Lab 04

Analog To Digital Converter

Logan Barber, Ian Nail

February 25, 2022

Ian Nail

Logan Barber

ME-4370 - Stephen Canfield

1 Executive Summary

In this lab, the objective is to use the Arduino Mega 2560 to read in analog values from an infrared distance sensor. A program was written to convert the analog values to the distance in centimeters. To determine an equation that will successfully convert the raw analog values to a distance, distance and analog value measurements were taken. The data was plotted into Excel where a line of best fit was determined.

The minimum objectives were:

1. Application must use Sharp GP IR ranger with analog output
2. Must demonstrated a resolution of .5 cm or better
3. Must display measured distance in cm (best) or inches
4. Must have some output that makes use of the range sensor information
5. Implement and demonstrate in a product application
6. Demo to lab assistant, instructor or other person with mechatronics background, document in your report.
7. Test your product on someone outside the class, record their use of your product, discuss outcomes in your report.

2 Analog to Digital Conversion

To create a function to convert the raw data from the analog input a bunch of data points were measured. The IR sensor was setup and the input raw input was read from the serial monitor. 14 different distances and points for taken. This data was placed into Excel. A scatter plot was made from the points.

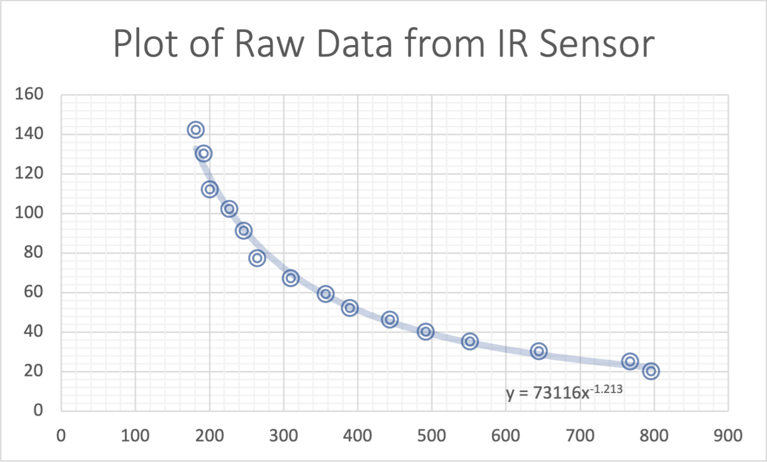

Figure 1: Plot of Points from Raw ADC data

The equation in the figure is a line fit power equation from Excel. This equation was used in the c code to fit change the sensor data into a centimeter measurement.

