Wireless Servo

Project Report

Vlad Dragoi, Ryan Bowman, Rebecca Slota
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
Email: vdragoi@oakland.edu, rmbowman@oakland.edu, rlslota@oakland.edu

Table of Contents

I. Introduction

The objective of this project was to control a servo through a wireless communication system. The wireless system utilized two XBee modules to transmit and receive data through point to point communication between the Dragon12-Light board and Arduino Mega 2560. The encoded commands sent by the Dragon12-Light board are interpreted by the Arduino. The Arduino uses the commands to manipulate the servo.

This project covers subjects learned in class such as utilizing a rangefinder, serial communication, interrupts, timer modules, and mixed assembly code. This project also implements wireless communication using wireless technology.

The applications of this project include the ability to control motors wirelessly through gesture. This project could be further developed to apply to wireless gesture controlled door systems, coffee makers, toys etc.

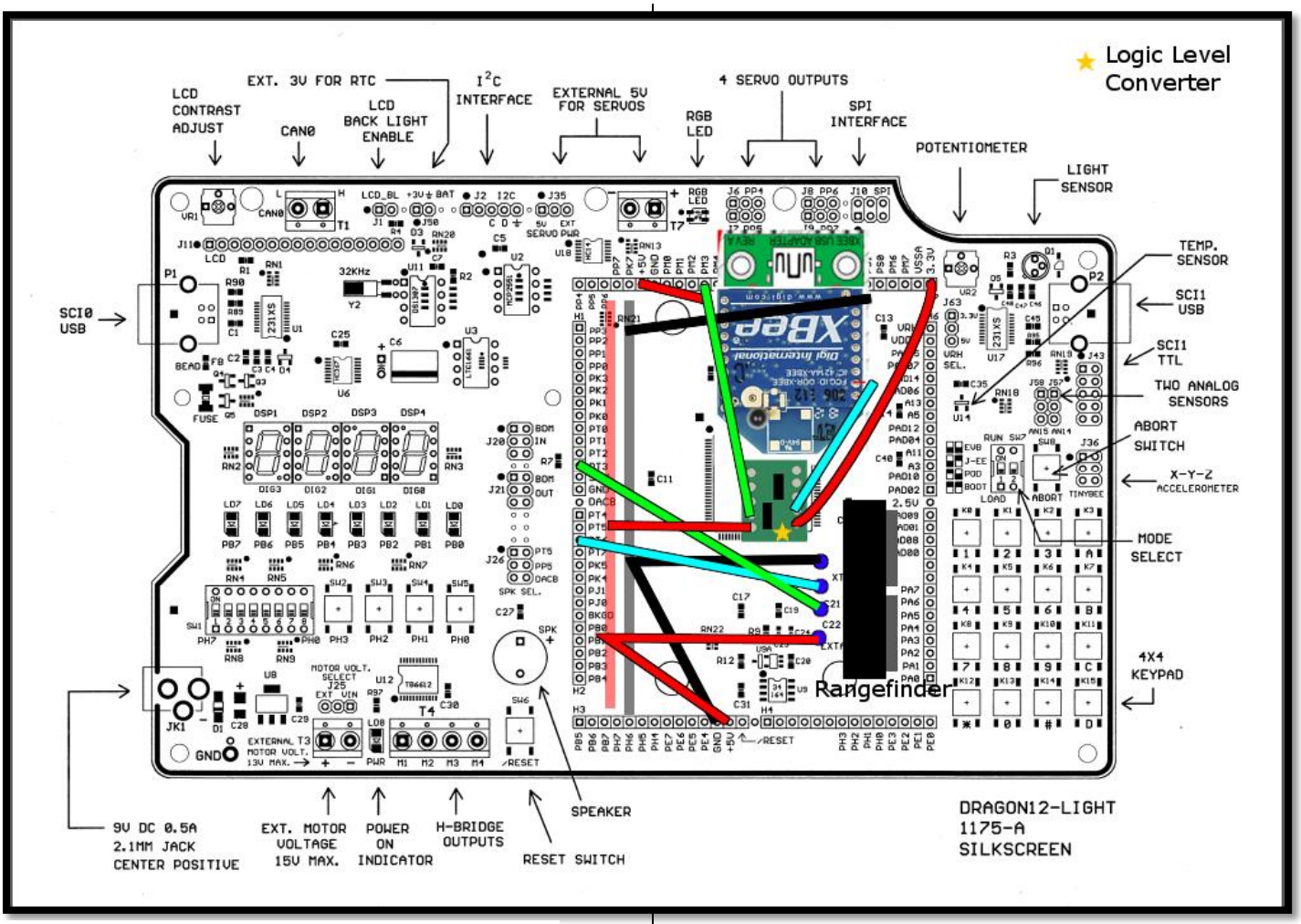

Figure 1

II. Methodology

Dragon12-Light Board

The Dragon12-Light Board implements an ultrasonic range finder, HC-SR04, in order to transmit data across the wireless system. The ultrasonic range finder is a sensor that detects distance based on a light transmitter

