**5/6/2015**

**ELTN 117 – Survey of Digital Logic and Microcontrollers**

**Final Laboratory Project: Photo-Deflector Rotary Encoder**

**Luis Gomez**

**Instructor: Thomas Thoen**

**School of Business, Engineering and Technology**

**Pasadena City College**

**Pasadena, CA**

Table of Contents

[Introduction 1](#_Toc417287609)

[Hardware Description 2](#_Toc417287613)

[Software 3](#_Toc417287614)

[Libraries / Functions 3](#_Toc417287615)

[Budget / BOM 5](#_Toc417287616)

[What was learned / Conclusion 6](#_Toc417287617)

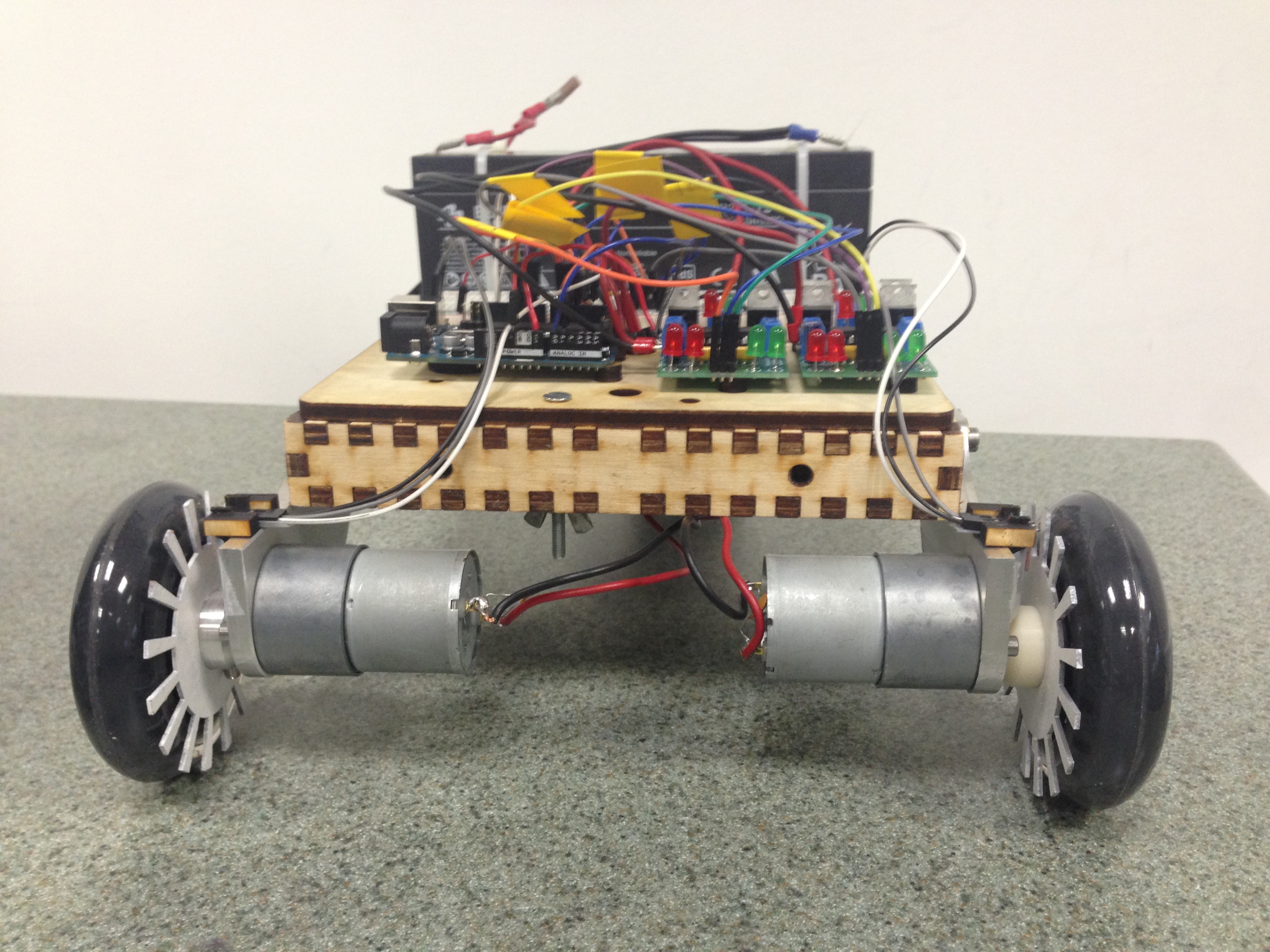
# Introduction

**The purpose of this project was to create an inexpensive Arduino compatible rotary encoder system that could be mounted on wheeled robots for accurate linear movement. For the project a spoked disc, coated in glossed white primer, was fixed onto a motorized wheel assembly within range of a photo-deflector. The purpose of the photo-deflector was to detect the passing of individual spokes as the wheel rotated, and to communicate this as both an analog & digital signal to a microcontroller directing the robot’s movement.**

