NORTHERN ARIZONA UNIVERSITY

To: Dr. David Willy

From: Martin Dorantes

Date: September 3, 2021

Re: Self-Learning Assignment

Introduction

Engineers are always learning new concepts to adapt to new problems. It is crucial to an engineer's success to adapt as new obstacles come in the path of a project. This memo contains practice and application of a self-taught skill regarding safety to the HPVC capstone project. Two Udemy courses were used to guide the practice and YouTube videos were used to guide the application.

The team is building a human-powered vehicle (HPV) for small kids, from kindergarten to 8th grade, to inspire young students to pursue an engineering career path in the future. Safety is the team's top priority, and the Arduino application from this self-learning assignment is meant to aid in more safety for the driver regarding speed limits. The team is designing the HPV to go no more than 20 mph to avoid injury in the case a situation goes south. Adding a speedometer to the HPV will visually show the driver the speed of the HPV and will blink with a message to slow down if the driver exceeds 20 mph. Cline Library and Professor Willy provided most of the hardware required for the project. Hardware used for the project includes: 16x02 LCD display, hall effect sensor, Arduino UNO R3, and jumper wires.

New Skill

Introduction to Embedded Systems was the first enrolled course from Udemy. This course teaches embedded control of the Arduino and using digital and analog inputs/outputs. This course was used to familiarize the student with digital input, serial communication, and transmitting and receiving data.

Arduino Bootcamp: Learning Through Projects was the second enrolled course from Udemy. This course teaches hands-on, project-based concepts with real-world applications. This course was used to familiarize the student with sensor usage and how to display data on an LCD screen. Figure 1 shows completion certificates validating the completion of both courses.

YouTube was another resource used to prompt the LCD screen to display speed and using a hall effect sensor. The video referenced is a basic tutorial for how hall effect sensors work and how to properly connect to Arduino and include in the code. [1] The hall effect sensor was used as a tachometer to measure rpm, which can be rewritten as speed in mph if circumference of the wheel is known. Speed output in mph was displayed using Equation 1 and included in the code to display speed in mph.

$$Speed = \frac{\pi * d * RPM * 60}{5280} \quad (1)$$

Figure 1 – Course completion certificates

Application

Upon course completions, the new skills were applied to create a prototype tachometer. Once the LCD would display the rpm of the spinning wheel, the code was rewritten with Equation 1 to display speed of travel in mph, where d is the diameter of the tire and RPM is the output from the sensor to get a speed in mph. Figure 2 shows the Arduino setup using jumper wires, LCD display, and hall effect sensor. The sensor will hang at the top of the wheel to sense an attached magnet on the rim of the wheel. Every revolution of the tire will be picked up by the hall effect sensor and the calculated speed will be shown on the LCD display, anything over 20 mph will prompt the driver to slow down back to below 20 mph. Appendix A shows the code for the HPV speedometer. For the prototype, a makeshift wheel was made by taping a magnet on the inside of a plastic cup. The diameter of the cup was included when the code was written to obtain an accurate measure of speed.

The active speedometer is shown in Figure 3 with the indication that a speed over 20 mph is detected. Figure 4 shows the final steps to power the speedometer with a 9V battery prior to attaching on the HPV. Creating an enclosure for the speedometer would help with aesthetics of the wiring and connections. Longer wire connections are required to send the data back to the LCD display at the handlebar of the HPV.

