Erick Chacon
800696361

Familiarize Students with the LabVIEW Project and introduce students to the DaNi robot. ________

Write a small program to set up the Parallax PING ultrasonic sensor. ________

Write a second program that allows the user to remotely control the direction the Parallax PING ultrasonic sensor on a 180 degree rotational axis. ________

Write a third program to make the led light on the back of the DaNi robot blink at any speeds specified by the user. ________

General Learning Objectives:

The general learning objectives of this lab was to enhance the users' skills in the use of LabVIEW. By creating a program in which the DaNi robot's ultrasonic sensor and LED were controlled, the users were able to become more familiar with LabVIEW. The users also learned the different ways in which the robot could be controlled and how the interfaces in the LabVIEW program show what the robot is sensing and what position the servo is at.

General Steps Needed to Complete the Lab:

The general steps needed to complete this lab were to follow the tutorial provided to us. The first part of the experiment included to write a program in which the ultrasonic sensor was setup. Once the program was created, it was executed and a graph showed the distance measured, in real time, of a nearby object. The second portion of the first program included controlling the angle of the servo in which the ultrasonic sensor was mounted. For the final part of the lab, a second program was created to control the blinking of an LED. The rate at which the LED blinked was controlled through this program.

Procedure/ Detailed Steps to Complete the Lab:

The lab experiment was broken down into **X** steps:

1. The DaNi 2.0 robot was connected to the computer using a LAN cable. It was ensured that the motors were switched off and only the main supply was on. The successful connection of the robot was indicated by the LAN icon on the windows task bar.
2. LabVIEW was then opened and then using the Robotics 2011 plugin, the hardware was detected. Furthermore, steps 2-8 were followed and proper function of the motor, ping sensor and servos was confirmed.
3. In order to setup the parallax PING ultrasonic sensor, a time loop was selected. A stop control and a waveform chart were added to the time loop. A Bitfile for starter 2.0 kit was

selected then a constant was created for the sbRIO using the open FPGA reference. An interface pertaining to the PING sensor was selected using FPGA read/write VI. This interface was 'PING)))_0_dist'. Change All to Read functionality was selected from the FPGA read/write and the sensor output from the read/write VI of 'PING)))_0_dist' was connected to the waveform chart. Closed FPGA VI reference was added to the outside of the loop. Connections were made between the sbRIO interface, the read/write interface, and the closed FPGA VI reference. Figure 1 shows the completed VI for the front panel and Figure 2 shows the completed VI for the back panel. Thus, a small program to set up the Parallax PING ultrasonic sensor was written.

