Государственное бюджетное образовательное учреждение г. Москвы Школа №354 им. Д. М. Карбышева

Проект “Smart City Parking”

Авторы: Воля Максим

Девятов Леонид

Рынин Дмитрий

Руководитель: Богачёва Татьяна Петровна

Москва 2020–2021

Цели:

Наша цель разработать умную парковку с зарядными станциями для электромобилей, чтобы сделать использование электромобилей более простым и удобным, тем самым увеличить спрос на них, уменьшить количество машин с бензиновыми двигателями и следовательно углеродные выбросы в атмосферу.

Задачи:

- Узнать информацию о зарядных станциях для электромобилей и о них самих
- Освоить вёрстку и программирование сайтов, создать http сервер для обработки “get” запросов
- Разработать робота “заправщика”
- Разработать робота “парковщика”
- Создать модель зарядочного комплекса
- Освоить создание баз данных и работу с ними
- Отладить связь между блоками EV3, Raspberry и Ардуино

Актуальность

Стало известно, что зарядные станции могут появиться возле всех новых торговых центров в столице. Московские власти и Министерство энергетики готовят поправки в градостроительный кодекс, согласно которым ни один новый торговый центр не может быть сдан в эксплуатацию без оборудования для зарядки аккумуляторов «зеленых» автомобилей.

Количество зарядных станцией зависит от региона и страны, например Китае активно строят инфраструктуру – сейчас в стране более 270 тыс. зарядных станций, а для сравнения, в России их чуть больше 350. Тем не менее, доля электромобилей от общего количества машинного парка в мире составляет ничтожные 0,5%. Количество электромобилей стремительно растёт, в то же время зарядные станции растут гораздо медленнее и на одну зарядную станцию в России приходится 30–40 электромобилей, в то же время в среднем по миру данный показатель - 10. На данный момент в России функционирует около 200 коммерческих электрозаправочных станций.
Эксперты, опрошенные ТАСС, одной из основных проблем развития рынка электромобилей назвали сложности с зарядной инфраструктурой в ряде регионов.

А сложности с зарядкой вынуждают владельцев электрокаров обратно пересаживаться на машины с бензиновыми двигателями, хотя эксплуатация неэкологичного авто выходит дороже и вредит экологии.

Мы решили это изменить и сделать использование электромобилей удобным и практичным.

Исследования:

Электрокары в настоящее время всё же распространены пока ещё не так, как транспортные средства с двигателями, работающими на бензине или дизельном топливе, поэтому производители продолжают работать над совершенствованием технологии зарядки.

Зарядка электромобиля возможна одним из четырех способов:
1. С помощью обыкновенной розетки с напряжением 220 В. Правда, данный вариант используется всё реже ввиду своей ненадежности.
2. От бытовой электросети, через которую проходит переменный ток. Этот способ более предпочтителен, чем предыдущий, поскольку кабель, покупаемый вместе с машиной, имеет внутри специальную защиту.
3. Трехфазная зарядка, являющаяся самой безопасной. Её основное преимущество – возможность полного контроля над процессом.
4. Быстрая зарядка электрокара.

Количество электрозаправок в мире:

Инфраструктура для электромобилей продолжает стремительно развиваться. Большая часть новой инфраструктуры построена в Китае и Европе. С немалым отставанием на третьем месте по количеству электрозаправок расположились США.

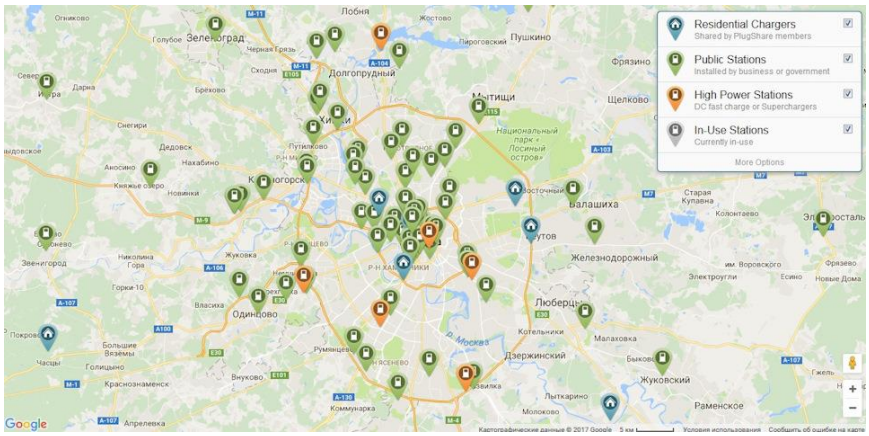
Наиболее агрессивную политику в отношении развития электромобилей проводит Китай. На него приходится свыше половины всех мировых зарядок. В Европе лидирует Франция.

