Security PUFfins
Overview Presentation

GTA: Jordan Beverly Client: Julie H.

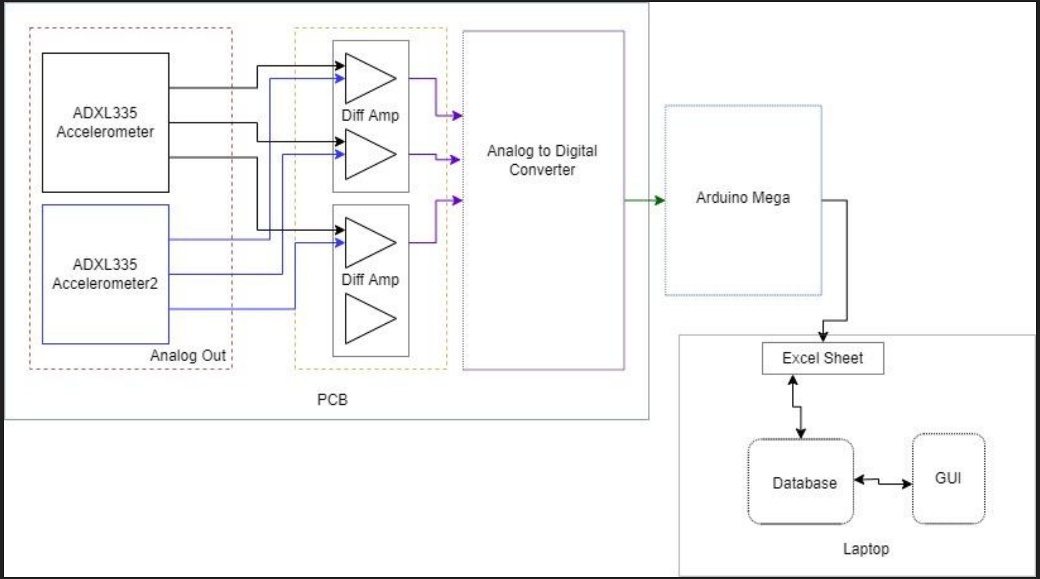
Benjamin Assmann, Sharley Fabro, and Traigh Kirkeeng

09 September, 2022

Overview

- Create four individual Physically Unclonable Functions(PUFs) from four individual circuits using sensor pairs of:
    - Accelerometers
    - Magnetometers
    - Gyroscopes
    - Current Sensors
- Amplify the difference between sensor pair readings due to their natural manufacturing differences
- Amplifying the changes creates a PUF as the difference is specific to the pair of sensors creating a sort of security code that tracks what value difference should be expected for the pair.

