System Details:

As discussed previously, the robot functions by either moving forward or backwards, depending on the user's input through a joystick. The device will detect an object (in this case a piece of Lego) once the latter is in the range of proximity sensor used. After that, a camera will be able to view the Lego, and will input an image into the program deployed on the device. The program will then detect the color, and will allow the robotic arm and clamp to move accordingly. In this section, we will give a more detailed look into the device's features and key-functions. First and foremost, we will start with overall illustrations of the device, and then move to more detailed illustrations of individual parts of the device.

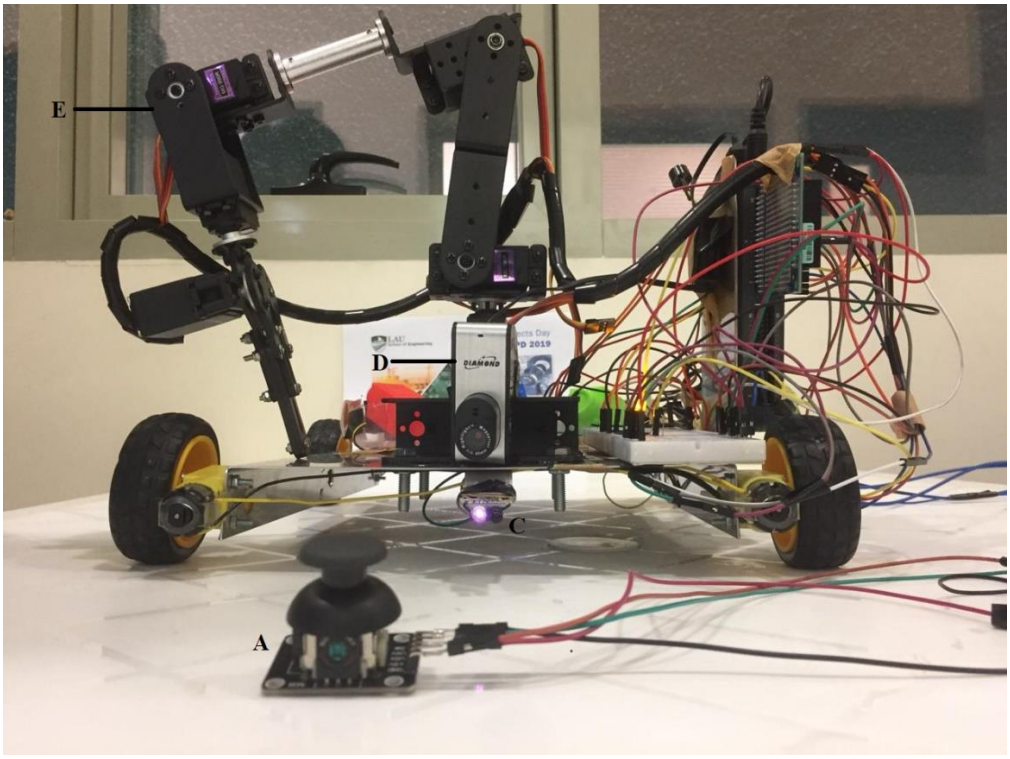

Figure 1: Overall Picture of the Device.