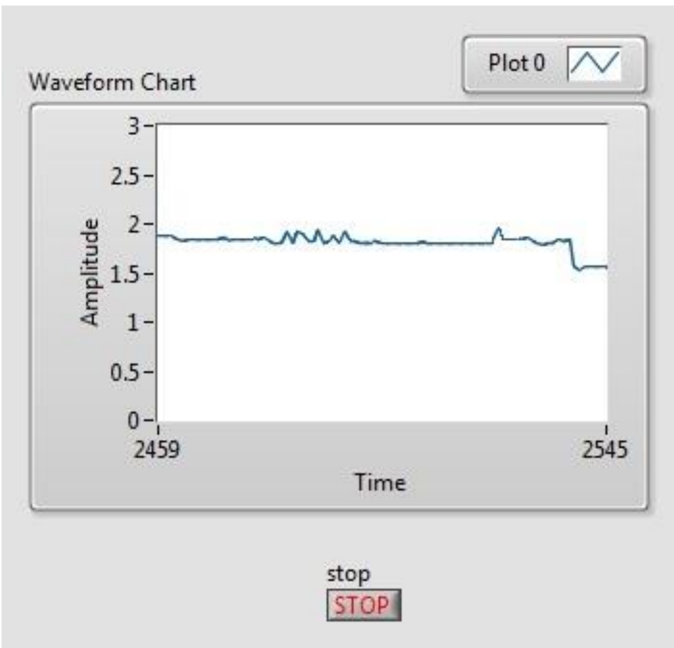

Figure 1: Completed Front Panel VI for Ultrasonic Setup

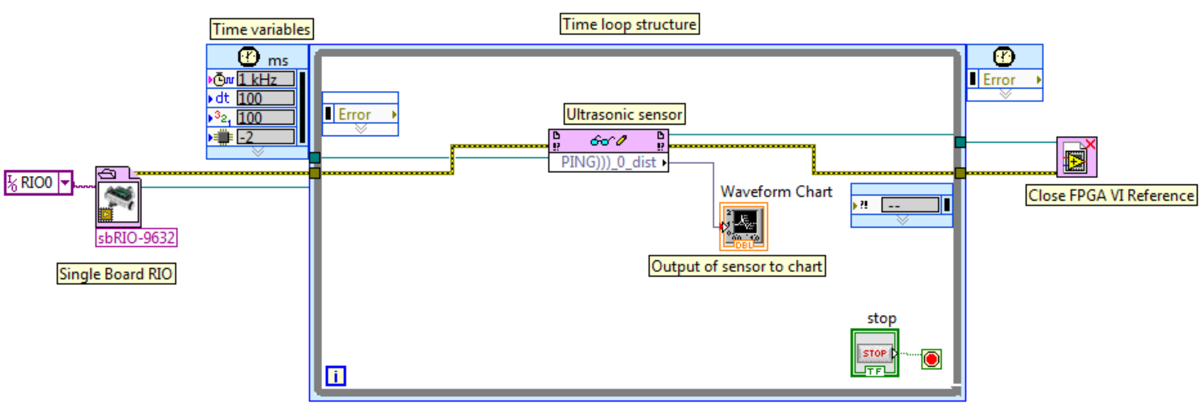

Figure 2: Completed Back Panel VI for Ultrasonic Setup

4. After the graphical code was successfully compiled, the output on the waveform chart in response to the obstacle proximity was observed. As shown in Figure 3, a dip was noticed in the graph when the object/obstacle was in close proximity of the sensor as compared to the time when there was no object in close proximity of the ping sensor. These descriptions are labeled in the figure.

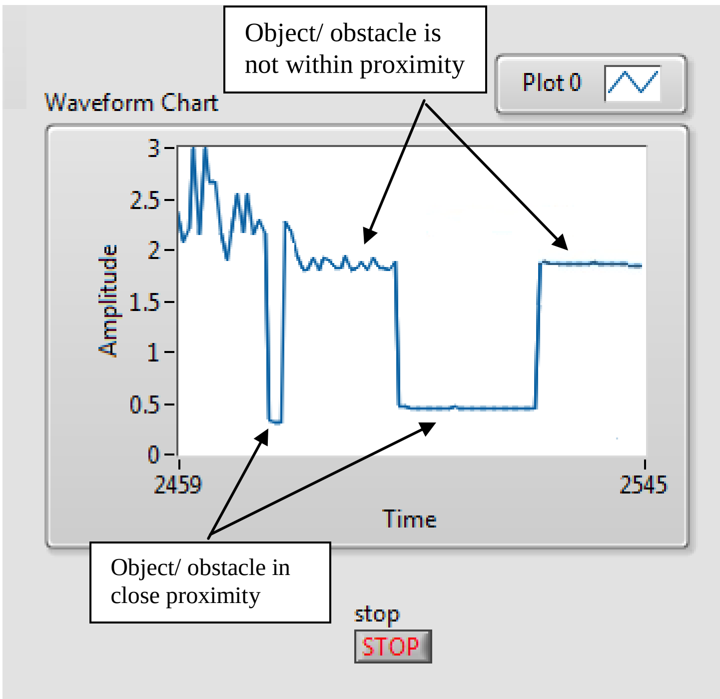

Figure 3: Describing chart output from sensor reading

5. In order to control orientation of the ping sensor a control interface 'hs400_0_ccw angle' was added to the Open FPGA Read/Write block. This was created to pass the angle of the motor to the FPGA Bitfile that implements the driver. A slider was added to the front panel for feeding in values to the sbRIO. Since the driver only accepts values in radians a conversion block from degrees to radians was added in between the slider output and the angle control interface on the FPGA Read/Write block. The completed front panel is shown in Figure 4 which shows the servo at 90 degrees. The response of the servo in reference to the slider control on the front panel is shown in Figure 5. The front panel is shown in Figure 6 which shows the servo at -90 degrees. The response of the servo in reference to the slider control on the front panel is shown in Figure 7. The back panel is shown in Figure 8. Thus, the requirement which allows the user to remotely control the direction of the Parallax PING ultrasonic sensor on a 180 degree rotational axis was achieved.