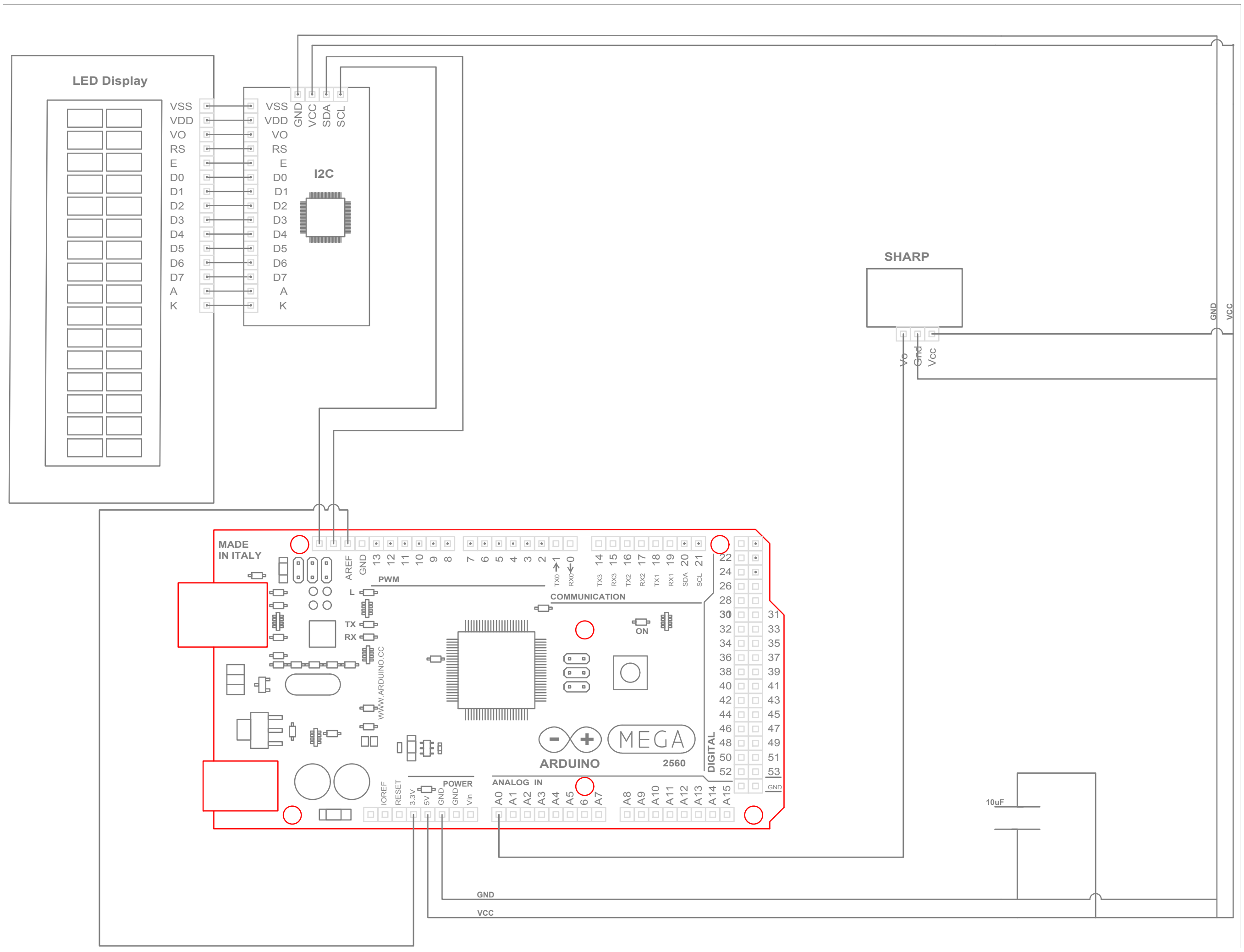

Figure 2: Circuit Diagram

3 Source Code

```
/* AUTHORS: A. LOGAN BARBER; IAN NAIL
 * FILE NAME: Lab04.ino
 * LAST UPDATED: 25 February 2022
 *
 * PURPOSE: THIS IS THE MAIN FILE FOR TAKING AN ANOLOG INPUT FROM A IR DISTANCE
   SENSOR AND DISPLAYING THE DISTANCE ON THE LCD.
 * The sensor is an SHARP 2Y0A02 F 18
 * Measuring distance: 20 to 150 cm
 * Matlab code for the measurements of IRsensor
 * IRsensor = [1/796 1/768 1/645 1/552 1/492 1/444 1/390 1/357 1/310 1/265 1/247
    1/227 1/201 1/193 1/182];
 * distance = [20 25 30 35 40 46 52 59 67 77 91 102 112 130 142];
 *
 */
// INCLUDE LIBRARIES
#include <LiquidCrystal_I2C.h>
#include <stdio.h>
#include <math.h>
#define TRUE 0x01
#define FALSE 0x00


// Global Variables
float value_ADC0 = 0;
float Distance_ADC0 = 0;
uint32_t low = 0;
uint32_t high = 0;

// DEFINE MACROS FOR LCD SERIAL
#define ADDRESS 0x27
#define LCDCOLS 16
#define LCDROWS 2
// CREATE LiquidCrystal OBJECT
LiquidCrystal_I2C lcd(ADDRESS, LCDCOLS, LCDROWS);

void setup() {

// Setup ATD:
// We used external AVREF as ADC reference with 3.3V wired to the pin, 10 bit
   reading, Set MUX 4-0 as 0 for ADC pin 0
  ADMUX = 0b00000000;
// Enable ADC
// Don't start conversions yet
// DOn't autrigger, clear flag, Dont enable interrupt
// Prescalers are 000 for divide by 2 prescale
  ADCSRA = 0b10010000;
  ADCSRB = 0b00000000; // for ADC0
// Begin serial and confirmation message
  Serial.begin(9600); // init serial
  Serial.println("Serial Connected");

    // INITIALIZE THE LCD SCREEN
  lcd.begin();
  // turn on the backlight
  lcd.backlight();
}
```

```
void loop() {
 value_ADC0 = average_Distance(); // Gets the average value from 15000 samples
 Distance_ADC0 = 73116*pow(value_ADC0,-1.213) + 6; // equation calculated in excel

 //Distance_ADC0 = 26632.2/value_ADC0 - 14.3366 + 10;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Distance ");
  lcd.print(Distance_ADC0);
  delay(200);
}

/*
 * TYPE: FUNCTION
 * NAME: average_Distance
 * RETURN: uint32_t
 * NUMBER OF PARAMETERS: 0
 * PARAMETER NAMES: void
 * PURPOSE: This function returns the average reading from the IR sensor from 15000 samples
 */
uint32_t average_Distance(){
  uint32_t u32_index;
  uint32_t u32_average_distance = 0;
  for(u32_index = 0; u32_index < 15000; u32_index++) {
    ADCSRA |= 0b01000000; // Start ADC Conversion
    while((ADCSRA & 0b00010000)==0); // Stays in while loop while conversion is happening
    low  = ADCL;
    high = ADCH;
    value_ADC0 = (high << 8) | low;
    u32_average_distance = u32_average_distance + value_ADC0;
  }
  u32_average_distance = u32_average_distance/15000;
return u32_average_distance;
}
```

../Lab04.ino