In order to communicate to the XBee, a logic level converter was needed to prevent overvoltage on the RX XBee pin. This is because the XBee runs on 3.3 V, while the rangefinder and Dragon12-Light Board run on 5 V. The logic level converter allows data transmission between two different voltage levels and requires two voltage inputs, as shown in Figure 1. The XBee data line is connected to the 3.3 V while PM3 is connected to the 5 V on the logic level converter. PM3 is a general purpose I/O pin on the Dragon12-LightBoard.

XBee and Arduino

The XBee is used to communicate wirelessly between the Dragon12-Light board and the Arduino. The XBee is able to communicate up to 300 ft away.

The XBee on the Dragon12-Light Board is responsible for transmitting data to the receiving XBee. The receiving XBee is connected to the Arduino which is used to manipulate the servo based on the data transmitted to it.

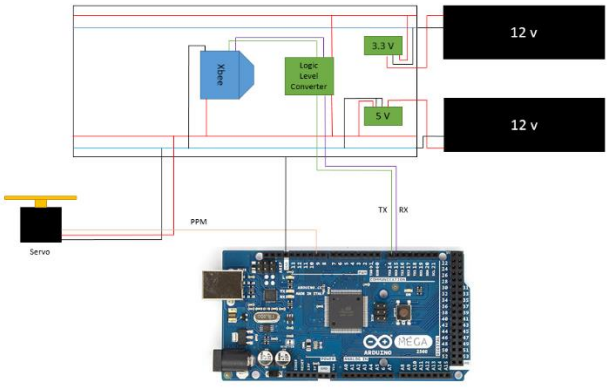

Figure 2

The Arduino manipulates the servo using pulse position modulation (PPM). This is different from pulse width modulation (PWM) which modulates the duty cycle. The servo library for the Arduino utilizes PPM in order produce servo pulses independent of the duty cycle determined only by the length of the pulse [1].

Data was transmitted one byte at a time to the Arduino, using a baud rate of 9600, and was observed using the Arduino serial monitor.

III. Experimental Step

Hardware

Hardware used to operate the project includes two wireless XBee's. Two Parallax XBee USB

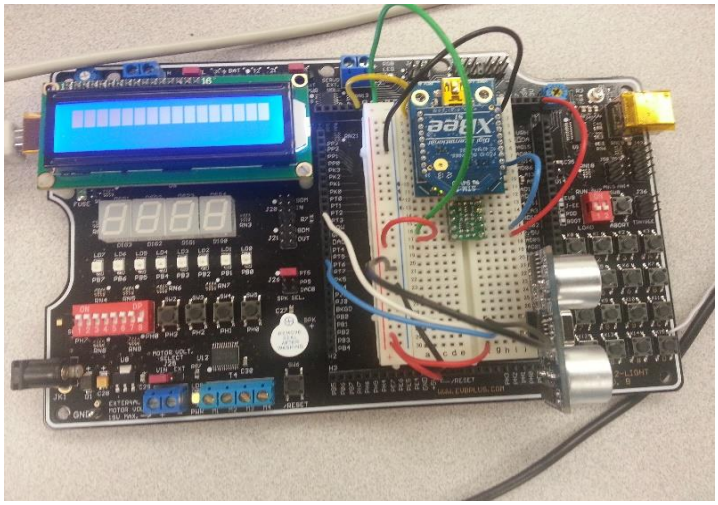

Figure 3

adapters were used to mount the XBee to the breadboard. The adapters made it easy to interface between the microcontroller and the XBee. The adapters included a 3.3 V down voltage regulator, allowing use of a 5 V source. Two 3.3 V to 5 V logic level converters were used between the XBee and the external data line source. To power the Arduino/XBee circuit, a 3.3 V and 5 V down voltage regulator were used with two 8 AA battery packs (12 V source).

