Handoff Documentation

# Description

This document serves to inform the user on how to set up the Driver Counter using the provided source files and an Arduino Uno.

# Files needed

All the files are in the git repository:

<https://github.com/JonathanGWesterfield/TamuDriver>

Submission files are in the ‘Submission’ folder

The backend website monitors data and displays statistics. An example can be accessed here (this is what it should look like in the end):

<http://students.cse.tamu.edu/jgwesterfield/>

PHP/MySQL API files:

* CommonInterface.php
* CommonMethods.php
* PHPtoSQL.php
* PHPtoSQLInterface.php

Python/MySQL API files:

* DataHandler.py
* DBHandler.py
* FileHander.py

Web Page files

* chart-data.php
* chart-locations.php
* chart-xlabels.php
* index.php
* regression.min.js
* Chart.min.js

Arduino Files:

* fullcounter.ino

For the website to display properly, all of these files MUST be in the same directory

* CommonInterface.php
* CommonMethods.php
* PHPtoSQL.php
* PHPtoSQLInterface.php
* chart-data.php
* chart-locations.php
* chart-xlabels.php
* index.php
* regression.min.js
* Chart.min.js

# Database Connection Information

All database connection credentials are on lines 23-26 in the CommonMethods.php file

$db="database.cse.tamu.edu"; // the database host name

$dbname="XXXXX-TamuDriver"; // specifying which database name

$user="XXXXX"; // The account used for the Database

$pass="XXXXX"; // Password for the database

These values can be changed depending on where you host the database (if you create your own).

# The Database Structure

To set up the database, run the DatabaseSetup.sql file in a program like, phpmyadmin or MySQL Workbench (the preferred option). This file will set up the table for the data and will also add any constraints to the table, in addition to setting up a database if one doesn’t exist. However, it would be easier to run the script if your database exists already.

# 

# 

Setup procedure:

1. Download and install the operating system for the Raspberry Pi onto your SD card:

<https://www.raspberrypi.org/documentation/installation/installing-images/>

1. Insert the SD card into the Pi, then connect it to a monitor, keyboard, mouse, and power supply. If you wish to run the Arduino through a PC instead of with a Raspberry Pi, you can do that as well.
2. To connect the Pi to the campus wi-fi, you will need to follow special instructions to modify operating system files. Instructions can be found in the following link. Reboot the Pi after completing this step: <https://kkremitzki.github.io/blog/connecting-a-raspberry-pi-to-texas-am-wifi/>
3. Now that you are connected to the internet, let’s start installing software. First, open a terminal window and verify you have the correct version of Python - 2.7 - installed by typing “python” and pressing enter. Press Ctrl+X to exit the python shell, then in the regular terminal type the following commands:
   1. sudo apt-get install Arduino
   2. pip install mysql-connector-python-rf
4. Next, open the Arduino IDE from the applications menu and load the fullcounter.ino sketch.
5. Once the sketch has been uploaded to the Arduino, and once you have set up the database, download the python files and run DataHandler.py. Provided the database has been configured properly, the script should now run and connect to both the database and the Arduino.
6. Walk past the sensors and observe the output in the Python shell while the scripts runs. You should see messages indicating that it detected you “entering/exiting” as you walked by.