**ДИПЛОМЕН**

**ПРОЕКТ**

**Тема: Разработване на управление на роботизирана количка чрез Wi-Fi. Създаване на макет на роботизирана количка, управлявана чрез Wi-Fi**

*Ученик:* ***Петко Петров Йонков***

***Професия:*** *код 481020 „Системен програмист“*

***Специалност:*** *код 4810201 „Системно програмиране“*

***Консултант:*** *инж. Константин Колев*

Сопот, 2023 г.

**СЪДЪРЖАНИЕ**

Увод…………………………………………………………………………………... 3 стр.

Цел и задачи на разработката………………………………………………….…… 3 стр.

**ГЛАВА I** ………………………………………………………………………..………………..4 стр.

1.1. Предпоставка за създаване на продукта………………………………………..4 стр.

1.2. Съществуващи решения и реализации………………………………………... 4 стр.

**ГЛАВА II**……………………………………………………………………………. 6 стр.

2.1. Избор и описание на развойна платка………………………………………… 8 стр.

2.2. Описание на развойната среда …………………………………………………8 стр.

2.3. Изчертаване на блокови схеми на алгоритмите……………………………… 9 стр.

2.4. Описание на алгоритмите…………………………………………………….. 10 стр.

**ГЛАВА III**………………………………………………………………………….. 11 стр.

3.1 Описание на използваните модули…………………………………………… 11 стр.

3.2 Изчертаване и описание на блок схема на връзките………………………… 12 стр.

3.3 Изчертаване и описание на принципна електрическа схема……….………. 12 стр.

3.4 Схема на опитна постановка………………………………………………….. 13 стр.

3.5 Резултати и тестове от реализирания експериментален модел…………… 14 стр.

**ЗАКЛЮЧЕНИЕ**………………………………………………………………….... 15 стр.

4.1 Обобщение на постиженията в дипломната работа…………………………. 15 стр.

4.2 Приложение на разработката………………………………………………….. 16 стр.

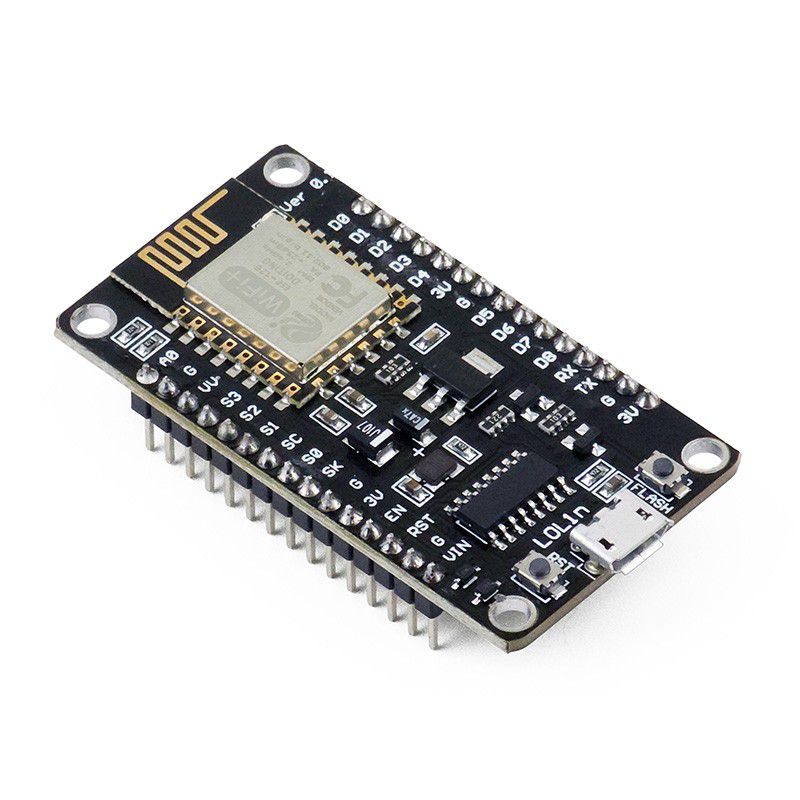
4.3 Тенденции за усъвършенстване на разработката……………………………. 17 стр.

**ИЗПОЛЗВАНИ ТЕРМИНИ И СЪКРАЩЕНИЯ**……………………………… 18 стр.

**ИЗПОЛЗВАНА ЛИТЕРАТУРА**……………………………………………… 18 стр.

**Увод**

В съвременния свят, където технологичните иновации изпъкват като катализатор за промените в различни области на живота, роботиката заема централно място в разработването на интелигентни системи и устройства. Дипломният проект, представен в този документ, не само отразява използването на високотехнологични средства, но и насочва вниманието към създаването на иновационна количка-робот, оборудвана с управление чрез ESP-8266.



**Фигура 1.1** Микроконтроллер ESP-8266

ESP-8266 е ключов компонент в този проект, представлява микроконтролер с вградена Wi-Fi поддръжка, който открива широк спектър от възможности за свързване и контрол на устройства по безжичен път. Този микроконтролер не само представя стабилно и бързо безжично свързване, но и съчетава гъвкавостта на програмируемите микроконтролери с удобството на безжичната комуникация. Този подход не само предоставя възможност за дистанционен контрол, но и създава почва за развитие на автономни функционалности и връзка с различни сензори, които биха подобрили възможностите на работа.