A- Arduino Joystick

C- Proximity Sensor

D- Camera

E- Robotic Arm and Clamp

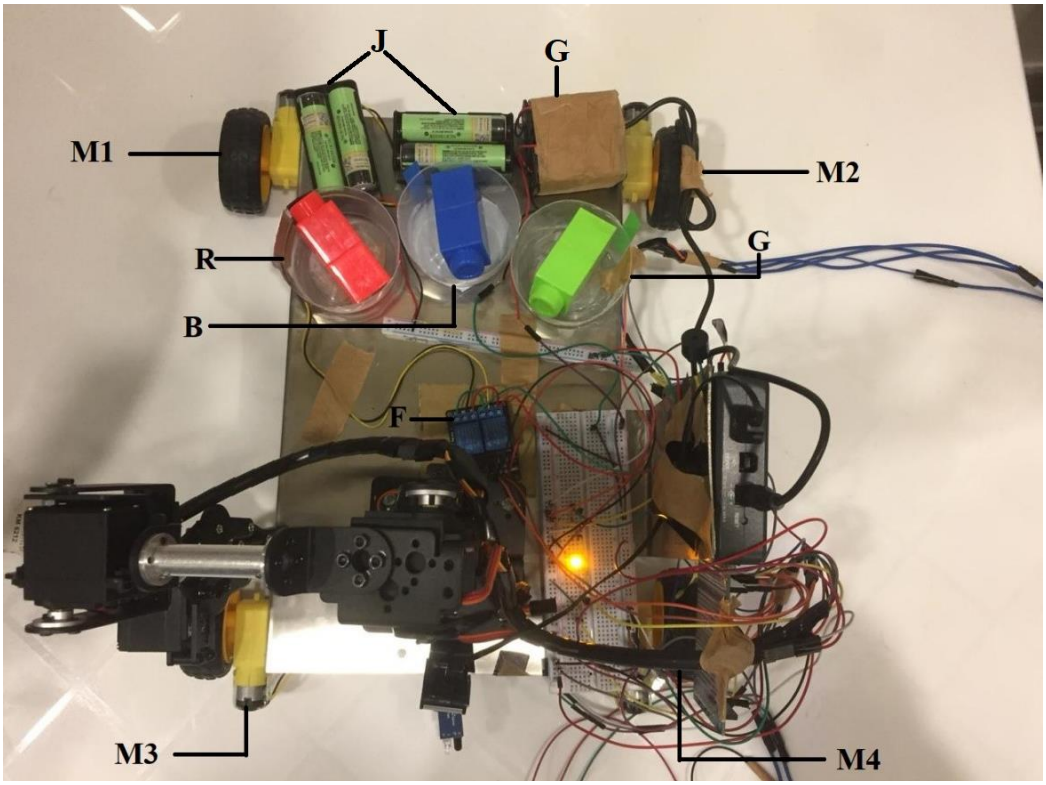

Figure 2: Overall Picture of the Device.

R- Red Lego Compartment

B- Blue Lego Compartment

G- Green Lego Compartment

F- Relay Board

M1, M2, M3, M4- DC motors attached to wheel

J- External Battery Supply

G- Power Supply (Batteries) for myRio

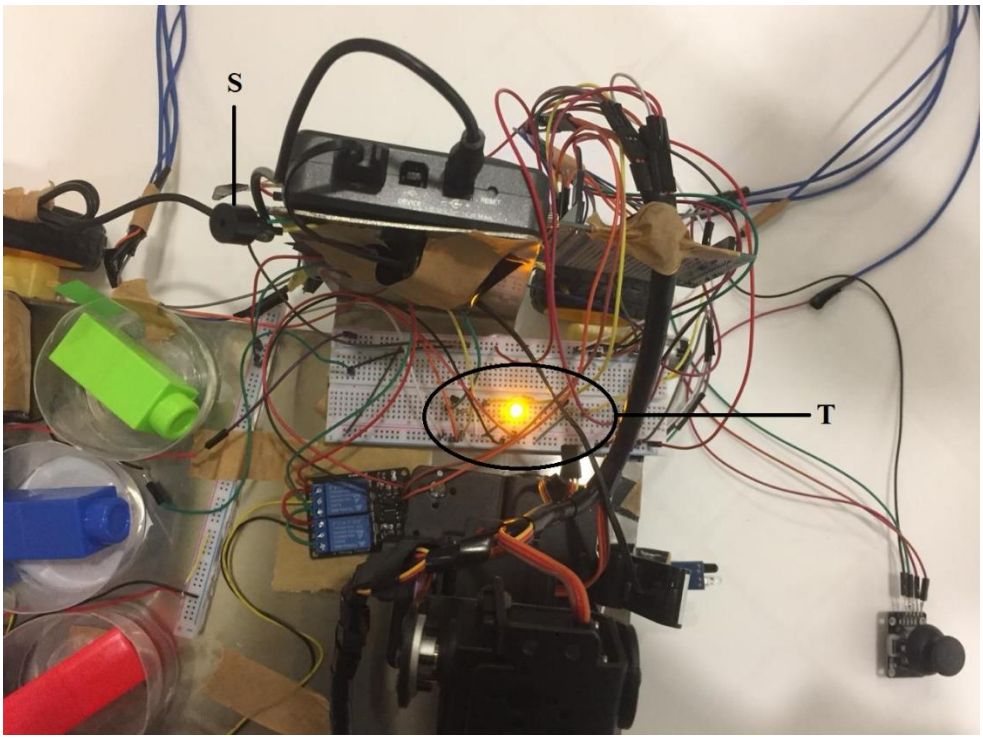

Figure 3: Overall Picture of the Device.

