

**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

КУРСОВ ПРОЕКТ

Тема: BattleBot - Страхил

Ученник:

*Илиян Антов, 11в клас, №20
Ивайло Шомев, 11в клас, №18*

Научен ръководител:

Росен Витанов

**СОФИЯ
2019**

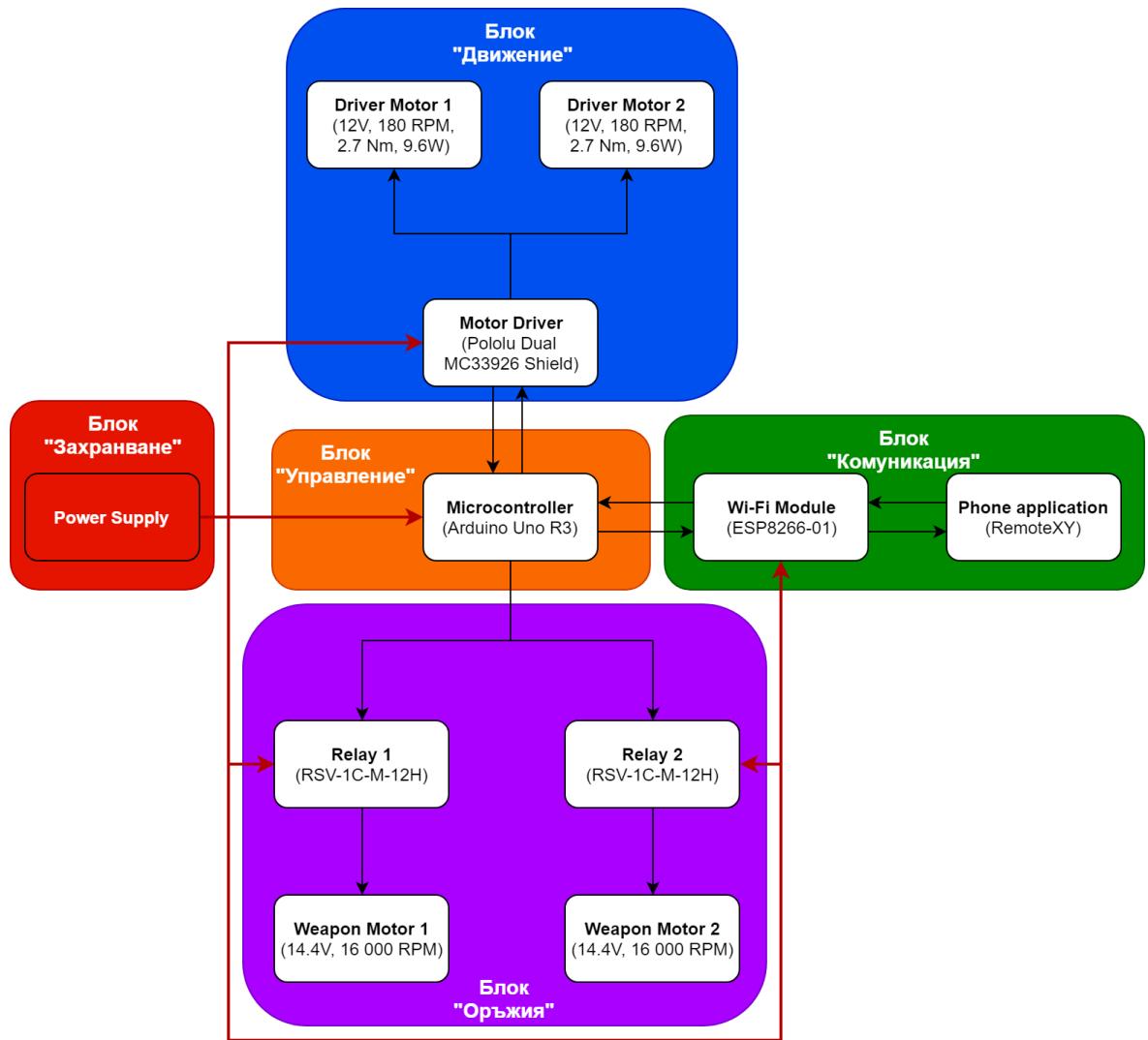
ЗАДАНИЕ

Да се проектира BattleBot, управляван чрез приложение за мобилен телефон. Да се реализира безжична връзка между телефон и wi-fi модул ESP8266-01, с която да бъде извършвана комуникацията. Да се реализира серийна комуникация между wi-fi модул ESP8266-01 и микроконтролер Arduino Uno R3 с цел изпълнение на командите, получени от мобилния телефон. Да се свърже и програмира драйвер за 2-та задвижващи мотора. Да се свържат и програмират 2 релета, позволяващи работата на 2-та мотора на оръжието.

1. ПРОЕКТИРАНЕ НА БЛОКОВАТА СХЕМА НА УСТРОЙСТВОТО

1.1. Блокова схема на устройството

На Фиг. 1.1. е показана блоковата схема на проектираното устройство.



Фиг. 1.1. Блокова схема на устройството

1.2. Описание на блоковете

Блок “Управление” – Служи за първоначално настройване на Wi-Fi модула ESP8266-01 от блок “Комуникация”, обработване на всички команди, получени от него, управление на драйвера на задвижващите мотори и управление на релетата, активиращи моторите на оръжието.

Блок “Комуникация” - Служи за комуникация с мобилното приложение и изпращане на получените от него команди към използваният микроконтролер Arduino Uno R3 от блок “Управление”.

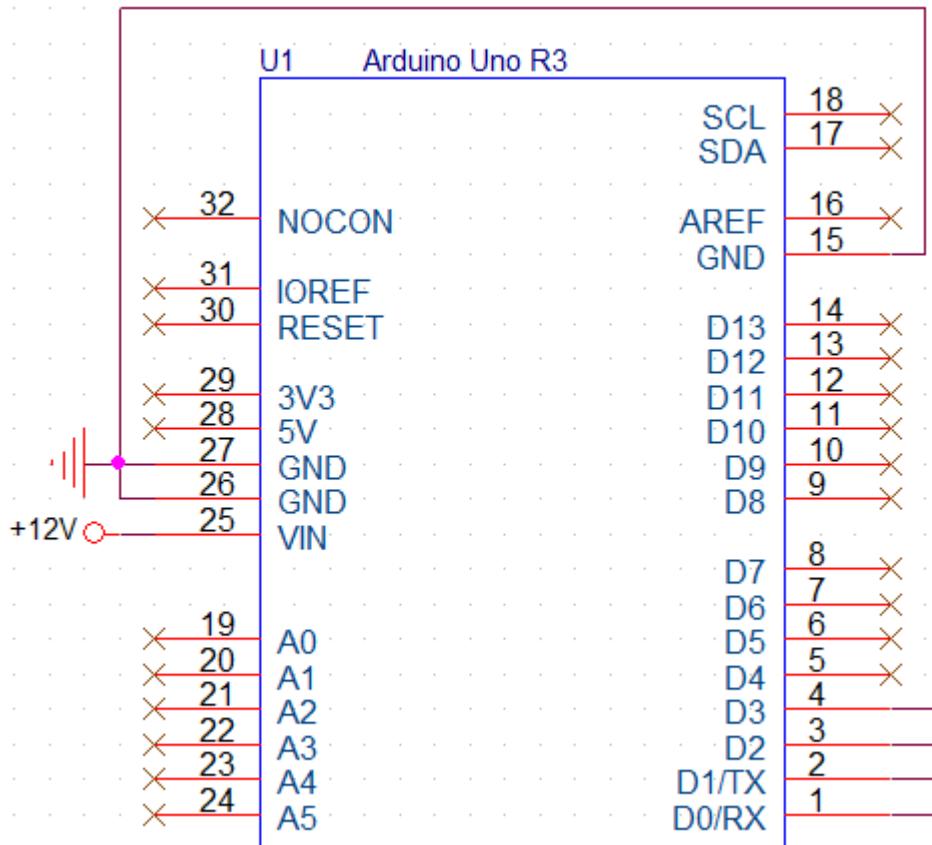
Блок “Движение” – Служи за управление на скоростта и посоката на движение на задвижващите мотори Driver Motor 1 и Driver Motor 2 в зависимост от получените команди от блок “Управление”.

Блок “Оръжия” – Служи за позволяване и забраняване на работата на двета мотора на оръжието Weapon Motor 1 и Weapon Motor 2 в зависимост от получените команди от блок “Управление”.

Блок “Захранване” – Служи за осигуряването на захранване на всички останали блокове.

