1) **Компоненты и цены**

- Машинка HSP Hunter 1:16 – 8250 рублей
- Аккумулятор Li-Ion Spard 2000mAh, 7,4V, 15C – 1690 рублей
- Raspberry Pi 3 Model B - ???
- Arduino Nano - ???
- Датчик напряжения x2 – 200 рублей
- Понижающий преобразователь DC-DC 6 - 32В до 5В с USB разъёмом – 150 рублей
- Raspberry Pi Camera - ???
- USB Camera - ???

2) **Компонентная база машинки**

- Коллекторный двигатель (4WD)
- ESC для коллекторного двигателя (от HSP)
- Приёмник 2.4GHz и 4 канала
- Сервопривод
- NiMH аккумулятор 7.4В 1100mAh
- Передатчик (пульт управления)

3) **Программная часть**

- Операционная система RPi – Raspberry Pi OS (Raspbian)
- ЯП сервера – Java 17
- Библиотека для работы с GPIO – PI4J v2
- Сервер для создания IP камеры – библиотека Motion

4) **Каналы связи**

- TCP для клиент-серверного обмена
- I2C для связи с Arduino

5) **Управление ESC**

Управление контроллером скорости осуществляется по такому же принципу, что и управление сервоприводом. Необходимо подать импульс определённой длины для установки необходимой скорости и направления (диапазон: [1000; 2000] мс). В ходе разработки получены следующие значения (нейтральное положение немного отличается от стандартного по причине особенностей управления серво через RPi + PI4J):

STOP = 1550;
FORWARD_BRAKE = 1300;
FORWARD_MAX = 1850;

FORWARD_MIN = 1610;
BACKWARD_BRAKE = 1590;
BACKWARD_MAX = 1490;
BACKWARD_MIN = 1518;

В данной модели регулятора для входа в режим reverse и brake необходимы дополнительные манипуляции. Обычно такие действия не требуются, необходимо читать инструкцию к ESC и проверять его работу на практике.

Переход в режим reverse (движение назад):

1) Подать импульс STOP
2) Задержка (40 мс)
3) Подать импульс BACKWARD_MIN
4) Задержка (40 мс)
5) Подать импульс STOP
6) Задержка (40 мс)
7) Подать импульс BACKWARD_MIN

Переход в режим brake (тормоз):

- Из движения вперед:

1) Подать импульс FORWARD_BRAKE
2) Задержка (50 мс)
3) Через 1000 мс подать импульс STOP

- Из движения назад:

1) Подать импульс BACKWARD_BRAKE
2) Задержка (250 мс)
3) Подать импульс BACKWARD_MIN
4) Через 1000 мс подать импульс STOP

---

6) Информация о соединениях компонентов

Больше информации представлено на схематическом рисунке.

Связь Arduino с RPi:

SDA1 I2C *(GPIO2, RPi)* ↔ **SDA** *(A4, Arduino)*

SCL1 I2C *(GPIO3, RPi)* ↔ **SCL** *(A5, Arduino)*

5V *(Power, RPi)* ↔ **VIN** *(Arduino)*

Ground *(RPi)* ↔ **GND** *(Arduino)*

Связь Arduino с датчиками напряжения (Voltage Sensor):

S *(Voltage Sensor)* ↔ **A0/A1** *(Arduino)*

+ *(Voltage Sensor)* ↔ **5V** *(Arduino)*

- *(Voltage Sensor)* ↔ **GND** *(Arduino)*

Связь ESC с RPi:

S *(ESC)* ↔ **GPIO13** *(RPi)*

Power Out *(ESC)* ↔ **Power In** *(Servo)*

Ground *(ESC)* ↔ **Ground** *(RPi)*

ESC питает сервопривод, как в оригинале при использовании приёмника.

Связь Servo с RPi:

S *(Servo)* ↔ **GPIO12** *(RPi)*

Power In *(Servo)* ↔ **Power Out** *(ESC)*

Ground *(Servo)* ↔ **Ground** *(RPi)*

Изображения с пинами Arduino и RPi будут приложены к документу.

7) Описание API

Заряды батарей передаются в RPi по I2C в формате ‘v1-v2’, где v – значение заряда в процентах от 0 до 100. Arduino работает в slave режиме. v1 – заряд аккумулятора двигателя, v2 – заряд аккумулятора RPi.

Сервер принимает по TCP два значения в формате ‘v1:v2\n’, где v – число в диапазоне [-1; 1]. v1 – значение скорости машинки, где 0 – тормоз, отрицательные значения – движение назад, положительные значения – движение вперед. v2 – значение угла поворота колес, где 0 – нейтральное положение, -1 – максимальное левое положение, +1 – максимальное правое положение. Оба числа в формате float.

Сервер отправляет по TCP два значения в формате ‘v1-v2\n’, где v – значение заряда в процентах от 0 до 100.