Количество электрозаправок в России:

В Москве больше находится половины станций “заправки” – около 100.

Санкт-Петербург на втором месте по количеству электрозаправок. Здесь их около 30.

По 2 электрозаправки в Екатеринбурге, Самаре, Краснодаре, Перми и Уфе.



Слабые места электромобилей

В итоге время зарядки электромобиля — одно из главных слабых мест электромобилей по сравнению с бензиновыми автомобилями

Для каждой аккумуляторной батареи обозначена максимальная сила зарядного тока. Если превысить ее, то это может сказаться не только на потере емкости батареи, но и

даже привести к полному выходу ее из строя.

Ультрабыстрые зарядные станции

Ультрабыстрые зарядные станции на сегодняшний день — самый быстрый способ зарядить электромобиль. Их можно встретить на автомагистралях или крупных публичных парковках. Такие станции обеспечивают постоянный или переменный ток большой мощности и могут зарядить автомобиль до 80% за 20–40 минут. В большинстве случаев ультрабыстрые станции отключаются, когда аккумулятор электромобиля заряжен примерно на 80%, чтобы защитить батарею и продлить срок её службы.

Ультрабыстрая зарядка может использоваться только на тех автомобилях, где возможность её применения предусмотрена изначально и присутствует специализированный тип зарядного разъёма.

Информация о разъёмах:

Во многих электромобилях под заправочным лючком можно обнаружить два разъема: один – для зарядки переменным током, второй – для подключения к постоянному току и быстрой зарядки.

Встречаются и машины на электротяге с одним зарядным портом. В этом случае порт либо совместим с переменным и постоянным током, либо используется для зарядки только переменным током, как правило медленной, хотя есть и исключения.

Единого разъема для зарядки электромобилей не существует до сих пор. Более того, виды разъемов зависят от страны или региона: Северная Америка, Европа, Китай и США — у всех разные.

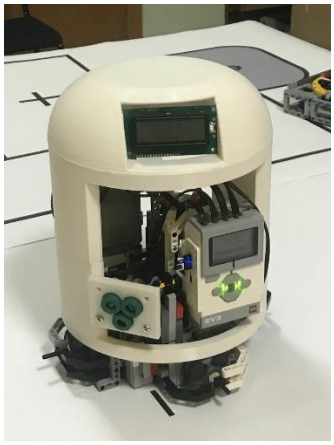
Type 1 J1772 – американский пятиконтактный разъем, разработанный еще в 2009 году и встречающийся практически на всех электромобилях, ввезенных в нашу страну из США. Рассчитан на напряжение 230 В и ток в 32	
Type 2 (Mennekes) Максимальная мощность для однофазной сети – 7,4 кВт, для трехфазной – 43 кВт. Однако чаще мощность при трехфазном подключении ограничена 22 кВт.	
CHAdeMO Рассчитан на постоянный ток в 125 А при напряжении 500 В и соответственно 62,5 кВт мощности.	
CCS Combo – еще один распространенный тип разъемов, используемый с 2012 года. Он может подключаться как к переменному, так и к постоянному току, что позволяет использовать его как с медленными, так и с быстрыми зарядками. При подключении к сети переменного тока происходит его выпрямление в постоянный.	
GB/T – Стандартом предусматриваются два типа разъемов: для медленной зарядки переменным током и для быстрой зарядки постоянным током.	

Tesla Supercharger – разъем, используемый в электромобилях марки Tesla. Максимальная зарядная мощность достигает 200-250 кВт при постоянном токе.	

