**Bootcamp Challenge**

**Vehicle State Estimator**

**Description:**

In automotive, a vehicle state is defined to be the state of different parameters relating to the current drive cycle. Some examples for vehicle state parameters are: vehicle speed, yaw rate, date/time…etc

In this exercise, we will be working with two ECU’s: The vehicle state estimator, and a measurement ECU.

The system is based on two AVR kits to simulate both ECUs, we will be injecting the current time once from a PC to the Vehicle State Estimator ECU, we will then be communicating data to the Measurement ECU which will perform some calculations and measurements over this data. This is an overview of the system:

Output data

PC:

Output screen

Initial Calibration data

AVR 2:

Measurement ECU

AVR 1:

Vehicle State Estimator ECU

PC:

Data injection

**System Requirements Specification:**

1. The measurement ECU should be able to fetch the current time for every press of a “fetch” button on the measurement ECU. This shall be considered the zero time of the measurement (start time). This time should be printed out on the measurement ECU Output screen.
2. The Vehicle State Estimator ECU should support increasing and decreasing the speed using the 2 buttons on the Vehicle State Estimator ECU.
3. The measurement ECU should be able to calculate the distance traveled since the zero time was fetched (time of measurement start) even if the speed changes during that time.
4. The system should be able to output speed, time and distance since the last time fetch every time the print button is pressed.
5. Communication between Vehicle State Estimator and Measurement ECUs should be implemented using SPI
6. Communication between any ECU and PC should be implemented using UART (TTL).
7. Final submission shall be on **HW** only.

**Deliverables:**

Each team is required to provide the following:

1. Static architecture.
2. Git Repository showing commits..etc.
3. An official release email with the video documentation.
4. A list of assumptions.
5. At the end of Saturday, a Git repo and a video demo of working UART/SPI drivers should be delivered.

**Hints:**

1. You receive **bonus** points if:

* You use interrupts to detect UART/SPI events

1. Deadline is final and there are no late delivery points. If the deadline is exceeded, the team receives no points for the final delivery.
2. Communication with me should be as a customer. Questions written in professional email format, calls are scheduled with agreement from the customer and are conducted in English
3. Simulation may be used to help with development. However, be very very careful with HW surprises. A lot of the development time is spent on actually getting things to work on HW.