За да илюстрираме значението на подобни технологии в реален контекст, можем да обърнем поглед към роувърът Opportunity (Опъртюнити) - един от двата марсохода, които са част от космическата програма за изследване на Марс. Opportunity, като символ на възможностите на роботиката, каца на повърхността на планетата Марс през януари 2004 г. и продължава на функционира в следващите 15 години. Свързвайки се с Земята чрез радиовълни, този робот представлява изключителен пример за това как безжичната комуникация и управление могат да бъдат от ключово значение в дългосрочните и изключително изискани мисии. Такъв пример подчертава не само технологичния напредък, но и възможностите за разширяване на хоризонта на роботиката в различни области на изследователската и индустриална дейност.

**Цели и задачи на разработката:**

1. Подробно проучване на информация за разработката и проектирането на WI-FI количката;
2. Проектиране на принципна схема;
3. Проектиране на изходен код;
4. Свързване на компонентите по проектираната схема;
5. Разработване на функционален макет;

**ПЪРВА ГЛАВА** ПРОУЧВАНЕ НА ПОТРЕБНОСТИТЕ, НУЖДИТЕ, ЦЕЛЕВА ГРУПА

**1.1. Предпоставка за създаване на продукта**

В нарастващата тенденция на технологичното развитие, иновационният продукт, представен в този текст, не само предоставя безкрайни възможности, но и има широко приложение в различни области на науката и техниката. Създаден с цел лесно интегриране на сензори, този продукт не просто отговаря на изискванията на различни сфери на науката, но ги превъзхожда, като предоставя гъвкавост и иновационен подход. От медицината и околната наука до тежките условия в индустрията, продуктът предоставя решения за разнообразни предизвикателства. Безопасната безжична комуникация не само осигурява надеждност, но и подчертава значението на продукта в работата в неблагоприятни за хората условия. В комбинация с лесно разширяемата функционалност на ESP-8266, този продукт предоставя несравними възможности за програмиране и настройка, които отварят нови хоризонти за инженерите и разработчиците.

Допълнително, продуктът може да бъде успешно приложен в областта на научните изследвания и космическите мисии. Възможността за изследване на далечни светове - от планети, луни, звезди, дори до далечни галактики - открива нови възможности за астрономите и космонавтите. Този продукт става ключов инструмент за научния напредък и разширяване на човешкото познание за Вселената. Освен това, продуктът може да се използва за мониторинг и управление на околната среда, включително следене на климата, екосистемите и важни екологични параметри. Такива приложения са от съществено значение за устойчивото развитие и опазването на природните ресурси на планетата.

В заключение, този иновативен продукт представлява не просто технологична стъпка напред, а цялостно решение с безгранични възможности и потенциал за приложение в различни научни, индустриални и космически сфери. Съчетавайки лесната интеграция, безопасната комуникация и програмируемостта, този продукт предизвиква новаторския подход към решаването на предизвикателства в разнообразни области на човешката дейност.

**1.2. Съществуващи решения и реализации**

Tesla е фирма създадена от Илън Мъск която произвежда електрически автомобили. Фабриките на Tesla които произвеждат тези иновативни коли са главно съставени от поточни линии и роботи които построяват колите вместо хората. Чрез този подход Tesla спестява пари от персонал и в същото време повишава производителността си, тъй като роботите работят по-бързо и по-ефективно от хората.



**Фигура 1.2** Роботизираната фабрика на Tesla за автомобили

Amazon е фирма която се специализира в продаването на стоки и услуги по интернет. Amazon е най-голямата компания по оборот в бранша. Такава огромна международна фирма разположена на няколко континента се нуждае от много голям персонал а това означава още по-големи разходи за фирмата. С цел намаляване на разходите и подобрение на качеството на работа Джеф Безос, основателят и главен изпълнителен директор на Amazon, прие план за разработването на роботизирани складове. Тези складове са снабдени с роботи наподобяващи колички и човекоподобни роботи които пренасят стоките от склада към камионите за доставки и обратното, от камионите до складовете за съхранение. Освен роботизираните складове Amazon има планове да премине на следващото което за тях е доставяне на пратки чрез дронове. Идеята е дрон да идва с пратката до домът ви, да го остави пред вратата ви и да се върне обратно в Amazon. Така доставките ще станат много по-бързи и ще се спестява време и ресурси. Освен това вие, клиентът, ще бъдете още по-доволни от дейността на Amazon.



**Фигура 1.3** Роботизиран склад на Amazon

След започването на пандемията от COVID-19 фирми като Glovol, Takeaway, UberEats и Panda Express специализиращи се в поръчка и доставка на храна и други битови консумативи разчитат на хора за доставките и успеха на бизнеса им. Обаче в някои държави се правят експерименти с роботи доставчици (Delivery Robots). Тези роботи са напълно автоматизирани служещи си с множество сензори за навигация. Чрез които роботите навигират улиците на градовете до дестинацията на която са изпратени. Разбира се тези роботи все още не са перфектни, случвало се е да засядат или да закъсват по пътя. Най-голямото препятствие за момента е непочистения сняг.