S- Buzzer

T- LED and Buzzer Circuits

The robot functions through a combination of hardware and software configurations. These configurations work interconnectedly to serve the robot's purpose. The LabView software program communicates with the device's hardware through the myRio and its available pins. The LabView code follows state machine architecture, and consists of four states (Initialization, Idle, Camera, and Exit).

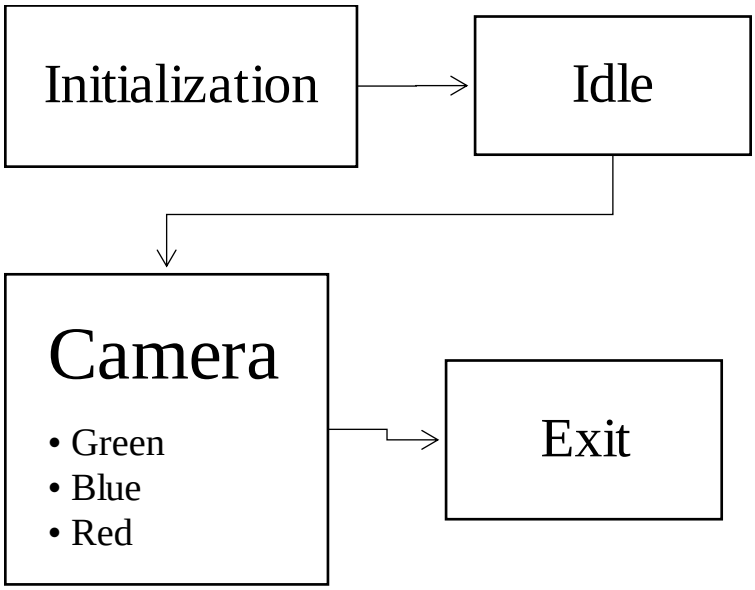

Figure 4: State Machine Flow Chart.

The program is deployed on the myRio, that is attached to the device, and runs whenever the myRio initiates. Initialization is the first state the device is in. We then enter the Idle state, where we are able to move the robot forward and backwards through the Joystick. Once the device detects a Lego piece, the LED and Buzzer turn on and we enter the Camera state. In that state, the robot will detect the color of the Lego, give a signal to the arm and clamp, which will place the Lego in its compartment (attached to the back of the device). We can then either go back to the Idle state where we can move the robot again, or we can stop the program by pressing the joystick. Each of these states are shown in the *Appendix, Part A, B, C, and D* respectively. To begin with, the hardware part consists of a relay board in coordination with a joystick, that is capable of moving the robot forward and backwards (through motors connected in parallel and to the relay board). This is illustrated below:

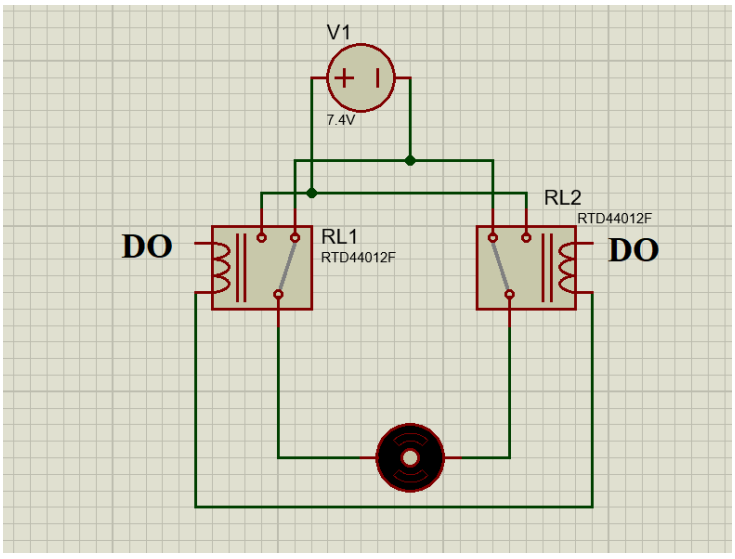

Figure 5: DC Motors with Relay Board Circuit.

The DO (Digital Output) connected to the relay are controlled from the LabView code, in the first case of the Flat Sequence (*Appendix Part B*).

