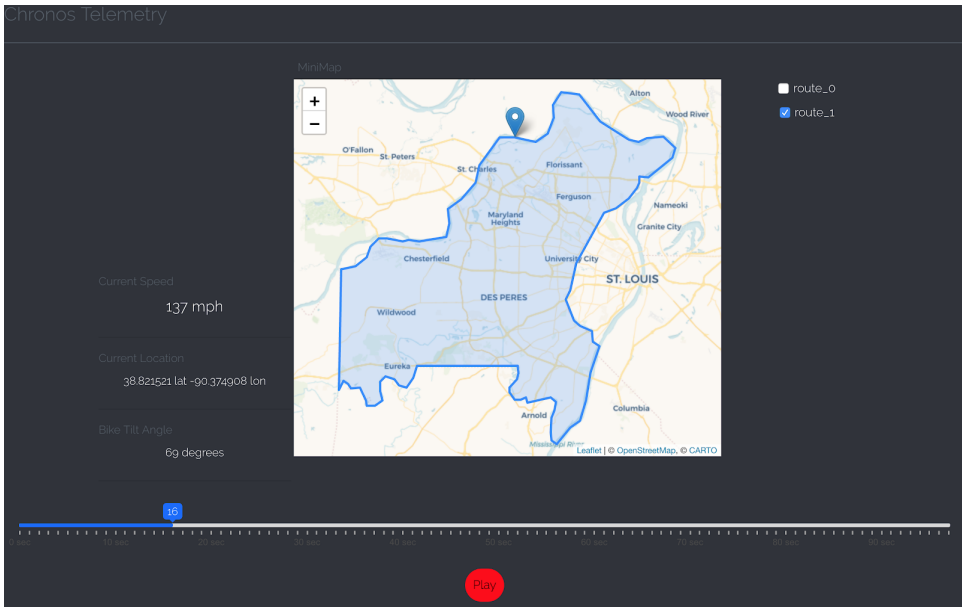
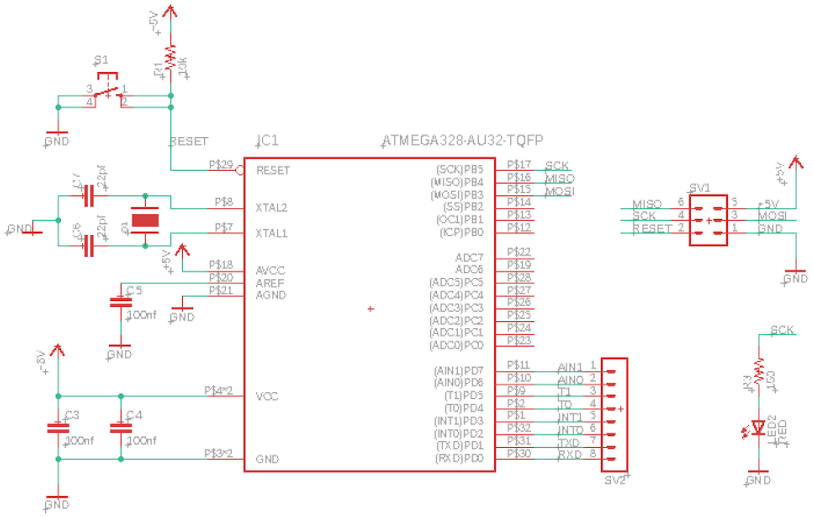
Arthur Zhang **Fall 2018**

Collegiate Electric Racing Team Telemetry System



I built a telemetry system for my collegiate electric motorcycle racing team (SPARK) to help us analyze our motorcycle performance after races. I used an Arduino Uno and various sensors (GPS, IMU, temperature) to collect the data on the motorcycle and clean it using sensor fusion algorithms before saving it to an sd card. This data collected can be viewed on the telemetry dashboard (shown above). The dashboard supports a playback feature that allows users to view how the data changes by the second.



Arduino and sensors schematic diagram

After selecting the sensors to collect data, I architected a system for cleaning sensor data and storing it. I created a separate library for each sensor, abstracting useful methods to reduce code clutter. By designing the data structure in this fashion, we were able to develop and test code for each sensor in parallel. I worked heavily on obtaining useful data from the gps sensor and developing the front end gui to view collected data. In the future, I plan on designing on a single electrical component that collects data and broadcasts it live for viewing during each race.

```
/*
  sensor_driver.ino - Sketch file for using GPS, temperature, and imu libraries
  Created by Arthur K. Zhang, November 14, 2018.
  Modified by:
  Property of SPARK el electrical
*/
// gps imports
#include <GPS.h>
#include <SoftwareSerial.h>
GPS gps(9, 10);

// imu imports
#include <IMU.h>
IMU imu;

// temperature imports
#include <TMP36.h>
TMP36 thermometer(A0, 3.3);

void setup() {
  // init thermometer
  thermometer.begin(9600);
  // init gps
  gps.begin(9600, 4800);
  // init imu
  imu.begin(9600);
}

void loop() {
  gps.printGPSInfo();
  imu.printIMUInfo();
  thermometer.printTempF();
```

Driver file saving sensor data by using sensor libraries I implemented