**Техническое задание Исполнитель**: Чеботарев М.М

**Задача**: разработать набор системных сервисов/демонов, обеспечивающих взаимодействие со следующими периферийными устройствами: модуль реального времени, микроконтроллер, дисплей, кнопки.

**План выполнения работ**:

1.1. Спроектировать взаимодействия:

1.1.1. Способ взаимодействия с RTC-модулем;

1.1.2. IIC-протокол взаимодействия с микроконтроллером DigiSpark;

1.1.3. Способ взаимодействия с GPIO.

1.2. С учетом требований и ограничений, предоставляемых/накладываемых каждым из трех направлений, разработать алгоритм работы все служб, так чтобы исключалась возможность гонки за ресурс;

1.2.1. Построить граф алгоритма rtc-демона;

1.2.2. Построить граф алгоритма демона Cerebro;

1.2.3. Построить граф алгоритма демона btn\_sound.

1.3. Разработать **RTC-демон**, поддерживающий синхронизацию при включении и выключении;

1.3.1. Спроектировать аппаратное подключение RTC-модуля;

1.3.3. Разработать V0.5 (синхронизация только при включении);

1.3.2. Отладить алгоритм синхронизации, определить оптимальный уровень загрузки демона;

1.3.3. Разработать V1.0 (синхронизация при вкл/выкл)

1.3.4. **Протестировать работу** rtc-модуля.

1.4. Разработать демон управления громкостью звука – **btn\_sound**;

1.4.1. Разработать класс управления GPIO - GPIOController;

1.4.2. Протестировать класс GPIOController;

1.4.3. Сгенерировать класс-адаптер OrgOfonoCallVolumeInterface с

сервисом ofono;

1.4.4. Реализовать алгоритм работы демона btn\_sound;

1.4.5. Написать тест и протестировать сервис.

1.5. Разработать прошивку микроконтроллера(Cerebellum):

1.5.1. Разработать прошивку микроконтроллера (МК);

1.5.2. Разработать бинарный протокол взаимодействия с МК по I2C;

1.5.3. Разработать набор тестов для МК;

1.5.4. Разработать принципиальную схему подключения батареи,

экрана и кнопки к MCU;

1.5.4. Произвести тестирование;

1.5.5. Отладить прошивку МК, внести изменения, подготовить V1.1;

1.5.6. Произвести повторное тестирование.

1.6. Разработать демон управления (Cerebro) микроконтроллером;

1.6.1. Разработать бинарный протокол взаимодействия с демоном через

unixSocket`ы;

1.6.2. Разработать набор тестов для демона;

1.6.3. Произвести тестирование;

1.6.4. Отладить прошивку МК, внести изменения, подготовить V1.1;

1.6.5. Отладить демон, внести изменения, подготовить V1.1;

1.6.6. Произвести повторное тестирование;

2. Установить все разработанное ПО на RPI Zero;

2.1. произвести окончательное тестирование всеми тестами;

**Используемое ПО:** блокнот Notepad++; компилятор g++; Dr. Memory (отладка);