**For this project, 2 photo-deflectors were mounted on opposing wheels. I designed & 2-D laser cut a mount assembled from ¼ inch birch plywood, on which the photo-deflectors were fixed onto. The ¼ inch birch plywood was inexpensive & sufficiently robust, while also protecting the sensors from contact with conductive surfaces on the robot. Two spoked discs were 2-D laser cut using 1/16 inch stock-board, an inexpensive material, and coated in white glossy primer paint for greater contrast & visibility. A Schmitt trigger was used to filter signals from the photo-deflector, as the interrupt functions in the code required a stable signal for reliable triggering.**

**This project was a continuation of an ongoing project from last semester. The previous semester’s project was the completion of a directional robot, on which the rotary encoder was mounted.**

**The software is built upon the work of my instructor Thomas Thoen, I incorporated the coding for the rotary encoder into his directional movement program, adding a few nuances along the way. The major additions included a user interface for directional commands, the creation of interrupt service routines for the sensors, and a pulse-count system recording and regulating wheel rotation at various speeds.**

****

# Hardware Description

The block diagram below shows the basic subsystems used in the system:

Right Photo-Deflector Sensor

Left Wheel Spoked Disc Assembly

Right Wheel Spoked Disc Assembly

Left Photo-Deflector Sensor

Right Wheel DC Motor

Left Wheel DC Motor

Right H-Bridge

Left H-Bridge

ARDUINO UNO

Schmitt Trigger

# Software

Reset Parameters & Hardware Interrupts

Compare Rotation State of Wheels

Prompt & Record User Directional Commands

Power DC Motors & Move Robot

Left Wheel Sensor ISR & Record Wheel State

Right Wheel Sensor ISR & Record Wheel State

**Figure 2**. Top Down diagram of program

## Functions

No additional libraries were used in the programming of this encoder system. A number of different functions were created to record and compare wheel rotation states, trigger sensor interrupts, to interface with users via the serial monitor, and to verify that users provided acceptable parameters. Chief among these functions are two portions of code. The first is the interrupt service routine responsible for counting pulses along a wheels rotation & triggering a pulse comparison, while the second is responsible for comparing the rotation states of wheels & displaying relevant data indicating rotational progress,

void rwheelSensor() // Intterrupt Service Routine

{

rwheelPulse++; // Increments Wheel Pulse by one when triggered

Rflag = 1; // Flags wheel as having been triggered, triggering a

// pulse comparison in main loop

}

**Figure 3** – Sensor Interrupt Service Routine

if (rwheelPulse<=lwheelPulse) analogWrite(rightOut, speed);

// Send values to H-bridges

if (lwheelPulse<=rwheelPulse) analogWrite(leftOut, speed);

if(Rflag == 1||Lflag == 1)

{

//This section contains serial print commands for displaying progress of rotation, triggered flags, pulse counts, analog sensor outputs

Lflag = 0;

Rflag = 0;

}

loopCount++;

}

// This segment runs at conclusion of loop, displaying relevant data such as pulse counts, loop count, triggered flags

speed = 255;

analogWrite(leftOut, speed);

