( 3 , -127 ) ;
SetServo ( 2 , 127 ) ;
Wait ( 1000 ) ;
Read_PD ( ) ;
Wait ( 1000 ) ;
if ( PD_sum < ambient_level  ) // If the Red Beacon Turns Off
{
    SetServo ( 2 , 0 ) ;
    SetServo ( 3 , 0 ) ;
    SetMotor ( 1 , -127 ) ;
    SetMotor ( 10 , -127 ) ;
    Wait ( 700 ) ;
    SetMotor ( 1 , 0 ) ;
    SetMotor ( 10 , 0 ) ;
    state =2;
```

Figure 8: State=1 of Code (Turn off Red Beacon)



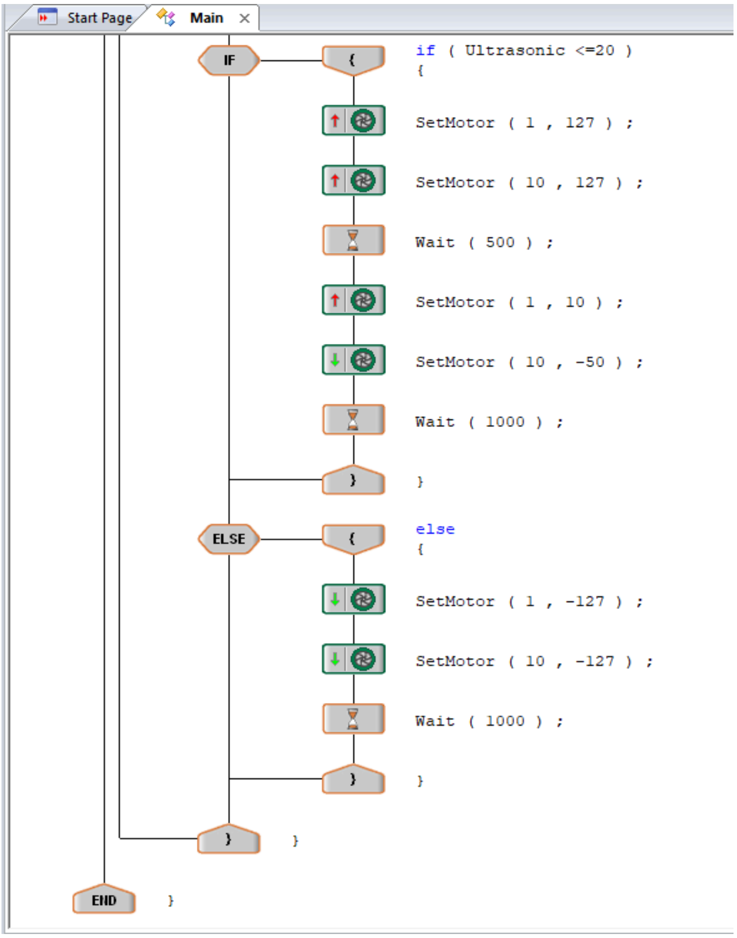
```
state =2;
}
}
freq =1;
Front_Bumper =1;
while ( state ==2 ) // Move to Green Beacon
{
SetDigitalOutput ( 10 , freq ) ;
Read_PD ( ) ;
find_max ( ) ;
move ( ) ;
if ( Front_Bumper ==0 )
{
SetMotor ( 1 , 0 ) ;
SetMotor ( 10 , 0 ) ;
Wait ( 1000 ) ;
state=3;
```

Figure 9: State=2 of Code (Move to Green Beacon)



```
    state=3;
    }
}
while ( state ==3 ) // Grab the Green Beacon
{
SetServo ( 2 , -127 ) ;
SetServo ( 3 , 127 ) ;
Wait ( 1000 ) ;
SetServo ( 2 , 0 ) ;
SetServo ( 3 , 0 ) ;
Wait ( 1000 ) ;
state =4;
}
while ( state ==4 ) // Leave the Arena
{
Ultrasonic = GetUltrasonic ( 5 , 6 ) ;
    if ( Ultrasonic <=20 )
    {
```

Figure 10: State=3 of Code (Grab the green beacon)



```
                if ( Ultrasonic <=20 )
                {
                SetMotor ( 1 , 127 ) ;
                SetMotor ( 10 , 127 ) ;
                Wait ( 500 ) ;
                SetMotor ( 1 , 10 ) ;
                SetMotor ( 10 , -50 ) ;
                Wait ( 1000 ) ;
                }
                else
                {
                SetMotor ( 1 , -127 ) ;
                SetMotor ( 10 , -127 ) ;
                Wait ( 1000 ) ;
                }
        }
}
```

Figure 11: State=4 of Code (Leave the Arena with Green Beacon)

Conclusions

Given how difficult it can be to create a proper logic code to run a functioning robot using an IR board to locate its target. Creating a programming logic code flowchart helps determine each movement of the robot. When the robot was stuck in a task, pinpointing the error in the programming code was simple, by following wherein the code logic flow chart the robot was stuck on. And determining which loop it was trying to complete. . In all, the robot was programmed to locate its target, the red beacon and go in front of it to turn it off. However, the robot was not able to turn it off because of the proper placement of the arms. It was also able to locate the green beacon and carry it out of the arena within 19 sec as its robot weight was 2.07 kg. It completed all but one task which was turning off the red beacon.

For future work, creating a proper code logic flow chart will help reduce the time it takes to create a running code. Also, to note when soldering delicate electrical components onto a circuit board the heat must be used to a minimum to just melt the solder to connect the electrical components. Because electrical components are extremely delicate, heat and static can cause resistors, diodes, and capacitors to short or change value. This was a problem for this robotic lab because the IR board was not functioning and had to be replaced with a new one. Lastly, measure the distance between the arm of the robot to the red beacon off switch to be able to turn it off.

References

[1] Hsu, P. (2020). *Robot Project Guidelines*. E10 Introduction to Engineering.

[2] Meng, Q., & Lee, M. H. (2005). Design issues for assistive robotics for the elderly. *Advanced Engineering Informatics*, *20*(2), 1–16. https://doi.org/10.1016/j.aei.2005.10.003

[3] Hofilena, John. (2013, April 29). Japan pushing for low-cost nursing home robots to care for elderly. *Japan Daily Press*. Retrieved from http://japandailypress.com/japan-pushing-for-low-cost-nursing-home-robots-to-care-for-elderly-2927943

[4] Rau, P. L. P., Li, Y., & Li, D. (2009). Effects of communication style and culture on ability to accept recommendations from robots. *Computers in Human Behavior*; *25(2)*, p. 587-595.

[5] Jost, D. (2019, July 29). *Fierce Electronic*. Retrieved from What is an IR sensor: https://www.fierceelectronics.com/sensors/what-ir-sensor