The hardware also consists of a LED and Buzzer circuit configuration that turn on once the robot detects a piece of Lego. The latter happens through a proximity sensor that senses an object in its range and then sends a signal to the software program that turns on the LED and Buzzer for a few seconds (through myRio's Digital Outputs "DO", in the second case of the Flat Sequence; *Appendix Part B*). In addition, the hardware part also includes a robotic arm and clamp, that move according to a PWM signal given to it from the LabView program (*Appendix Part C*). The hardware circuitry for the LED and Buzzer are as follows:

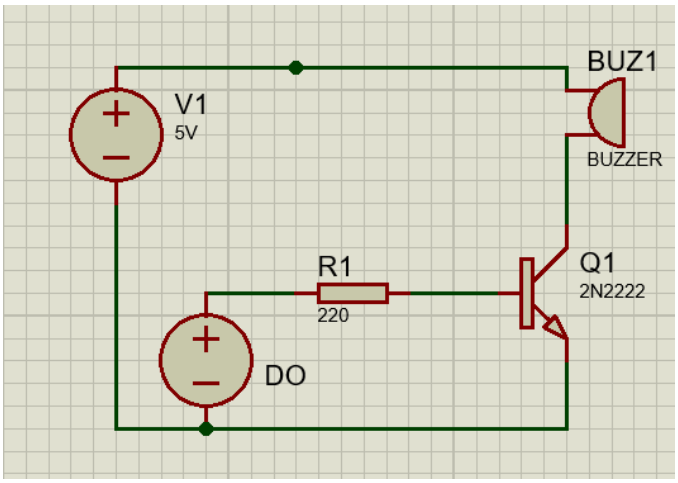

Figure 6: Buzzer Circuit.

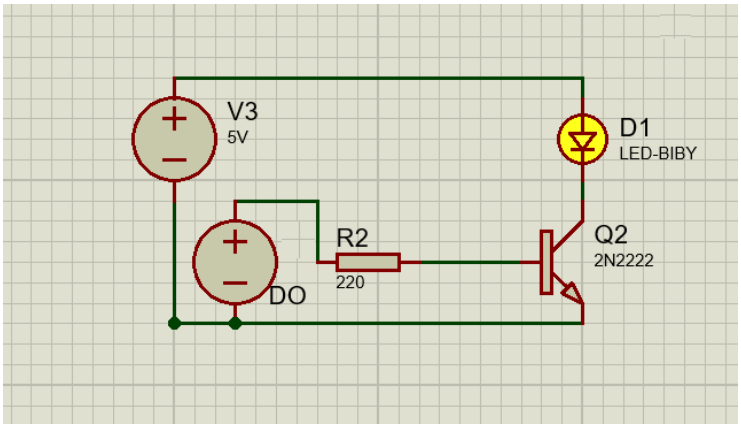

Figure 7: LED Circuit.

After that, the camera turns on and sends a picture of the Lego in front of it to the program. The program can then recognize the color through the code below (*Figure 8)*. The color of the Lego should fall between one of the three defined ranges (1-6, 8-20, or 30-50) for the corresponding color LED to turn on. Then, according to the color, the program (*Appendix Part C*) allows the arm and clamp to move, pick up the Lego piece then place in its corresponding compartment. The arm and clamp then go back to its initial position defined in the "initialization" state (*Appendix Part A*), where we can move the device again. We can also stop the program by pressing and holding down the joystick.

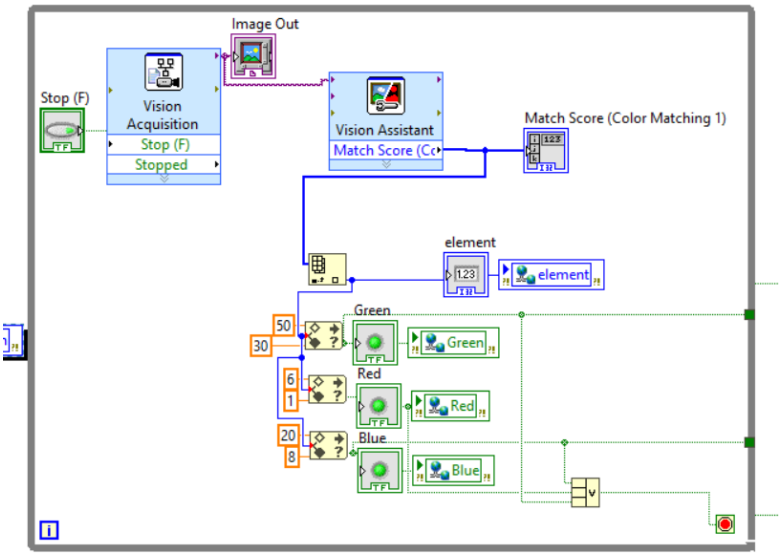

Figure 8: LabView Camera Code.