2. ПРОЕКТИРАНЕ НА УСТРОЙСТВОТО

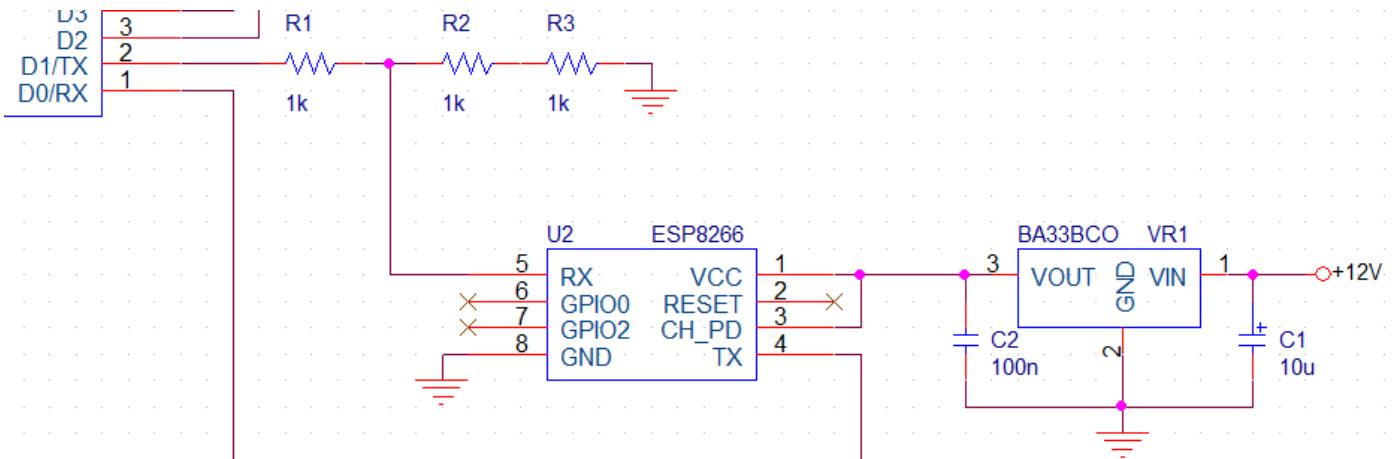
2.1. Принципна електрическа схема на Блок “Управление”



Фиг. 2.1. Принципна електрическа схема на блок “Управление”

На фиг. 2.1. е показана принципната електрическа схема на блок “Управление”. Блокът е базиран на микроконтролера Arduino Uno R3. Основната му задача е да извършва необходимите изчисления и да управлява блоковете “Движение” и “Оръжия”. Той също така служи за първоначалното настройване на Wi-Fi модула от блок “Комуникация”. Захранва се от 12V батерия, свързана към VIN и GND пиновете. Програмният код на микроконтролера е приложен и обяснен в [глава 3](#).

2.2. Принципна електрическа схема на Блок “Комуникация”



Фиг. 2.2. Принципна електрическа схема на блок “Комуникация”

Блокът служи за установяване на комуникация с мобилното приложение “RemoteXY”, получаване на команди от него и препращането им към блок “Управление” за обработка и изпълнение. Между Wi-Fi модула ESP8266-01 (U2) и микроконтролера Arduino Uno R3 е реализирана сериенна връзка посредством техните TX и RX пинове. Тъй като ESP8266-01 работи с високо ниво 3.3V на своите сериенни пинове, е нужно реализирането на напрежителен делител на RX пина му, за да се предотврати изгаряне. Той се определя от резисторите R1-R3 и се изчислява по формулата 2-1.

$$U_{2_{RX}} = U_{1_{TX}} * \frac{(R2+R3)}{(R1+R2+R3)}$$

$$U_{2_{RX}} = 5V * \frac{2k\Omega}{3k\Omega} \approx 3.3V$$

2-1. Формула за изчисление на входното напрежение на RX пина на Wi-Fi модула U2 при високо ниво на изхода на TX пина на микроконтролера U1

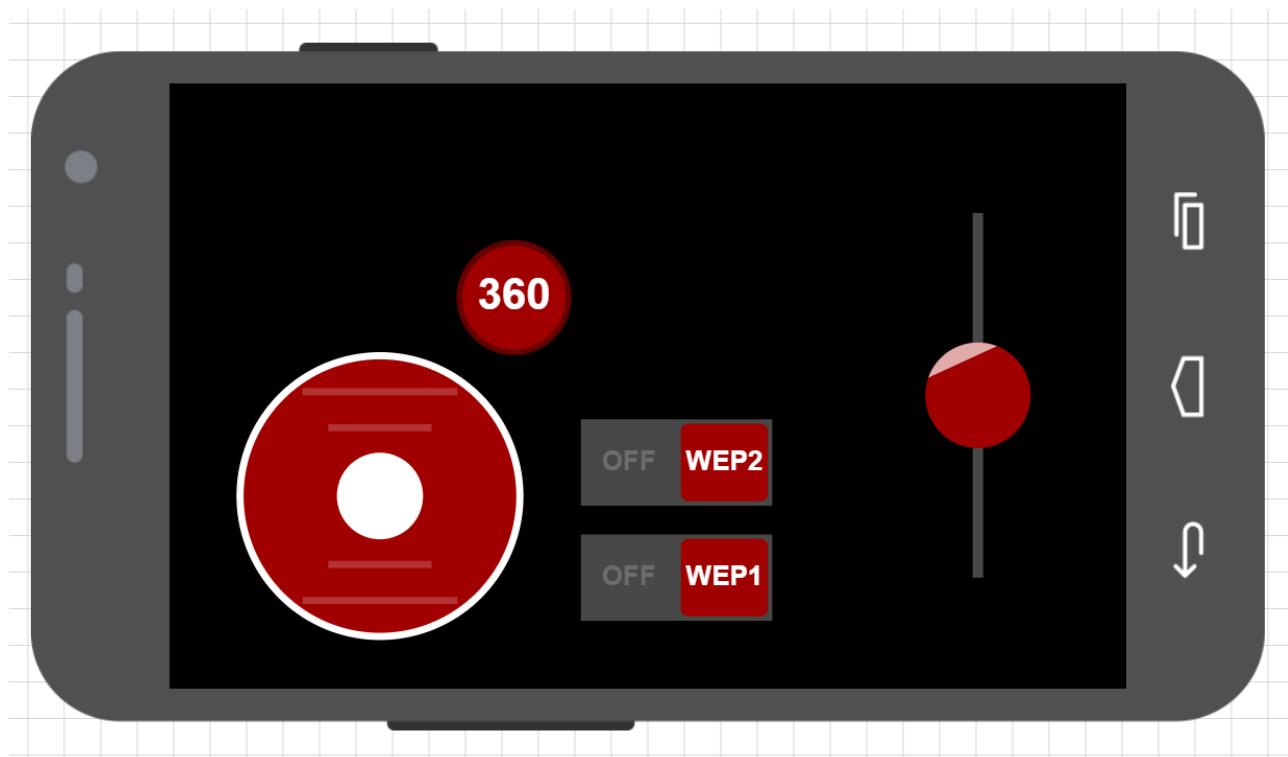
Тъй като Arduino Uno R3 възприема 3.3V за високо ниво, не е нужно реализирането на level shifter за другата връзка (TX пина на Wi-Fi модула към RX пина на Ардуиното).

Захранването на модула се извършва с помощта на регулатор BA33BCO (VR1), който превръща 12V на входа в нужните 3.3V на изхода си, свързан към VCC.

CH_PD пина, който играе ролята на enable, е свързан директно към VCC (неговото активно ниво), като по този начин се разрешава работата на модула.

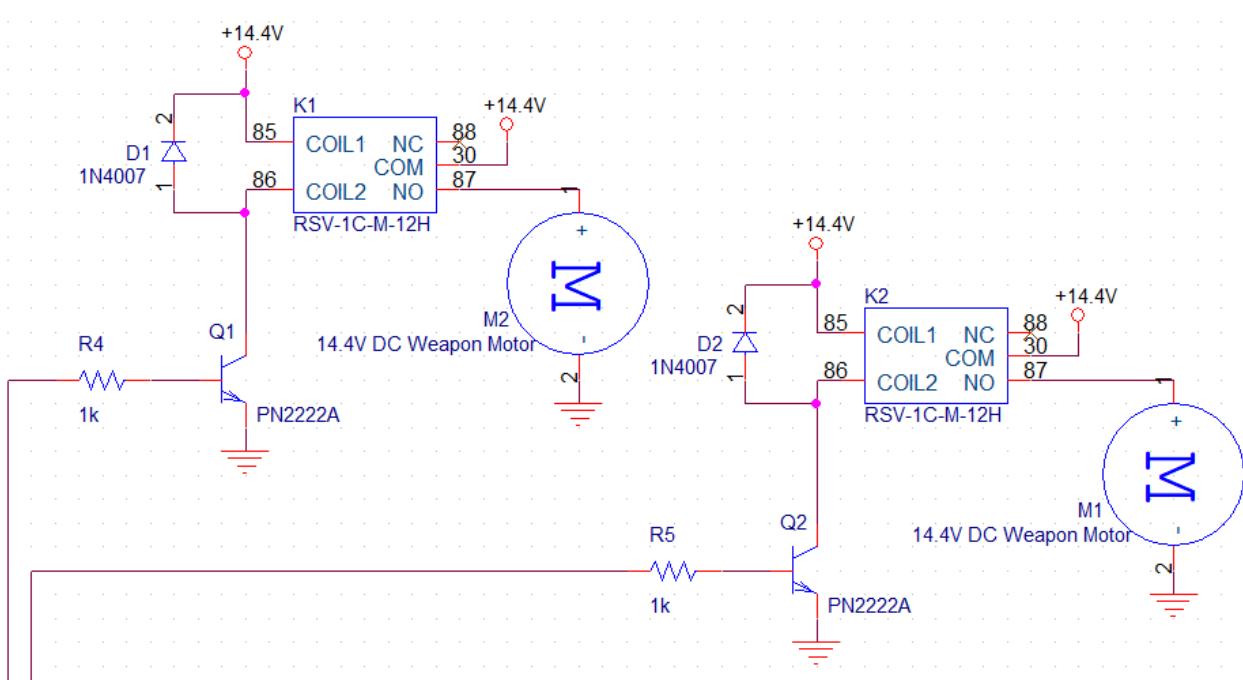
Reset пинът, както и GPIO0 и GPIO2 пиновете остават плаващи.