**Фигура 1.4** Робот доставчик на UberEats

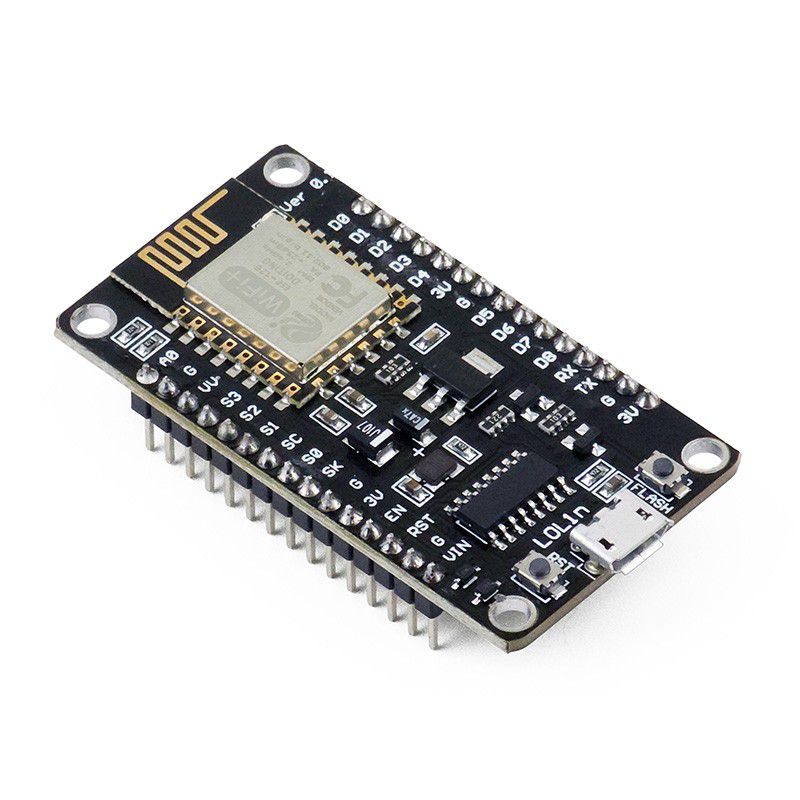
**ВТОРА ГЛАВА** ОПИСАНИЕ НА РАЗВОЙНАТА СРЕДА И АЛГОРИТМИТЕ НА РАЗРАБОТКАТА

**2.1. Избор и описание на развойна платка**

За разработката на количката се използва платката ESP-8266 (още познат като NodeMCU) в комбинация с ESP12E Motor Shield за контролиране на двигателите. ESP-8266 е универсален 32-битов микроконтролер от Espressif Systems с вграден Wi-Fi микрочип снабден с TCP/IP мрежови софтуер.

**2.1.1 Характеристики на ESP-8266**

* Процесор: L106 32-bit RISC микропроцесор базиран на Tensilica Diamond Standard 106 Micro работещ от 80 до 160 MHz.
* Работно напрежение: 3,3 V
* Входно напрежение: 7-12 V
* Памет:
  + Флаш памет: 4MB
  + SRAM: 64 KB
* Пинове:
  + 16 дигитални пина за вход и изход (DIO)
  + 1 аналогов входен пин (ADC)
* Serial Peripheral Interface (SPI)
* софтуерна имплементация на I²C
* I²S интерфейси с DMA (споделяне на изводите с GPIO)
* UART на специални изводи, както и възможност за активиране на UART само за предаване на GPIO2
* 10-bit ADC (преобразувател на аналогов към дигитален сигнал)

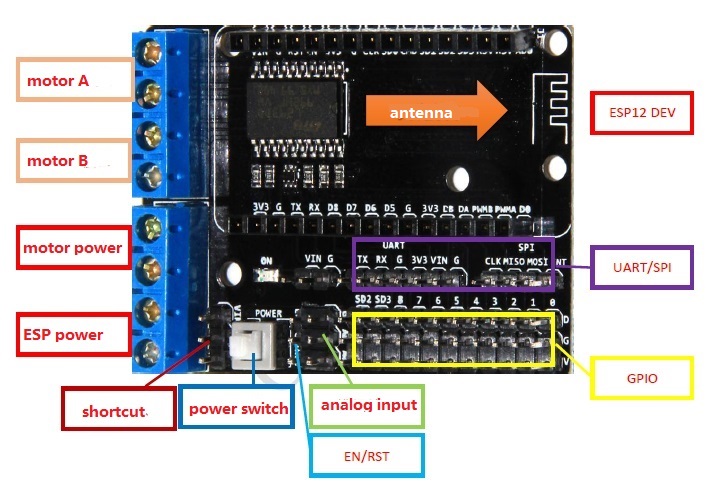


**Фигура 2.1** микроконтролер ESP-8266

**2.1.2 Описание на ESP12E Motor Shield**

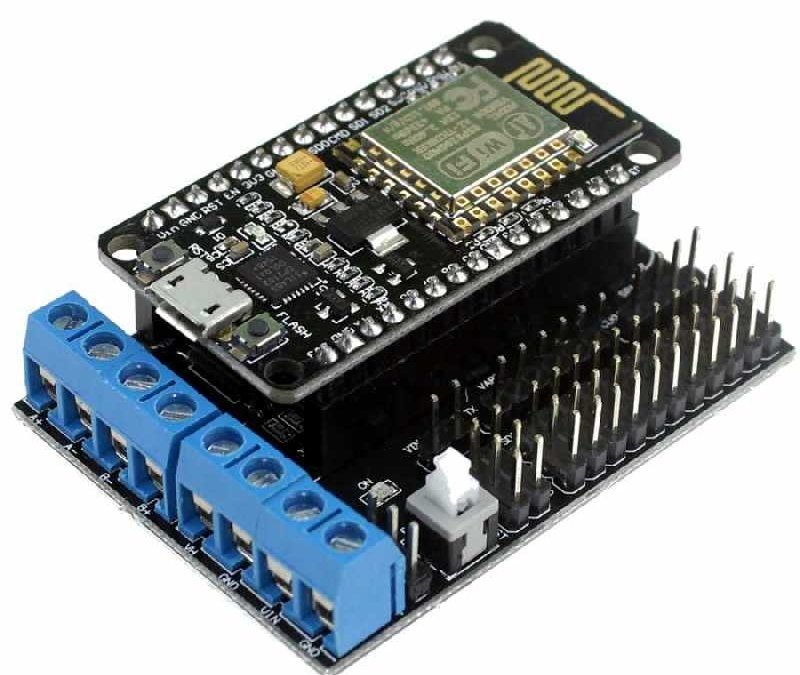
ESP12E Motor Shield е разширителна платка, която позволява на микроконтролера ESP-8266 да управлява два (постоянно такови / DC) двигателя или един стъпков двигател. ESP12E Motor Shield използва L293D двоен H-bridge, който му позволява да управлява посоката и скоростта на двигателите с номинално напрежение между 5 и 36V и електрически ток от 600mA до 1,2A с външен източник на напрежение. Тази платка ви позволява да използвате:

* До два DC мотора или един постъпков двигател
* аналогови сензори
* GPIOs
* I²C, SPI и UART шини

****

**Фигура 2.2** ESP12E Motor Shield

ESP12E Motor Shield е съвместим с ESP-8266 (още познат като NodeMCU) като се поставя директно на микроконтролера. Захранването на двигателите се свързва към терминален блок VM/GND, а това на платката - към терминален блок VIN/GND. Възможно е изводите VIN и VM да се свържат с мост ако захранването на двигателите е същото като захранването на ESP-8266. Двигателите се свързват към terminal blocks A+,A-,B+,B-.



**Фигура 2.3** ESP-8266 прикачено към ESP12E Motor Shield

**2.2. Описание на развойната среда**

Arduino Integrated Development Environment (IDE) е безплатна среда за

програмиране, която се използва за програмиране на микроконтролерите Arduino. Лесна и интуитивна за инсталация и конфигуриране. Arduino IDE също предоставя инструмент за конфигуриране на портовете и драйверите за микроконтролерите. Тя е съвместима с повечето операционни системи, включително Windows, Mac OS X и Linux.

Arduino IDE поддържа и работи с различни версии на микроконтролерите Arduino, като Arduino Uno, Arduino Nano, Arduino Mega и други. Arduino IDE има лесен за използване интерфейс, който позволява на потребителите да програмират микроконтролерите Arduino по лесно достъпен и разбираем начин. Също така поддържа много езици, включително английски, испански, френски, италиански, немски и др., което прави платформата още по-разбираема и достъпна за много потребители. Развойната среда съдържа всички необходими инструменти за програмиране и тестване на Arduino кода, включително текстов редактор, компилатор, монитор за серийна комуникация и др.

Един от главните компоненти на Arduino IDE е текстовият редактор. Той е прост за използване и има функции за подсветка на синтаксиса, автодовършване на кода и други полезни функции. Arduino IDE използва C++ като основен език за програмиране на микроконтролерите.

Arduino IDE има вградени библиотеки, които съдържат функции за

управление на различни периферни устройства на микроконтролерите Arduino. Тези библиотеки могат да бъдат използвани за управление на дисплеи, сензори, мотори, светодиоди и други.

Arduino IDE има инструменти за компилиране на кода и качване на

програмата в микроконтролера. Това позволява на потребителите да проверят кода си и да го качат на микроконтролера в една лесна стъпка.

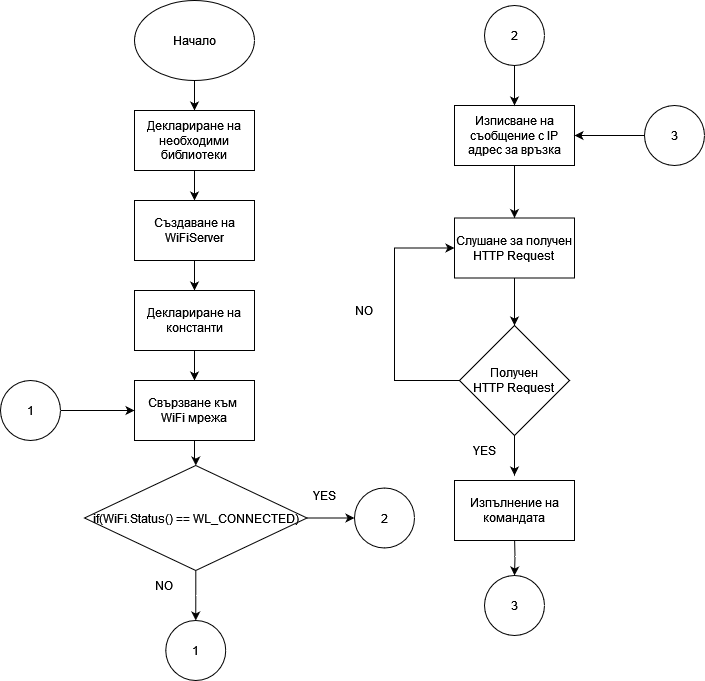
Arduino IDE също така има монитор за серийна комуникация, който позволява на потребителите да изпращат и получават данни през серийната връзка на микроконтролера. Това е полезно за тестване и отстраняване на проблеми в кода.

В допълнение, Arduino IDE е отворен код и много от потребителите могат да създават и споделят свой библиотеки и програми. Това прави платформата много популярна сред инженерите, студентите и хобистите, които търсят лесен начин за

програмиране на микроконтролерите Arduino.

