#### Условия

Современные технологии проникают не только в быт, но и в спортивные состязания. Примером может служить VAR (Video assistant referee) в футболе. В лёгкой атлетике фотофиниш применяется еще с первой половины 20-го века, а начиная с 1992 года используются цифровые технологии для определения победителя в забеге. Автогонки не стали исключением и с появлением скоростных камер, ими оснащается каждая трасса, на которой проводятся высокоскоростные гонки.

В последние годы широкое распространение получил дрифт, как дисциплина в автоспорте. С учетом специфики дисциплины, фотофиниш не позволяет выявить победителя ни в квалификации, ни в парном заезде. Основными критериями оценки заезда являются:

* угол заноса;
* скорость движения автомобиля;
* траектория движения автомобиля.

Баллы за все три критерия выставляются судьями, но скорость и угол в определенных участках трассы имеется возможность измерить используя сенсоры, обработать с помощью программного обеспечения и предоставить судейской коллегии в виде инфографики.

Вам предлагается реализовать программно-аппаратный комплекс, способный дать максимально объективную оценку проездов пилотов по критерию «Угол».

#### Техническое задание

Для реализации комплекса вам необходимо разработать аппаратную и программную части.

Аппаратная часть должна состоять из сенсора (гироскоп и/или акселерометр), вычислительного устройства (Arduino, Raspberry Pi, ESP) и устройства, передающего по беспроводному каналу измеренную сенсором величину. В зависимости от выбранного вычислительного устройства, может варьироваться способ беспроводной передачи информации. Требования к конструктивным

особенностям и питанию аппаратной части не предъявляются.

Программная часть должна быть запущена на персональном компьютере или телефоне и иметь возможность соединения по беспроводному каналу с аппаратной частью.

Необходимо предусмотреть в программной части добавление соревнования, в рамках которого будет происходить последующие заезды. Информация о соревновании должна включать в себя:

* наименование соревнования;
* дату;
* наименование организатора;
* место проведения.

Перед началом заезда по инициативе программной части активируется аппаратная часть. При этом пользователем указываются порядковые номера пилота(ов) (число от 1 до 99) и один из типов заезда:

* qualifying (квалификация);
* top 32 (1/16)
* top 16 (1/8)
* top 8 (четвертьфинал)
* semifinal (полуфинал)
* battle for 3rd place (заезд за 3 место)
* final (финал)

Аппаратной частью осуществляется измерение необходимой телеметрии с сохранением её в памяти аппаратной части. Измерения должны производится не менее 10 раз в секунду. Длительность заезда не должна превышать 60 секунд.

После окончания заезда по инициативе программной части необходимо осуществить передачу по беспроводному каналу измеренных данных с аппаратной части с последующей очисткой памяти аппаратной части и сохранением на программной части информации о заезде:

* тип заезда;
* номер(а) пилота(ов);
* время начала заезда;
* время окончания заезда;
* данные телеметрии.

Непосредственно после передачи данных необходимо осуществить демонстрацию полученных данных программной частью в виде графика. График должен представлять зависимость угла от общего времени заезда. В случае квалификационного заезда отображается график телеметрии одного пилота, в случае парного заезда отображаются графики телеметрии обоих пилотов с наложением друг на друга.



*Пример графика зависимости угла от времени заезда*

После прохождения всех заездов необходимо предусмотреть возможность выбрать любой из заездов для просмотра данных телеметрии по нему в табличном виде и в виде графика.

Необходимо предусмотреть возможность архивации всех заездов соревнования.

#### Рекомендации к выполнению

* Рекомендуется предусмотреть модульность программной части, а именно серверную и клиентскую части.
* Необходимо предусмотреть сохранение данных, получаемых от аппаратной части в системе управления базами данных (СУБД). Выбор СУБД не регламентируется.
* Разработку аппаратной и программной частей рекомендуется вести с помощью системы контроля версий git.
* Рекомендуется использовать unit-тестирование при разработке программной части.

#### Требования к документации

* Титульный лист (с указанием названия кейса и перечислением членов команды);
* Анализ технических требований;
* Обоснование выбора языка программирования и используемых программных средств;
* Структурная и функциональная схемы программного продукта;
* Блок-схема работы основного алгоритма;
* Схема базы данных;
* Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);
* Программный код (ссылка на репозиторий).

#### Регламент испытаний

Для демонстрации работы комплекса, в частности аппаратной части, допускается использование моделей автомобилей масштабе не более 1/8.

* Производится запуск комплекса
* Производится добавление соревнования
* Производится добавление и старт квалификационного заезда
* Производится окончание заезда и просмотр графика телеметрии
* Производится добавление и старт парного заезда
* Производится окончание заезда и просмотр графиков телеметрии
* Производится просмотр графика(ов) по уже прошедшему заезду
* Производится архивация соревнования и добавление нового

#### Примерный перечень средств и инструментов для выполнения задания

* https://www.python.org/
* https://sqlite.org/
* https://www.postgresql.org/
* https://flask.palletsprojects.com/en/2.0.x/
* https://vuejs.org/
* https://www.raspberrypi.com/
* [https://www.espressif.com/en/products/socs/esp32](https://www.espressif.com/en/products/socs/esp32З)