Project design

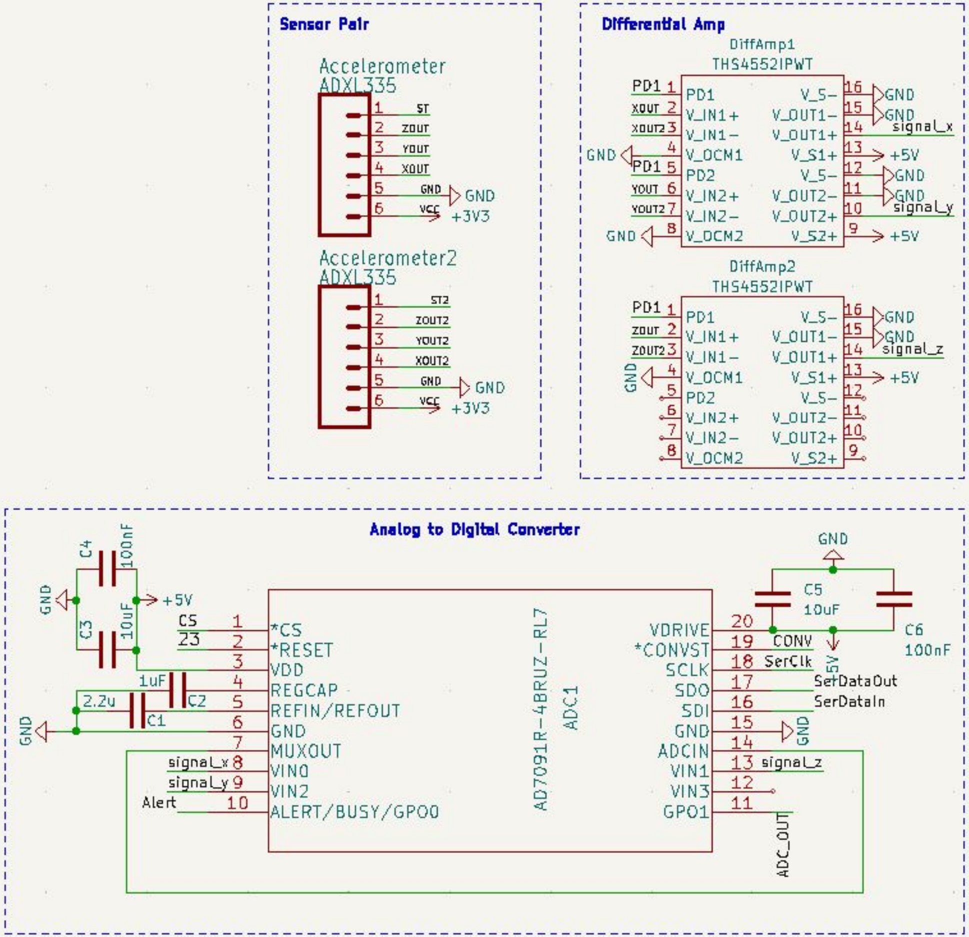


- Have arduino mega power, control and send/receive data for circuit
- Send data to excel file to be imported into database
- Create graphic user interface to implement and display data without using database

Previous Work

PCB Circuit Design

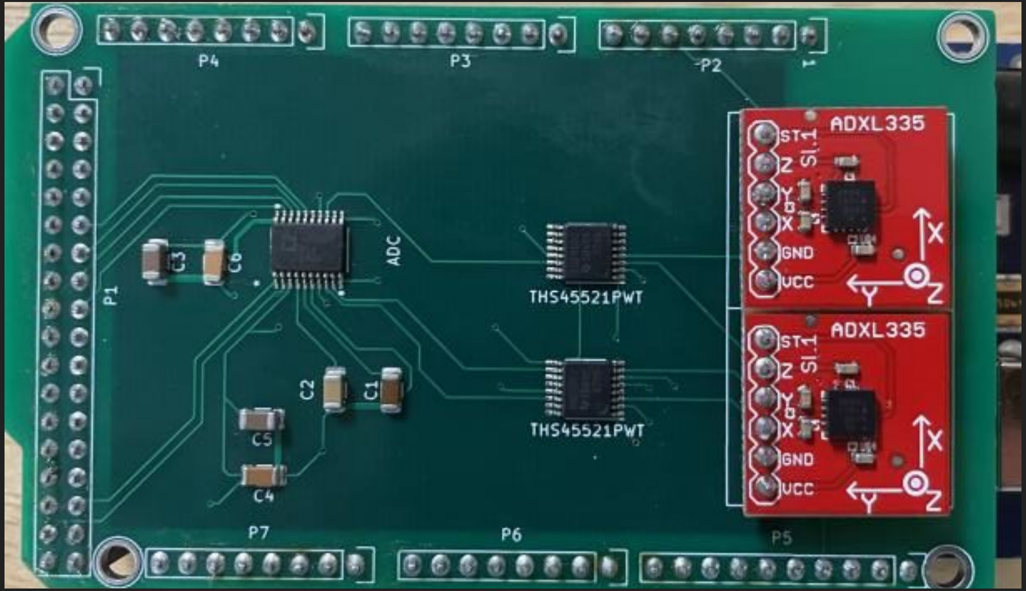
- The accelerometer is supplied 3.3 V
- X,Y, and Z out from sensor 1 and 2 are connected into Diff Amp inputs
- 5V supply is routed to the Diff Amps and ADC
- Enables for the Diff Amps are connected together into one toggle
- ADC outs are passed into the Arduino Mega
- All capacitors stabilize the supply voltage to the ADC



Circuit Design

- Arduino reads values and processes them
- Outputs a Security PUF to computer
- Arduino connected to PCB via female-male header pins noise control
- Through testing it was found that the Arduino 5V supply was not reliable enough for the entire PCB