NORTHERN ARIZONA UNIVERSITY

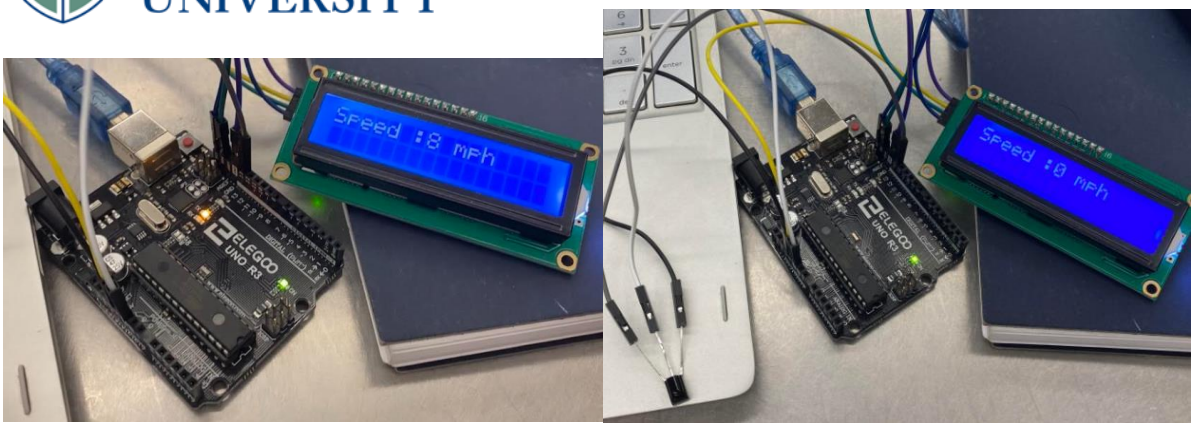

Figure 2 – Active & Inactive Speedometer Schematic

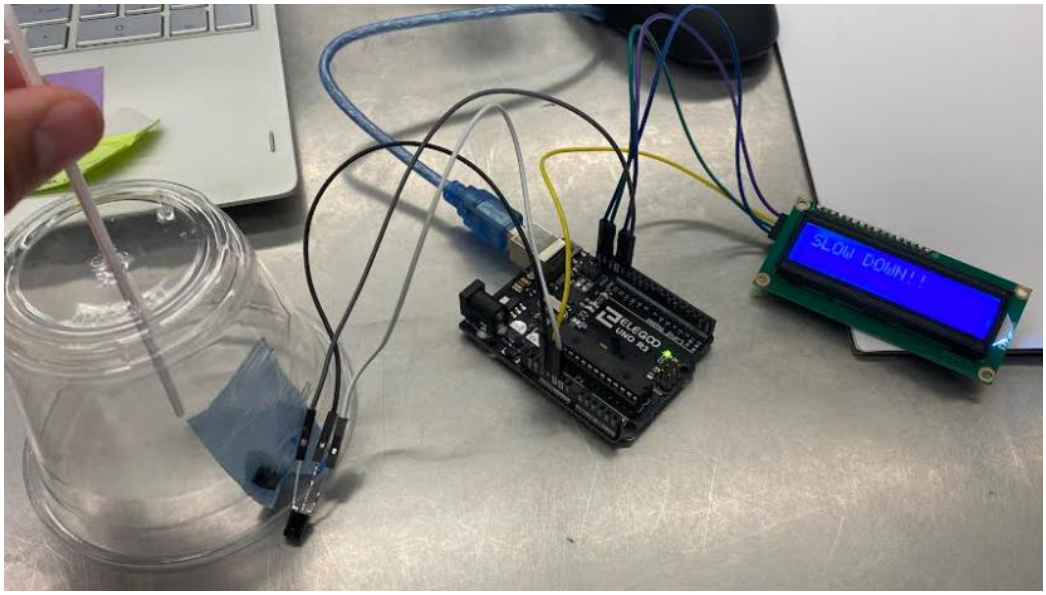

Figure 3 – Speedometer Prototype with Active Warning

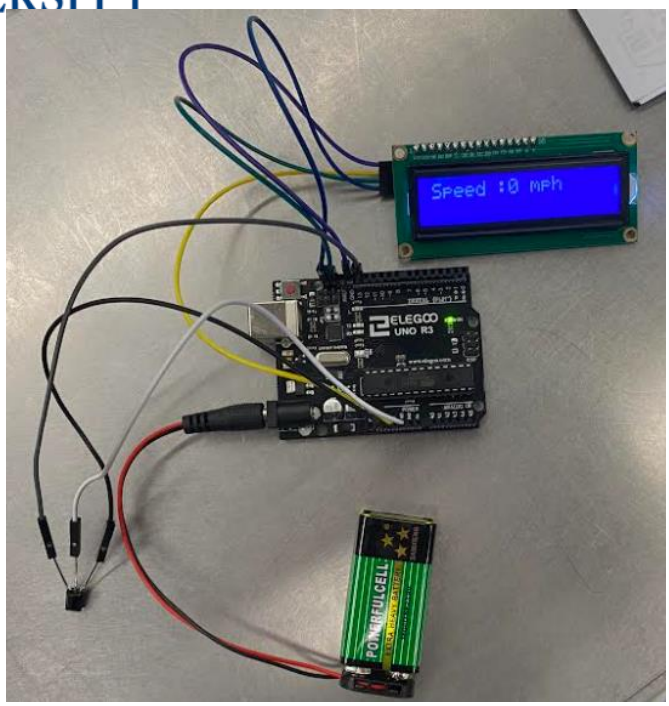

Figure 4 – Battery-Powered Speedometer Prototype

Conclusion

It is crucial that the operators are kept safe while driving the HPV. Adding the speedometer with a warning is meant to keep the young students safe. Showing children that there are restrictions, like speed limits, while operating a vehicle will help familiarize them when the time comes to operate an actual car on the streets. Using the LCD screen helps with aesthetics to captivate more inspiration to pursue an education in engineering within younger students.

NORTHERN ARIZONA UNIVERSITY

References

[1] educ8s.tv, “Arduino Tutorial: Hall Effect Sensor from Banggood.com and Arduino Uno,” *YouTube*, Nov. 14, 2015. [Video Recording]. Available: https://www.youtube.com/watch?v=4eqi0G7uY_4. [Accessed: 2-Sep-2021].

NORTHERN ARIZONA UNIVERSITY

Appendix A – Speedometer Arduino Code

```
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
#define DISPLAY_W 16

LiquidCrystal_I2C lcd(0x27,DISPLAY_W,2);

float revolutions=0;
int rpm=0;
int spd=0;
diam=0.4                                    //EDIT TO CURRENT DIAMETER IN FEET
long  startTime=0;
long  elapsedTime;

void setup()
{
   pinMode(2, INPUT_PULLUP);
   lcd.init();
   lcd.backlight();
   lcd.setCursor(0,0);
   lcd.print("Speed :");
   Serial.begin(9600);
}

void loop() {
revolutions=0; rpm=0; spd=0;
startTime=millis();
attachInterrupt(digitalPinToInterrupt(2),interruptFunction,RISING);
delay(1000);
detachInterrupt(2);

elapsedTime=millis()-startTime;     //finds the time

if(revolutions>0)
{
 rpm=(max(1, revolutions) * 60000) / elapsedTime;       // rpm calc
 spd=(3.1415*diam*rpm*60)/5280;                         // speed calc from rpm

}


lcd.setCursor(0,0);
if(spd>20)                                              // if speed goes over 20 mph
  {
   String slowmsg = String("SLOW DOWN!!");
   fillMessage2DisplayWidth(slowmsg);
   lcd.print(slowmsg);
   Serial.println(slowmsg);
```

NORTHERN ARIZONA UNIVERSITY

```
 }
String outtMsg = String("Speed :") + spd + " mph";
fillMessage2DisplayWidth(outtMsg);
lcd.print(outtMsg);
Serial.println(outtMsg);

}
```