При включване на устройството, микро контролерът от блок “Управление” изпраща поредица от AT команди, които служат за първоначална настройка на Wi-Fi модула U2. Той започва да работи в режим на Access Point (AP), като същевременно се променят и името и паролата на access point-а. При свързване на мобилно устройство към създадения от модула AP, става възможно отварянето на връзка на предварително дефиниран в програмата порт през RemoteXY приложението. При свързване, модулът изпраща нужната конфигурационна информация на приложението и то приема предварително дефинирани от потребителя външен вид и функционалност. Това включва всички бутони и слайдери, нужни за управлението на робота. След установяване на връзка, Wi-Fi модулът започва да приема въвежданата информация и да я пренасочва към приложението, работещо на микроконтролера, посредством серийната връзка. По този начин в реално време се обработват и изпълняват командите, въвеждани от управляващия робота. На фиг. 2.3. е показана конфигурацията на приложението, която използва устройството. Управлението на движението става чрез слайдер за скорост и джойстик за завиване. Добавен е и бутон за завъртане на 360 градуса, който настройва моторите да се движат в противоположни посоки. За управление на оръжието на робота се използват два ключа.



Фиг. 2.3. Потребителски интерфейс на приложението *RemoteXY*

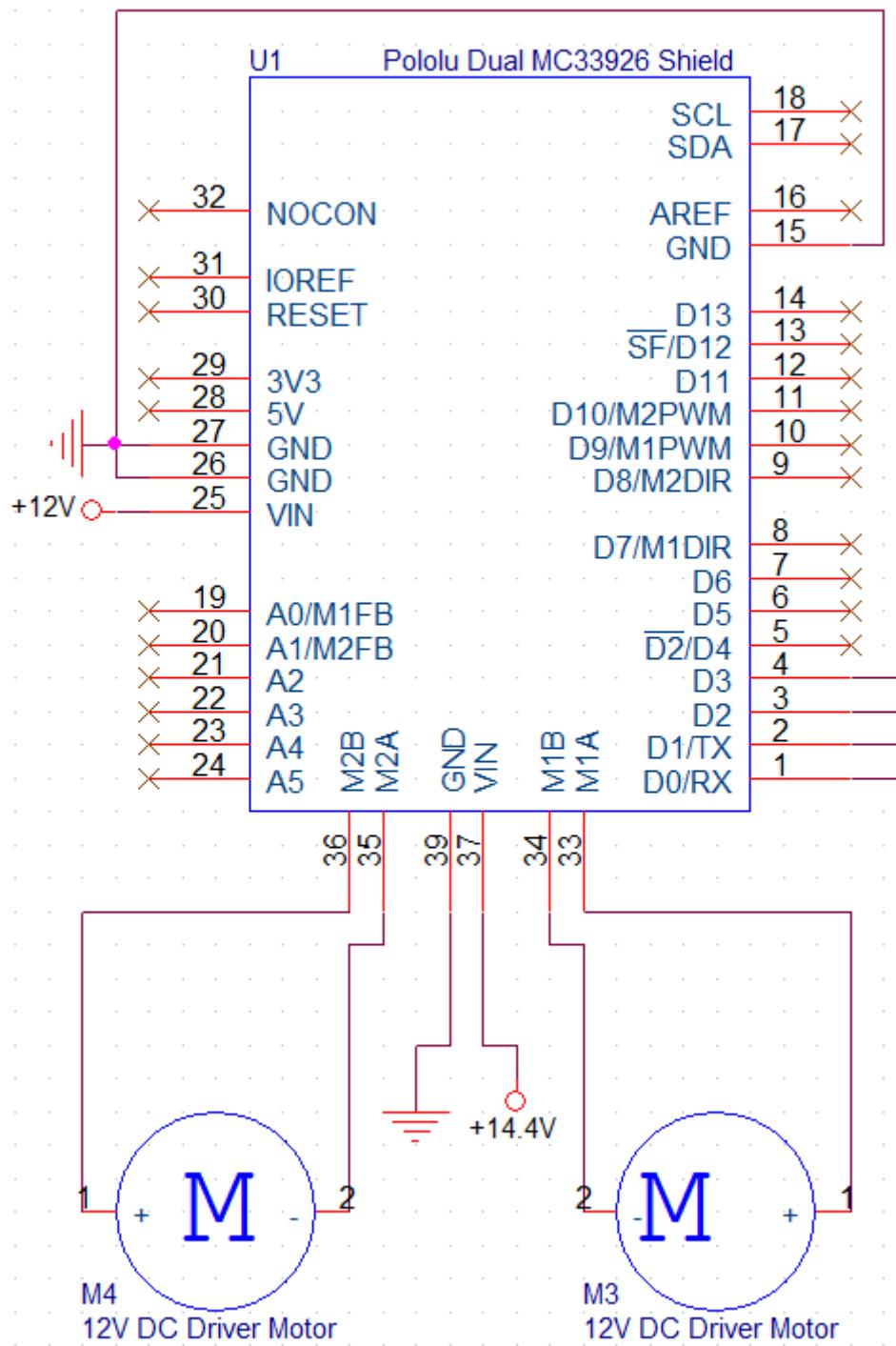
2.3. Принципна електрическа схема на Блок “Оръжия”



Фиг. 2.4. Принципна електрическа схема на блок “Оръжия”

Блокът служи за позволяване и забраняване на работата на двата мотора, задвижващи оръжието на робота (2 бургии). Базите на транзисторите Q1 и Q2 са свързани към управляващите оръжието пинове на микроконтролера от блок “Управление”. При установяване на високо ниво на тези пинове, транзисторите се отпушват. По този начин те позволяват протичането на ток през бобините на релетата K1 и K2. Това активира релетата и те от своя страна позволяват протичането на ток през моторите на оръжието M1 и M2 и те се завъртат. При установяване на ниско ниво на управляващите пинове, транзисторите се запушват. По този начин спира протичането на ток през бобините на релетата и ключовете в изходните им вериги се отварят. Това забранява протичането на ток през моторите и тяхното въртене спира. Към бобините са включени и flyback диоди, които служат за елиминиране на пиковете на напрежение, които се получават при запушването на транзисторите. По този начин се защитават транзисторите и останалите елементи на схемата от изгаряне.

2.4. Принципна електрическа схема на Блок “Движение”



Фиг. 2.5. Принципна електрическа схема на блок “Оръжия”

Блокът служи за управлението на скоростта и посоката на въртене на двета задвижващи мотора. Шийлдът U1 получава команди от микроконтролера и позволява движението на моторите M3 и M4 в нужната посока и с нужната скорост. Шийлдът се захранва от 14.4V батерия, свързана към VIN и GND пиновете му. Той от своя страна захранва включените към него мотори. Управлението става чрез пиновете D7/M1DIR и D9/M1PWM за първия мотор и D8/M2DIR и D10/M2PWM за втория мотор. MxDIR пиновете определят посоката на въртене на моторите. Скоростта им се определя чрез подаване на PWM сигнал на MxPWM пиновете и се изчислява по формулата 2-2