Appendix:

Part A

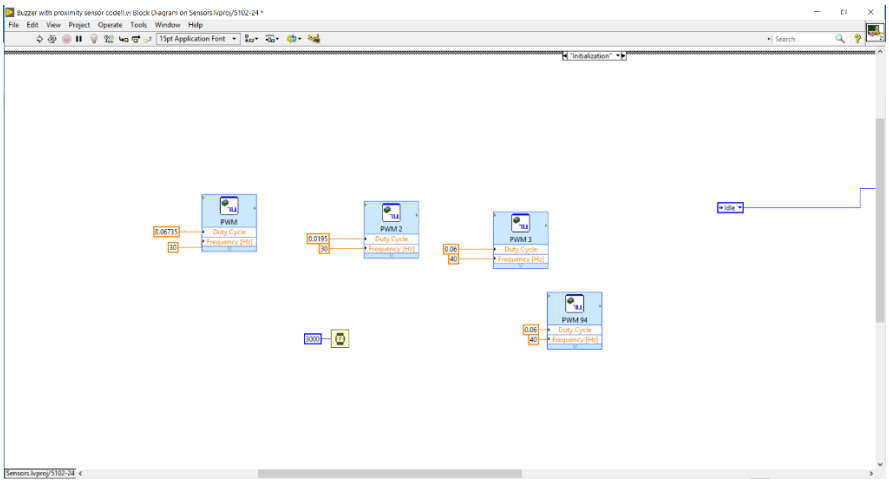

Figure 9: Initialization State.

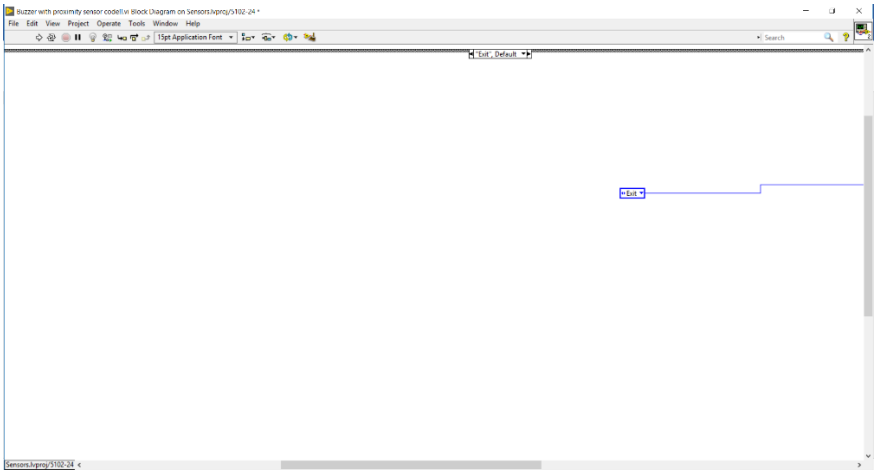

Figure 10: Exit State.