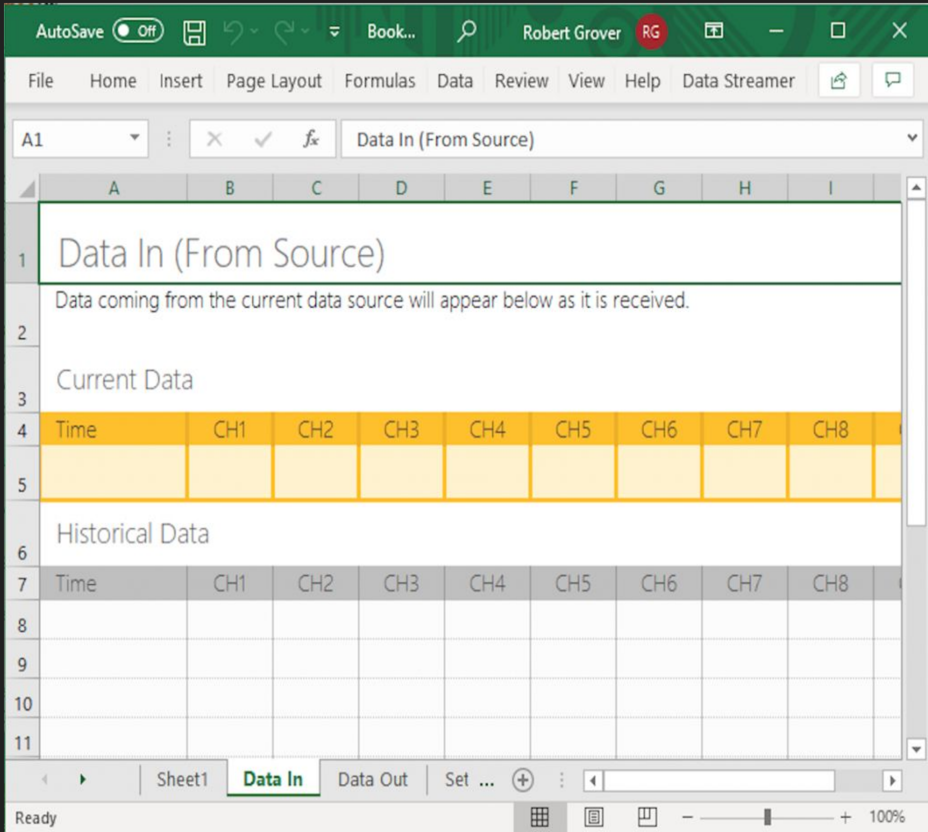
Our Printed Circuit



Arduino Code and Output

- Arduino code is simple for readability
- Uses one pin for power, ground, and 6 pins for reading analog input
- PUF is created by taking difference of two, same axis readings
- Outputs to serial monitor and to data streamer for excel
- Formatted excel spreadsheet for input of data to database

Data Streamer example



Data In (From Source)

Data In (From Source)

Data coming from the current data source will appear below as it is received.

Current Data

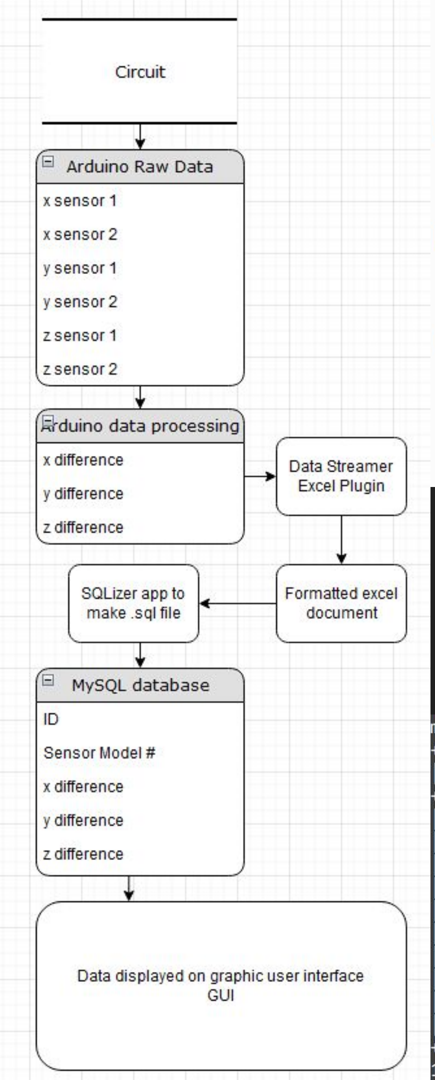
Time	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8