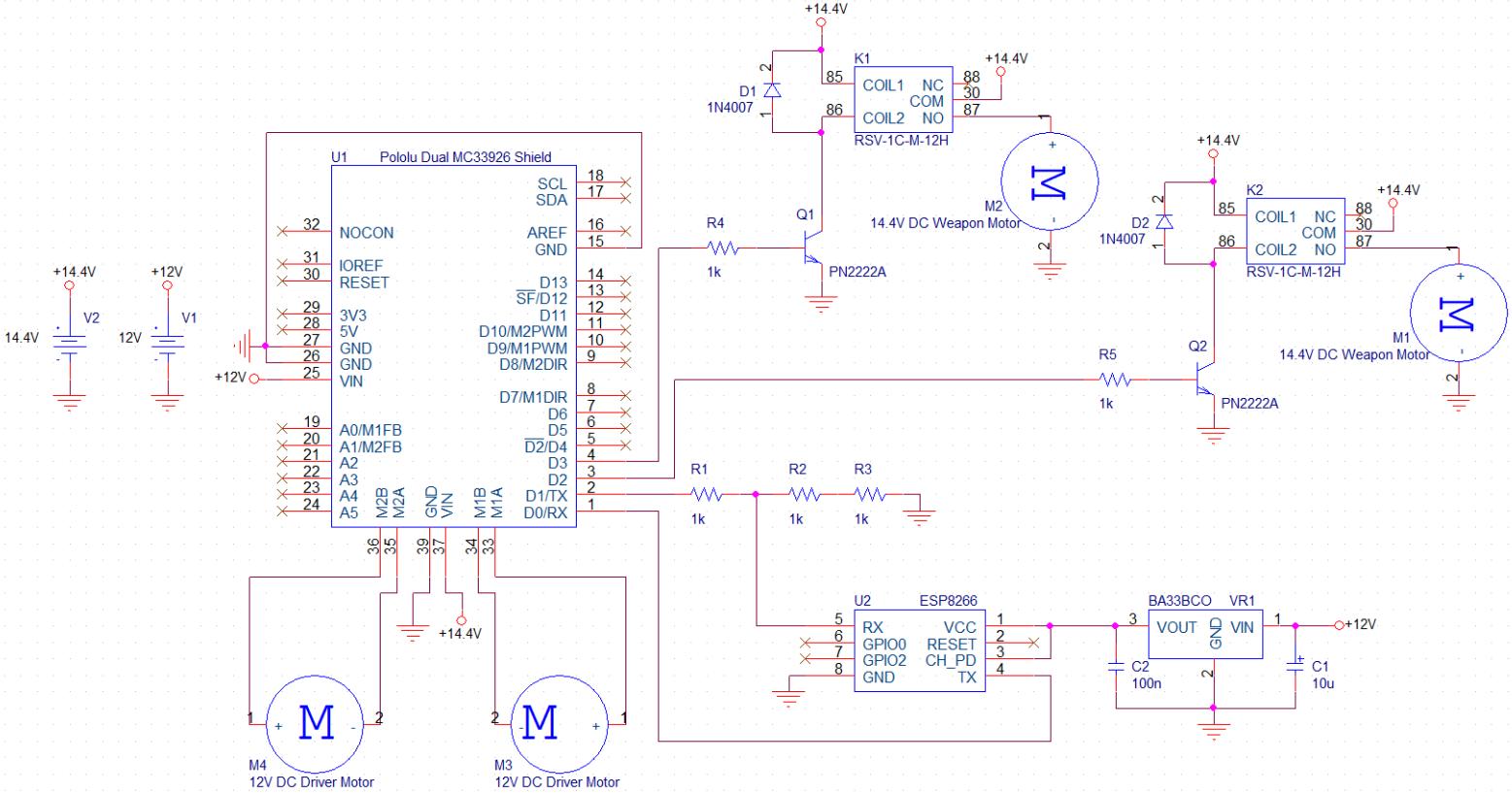
$$M_{speed} = M_{max\ speed} * K$$

2-2. Формула за изчисление на скоростта на мотор (K = коефициент на запълване)

Работата на шийлда се разрешава чрез подаване на ниско ниво на D2bar/D4.

2.5. Пълна принципна електрическа схема на устройството

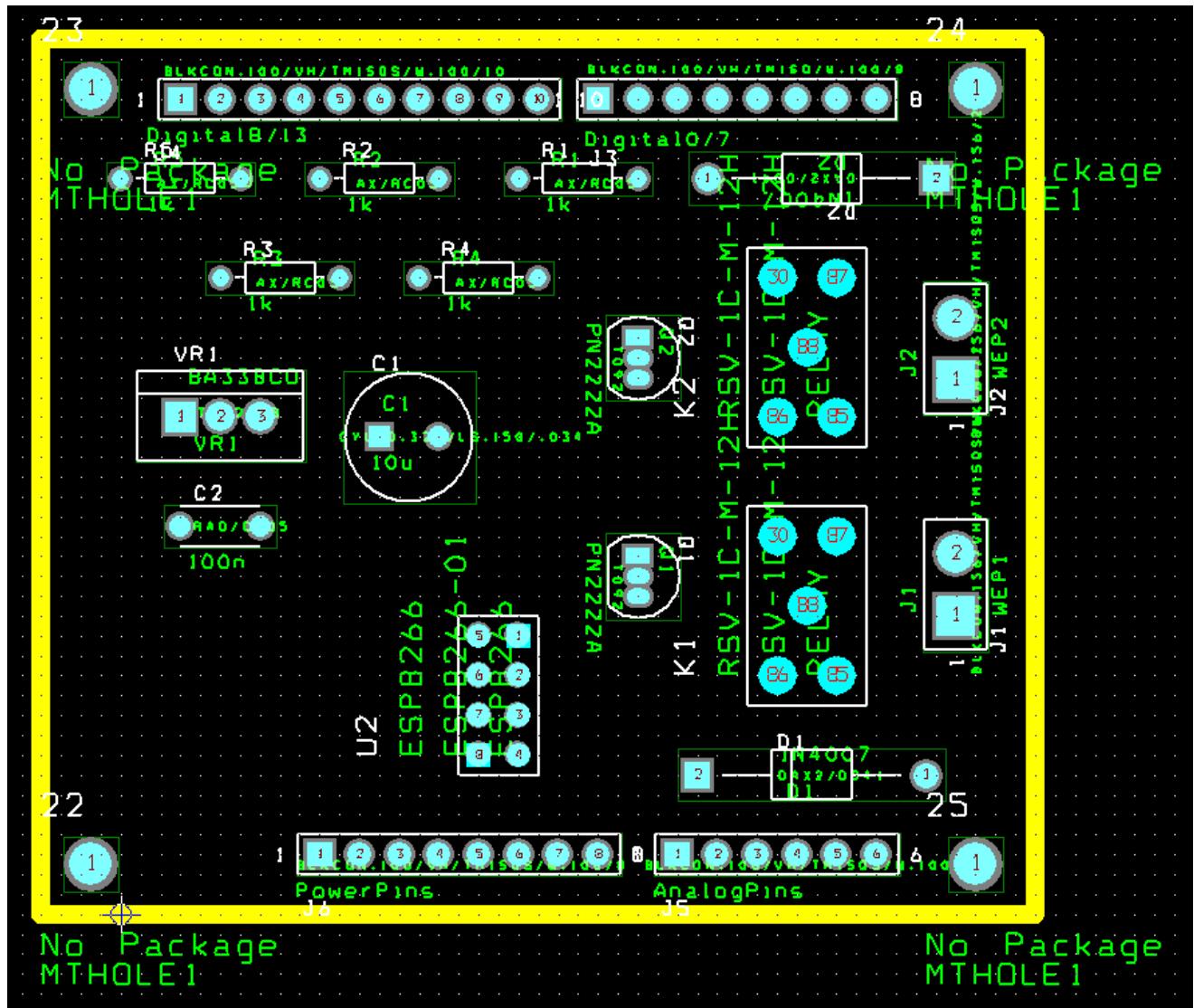
На Фиг. 2.1. е показана пълната принципната електрическа схема на проектираното устройство.