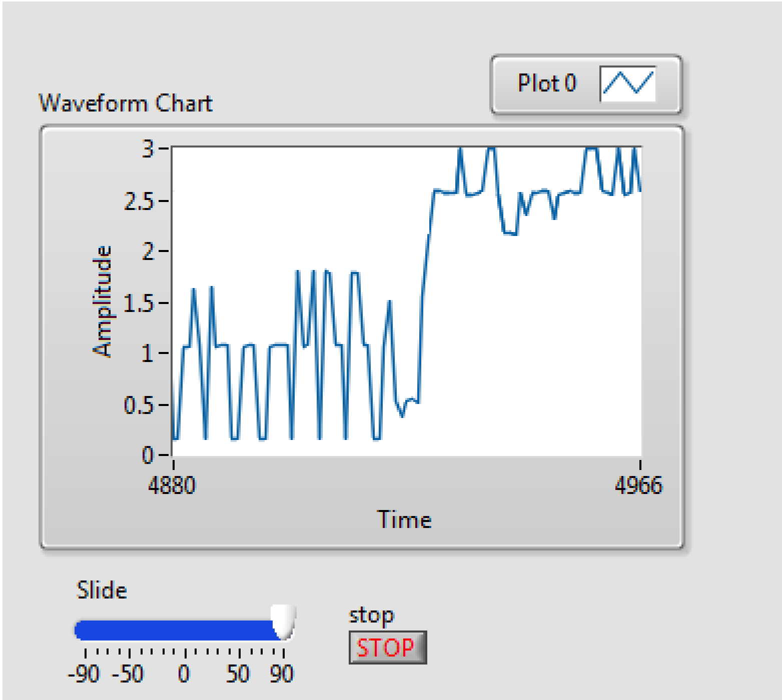

Figure 4: Completed Front Panel VI for Ultrasonic Setup with Servo Control at 90 degrees