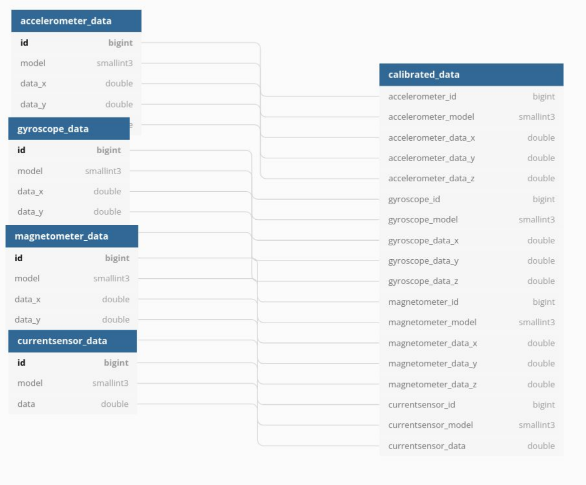
Historical Data

Time	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8

Database

- Originally used MySQL database using C++ connector, though licensing changed
- Instead using Microsoft SQL server 2019
- Export data from arduino to excel sheet to import into database
- Check data, go through the calibration process, and union to calibrated data
- Support for large datasets built into functionality





accelerometer_data	
id	bigint
model	smallint3
data_x	double
data_y	double

gyroscope_data	
id	bigint
model	smallint3
data_x	double
data_y	double

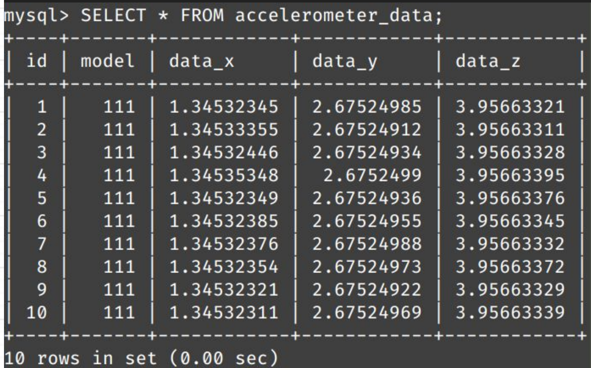
magnetometer_data	
id	bigint
model	smallint3
data_x	double
data_y	double

currentsensor_data	
id	bigint
model	smallint3
data	double

calibrated_data	
accelerometer_id	bigint
accelerometer_model	smallint3
accelerometer_data_x	double
accelerometer_data_y	double
accelerometer_data_z	double
gyroscope_id	bigint
gyroscope_model	smallint3
gyroscope_data_x	double
gyroscope_data_y	double
gyroscope_data_z	double
magnetometer_id	bigint
magnetometer_model	smallint3
magnetometer_data_x	double
magnetometer_data_y	double
magnetometer_data_z	double
currentsensor_id	bigint
currentsensor_model	smallint3
currentsensor_data	double

MySQL Schema

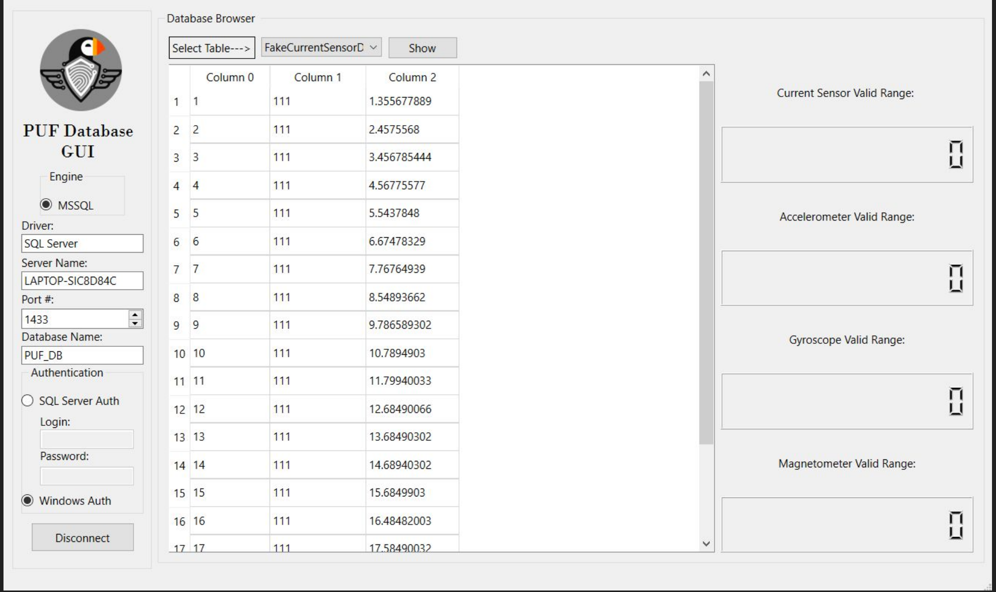
Example of data in database



```
mysql> SELECT * FROM accelerometer_data;
+----+-------+------------+------------+------------+
| id | model | data_x     | data_y     | data_z     |
+----+-------+------------+------------+------------+
|  1 |   111 | 1.34532345 | 2.67524985 | 3.95663321 |
|  2 |   111 | 1.34533355 | 2.67524912 | 3.95663311 |
|  3 |   111 | 1.34532446 | 2.67524934 | 3.95663328 |
|  4 |   111 | 1.34535348 |  2.6752499 | 3.95663395 |
|  5 |   111 | 1.34532349 | 2.67524936 | 3.95663376 |
|  6 |   111 | 1.34532385 | 2.67524955 | 3.95663345 |
|  7 |   111 | 1.34532376 | 2.67524988 | 3.95663332 |
|  8 |   111 | 1.34532354 | 2.67524973 | 3.95663372 |
|  9 |   111 | 1.34532321 | 2.67524922 | 3.95663329 |
| 10 |   111 | 1.34532311 | 2.67524969 | 3.95663339 |
+----+-------+------------+------------+------------+
10 rows in set (0.00 sec)
```

