

Project - LapKart

The objective of the project is to design a lap timer consisting of:

- Raspberry Pi to run the software
- LED matrix display

Requirements

LED Matrix

1. Display shall be readable from 10 feet away
2. Display should be bright enough to be read on a sunny day

Please consider using 128x64 or another large size display:

- <https://www.adafruit.com/product/2278>.
- <https://www.sparkfun.com/products/14646>Links to an external site.

Sensor

There shall be two distance sensors such that we can detect a vehicle passing by in one direction.

1. Sensor logic shall be able to detect a vehicle moving north, versus a vehicle moving south in the other direction
 - Vehicle moving south should not trigger a lap. This can happen when a driver has finished their laps and they are coming back around to park the vehicle
2. When Sensor 1 triggers, immediately followed by Sensor 2 trigger, that shall indicate a driver passing by
 - Timing of the sensor shall be taken into account such that bystanders walking by the project will not trigger a driver passing by

In a nutshell, your sensor logic should ideally sense only when a driver is passing forwards, and filter out everything else. You can assume that a sensor is triggered when someone is within 10 feet of the sensor.

User Interface

A good way to interface with the user is through voice commands. This way, we would not need to build

custom button-menu or use the keyboard. For example, a user can hold a button and speak a command.

You must support the following commands:

1. Reset (should not restart Raspberry Pi, and instead should restart your logic loop)
 - A good way might be to kill the application and have systemd restart it
2. Set maximum laps: #
3. Set maximum drivers: #
 - Set driver 1 name: Preet
 - Set driver 2 name: John
4. The settings above must persist across power cycles
 - Save the configuration to a file, and load configuration upon startup of your software

Packaging

1. There shall be a single power cable entering the project package
 - You could have a 12v input that is then converted to 5v and other voltage levels internally
 - You could also have an AC wire plugged in, and internally you can use 12v transformer or other power conversion hardware
2. Package should be sturdy enough to not fall apart easily ;)
 - Ensure that your hardware is secured properly
3. The Raspberry Pi hostname should be listed on your package
 - In case I wish to edit your code, I should be able to ssh into Raspberry Pi to alter any logic
 - There should be a README file at the root indicating where your source code is located

Revision #4

Created 1 month ago by [Preet Kang](#)

Updated 1 month ago by [Preet Kang](#)