Фиг. 2.6. Пълна принципна електрическа схема на устройството

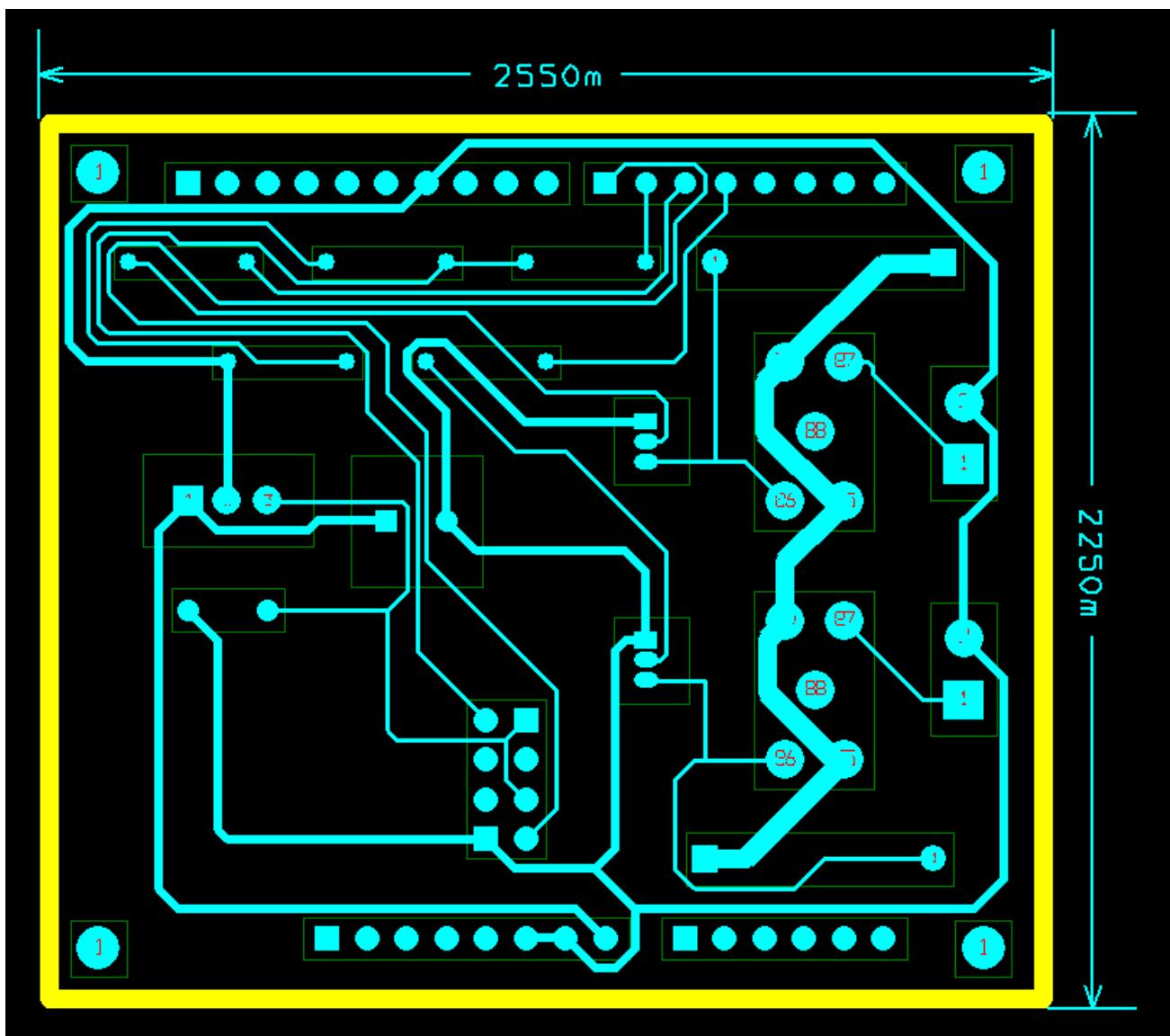
Използваният драйвер за задвижващите мотори е Pololu Dual MC33926 Shield (U1). Информация за този елемент е приложена в [1]. Информация за Wi-Fi модула ESP8266-01 (U2) е приложена в [2]. Информация за релетата RSV-1C-M-12H (K1 и K2) е приложена в [3]. Информация за регулатора BA33BCO (VR1) е приложена в [4]. Моторите, използвани за оръжието на робота са RS-550 DC 6-14.4V (M1 и M2). Информация за тях е приложена в [5]. Информация за задвижващите мотори VFGA60FHH-22i A802 (M3 и M4) е приложена в [6]. Информация за транзисторите PN2222A (Q1 и Q2) е приложена в [7]. Информация за резисторите R1-R5 е приложена в [8]. Информация за кондензаторите C1 и C2 е приложена в [9] и [10]. Информация за батериите V1 и V2 е приложена в [11] и [12]. Информация за диодите 1N4007 (D1 и D2) е приложена в [13].

2.6. Печатна платка на устройството



Фиг. 2.7. Разположение на елементите на печатната платка

На Фиг. 2.7. е показана проектираната печатна платка с разположени върху нея елементи. Изработена е във вид на шийлд за Arduino Uno R3, с цел да се спести място и да бъде по-евтина. Добавени са и 4 mounting hole-а в четирите ъгъла на платката за лесно монтиране.

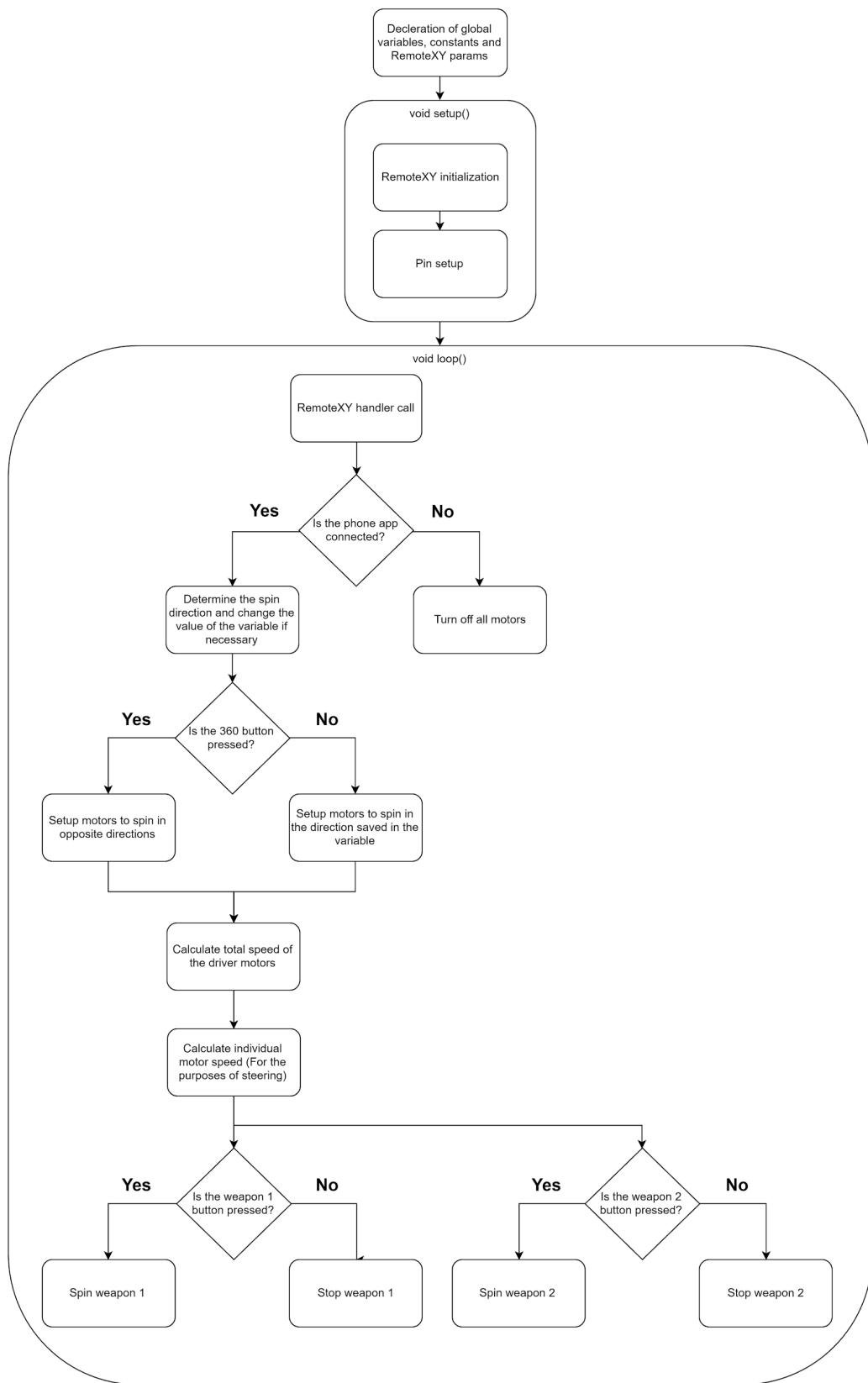


Фиг. 2.8. Опроводяване на печатната платка

На Фиг. 2.8. са показани опроводяването на печатната платка и нейните размери. Използваните захранващи писти са 48 mil за 14.4V захранване и 24 mil за 12V захранване. Пистите за данни са с размер от 12 mil.

