Dumptruck’s Mechatronics Lab Report 4

02.25.2022

Nick Morse

Ryan Colón

**Summary:**

The goal of lab 4 was to explore how to use the Analog to Digital converters on the ATMega 2560. In this lab, we used the GP2D12 and GP2Y0A21 IR ranging sensors, which output an analog voltage dependent upon the distance to the object. We then converted the voltage measurement to a measurement of distance in cm and displayed that to an LCD. 5 V at the AREF pin was used as the reference for the AtoD and we decided to use Analog channel 7 just because it was as close as we could get to our breadboard.

**First Design:**

The initial design worked partially, but the sensor was very finicky and difficult to work with. It would sometimes disconnect and crash the program or take erroneous measurements and a lot of times this would easily be fixed by giving the sensor a soft tap. We took various measurements of distances and voltages and tested the various formulas that emerged as trendlines from those measurements. Most of them were fairly accurate at close proximity, however when we moved an object further away, the readings would stop at around 40 cm.

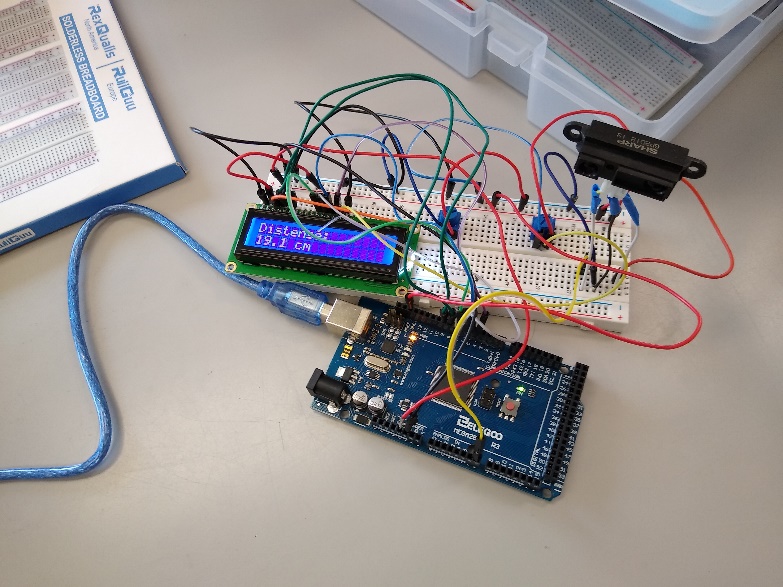


Figure Circuit of First Design

Also, occasionally a random very small distance would be shown. It appeared that the sensor was not completely functional. In the second design, the GP2Y0A21 was used. Imaged on the next page is the circuit diagram for this design.