Самые распространённые виды разъёмов:

Описание:

Робот «Заправщик»:



- 2 блокаev3,
- raspberry pi 3,
- Arduino uno,
- 3 мотора для откачки воздуха,
- ЖК дисплей,
- аккумулятор.

Робот “Заправщик” с помощью камеры определяется местоположение крышки заправочного люка конкретно данного автомобиля, после чего посылаются координаты позиционирования манипулятору, который с помощью вакуумной присоски бережно открывает крышку люка. Затем с помощью камеры определяется тип разъема и подбирается к нему соответствующий штекер.

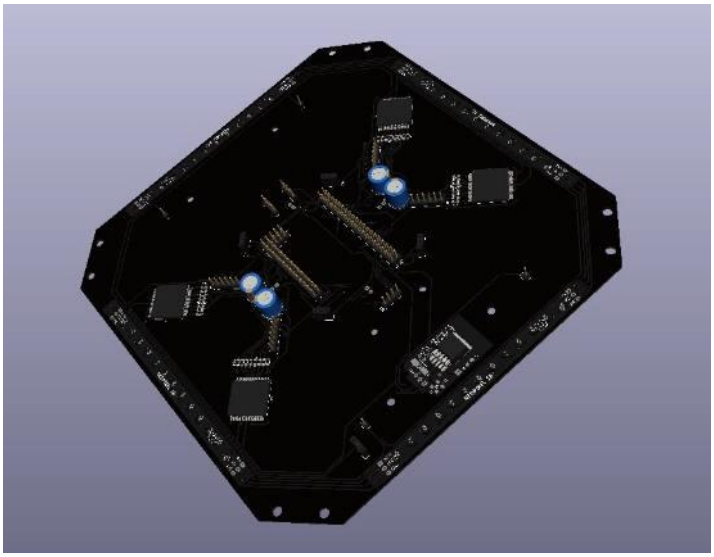
Заправщик присоединяет нужный штекер к разъему автомобиля и начинается зарядка

Робот «Парковщик»:

Робот парковщик, состоит из:

4 моторов на движение, что позволяет ему двигаться в разных направлениях за счет всенаправленных колес, моторы управляются Arduino Mega через специальные драйверы. Робот поднимет автомобиль с помощью 4 сервоприводов что позволяет ему поднять без перекосов. Для ориентации робот имеет гироскоп, энкодоры и датчики

освещённости. Робот парковщик подает автомобиль для зарядки на специальную вращающуюся платформу.



“**Raspberry”**

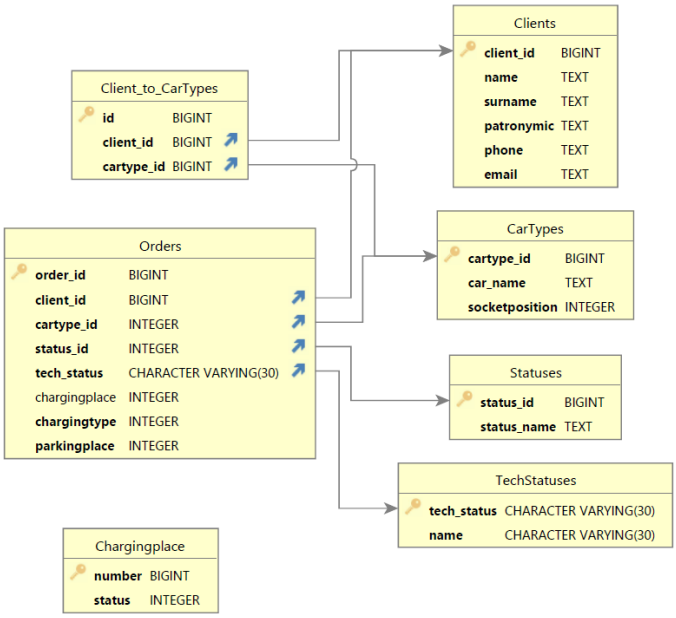
Raspberry с подключённой к ней USB камерой, используется для считывания QR-кода и определения разъёма, установленного в автомобиль(с помощью видео зрения, написанного на OpenCV)

Сайт

В ходе работы над проектом также был создан сайт, на котором пользователь регистрируется, вводя помимо стандартной информации и информацию об автомобиле. Оплата зарядки, указание номера станции и парковочное место автомобиля происходят на нём. ОН сделан на: html, CSS, java script

База данных

Все данные по заказам хранятся в базе данных (такие как: информация о пользователях, данные по каждой модели электромобиля, статусы для пользователя, технические статусы и многое другое. Для этого используется несколько “таблиц”, между которыми имеются связи). После занесения данных в базу, специальная логика определяет, что нужно начать зарядку, подбирает оптимальное зарядочное место отправляет сообщение “парковщику” с необходимыми для его работы параметрами.