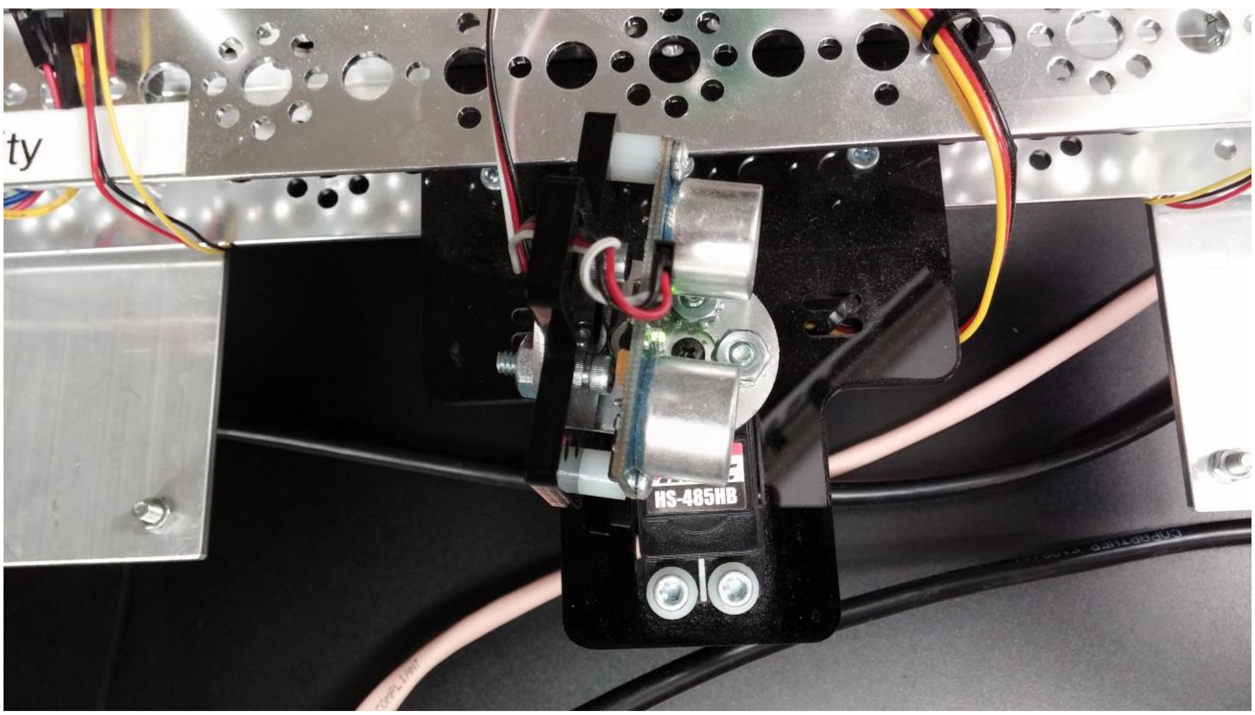

Figure 5: Response of the Servo with respect to Slider Value of +90

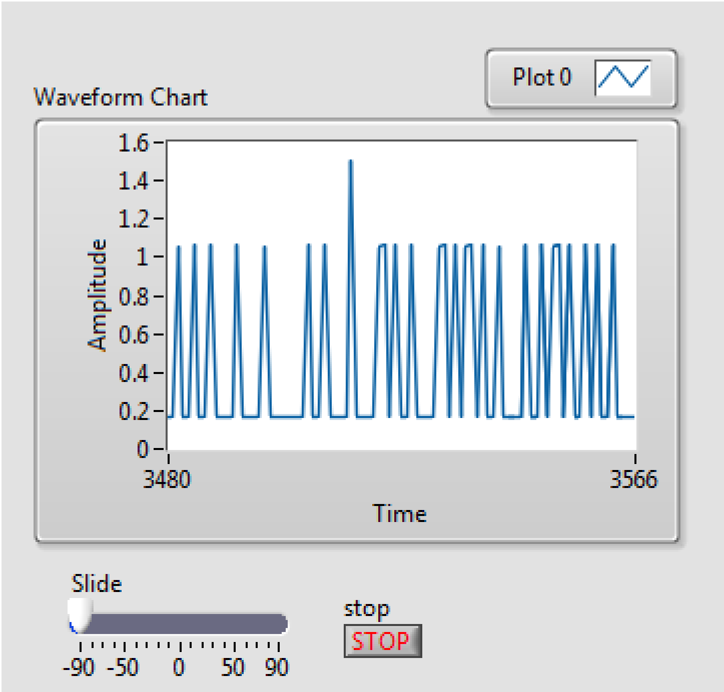

Figure 6: Completed Front Panel VI for Ultrasonic Setup with Servo Control at -90 degrees