**Block Diagram:**

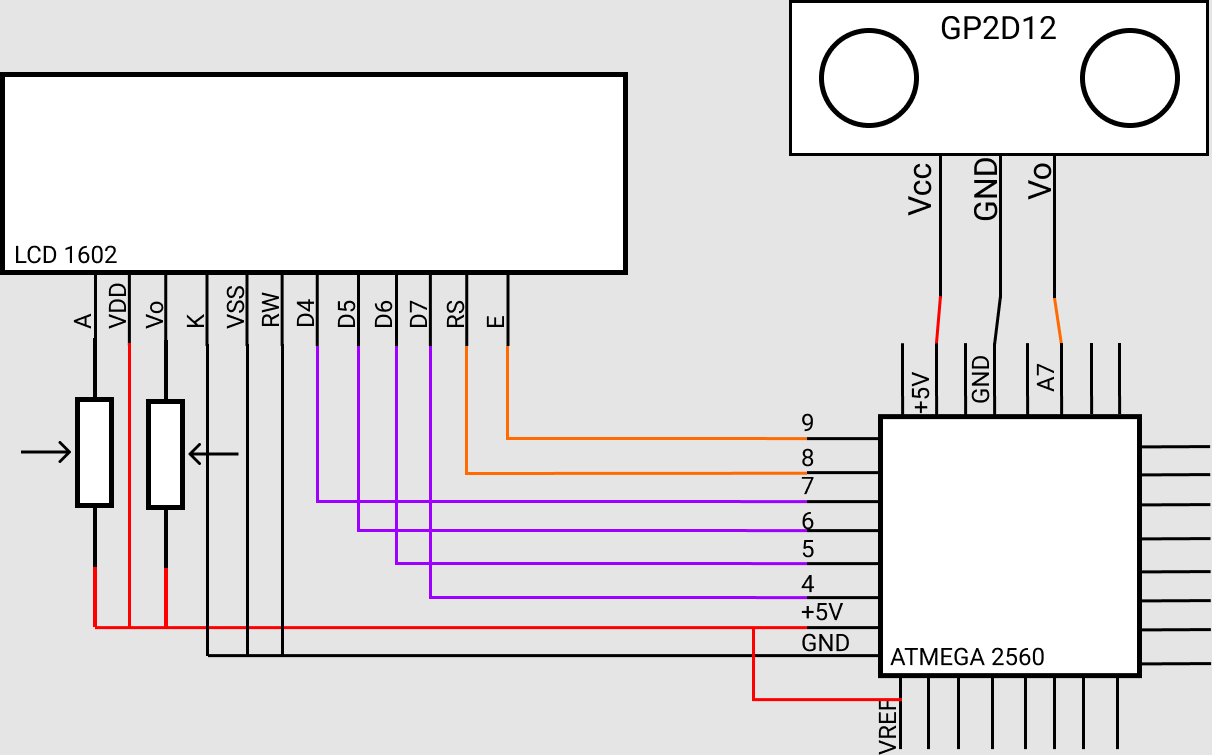


Figure Block Diagram

**Second Design:**

For this design, the same circuit topology was used, we merely replaced the GP2D12 with the GP2Y0A21. Testing this circuit showed that the sensor worked more consistently than the previous one, although it still gave rough measurements. The accuracy of this design is closer to +- 0.75 cm than +- 0.5 cm, but it was 100% accurate down to 1 cm. We also had the same issue of total range, where this one was accurate down to 6 cm but could only go up to ~36 cm. This sensor would also at random times give low measurements less than 10 cm, and sometimes it wouldn’t. This design was the one showcased in the video submitted with the report.

Since we were using a more recent and better supported sensor, this program was compared to using a sensor library available from SharpIR for Arduino. Testing the performance of this library showed the exact same behavior as our program, leading to the conclusion that this is most likely an issue with our overall program (unrelated to the sensor) or is an issue with the way we are using the sensor itself. The library is SharpIR-2.0.1 and can be found at the link below:

https://www.arduinolibraries.info/libraries/sharp-ir

**Code:**

/\*

Intelligent Machines Lab 4 Distance Measuring and The Screams of the Damned

Ryan Colon & Nick Morse

02.24.22

\*/

#include <LiquidCrystal.h>

#include <stdlib.h>

const int AREF = 5;

const int NumLevel = 1023;

const int rs = 8, en = 9, LCDd4 = 7, LCDd5 = 6, LCDd6 = 5, LCDd7 = 4;

LiquidCrystal lcd(rs, en, LCDd4, LCDd5, LCDd6, LCDd7);

void setup() {

//Setup the LCD

lcd.begin(16,2);

lcd.clear();

Serial.begin(9600);

//Setup the ADC

ADMUX = 0x47; //Select ADC channel 7, right justified, AREF = 5V @ AVCC

ADCSRA = 0x94; //Dont Convert, Don't Autotrigger, Clear flag, Interrupt Disabled

ADCSRB = 0;

}

void loop() {

//Acquire Voltage

ADCSRA |= 0x40; //Start reading

while(!(ADCSRA & 0x10));//wait for read to finish

int adcRead = ADC; //read value

float cm = 4800 / (float)(adcRead - 20);

//Print to LCD

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Distense:");

lcd.setCursor(0,1);

lcd.print(cm, 0);

lcd.print(" cm");

delay(500);

}