The servo is powered by the Arduino/XBee circuit, requiring a 5 V source. The servo data line is connected to Pin 9 on the Arduino. The ultrasonic range finder used requires a 5 V source. The ECHO pin on the range finder is connected to PT6 on the Dragon12-Light Board and the TRIG pin on the range finder is connected to PT3.

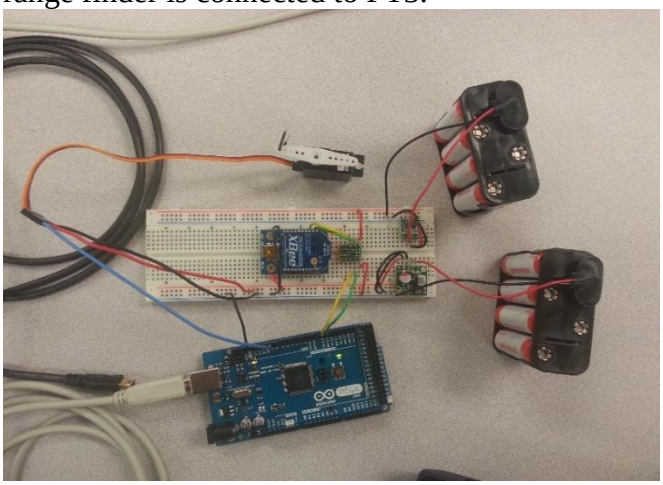

Figure 4

A limitation of the XBee is one mode can send one byte at a time, while the other mode can send one packet at a time. Only one mode can be used at a time.

Software

Software used includes the CodeWarrior IDE, Arduino software, and XCTU. CodeWarrior is a development environment that was used to program the Dragon12-LightBoard. XCTU is software platform developed by Digi to allow interaction with the XBee module.

In CodeWarrior, interrupts and timers were used to capture input from the ultrasonic range finder (Channel 6). Channel 3 was used as an output compare in TOGGLE mode. Since it was placed to capture on rising only, a width variable was introduced in order to calculate the difference between the edges on Channel 6. Width is then converted into centimeters and placed as the distance variable. The ASM function was then implemented in order to change the 16-bit distance into 8-bits. The flipped 8-bits is sent to the PM3 pin, a general purpose IO pin on the Dragon12-LightBoard.

The Arduino software is a specialized implementation of C that allows coding for the Arduino hardware.

IV. Results

The project was able to meet the defined objective. The Dragon12-Light Board was able to

receive information from the ultrasonic rangefinder and send information over the XBee wireless system. The Arduino was able to receive information from the Dragon12-Light Board and manipulate the servo.

Issues

Issues were encountered when using the ultrasonic rangefinder, XBee's, and other hardware.

The rangefinder was highly susceptible to noise while measuring distance. Since there was no easy way to resolve this issue, the reading was averaged over five values. To stop the servo from reacting too sporadically, a limiter was placed in code to prevent the difference between the last averaged reading and the current averaged reading from being too far apart.

The issue with XBee was interfacing to a general purpose I/O pin on the Dragon12-Light Board. Using a general purpose I/O pin proved to be a large issue, since the least significant bits were never sent. The loss of data caused inaccuracies in the data sent to the Arduino to control the servo.

If the project had started with the use of an actual serial communication port, the design of the project would have generated more accurate data as well as allowed more effective control over the servo. Occasionally, the ground line between the Arduino and breadboard would be forgotten when testing the servo. This was remedied fairly quickly when it was noticed that the servo did not respond appropriately to the given commands. The ground was required for the PWM channel on the Arduino to work appropriately.

The inefficient and often problematic communications had been unexplainable during the project and was not noticed early on in the project. After spending considerable time on error checking, the discovery was that the general purpose I/O pin was not sending the least significant bits.

V. Conclusion

The project met the objectives. Given more time and finances, significant improvements could have been made. The most important improvement would have been getting a serial adapter to establish a serial connection between the Dragon12-LightBoard and Arduino. Another idea for the project was to use an LED ring that would change colors based on the servo's position. An ultrasonic rangefinder that was less susceptible to noise would have improved the project significantly.

Overall, the project accomplished manipulation of the servo based on a wireless communication system. The project utilized interrupts, a timer module, and assembly functions that were learned in class.

References

[1] B. Evans, *Beginning Arduino Programming*. New York: Apress, 2011, pp. 156-157.