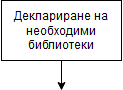
**Фигура 2.4** Arduino IDE

**2.3. Изчертаване на блокови схеми на алгоритмите**

****

**Фигура 2.5** Блокова схема на кода

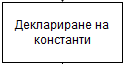
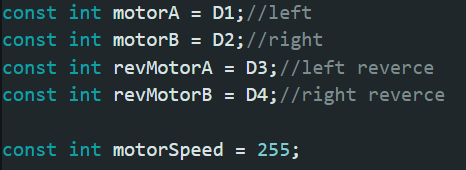
**2.4. Описание на алгоритмите**

**2.4.1 Импортиране на необходими библиотеки**

**Фигура 2.6** Блок и код за импортиране на библиотеки

Изобразения код импортира библиотеки, които предоставят функции за свързване на ESP-8266 с Wi-Fi.

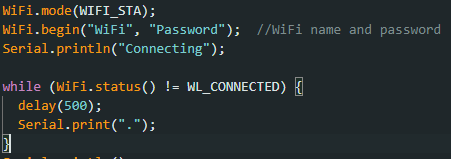
**2.4.2 Деклариране на константи и променливи**



**Фигура 2.7** Блок и код за деклариране на променливи и константи)

* MotorA е променлива служеща за контролиране на двата леви DC двигателя, като с нея те се задвижват напред. Както името подсказва тази променлива използва PWM или ШИМ на български за контролиране на скоростта на двигателите.
* MotorB e променлива контролираща двата десни двигателя, като тя задвижва двигателите напред. PWM също се използва за контролиране на скоростта на двигателите.
* revMotorA е променлива контролираща двата леви двигателя, само че тя се използва, когато искаме да накараме двигателите да се движат на обратно. Тази променлива не използва PWM
* revMotorB е променлива контролираща двата десни двигателя също както предходната константа тя се използва, когато искаме да накараме двигателите да се движат на обратно. Тази променлива не използва PWM
* motorSpeed задава скоростта, с която ще се движат двигателите.

**2.4.3 Свързване към Wi-Fi мрежата**

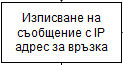


**Фигура 2.8** Блок и код за свързване към WiFi мрежа

Показаният код свързва ESP-8266 със Wi-Fi мрежа нека го разгледаме подробно.

* WiFi.Mode(WIFI\_STA); - Е метод, който задава режима на работа на ESP-8266. Параметъра WIFI\_STA настройва ESP-8266 в режим Station. В режим Station ESP-8266 се свързва към съществуваща Wi-Fi мрежа. По същество това е режимът, в който се присъединява към съществуваща мрежа, вместо да създава своя собствена точка за достъп.
* WiFi.Begin(“Wi-Fi”, “Password”); - Това е метод с два параметъра, който започва връзката между ESP-8266 и Wi-Fi. Двата параметъра са за името и паролата на Wi-Fi мрежата с която трябва да се свърже.
* Serial.println(“Connecting”); изписва на сереиния монитор съобщение, че е започнало свързване с мрежата.
* Останалата част от кода проверява през 500 милисекунди дали връзката е установена.

**2.4.4 Изписване на съобщение с IP адрес за връзка**





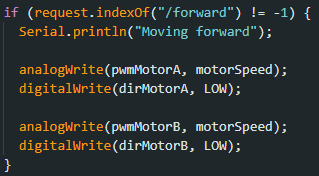
**Фигура 2.9** Блок и код за изписване на съобщение с IP адрес за връзка

Показаният код на снимката 2.4.4 служи за изписването на IP адреса на, който се намира сайта през, който ще можете да управлявате количката робот. WiFi.localIP() е метода, който показва IP-то. А espServer.begin(); се използва за да се стартира web сървър на ESP-8266.

**2.4.7 Получаване и изпълняване на команда - HTTP request**

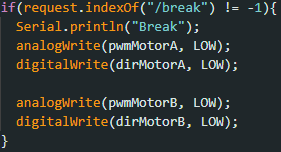
След като от сайта, с които се управлява количката бъде натиснат бутон (било то за посока на движение или за спиране) се изпраща HTTP request към микроконтролера и се проверява каква е командата и се изпълнява.

Ако е HTTP request за движение напред. На серииния монитор се изписва съобщение с посоката на движинеие. Това съобщение служи за по-лесно програмиране и тестване на управлението на количката за това то не се изписва никъде другаде освен на серииния монитор, до който имат имат достъп само разработчиците. След изписването на съобщението се задава с метода analogWrite() скоростта на мотор А и мотор Б и се спира движението назад чрез метода digitalWrite() като му се придаде LOW за втори параметър.



**Фигура 2.10** Код за движение на двигателите напред.

При получена заявка за спиране, на всички променливи не зависимо дали са за движение на двигателите напред или назад се задава ниско активно ниво LOW. По този начин двигателите биват спрени. Отново се изписва на серииния монотор състоянието на двигателите с цел по-лесно разработване.

