**Hardware:**

1. Livestream cameras manual

- http://doc.arh.hu/parkit/parkit\_users\_manual\_v3.pdf

- http://doc.arh.hu/freewaycam/freewaycam\_users\_manual\_v3.pdf

- <https://www.npmjs.com/package/onvif-snapshot>

2. DataSheets for MOXA

- Datasheets / МОХА

3. Vehicle detector

- Datasheets / VehicleDetector

**Software:** Cафронов

Сервер на объекте (Pentagon)

1. Точка входа: pg\_main

- Это демон для сервиса захвата изображений, запускаемый на компьютере на объекте

Запрашивает у камеры снимок, когда машина подъехала, сохраняет снимок на диск в

отдельную папку и отсылает на удалённый сервер для распознавания (по http, предварительно шифруя AES поток байтов)

Возвращает json с результатом распознавания и путём до фото

2. Драйвер камеры: pg\_camrea\_driver

- Клиент, запрашивающий у ip камеры снимок

3. Драйвер MOXA: pg\_moxa\_driver

- MOXA опрашивается сервером каждые 0,5 сек (500 мс) для получения информации об активных пинах

- Есть второй вариант получения информации с MOXA: *ActiveTag –* сервер подписывается на обновления MOXA и тем самым не нагружает микроконтроллер частыми опросами

4. HTTP-клиент: pg\_http\_client

- Клиент для связи с Godfather

5. База данных: pg\_databse

- СУБД SQLite

- При взаимодействии с базой данных используются очереди

- Сохраняется сам номер в виде теста(‘plateno’), координаты 4-х углов

маски(‘plate\_frame’), изображение маски(‘plate\_image’), фото полностью (‘full\_image’)

Удалённый сервер (Godfather)

1. Точка входа: gf\_main

2. HTTP-сервер: gf\_http\_server

- Многопоточный http-сервер

3. Recognition engine: смотри  *Сафронов / Recognition architecture*

Распознавание номера осуществляется с помощью двух нейросетей, имеющих архитектуру UNET:

- Первая сеть (plate\_frame\_model) отвечает за сегментацию изображения. На выходе получаем маску(белым цветом обозначено то, что относится к классу платы/номера; чёрным – то, что относится к классу фона).

- Вторая сеть отвечает непосредственно за классификацию платы – распознавание символов на плате (37 классов: a-z, 0-9, фон). На выходе получаем строку json, содержащую следующие поля: 'status' (1 – если распознавание прошло успешно, 0 – если нет); ‘number’ - сам номер (текст); ‘frame’ - координаты 4-х углов маски; ‘confidence’ – степень уверенности сети в результате распознавания.

**Software:** Фрилансер

Сервер на объекте (Pentagon)

1. Точка входа: pg\_main

2. Драйвер камеры: pg\_camera\_driver

- Клиент, запрашивающий у ip камеры серию (из 9) снимков с интервалом 0,05 сек (50 мс)

3. Драйвер MOXA: pg\_moxa\_driver

- MOXA опрашивается сервером каждые 0,5 сек (500 мс) для получения информации об активных пинах

- Есть второй вариант получения информации с MOXA: *ActiveTag –* сервер подписывается на обновления MOXA и тем самым не нагружает микроконтроллер частыми опросами

4. HTTP-клиент: pg\_http\_client

- Клиент для связи с Godfather

Удалённый сервер (Godfather)

1. Точка входа: gf\_main

- Демон, запускаемый на компьютере с GPU для распознавания номеров с изображения.

Связь с компьютером на объекте поддерживается по http

2. База данных: gf\_sqlite

- СУБД SQLite

- При взаимодействии с базой данных используются очереди

3. HTTP-сервер: gf\_http\_server

- http-сервер для связи с Pentagon

4. Модуль распознавания: gf\_predict (смотри: (1)Фрилансер / Predict, (2)Фрилансер / CarplateEngineDoc.doc)

- Распознавание осуществляется с помощью гугл библиотеки openALPR

- На выходе получаем строку json, содержащую поля: ‘plate’ – распознанный номер, ‘status’ - 0 или 1

**Вариант улучшения технического зрения:**

1. На вход openALPR подавать результат первой нейронной сети Сафронова (сегментированную плату)