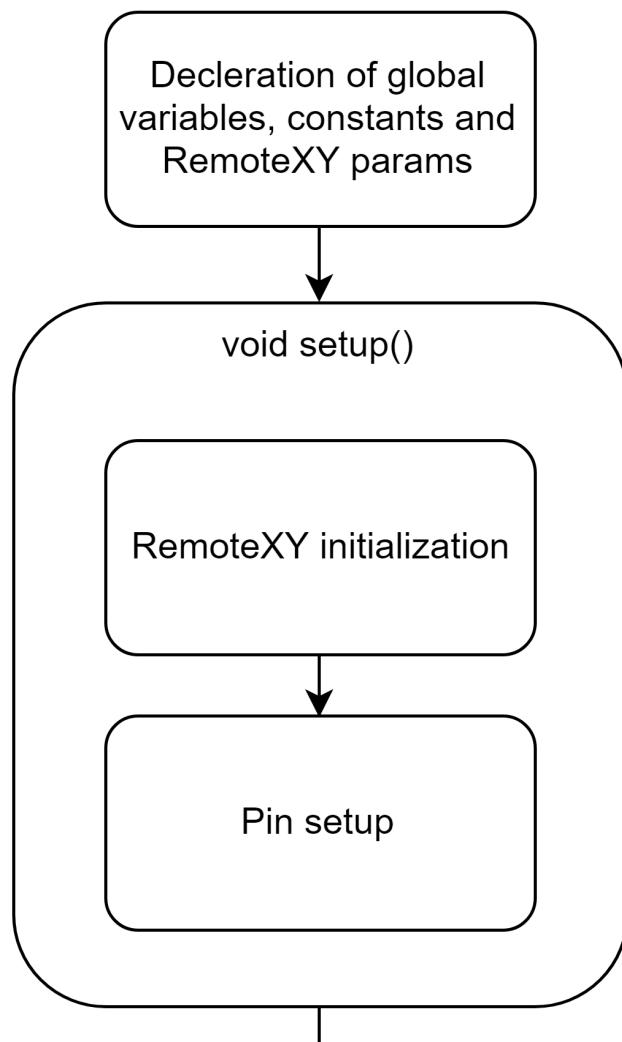
3. ПРОГРАМЕН КОД НА УСТРОЙСТВОТО

3.1. Блокова схема на кода



Фиг. 3.1. Блокова схема на кода на устройството

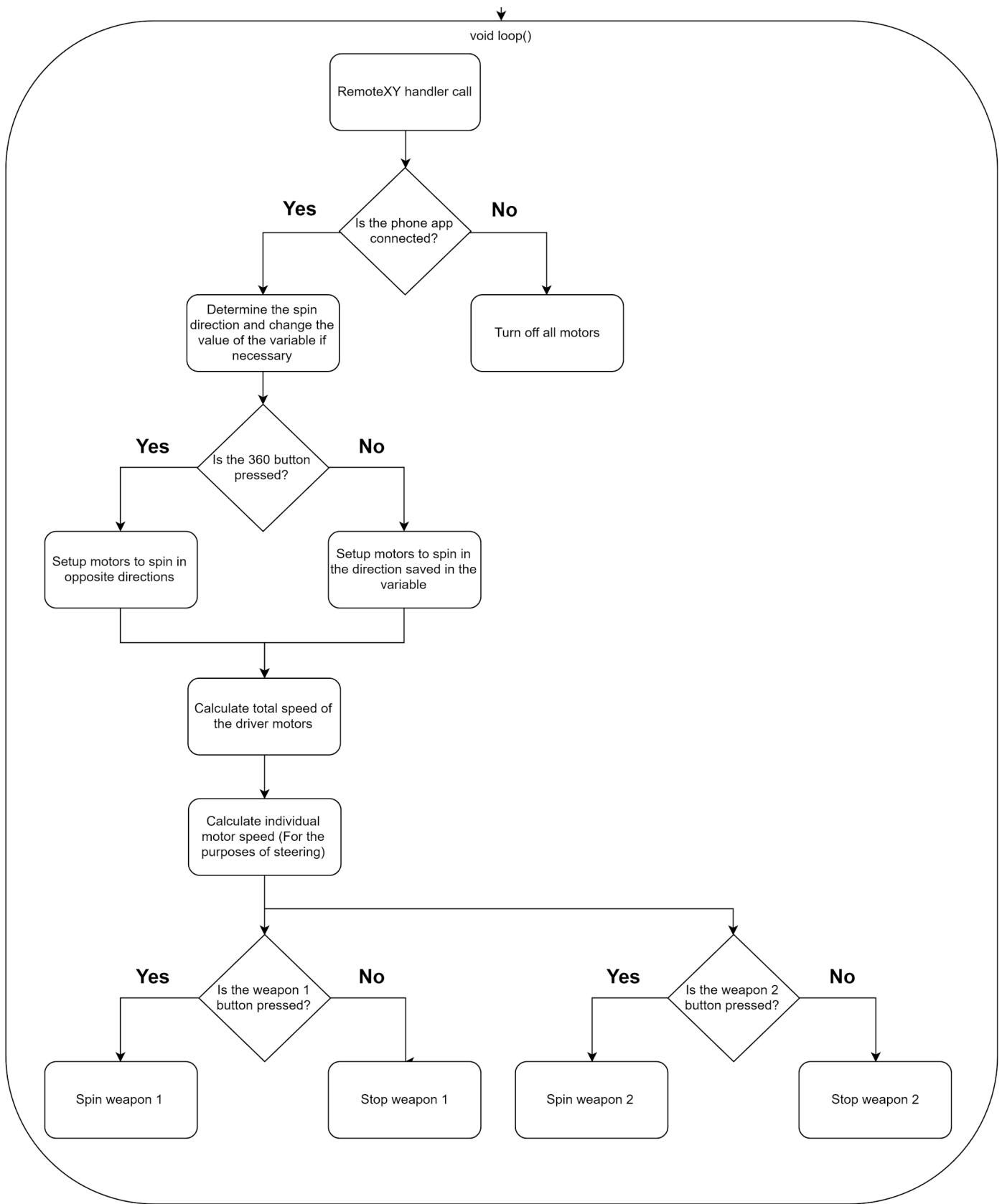
3.2. Блокова схема на инициализацията



Фиг. 3.2. Блокова схема на кода за инициализация (`void setup()`)

На Фиг. 3.2. е показана блоковата схема на кода за инициализация на устройството. През първият етап от инициализацията се извършва деклариране на глобални променливи, константи и параметри на RemoteXY библиотеката. Втория етап е изпълнението на `setup()` метода, като в него се инициализира библиотеката RemoteXY и се декларират режимите на работа на използваните пинове. В края на инициализацията всички пинове се установяват в тяхното начално състояние, като по този начин се гарантира предвидимата работа на устройството.

3.3. Блокова схема на основния цикъл



Фиг. 3.3. Блокова схема на основния цикъл (void loop())

На Фиг. 3.3. е показана блоковата схема на основния цикъл. В началото на всяка итерация се извиква RemoteXY handler функция, която служи за получаване на информация от мобилното приложение и актуализиране на стойностите на променливите на библиотеката.

Първата стъпка е проверка дали модулът има установена връзка с мобилното приложение. Ако такава няма, или ако тя е загубена за повече от 5 секунди, въртенето на всички мотори се спира и цикълът се повтаря. В противен случай се пристъпва към втора стъпка - проверка за посоката на движение. В случай че тя е различна от предходната, записана в spinDirection , в променливата се записва новата посока на движение. Третата стъпка е проверка дали 360 бутоњът е натиснат. Ако това е изпълнено, двата мотора се настройват да се движат в противоположни посоки, а ако не е - и двата мотора се настройват да се движат в посоката, записана в spinDirection променливата. Четвъртата стъпка е изчисляване на скоростта, с която трябва да се движат моторите, спрямо позицията на слайдера за ускорение. След като бъде изчислена общата скорост се пристъпва към пета стъпка - определяне на скоростта на двата мотора поотделно спрямо позицията на джойстика за завиване. При изместване на джойстика по x оста роботът трябва да завие, т.е. ускорението на двата мотора трябва да бъде несиметрично. Това бива съобразено в тази стъпка преди крайният резултат да се подаде на MxPWM пиновете на драйвера и моторите да се задвижат с нужната скорост. Основният цикъл завършва с установяване на моторите на оръжиета. В мобилното приложение се проверяват ключовете за активиране и всяко оръжие поотделно се завърта или спира съобразно с прочетените от приложението данни.

В т. 3.4. е показан програмният код за управление на микроконтролера на устройството (Arduino Uno R3). На фиг 3.4. и фиг. 3.5. е показан кодът за инициализация на устройството, а на фиг. 3.6. и фиг. 3.7. са показани главният цикъл и функциите, които той използва.

3.4. Програмен код на микроконтролера

```
//////////  
//      RemoteXY include library      //  
//////////  
  
// RemoteXY select connection mode and include library  
#define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT  
  
#include <RemoteXY.h>  
  
// RemoteXY connection settings  
#define REMOTEXY_SERIAL Serial  
#define REMOTEXY_SERIAL_SPEED 115200  
#define REMOTEXY_WIFI_SSID "NaStrahilWiFito"  
#define REMOTEXY_WIFI_PASSWORD "battlebotpassword"  
#define REMOTEXY_SERVER_PORT 6377  
  
  
// RemoteXY configurate  
#pragma pack(push, 1)  
uint8_t RemoteXY_CONF[] =  
{ 255, 6, 0, 0, 0, 70, 0, 8, 24, 0,  
 5, 32, 6, 27, 30, 30, 30, 34, 31, 4,  
 48, 78, 11, 11, 43, 34, 30, 1, 0, 14,  
 10, 13, 13, 34, 31, 51, 54, 48, 0, 2,  
 0, 41, 29, 20, 10, 34, 26, 31, 31, 87,  
 69, 80, 50, 0, 79, 70, 70, 0, 2, 0,  
 41, 44, 20, 10, 34, 26, 31, 31, 87, 69,  
 80, 49, 0, 79, 70, 70, 0 };  
  
// this structure defines all the variables of your control interface  
struct {  
  
    // input variable  
    int8_t steering_x; // =-100..100 x-coordinate joystick position  
    int8_t steering_y; // =-100..100 y-coordinate joystick position  
    int8_t acceleration; // =-100..100 slider position  
    uint8_t rotate; // =1 if button pressed, else =0  
    uint8_t spin_weapon_2; // =1 if switch ON and =0 if OFF  
    uint8_t spin_weapon_1; // =1 if switch ON and =0 if OFF  
  
    // other variable  
    uint8_t connect_flag; // =1 if wire connected, else =0  
  
} RemoteXY;  
#pragma pack(pop)  
  
//////////  
//      END RemoteXY include      //  
//////////
```