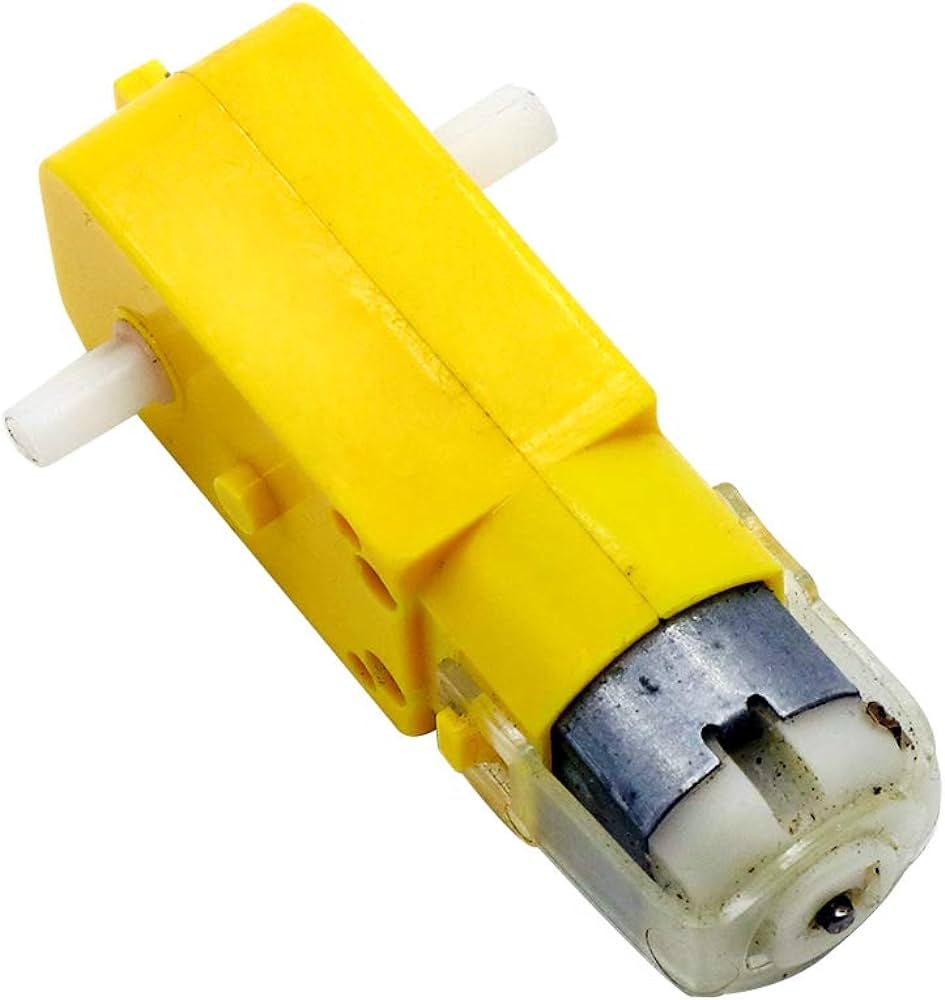


**Фигура 2.11** Код за спиране на двигателите

**ТРЕТА ГЛАВА** ОПИСАНИЕ НА РЕАЛИЗИРАНИЯ ЕКСПЕРИМЕНТАЛЕН МОДЕЛ

**3.1 Описание на използваните модули**

* **DC Двигател -** Постояннотоков електрически двигател



**Фигура 3.1** DC двигател

Постояннотоков електрически двигател (DC двигател) е електрически двигател, захранван с постоянен ток. Тази група двигатели се разделя на два вида:

1. Колекторни електродвигатели
2. Безколекторни електродвигатели

Колекторните електродвигатели са първите електродвигатели, намерили практическо приложение. Тъй като четките им се износват по време на работа и се нуждаят от честа подмяна, откритите впоследствие асинхронни двигатели, работещи с променливо напрежение, са заели местата на колекторните постояннотокови двигатели в много сфери. Въпреки това колекторните постояннотокови двигатели продължават и до днес да бъдат използвани без алтернатива на много места - автомобилни стартери, машини за хартия и други. Колекторните DC електродвигатели лесно могат да променят скоростта си чрез промяна на работното напрежение или силата магнитното поле.

Безколекторният електродвигател е модерен синхронен двигател, работещ на принципа на честотното регулиране със самосинхронизация, чрез изменение на вектора на магнитното поле на статора в зависимост от положението на ротора. Вентилните двигатели се определят като безколекторни двигатели на постоянен ток, понеже такъв двигател най-често се захранва от постоянно напрежение. Двигателят е създаден с цел подобряване свойствата на класическите постояннотокови мотори - избягване на бързоизносващия се колекторен блок с четки. За разлика от традиционния постояннотоков двигател, комутацията във вентилния се осъществява и управлява с електроника. Високите изисквания към изпълнителните механизми са обусловили употребата на тези специфични безконтактни трифазни постояннотокови двигатели.

В този проект се използват първия вид DC двигатели - Колекторни електродвигатели.

* **Захранване на ESP8266 и ESP12E Motor Shield**



**Фигура 3.2** Батерии за платките

Платката ESP12E Motor Shield може да управлява двигатели с волтаж от 5 до 35V с външно захранване, обаче в този проект захранването на DC двигателите и на ESP8266 е обединено следователно е необходимо и двете платки да работят с едно и също захранване. ESP8266 работи с максимум 10V затова е важно захранването на не го надвишава.

За целта са избрани 4 x 1.5V батерии общо 6V с които ще може да се захранват и двете платки без опасност за повреда на ESP8266. Разбира се при необходимост може да се използва и една 9V батерия без опасност от повреда. Ако желаем да използваме пълния потенциал на ESP12E Motor Shield ще трябва да използваме отделно захранване на платката ESP8266 и моторите.