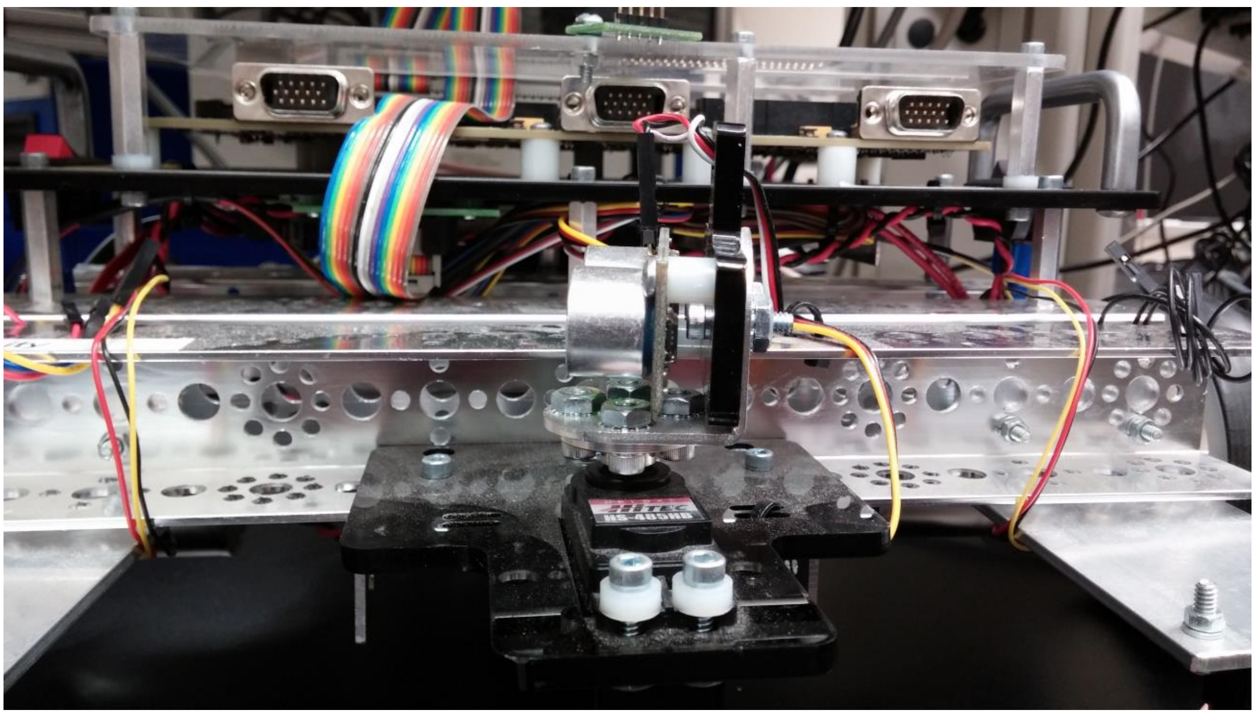

Figure 7: Response of the Servo with respect to Slider Value of -90

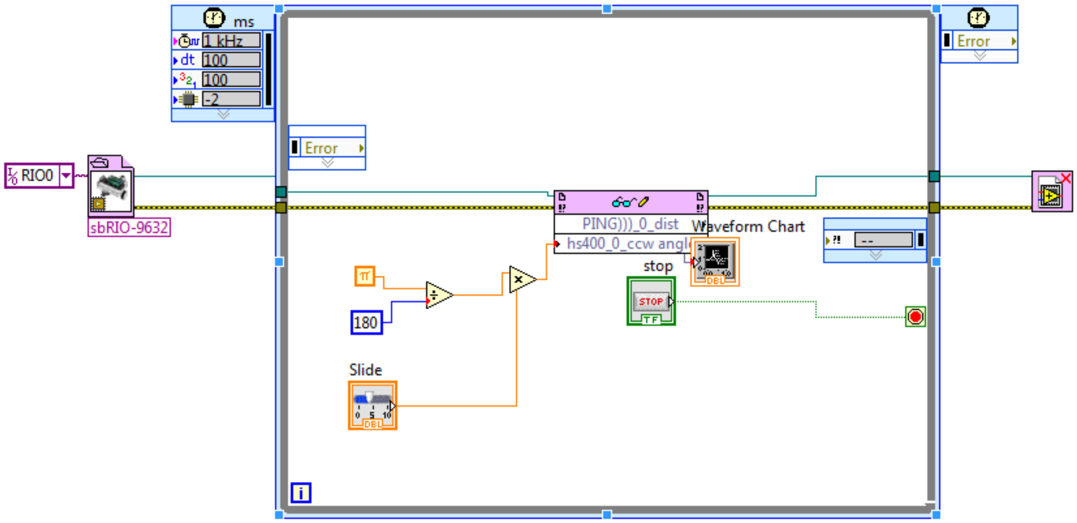

Figure 8: Completed Back Panel VI for Ultrasonic Setup with Servo Control

6. In order to blink the LED continuously, a while loop was added to a blank VI. A constant was created using open FPGA reference and the sbRIO was selected. A read/write VI was added to the loop and the FPGA LED interface was selected from the drop-down menu. The closed FPGA VI reference was added outside the while loop. Shift registers were added to the while loop. Output from one of the shift registers was wired to the input to the NOT gate. The NOT gate output was branched and one of it was given as input to the other end of the shift register. In order to toggle the voltage to the FPGA LED, the other NOT gate output was given to the input of the FPGA. Boolean constants were added to control the while loop and the shift register. In order to determine the time duration between blinking LEDs, a timer with millisecond input value was added to the while loop. Connections were made between the open FPGA reference constant, the read/write FPGA interface (FPGA LED) and the closed VI reference. The completed front panel VI is shown by Figure 9 which shows the timer at a value of '100' milliseconds. The completed back panel is shown in Figure 10. The program was successfully compiled and the blinking of the LEDs was observed as shown in Figure 11 and Figure 12. The value of '1000' millisecond as shown by Figure 13 was fed through the front panel & the program was compiled. Thus, a decreased frequency of LED blinking was noticed for this value of counter.