Фиг. 3.4. Деклариране на променливите и константите на библиотеката.

```

#define M1PWM 9
#define M2PWM 10

#define M1DIR 7
#define M2DIR 8

#define WEAPONMOTOR1 2
#define WEAPONMOTOR2 5

#define D2BAR 4

int spinDirection = 1;
int motorSpeed[2] = {0, 0};
int totalSpeed = 0;

void setup()
{
    RemoteXY_Init ();

    pinMode(M1PWM, OUTPUT);      //Setting up driver pins
    pinMode(M1DIR, OUTPUT);
    pinMode(M2PWM, OUTPUT);
    pinMode(M2DIR, OUTPUT);
    pinMode(WEAPONMOTOR1, OUTPUT);
    pinMode(WEAPONMOTOR2, OUTPUT);

    pinMode(D2BAR, OUTPUT);

    digitalWrite(M1DIR, LOW);     //Setting up initial rotation direction
    digitalWrite(M2DIR, LOW);

    digitalWrite(M1PWM, LOW);     //Making sure the motors are stopped
    digitalWrite(M2PWM, LOW);

    digitalWrite(WEAPONMOTOR1, LOW); //Making sure the weapon is stopped
    digitalWrite(WEAPONMOTOR2, LOW);

    digitalWrite(D2BAR, HIGH);
}

```

Фиг. 3.5. Деклариране на глобалните променливи, константи и режими на работа на пиновете, както и техните начални състояния.

```

void loop()
{
    RemoteXY_Handler();

    if(RemoteXY.connect_flag){

        if(spinDirection != DetermineSpinDirection(RemoteXY.acceleration)){ //Reverse the direction if needed
            spinDirection = !spinDirection;
        }

        if(RemoteXY.rotate){ //Checking if the 360 button is pressed
            Rotate360();
        }
        else{
            DetermineDirection(spinDirection);
        }

        totalSpeed = CalculateTotalSpeed(RemoteXY.acceleration); //Calculate total motor speed
        CalculateIndividualMotorSpeed(totalSpeed, RemoteXY.steering_x);

        analogWrite(M1PWM, motorSpeed[0]);
        analogWrite(M2PWM, motorSpeed[1]);

        SpinWeapon(); //Checking if any of the weapon buttons are pressed
    }

    else{
        digitalWrite(M1PWM, LOW); //Making sure the motors are stopped
        digitalWrite(M2PWM, LOW);

        digitalWrite(WEAPONMOTOR1, LOW); //Making sure the weapon is stopped
        digitalWrite(WEAPONMOTOR2, LOW);
    }
}

int DetermineSpinDirection(int acceleration){ // 0 for backward, 1 for forward
    return (acceleration < 0) ? 0 : 1;
}

int DetermineTurnDirection(int turnModifier){ // 0 for left, 1 for right
    return (turnModifier < 0) ? 0 : 1;
}

```

Фиг. 3.6. Основен цикъл и функции

```

void DetermineDirection(int spinDirection){
    if(spinDirection > 0){           //Forward
        digitalWrite(M1DIR, LOW);
        digitalWrite(M2DIR, LOW);
    }
    else{                          //Backwards
        digitalWrite(M1DIR, HIGH);
        digitalWrite(M2DIR, HIGH);
    }
}

int CalculateTotalSpeed(int acceleration){
    return map(abs(acceleration), 0, 100, 0, 255);
}

void CalculateIndividualMotorSpeed(int totalSpeed, int turnModifier){
    int turnDirection = DetermineTurnDirection(turnModifier);
    int modifier = map(abs(turnModifier), 0, 100, 0, 255);
    if(turnDirection == 0){
        motorSpeed[0] = totalSpeed;
        motorSpeed[1] = totalSpeed - modifier;
    }
    else{
        motorSpeed[0] = totalSpeed - modifier;
        motorSpeed[1] = totalSpeed;
    }
}

void Rotate360(){
    digitalWrite(M1DIR, HIGH);
    digitalWrite(M2DIR, LOW);
}

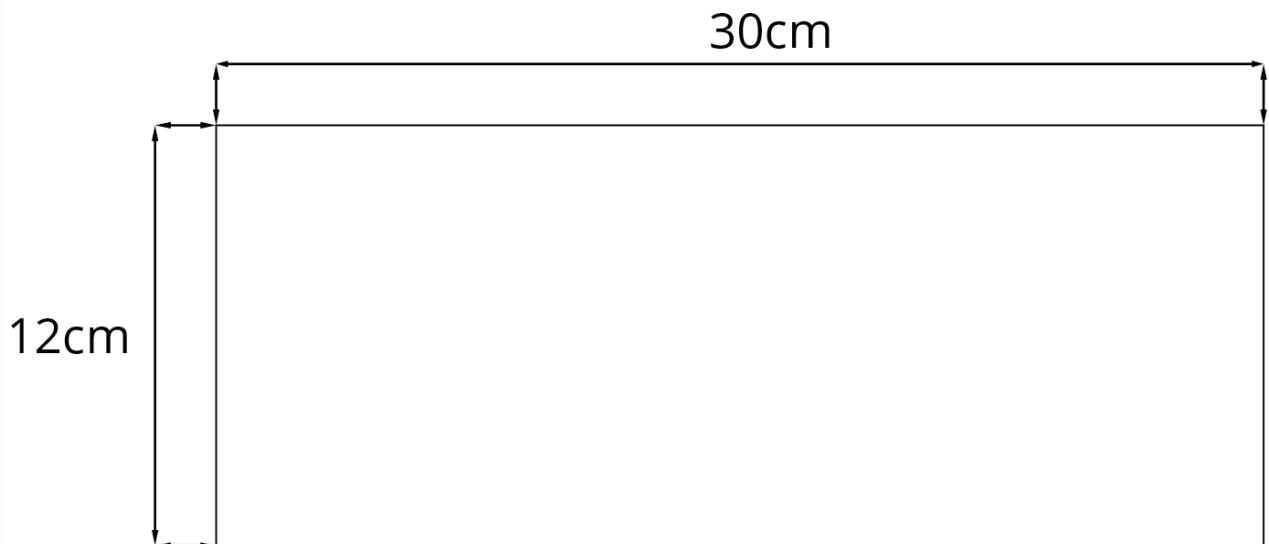
void SpinWeapon(){}
    if(RemoteXY.spin_weapon_1){
        digitalWrite(WEAPONMOTOR1, HIGH);
    }
    else{
        digitalWrite(WEAPONMOTOR1, LOW);
    }
    if(RemoteXY.spin_weapon_2){
        digitalWrite(WEAPONMOTOR2, HIGH);
    }
    else{
        digitalWrite(WEAPONMOTOR2, LOW);
    }
}

```

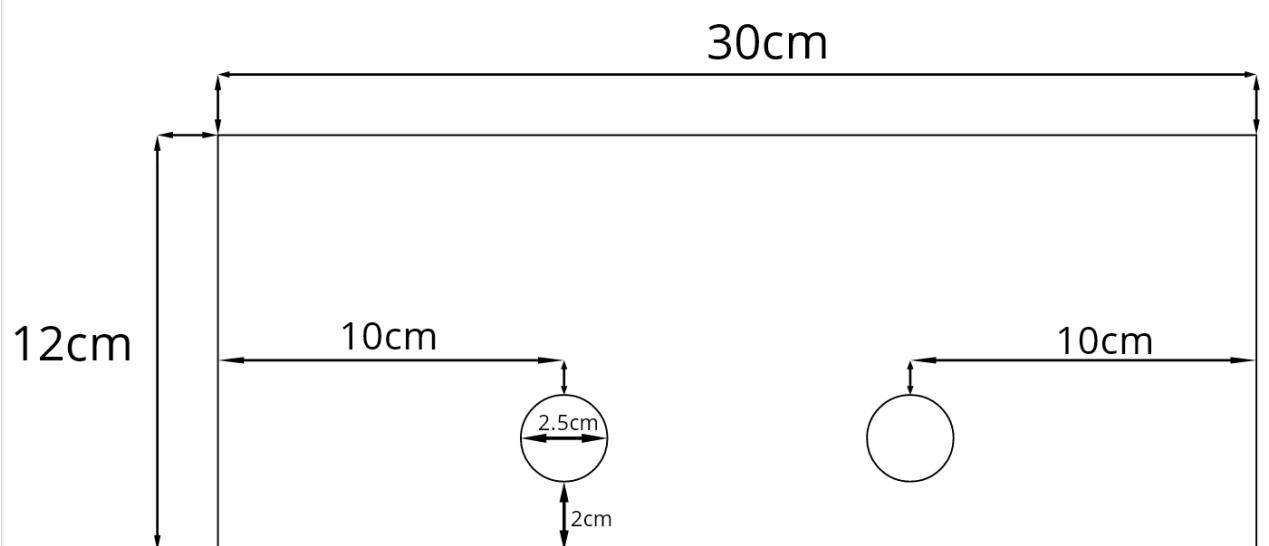
Фиг. 3.7. Функции, използвани в главния цикъл