Part B

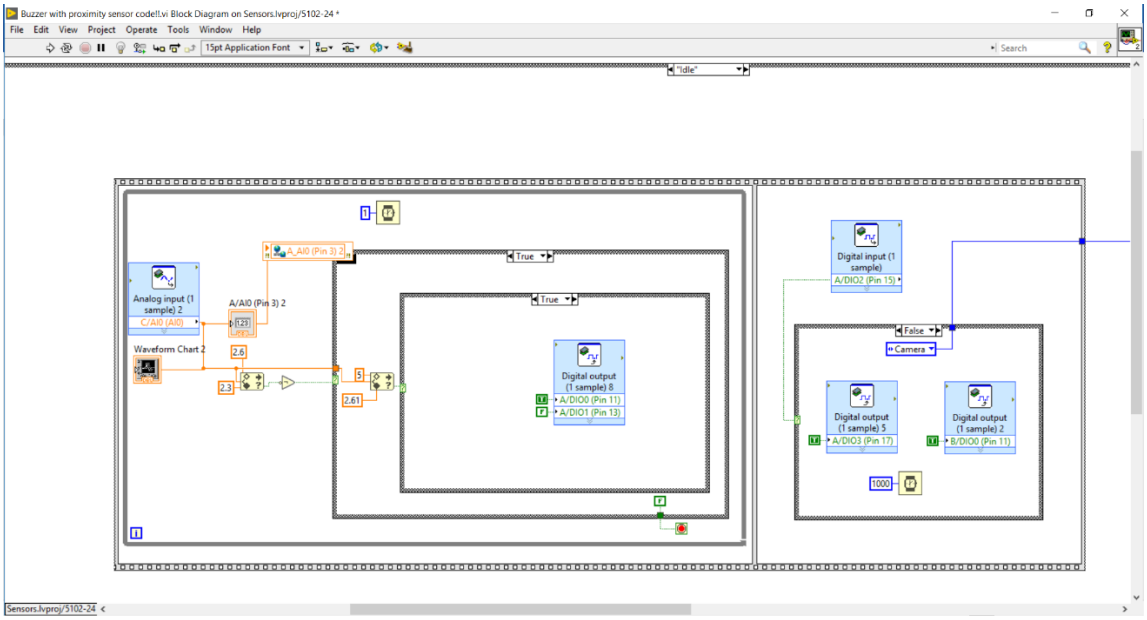

Figure 11: Idle State (Once an Object is Detected).

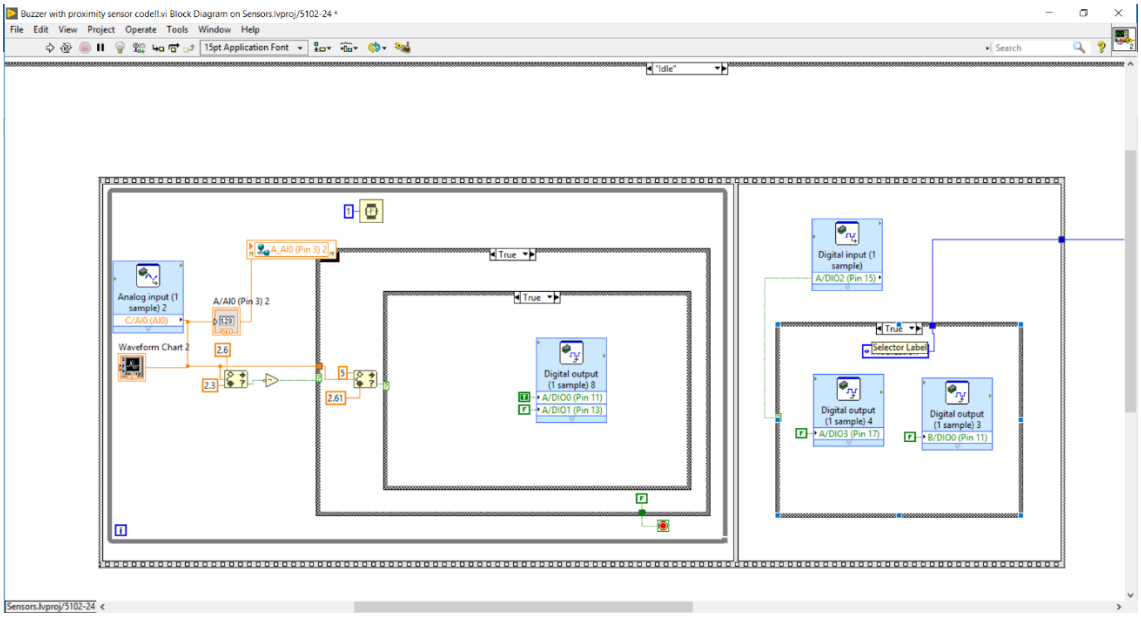

Figure 12: Idle State (When Nothing is Detected).