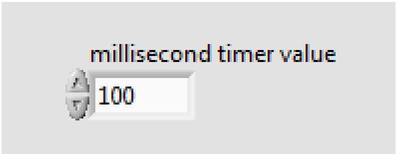

Figure 9: LED Front Panel Timer Value of 100 Milliseconds

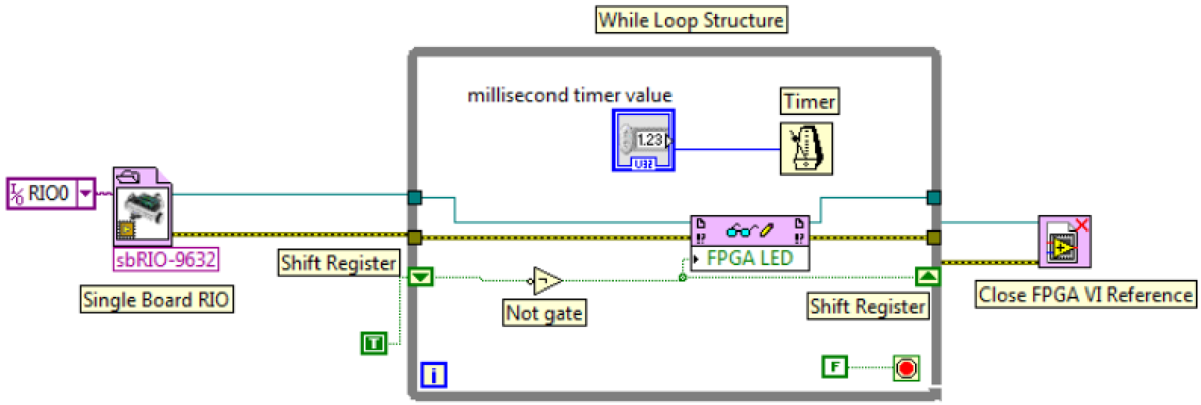

Figure 10: LED Completed Back Panel

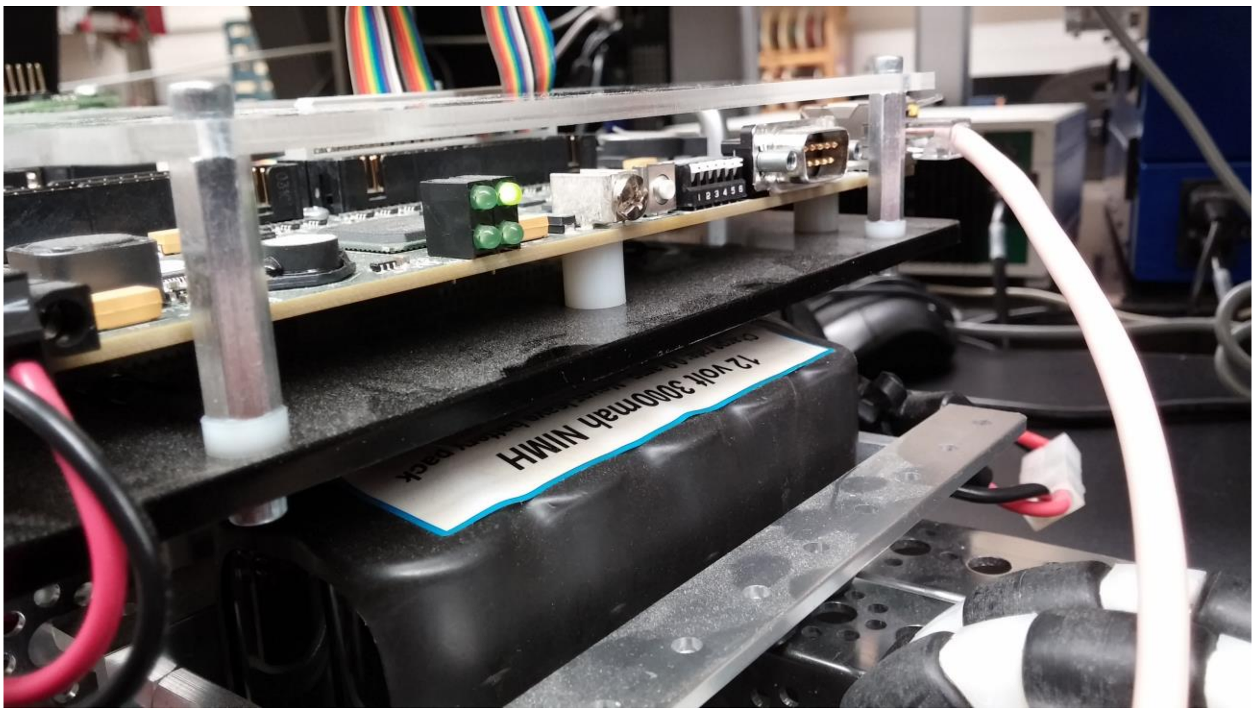

Figure 11: LED Off

Tanmay Sane
800799458
Erick Chacon
800696361

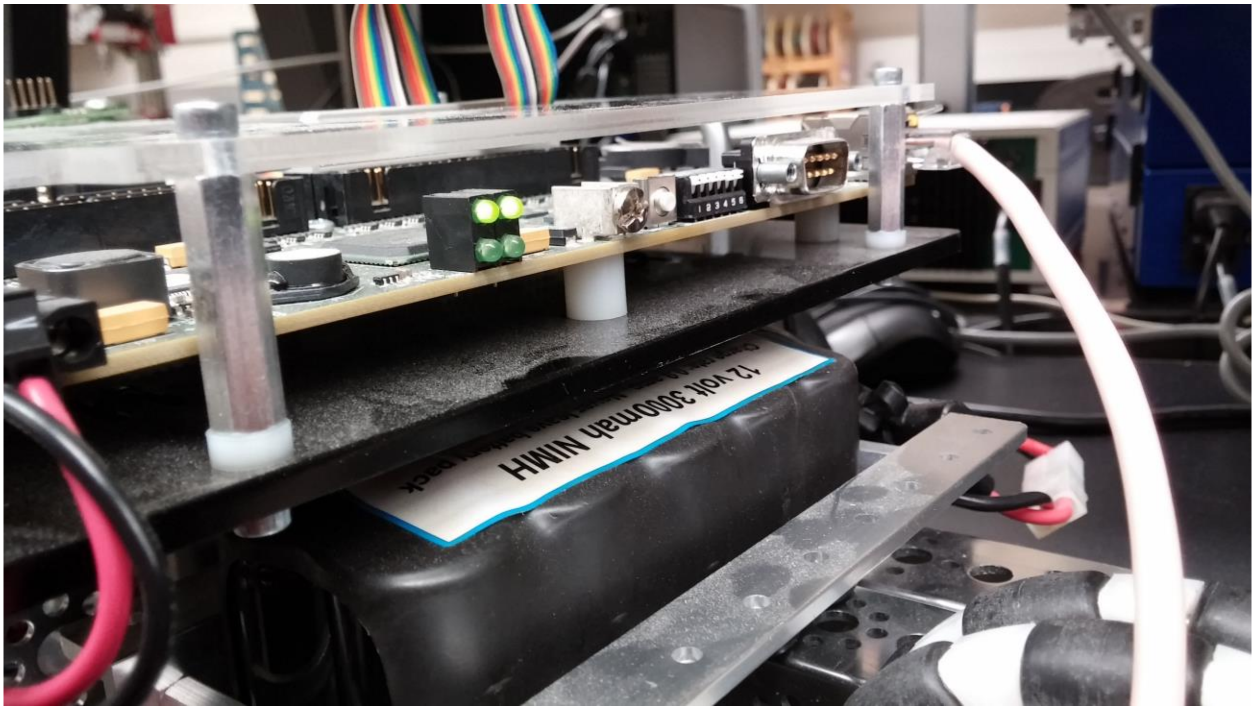

Figure 12: LED On

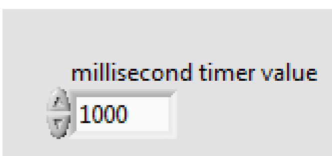

Figure 13: LED Front Panel Timer Value of 1000 Milliseconds

Observations while completing/testing the Lab:

There were several observations that were made while completing this lab experiment. Some important observations that were made were that the LabVIEW programming language is simpler than expected. After analyzing and interpreting what each of the graphical icons and structures mean, it is simple to create complex programs to perform a various controlling operations. Another observation that was made was that if both programs are placed under one VI, it runs both of them simultaneously. This observation was very important so that it could be used for future reference in upcoming labs .Thus it was inferred that various smaller building blocks can be integrated to form bigger operational modules. One final observation was that the changes in the waveform were in accordance to the distances of objects from the ultrasonic sensor.

Lesson Learned:
In this lab we learned the following concepts & implementations.
a. Further usage of LabVIEW and its different graphical icons and robotics settings.
b. Ability to program a controller for the servo and LED on the DaNi.
c. The use of timers, shift registers and horizontal pointer sliders.
d. Programming a FPGA based robot over the Ethernet as compared to the usual micro USB on other microcontroller boards.

References:
1. Experiment No. 3 – Introduction to LabVIEW Manual
2. LabVIEW 2011 Robotics Kit DaNI 2.0 Quickstart Guide