В базе данных находится 7 таблиц:

1. Информация о модели электромобилей
2. Таблица с зарядочными местами и их статусами
3. Информация о клиентах
4. «Словарь» статусов для пользователя
5. «Словарь» технических статусов
6. Принадлежность моделей автомобилей пользователям (связь многие ко многим)
7. Таблица с заказами (8 колонн:

@ номер заказа - колонна с самозаполнением, является первичным ключом

@ клиент - колонна, связанная с таблицей с информацией о клиентах вторичным ключом (для таблицы с клиентами это первичный)

@ модель автомобиля - колонна связанна с таблицей с информациях о моделях электромобилей вторичным ключом (для таблицы с информацией о электромобилях это первичный)

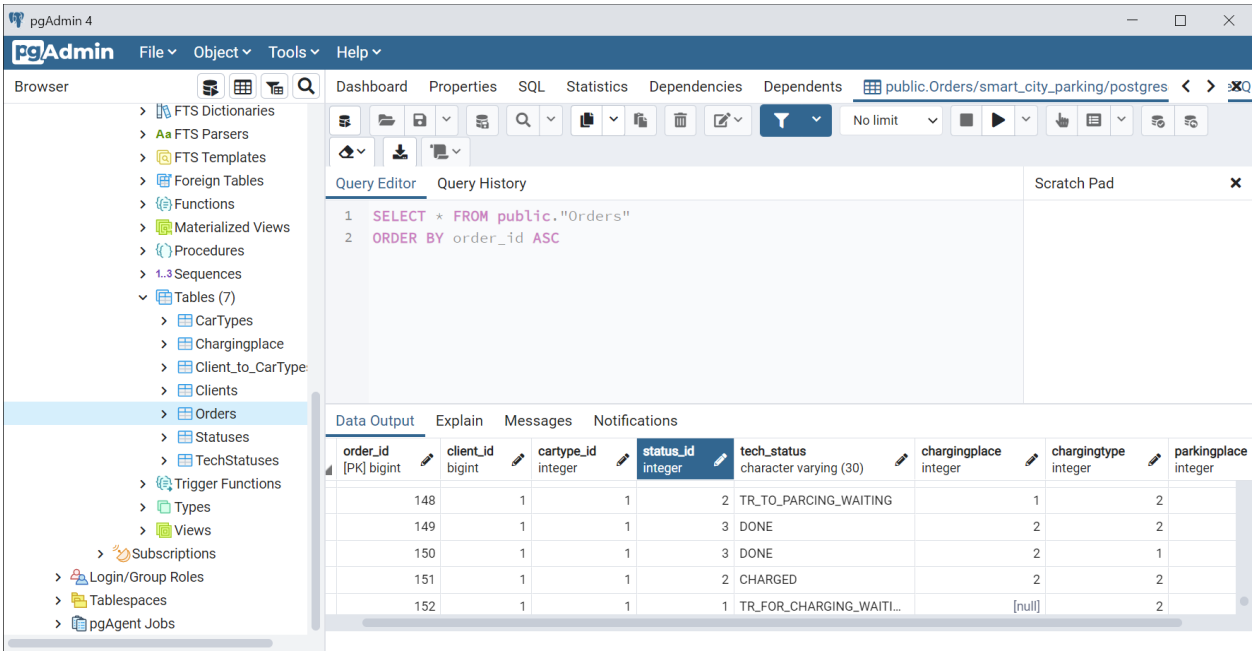
@ статус для клиента - колонна, связанная вторичным ключом со «словарём» статусов для клиента (для «словаря» это первичный ключ)