Graphic User Interface(GUI)

- Originally created GUI using C++ gtkmm and gtkmm-plot libraries
- Due to QT6 being available under special terms for the project, switched backend
- Easy access to at least last 100 data points of calibrated data for each sensor in table form
- Fetch data to find statistics
- Have no reason to access database manually besides adding and removing data



Engine
● MSSQL

Driver: SQL Server

Server Name: LAPTOP-SIC8D84C

Port #: 1433

Database Name: PUF_DB

Authentication
○ SQL Server Auth
Login:
Password:
● Windows Auth

Select Table---> FakeCurrentSensorD

	Column 0	Column 1	Column 2
1	1	111	1.355677889
2	2	111	2.4575568
3	3	111	3.456785444
4	4	111	4.56775577
5	5	111	5.5437848
6	6	111	6.67478329
7	7	111	7.76764939
8	8	111	8.54893662
9	9	111	9.786589302
10	10	111	10.7894903
11	11	111	11.79940033
12	12	111	12.68490066
13	13	111	13.68490302
14	14	111	14.68940302
15	15	111	15.6849903
16	16	111	16.48482003
17	17	111	17.58490032

Current Sensor Valid Range: 0

Accelerometer Valid Range: 0

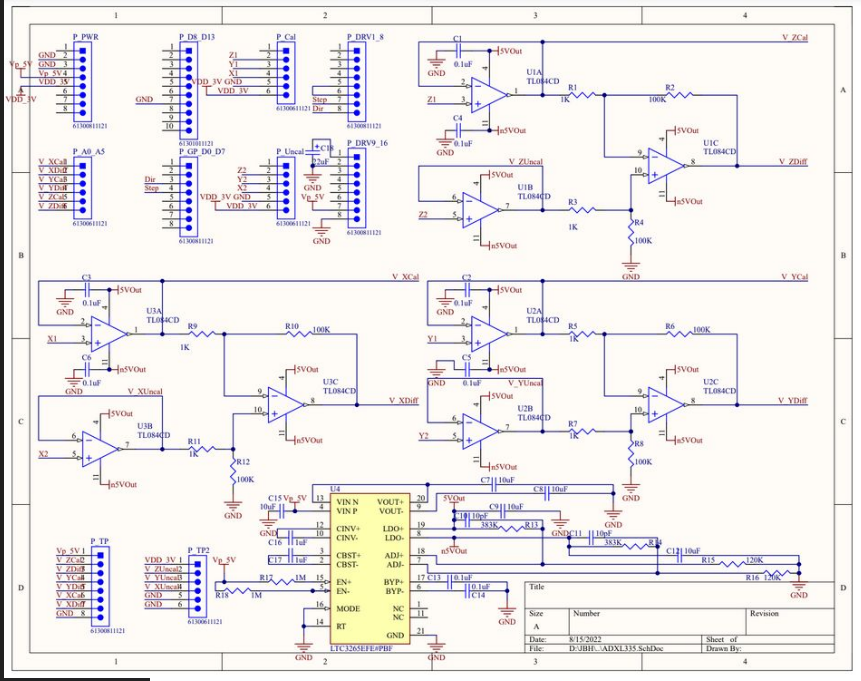
Gyroscope Valid Range: 0

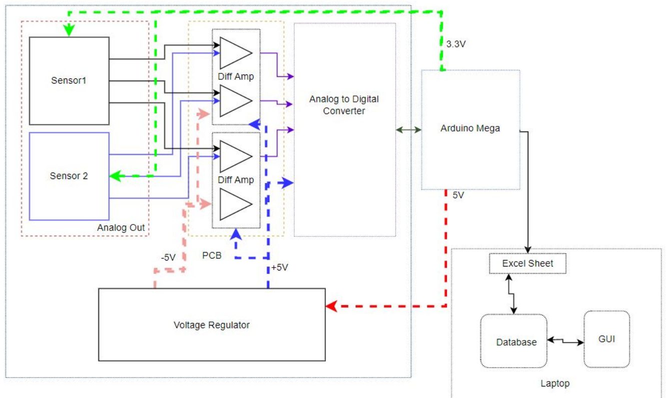
Magnetometer Valid Range: 0

What needs to be done