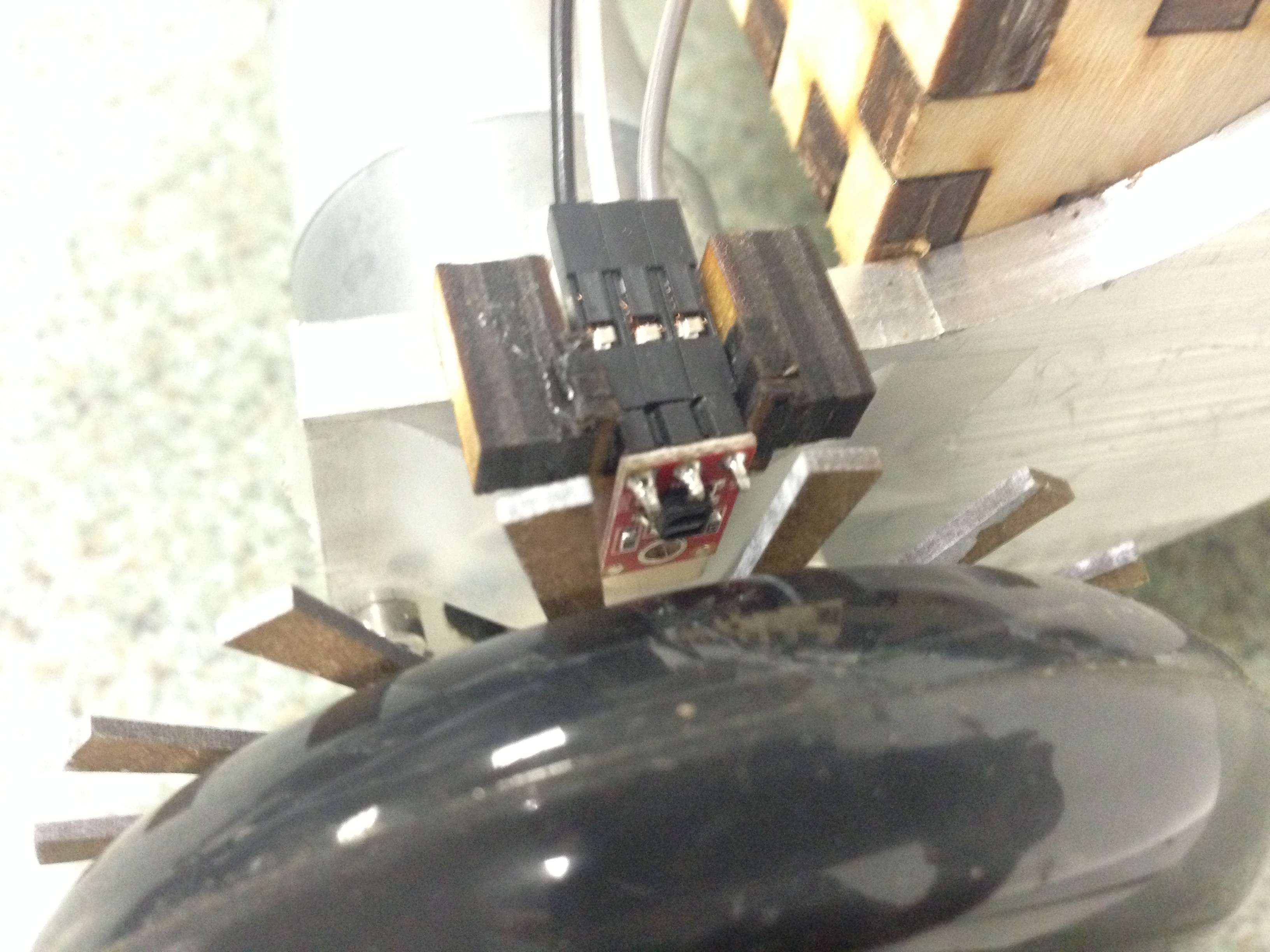
analogWrite(rightOut, speed);

}

**Figure 4** – Abbreviated version of Pulse Count Comparison Function

# Budget / BOM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Line item | Quantity | Description | Part number | Cost | Extended Cost |
|  |  |  |  |  |  |
| 1 | 1 | Frame/Mount Material | 4X8 ¼” Birch Plywood | $30 | $30 |
| 2 | 2 | Photo-Deflectors |  | $0 | $0 |
| 3 | 1 | Spoked Disc Material | 2X2 1/16” Stock Board | $5 | $5 |
| 4 | 2 | Schmitt Trigger | CD74HC14E | $0 | $0 |
| 5 | 1 | Spoked Disc Paint | Glossy White Spray Paint | $10 | 10 |
| 6 | 1 | Jumper Cables | M/M Heads | $5 | $5 |
| 7 | 1 | Development Board | Arduino UNO | $27 | $27 |
|  |  |  | Total |  | $77 |

**Figure 5**: Bill of Materials w/ costs

# What was learned / Conclusion

To a degree, the project was a success. My initial hopes for the project and the work I could complete in a reasonable amount of time were a bit too ambitious for the resources at my disposal but nonetheless; I created a functional rotary encoder for my robot. I had initially planned to assemble a robot with simple articulated limbs, but as costs began to increase for the project it became clear that such an assembly would not be possible.

In the process of planning the project, it became clear that there were structural deficiencies with the robot I had planned to incorporate the encoder onto. The poorly welded steel frame the robot was built upon was misaligned and caused the wheels to wobble, preventing accurate & consistent readings from the light sensor subsystem. This led me to redesign the frame using ¼” plywood, a somewhat inexpensive material that successfully replaced the steel frame and aligned the wheels properly with the sensors. This phase led me to further develop familiarity with computer aided design software, which proved instrumental in the design of the sensor mount & spoked disc later on in the project.

Prior to all this, I had begun working on the code for the project. I built upon the directional movement code provided by my robotics instructor Prof. Thomas Thoen. Within a week, I had completed what in my mind amounted to a functional program for the interrupts; however, after completing the assembly of critical hardware components, testing the code led to my discovery that the interrupts were not working properly. The sensors were much too sensitive, triggering the ISR continuously. I tested delays both within the main loop and ISR to no avail. Prof Thoen successfully proposed the use of a Schmitt trigger to filter the signals from the sensors, which fixed the problem of sensor hypersensitivity outright.

Another problem arose however; when one wheel failed to trigger the ISR in a similar manner as before the installation of Schmitt trigger. This wheel was skipping pulses, and this wheel was also assembled using a weak 3-D printed component causing the wheel to wobble. Unfortunately, access to the machine shop is limited, so I relied on the fabrication of another 3-D component to replace the faulty one. This component however was completely solid, and more robust for that matter. This ultimately worked and allowed me to test my final rendition of program, and concluding this project with its completion.

Time & money were the most influential & scarce resources