@ технический статус заказа - колонна, связанная вторичным ключом со «словарём» технических статусов (для «словаря» это первичный ключ)

@ номер зарядочного места (единственная колонна в таблице, где допустимо значение «HULL»)

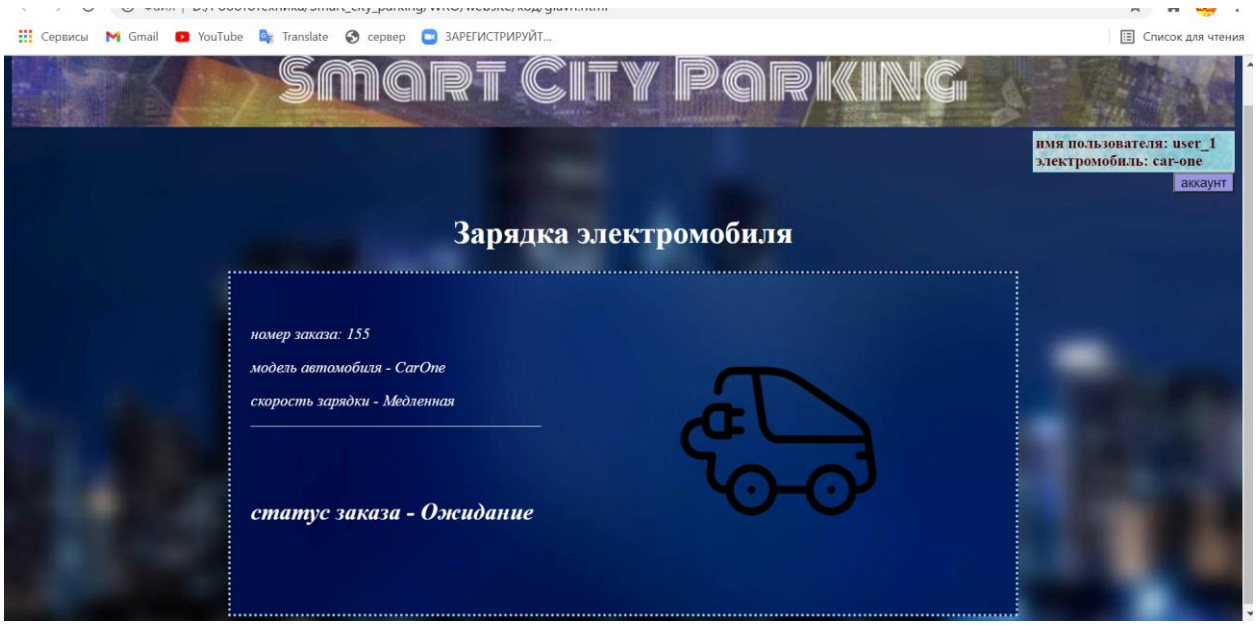
@ номер парковочного места

)



Сервер

Сервер работает одновременно с сайтом, роботами и базой данных. На страницу пользователя сервер отправляет данные, необходимые для отображения информации по заказу. также с помощью сервера происходит вход в аккаун (в базе данных хранятся логин и пароль, привязанные к конкретному пользователю, которые сервер получает оттуда и анализирует, после чего отправляет необходимые данные на сайт). На сервере происходит оформление заказа: он получает все необходимые параметры(выбранные пользователем), остальные уже хранятся в базе данных и привязываются к заказу, после оформления заказа, на сайте выводится его актуальный статус(заказа), их на данный момент 3(статуса): “ожидание”(электромобиль стоит на парковке и находится в очереди), “заряжается” и “готово”.



Также есть технические статусы заказа (их куда больше чем статусов для пользователя и нужны они для лучшей работы очереди), когда приходит время перевозить ставить на зарядку или снимать с неё электромобиль (что именно сделать и с каким электромобилем, определяется с помощью специальной логики в которой используются в том числе технические статусы заказа, статусы зарядочных мест, расположение заправочного люка на электромобиле и многое другое), сервер отправляет сообщение необходимому роботу(“Парковщику” или “Заправщику”) с необходимыми параметрами для его работы.

