**Accuracy without PPS at room temperature**

Every minute 0.36ms aka every hour 21.6ms

1ms accuracy with lap times <1minute or with gps

10ms accuracy with lap times >1minute and no gps

A diagram of a computer chip

AI-generated content may be incorrect.

A computer chip with many colored wires

AI-generated content may be incorrect.

Planning out the whole system.

The Timing Gate will have a Wifi Antenna, GPS Module, and Beam Break trigger. The Timing gate sends and receives TCP messages on port 5000. Sends ETC time of the gate event. Can be paired with multiple gates that also send their trigger time. Each is identified by “Serial” number. Server side software logs the raw events in a file for autosave ability and calculates sector and lap times from the raw events. Simple labels in a menu will show connected devices and allow to select where it is placed. Maybe in the future it’ll show gate locations.

In the same application I would also like to view the live telemetry of the car. This will consist of a UDP stream of filtered can data. This is in hopes to have a little lag as possible, and save compute for other functions. This module will also have a GPS antenna that will sync the internal clock and send the GPS coordinates of the car to the server using the same UDP connection. The Server will log both of these data streams and save to a file. Ideally the server would also graph the coordinates relative to the gates. The server will also send the Timing Gate times and any flags to the car to be transmitted over CAN to the vehicle’s dashboard. Flags will be transmitted as a simple integer.

0 – Green

1 – Yellow

2 – Red

3 – Checkered

4 – Black

Lap time will be sent in seconds and milliseconds. Both will only be sent on an update. Eventually this will also be able to send commands to change settings on certain devices in the car.

Eventually the wireless link might get switched to a lora system or 900mhz and just use wifi to create a high speed uplink. The live telemetry module could also work with just the long range link and then I have a separate data logger to micro sd card module that has wifi for high speed log starting, stopping, and file transfer.

Links 2 stuff

Esp32 with removeable antenna - <https://www.digikey.com/en/products/detail/olimex-ltd/ESP32-S2-WROVER-DEVKIT-LIPO-EA/21662599?gad_campaignid=20243136172>

Potential USB c to battery charger circuitry

ILTM Throughput calculations

Can Bus 500kbps

Spi Bus 8,000kbps

Can message 8 bytes + overhead, call is 128bit per message

500k/128 = 3.9k messages a second (Currently sending ~500 messages per second)

| **Layer** | **Max throughput (8-byte CAN payload)** |
| --- | --- |
| CAN bus 500 kbps | ~3,900 msg/sec |
| SPI 8 MHz + SD | ~62,500 msg/sec |
| WiFi UDP 2.4GHz | ~68,000 msg/sec |

Improving Data Acquisition for the UGA Motorsports IC Team

The UGA Motorsports formula team needed a way to accurately time and car during events such as Acceleration and Autox as well as be able to reliably view vehicle data while driving. Currently the team is using a stopwatch for timing which is quite inaccurate. Not only is the human reaction time slow and inconsistent, but the person cannot be at both the start and finish line, so the point at which they think the car crosses the line could, and normally is, inaccurate. A simple solution to this problem is a beam break timing system, but commercial products are often prohibitively expensive or don’t have the capability to do what the team needs. It would be beneficial to the team if the timing gates were able to be easily combined with engine data for easier analysis after the fact. Additionally, the current live telemetry system on the vehicle is dependent on the ECU, which limits the amount of data and speed at which it can be transmitted. The current system is also not very tolerant of bad connections, leading to it frequently going out.

I propose a system that combines all data acquisition into one convenient system/application that allows data to be quickly saved, viewed, synchronized, and expanded. A vehicle telemetry link capable of up to 2km (The max distance from the pit at Michigan International Speedway) for key engine statistics and timing gates capable of a typical acceleration test, or multiple sectors in an Autox test. Additionally, it should be a convenient way to take driver notes, record tire data, and organize test data. This will make testing documentation easier and improve the quality of data collected. Further, it will allow for easier analysis and review of testing data.

The core principle linking each device together is accurate GPS timing. All data will be sent with the UTC time accurate to the millisecond. This will used to calculate the time between gate triggers and to combine the engine data with those timing events. When GPS is lost, network time protocol will be used to synchronize the devices.

Below is the initial GUI mockup for the application.

A screenshot of a computer

AI-generated content may be incorrect.

Component/Functionality DiagramA whiteboard with black text and words

AI-generated content may be incorrect.