Part C

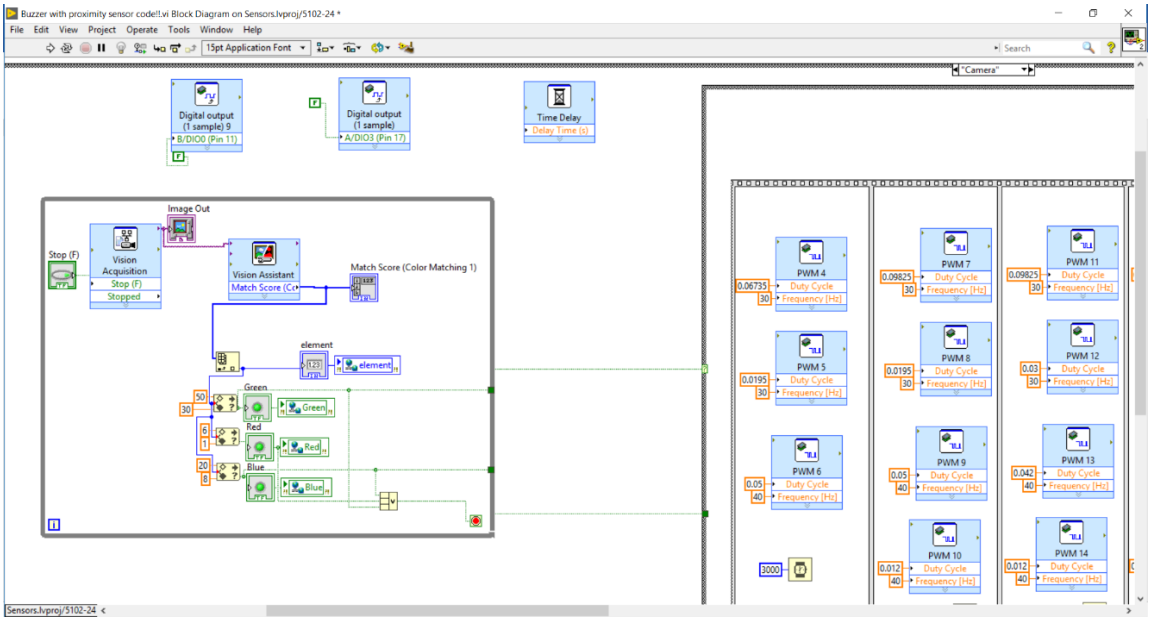

Figure 13: Camera State.

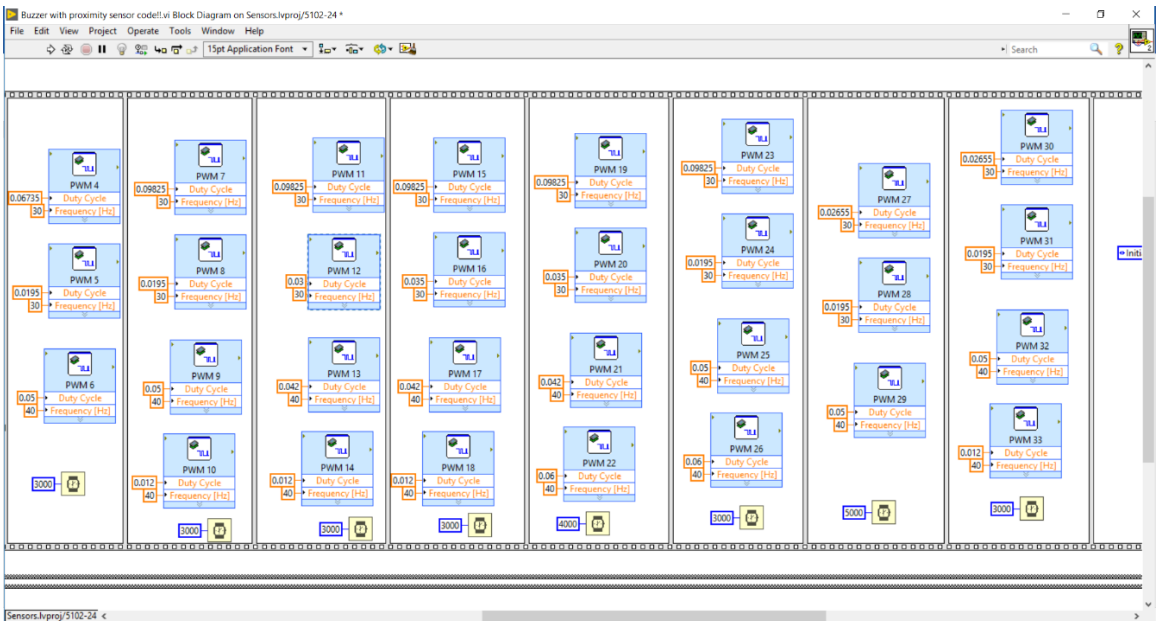

Figure 14: PWM Signals for Green Lego Detection.