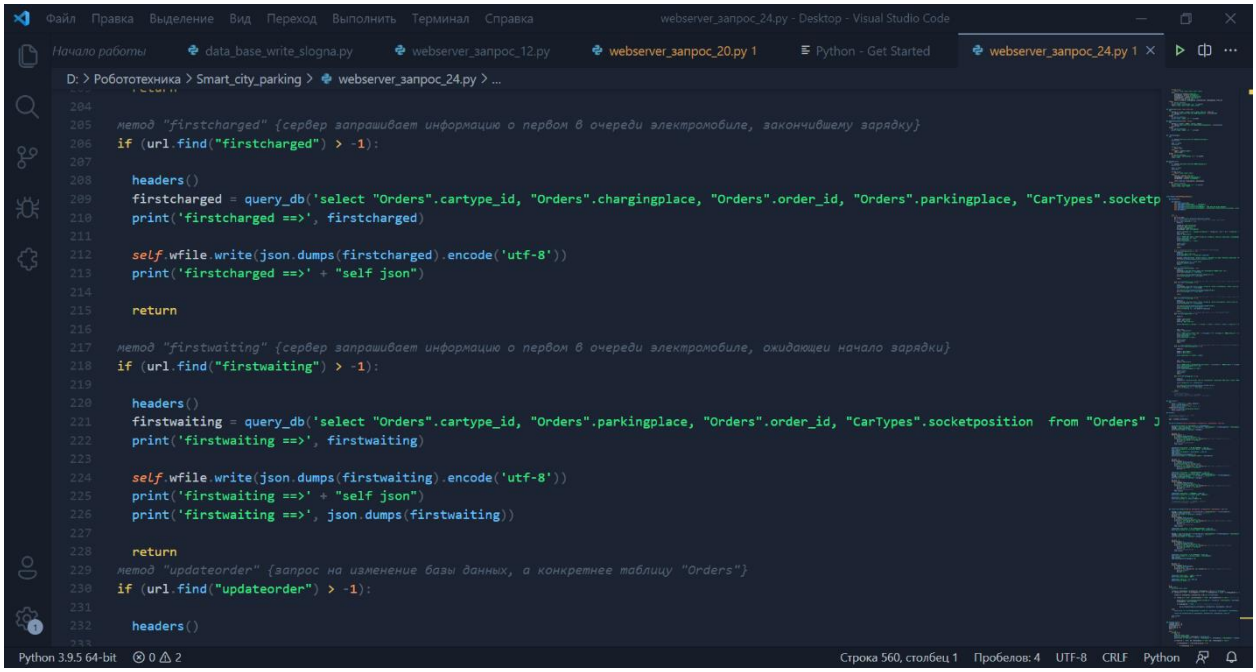
Итоги

Мы узнали многое об электромобилях и зарядных станциях для них, доработали вакуумную присоску, сконструировали, собрали и запрограммировали роботов: “заправщика” и “парковщика”. Создали веб-сайт и сервер, работающий с базой данных (созданной нами). Так же смоделировали и распечатали разъёмы, штекеры и другие части проекта. После этого мы создали макеты автомобилей и соединили все части проекта в заправочную станцию.

Приложение

(фрагменты кода)

(Web-сервер)



(robotNet)

```
from socket import *
def initMailBox():
     "This function creates mailbox and opens socket"
     mbx = socket(AF_INET, SOCK_DGRAM)
     mbx.bind(('',12345))
     return mbx
def sendMessage(message):
     s=socket(AF_INET, SOCK_DGRAM)
     s.setsockopt(SOL_SOCKET, SO_BROADCAST, 1)
     s.sendto(message.encode('utf-8'),('255.255.255.255',12345))
def getMessage(mailbox):
     return mailbox.recv(1024).decode('utf-8')
```