* **Държач на батерии**

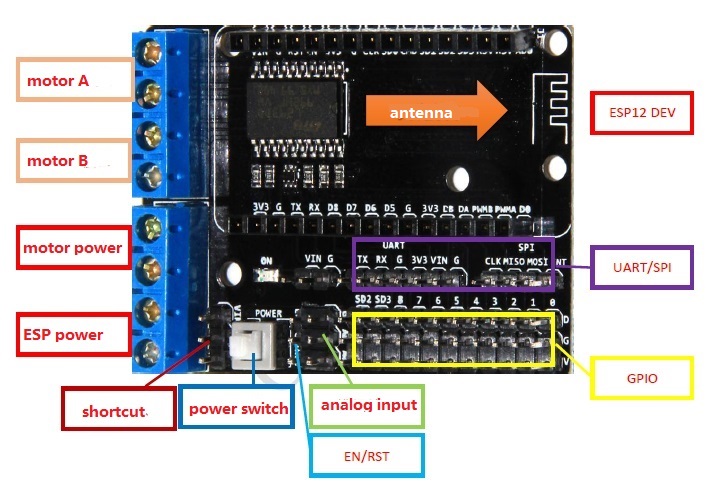
**Фигура 3.3** Държач за 4 батерии

Държачът за батерия е едно или повече отделения или камери за задържане на батерия. За сухи клетки държачът трябва също така да има електрически контакт с клемите на батерията. За мокрите клетки кабелите често се свързват към клемите на батерията, както се среща в автомобили или оборудване за аварийно осветление. Държачът за батерии е или пластмасова кутия с формата на корпуса, оформена като отделение или отделения, които приемат батерия или батерии, или отделен пластмасов държач, който се монтира с винтове, капси, лепило, двустранна лента или други средства . Държачите на батерии може да имат капак за задържане и защита на батериите или могат да бъдат запечатани, за да се предотврати повреда на вериги и компоненти от изтичане на батерията. Навита пружинна жица или плоски пластини, които притискат клемите на батерията, са двата най-често срещани метода за осъществяване на електрическа връзка вътре в държача. Външните връзки на държачите на батериите обикновено се осъществяват чрез контакти с щифтове, крачета за повърхностен монтаж, уши за запояване или проводници.

* **Описание на ESP12E Motor Shield**

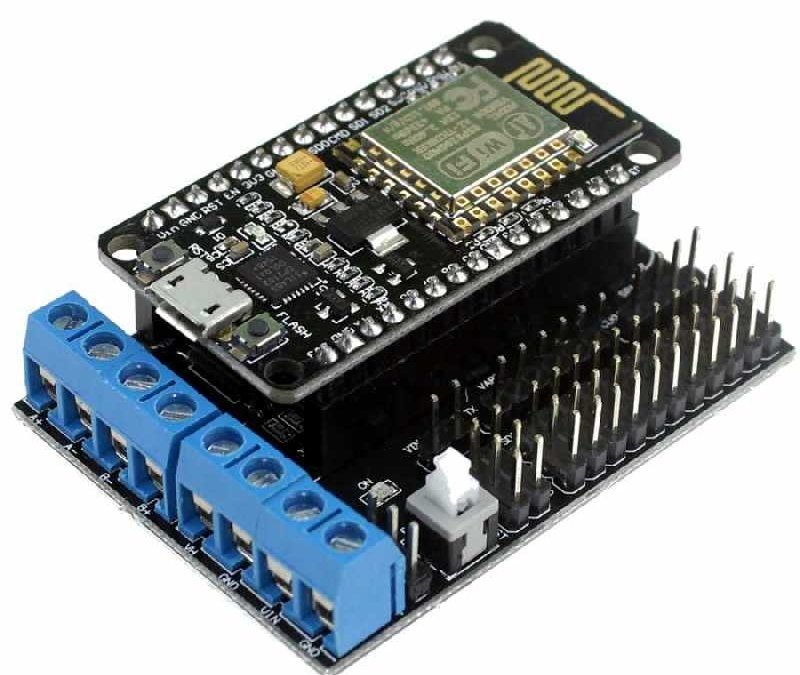
ESP12E Motor Shield е разширителна платка, която позволява на микроконтролера ESP-8266 да управлява два (постоянно такови / DC) двигателя или един стъпков двигател. ESP12E Motor Shield използва L293D двоен H-bridge, който му позволява да управлява посоката и скоростта на двигателите с номинално напрежение между 5 и 36V и електрически ток от 600mA до 1,2A с външен източник на напрежение. Тази платка ви позволява да използвате:

* До два DC мотора или един постъпков двигател
* аналогови сензори
* GPIOs
* I²C, SPI и UART шини

****

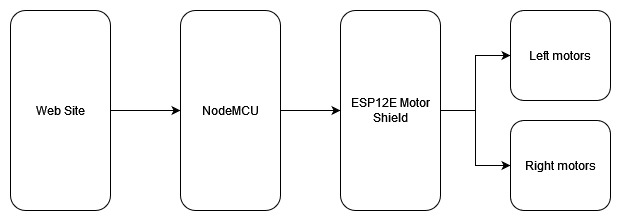
**Фигура 2.2** ESP12E Motor Shield

ESP12E Motor Shield е съвместим с ESP-8266 (още познат като NodeMCU) като се поставя директно на микроконтролера. Захранването на двигателите се свързва към терминален блок VM/GND, а това на платката - към терминален блок VIN/GND. Възможно е изводите VIN и VM да се свържат с мост ако захранването на двигателите е същото като захранването на ESP-8266. Двигателите се свързват към terminal blocks A+,A-,B+,B-.