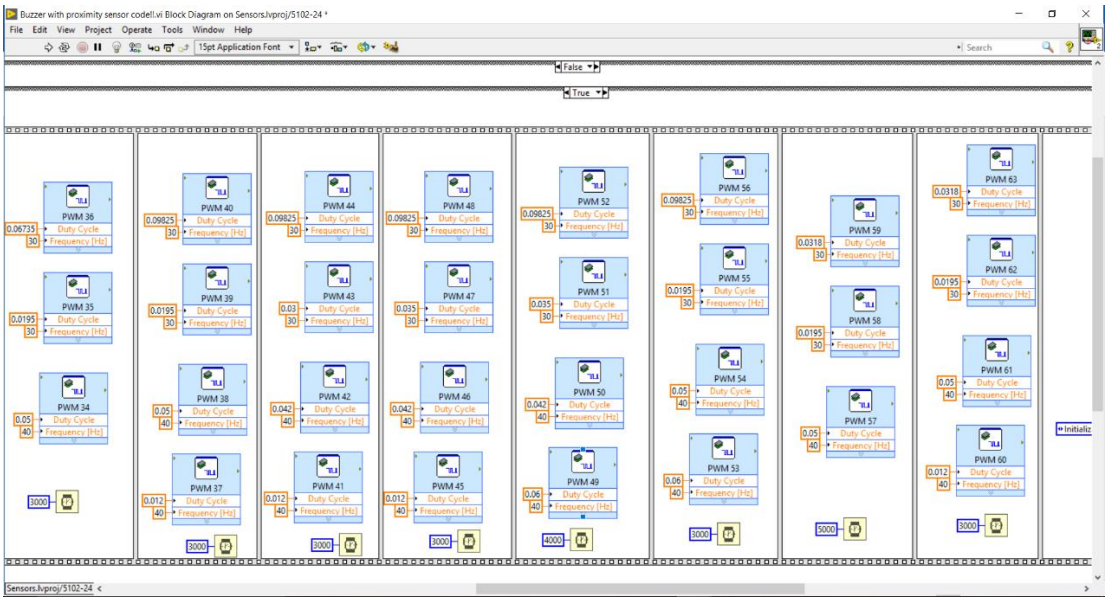

Figure 15: PWM Signal for Blue Lego Detection.

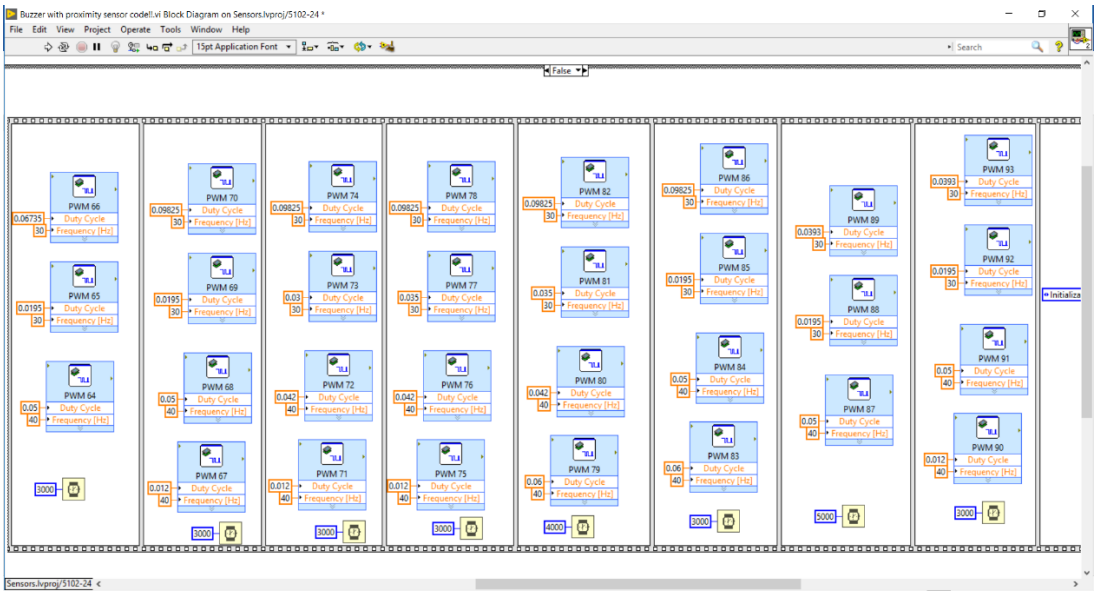

Figure 16: PWM Signal for Red Lego Detection.