(qr-code coordinates)

```
import cv2
import numpy as np
from subprocess import call


pic = '/home/pi/Downloads/foto.jpg'
```

```
def makePhoto(pic):
    call(['fswebcam',pic])
makePhoto(pic)

img = cv2.imread('foto.jpg')
height, width = img.shape[:2]
print(height,"---", width )

detector = cv2.QRCodeDetector()
# detect and decode
data, bbox, _ = detector.detectAndDecode(img)
print(data,bbox)
if bbox  is not None:
    print(bbox,data)
    x1 = bbox[0][0][0]
    y1 = bbox[0][0][1]
    x2 = bbox[0][2][0]
    y2 = bbox[0][2][1]
    print("x1= ",x1,"y1= ",y1,"x2= ",x2,"y2= ",y2)
    x = ((x2-x1) / 2) + x1
    y = ((y2-y1) / 2) + y1
    print("x= ",x,"y= ",y)
    cv2.circle(img, (int(x), int(y)), 10, (0, 0, 255), -1)
    if x < ((width / 3)):
        location = "left"
    elif x > ((width / 3))*2:
        location = "right"
    else:
        location = "center"
    print(location)
else:
    print("none qr")

cv2.imshow("img", img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Заправщик

```
from ev3dev.ev3 import *

from time import sleep
from subprocess import call
import robotNet
mA = MediumMotor('outA')
mB = MediumMotor('outB')
mC = MediumMotor('outC')
mD = MediumMotor('outD')
btn = Button()
#cl1 = LightSensor ('in1')
#cl2 = LightSensor ('in2')
```

```
#cl3 = LightSensor ('in3')
#cl4 = LightSensor ('in4')

print('go')

#mbox = robotNet.initMailBox()
#msj = robotNet.getMessage(mbox)
#msk = robotNet.getMessage(mbox)
#msg = robotNet.getMessage(mbox)
msg = int(input())
print(msg)

def povorot(a,b):
    mA.run_timed(time_sp= a, speed_sp=b)
    mB.run_timed(time_sp= a, speed_sp=b)
    mC.run_timed(time_sp= a, speed_sp=b)
    mD.run_timed(time_sp= a, speed_sp=b)
    return 0

def vpered (a,b):
    mA.run_timed(time_sp=a, speed_sp=-b)
    mC.run_timed(time_sp=a, speed_sp=b)
    return 0

def nazad (a,b):
    mA.run_timed(time_sp=a, speed_sp=b)
    mC.run_timed(time_sp=a, speed_sp=-b)
    return 0

def vlevo (a,b):
    mB.run_timed(time_sp=a, speed_sp=b)
    mD.run_timed(time_sp=a, speed_sp=-b)
    return 0

def vpravo (a,b):
    mB.run_timed(time_sp=a, speed_sp=-b)
    mD.run_timed(time_sp=a, speed_sp=b)
    return 0

def Vpravo (a,b):
    mC.run_timed(time_sp=a, speed_sp=b)
    return 0
```

“Парковщик”

```
const int inaPin1 = 7;
const int inbPin1 = 9;
const int inaPin2 = 13;
const int inbPin2 = 11;
const int inaPin3 = 40;
```

```
const int inbPin3 = 42;
const int inaPin4 = 36;
const int inbPin4 = 38;

const int pwmPin1 = 3;
const int pwmPin2 = 5;
const int pwmPin3 = 44;
const int pwmPin4 = 46;

const int diagaPin = 10;
const int diagbPin = 12;
const int buttonPin = 2;
const int trimPin = A0;
int _speed = 50;
#define SIG_A_PIN 3
#define INT_A_PIN 2

long encA = 0;
long encB = 0;
long i = 0;
const int message = 2;
int line = 470;

void vpered() {
  digitalWrite(inaPin3, HIGH);
  digitalWrite(inbPin3, LOW);
  analogWrite(pwmPin3, _speed);

  digitalWrite(inaPin1, LOW);
  digitalWrite(inbPin1, HIGH);
  analogWrite(pwmPin1, _speed);

  digitalWrite(inaPin2, LOW);
  digitalWrite(inbPin2, HIGH);
  analogWrite(pwmPin2, _speed);

  digitalWrite(inaPin4, HIGH);
  digitalWrite(inbPin4, LOW);
  analogWrite(pwmPin4, _speed);
}

void vpravo() {
  digitalWrite(inaPin3, HIGH);
  digitalWrite(inbPin3, LOW);
  analogWrite(pwmPin3, _speed);

  digitalWrite(inaPin1, HIGH);
  digitalWrite(inbPin1, LOW);
  analogWrite(pwmPin1, _speed);
```