**Фигура 2.3** ESP-8266 прикачено към ESP12E Motor Shield

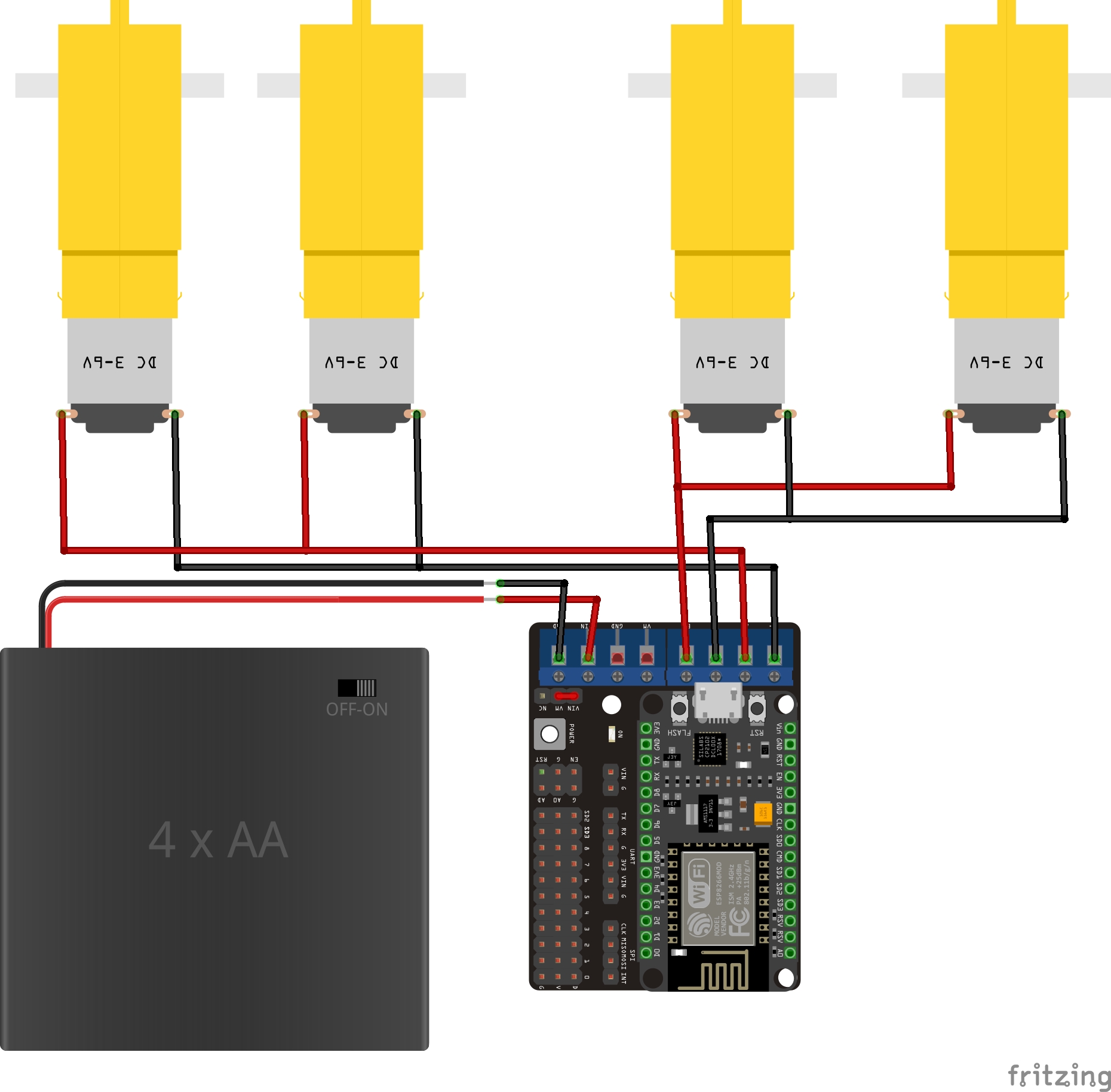
**3.2 Изчертаване и описание на блок схема на връзките**

****

**Фигура 2.3** Блок схема на връзките

**3.3 Изчертаване и описание на принципна електрическа схема**

**3.4 Резултати и тестове от реализирания експериментален модел**

**3.5 Схема на опитна постановка**

**Фигура 2.5** Схема на опитната постановка

**ЗАКЛЮЧЕНИЕ**

**4.1 Обобщение на постиженията в дипломната работа**

**4.2 Приложение на разработката**

**4.3 Тенденции за усъвършенстване на разработката**

**ИЗПОЛЗВАНИ ТЕРМИНИ И СЪКРАЩЕНИЯ**

SRAM

DIO

ADC

SPI

I²C

DMA

GPIO

UART

ШИМ / PWM

HTTP Request

**ИЗПОЛЗВАНА ЛИТЕРАТУРА**

[www.components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet](http://www.components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet)

[**https://en.wikipedia.org/wiki/DC\_motor**](https://en.wikipedia.org/wiki/DC_motor)

[**https://en.wikipedia.org/wiki/Battery\_holder**](https://en.wikipedia.org/wiki/Battery_holder)

Примери: /трябва да се следва точно формата/

1. Николов А., Програмиране на С++, Техника, София, 1998.

2. John A., Main Principles of C++ Programming, International Journal of Programming, Vol. 35, No 5, May 2001, pp. 112-183.

3. C++ Users’ Guide, [www.borlaland.com](http://www.borlaland.com/), 2002.