Circuit

- Redesign schematic
  - Modify the voltage regulator to supply ±5V
  - Rewire for on board motor driver
  - Modify to account for ADC
- Fabricate Boards
- Test board and ensure proper values
- Create the other PCBs for different sensors





Arduino

- With new schematic, pin I/O needs to be redone and reassigned.
- Program ADC timing with conversion start and data output
- Fully automate process with stepper motors
- Program motor driver for stepper motors

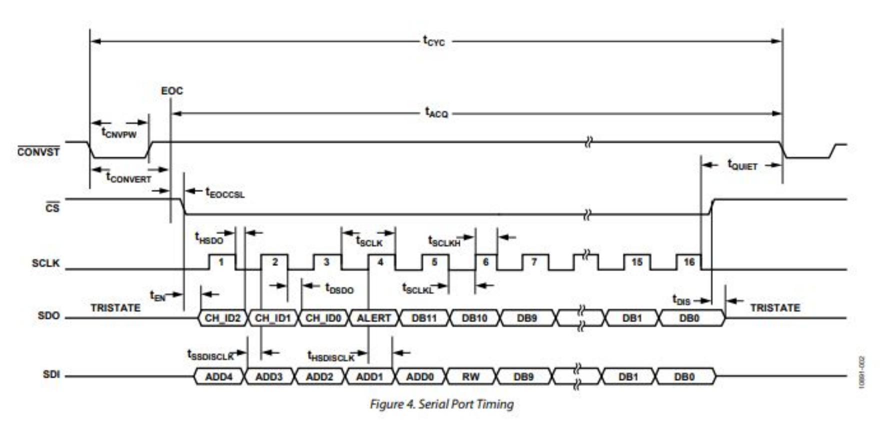
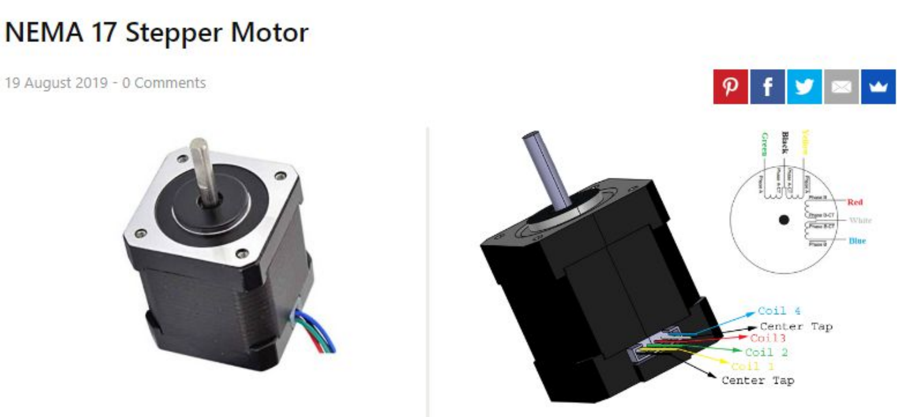


Figure 4. Serial Port Timing



NEMA 17 Stepper Motor

19 August 2019 - 0 Comments

Database

- Continue learning how Microsoft SQL Server works
- Ensure schema from mySQL translates to MSSQL correctly
- Learn the easiest way to implement data insertion from excel
- Get computer/laptop to run Database server locally instead of on my personal machine

Graphic User Interface

- Clean up database connection code
- Implement accepted sensor values functions by pulling database values into array
- Make sure C++ still complies after bug testing
- Write documentation so others can use it easily

Thank you for your time

Any Questions?