4. МЕХАНИКА НА РОБОТА

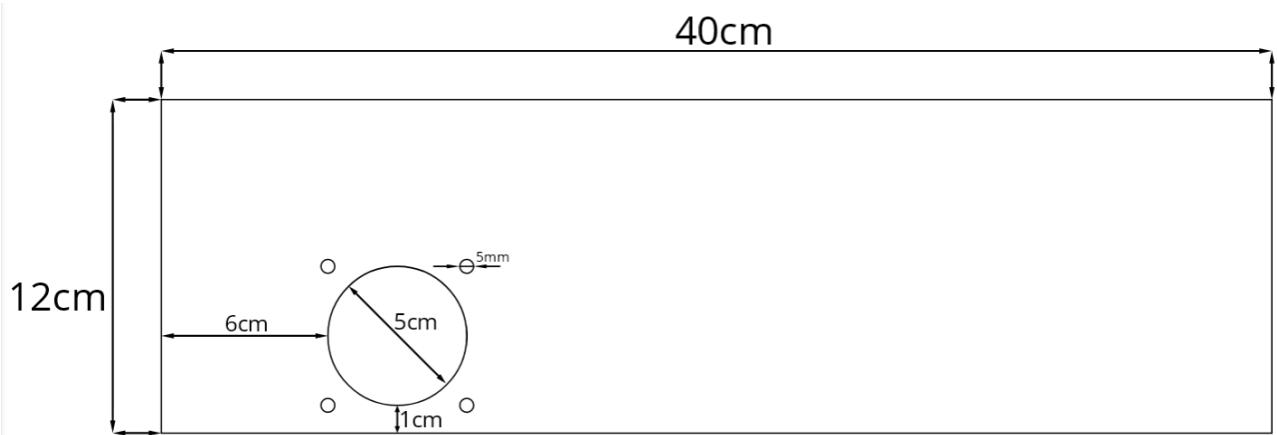
4.1. Чертежи на механиката



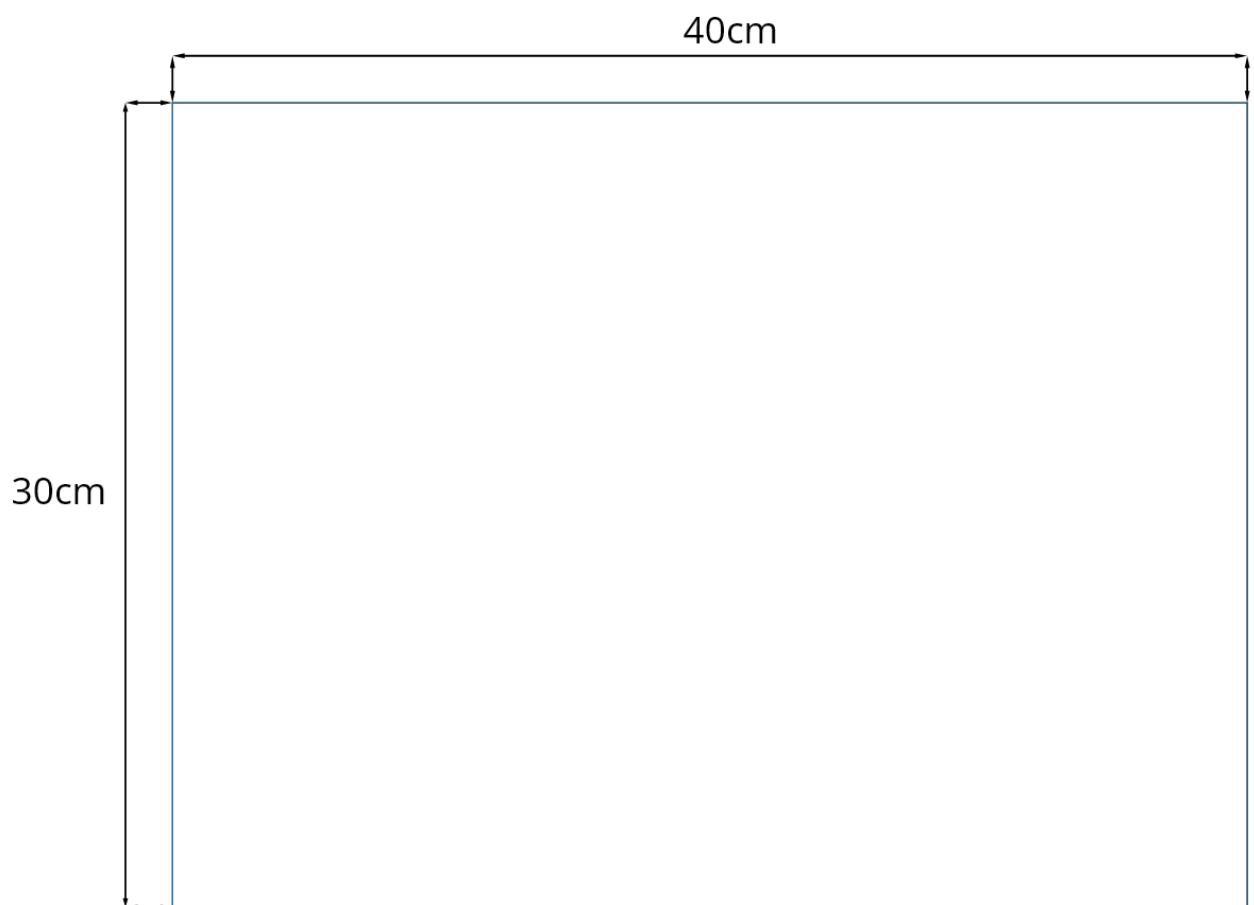
Фиг. 4.1. Задна стена



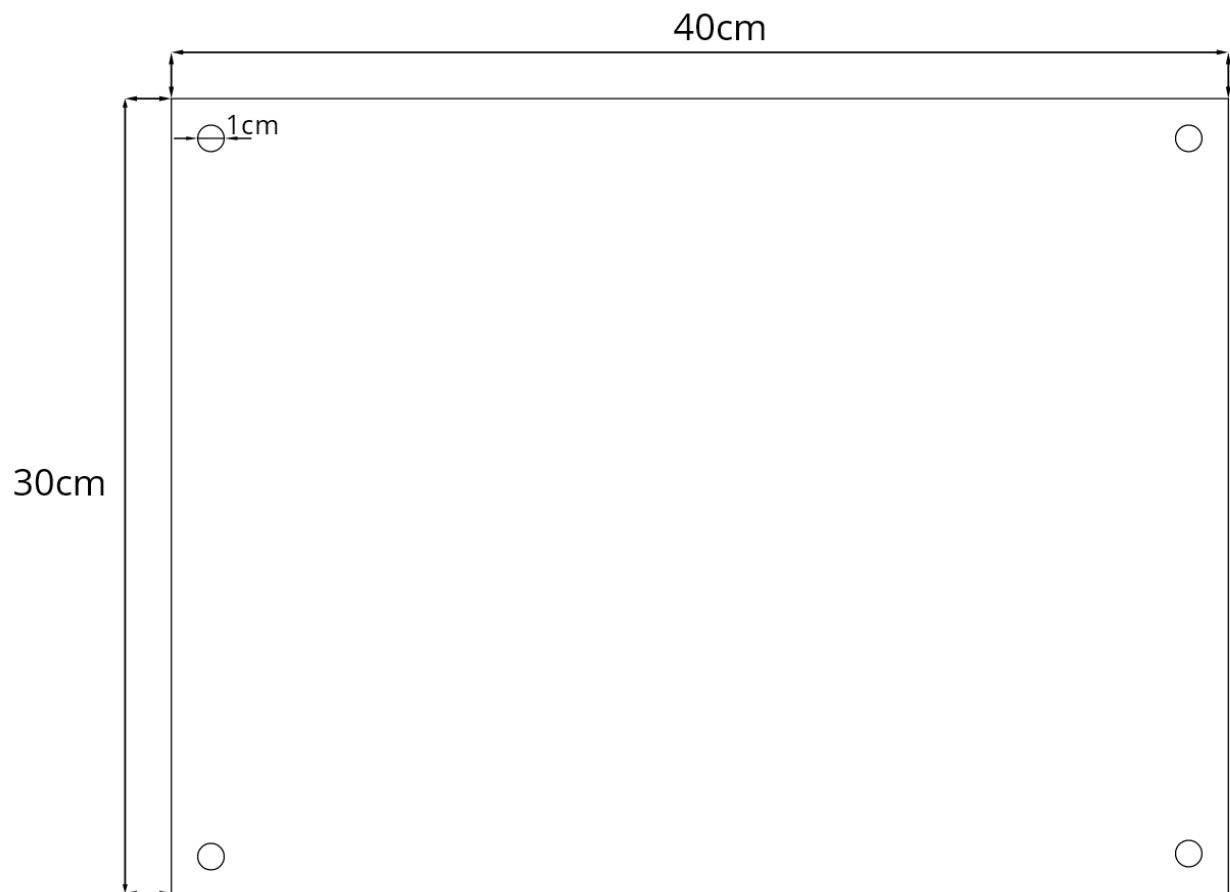
Фиг. 4.2. Предна стена



Фиг. 4.3. Лява и дясна стена



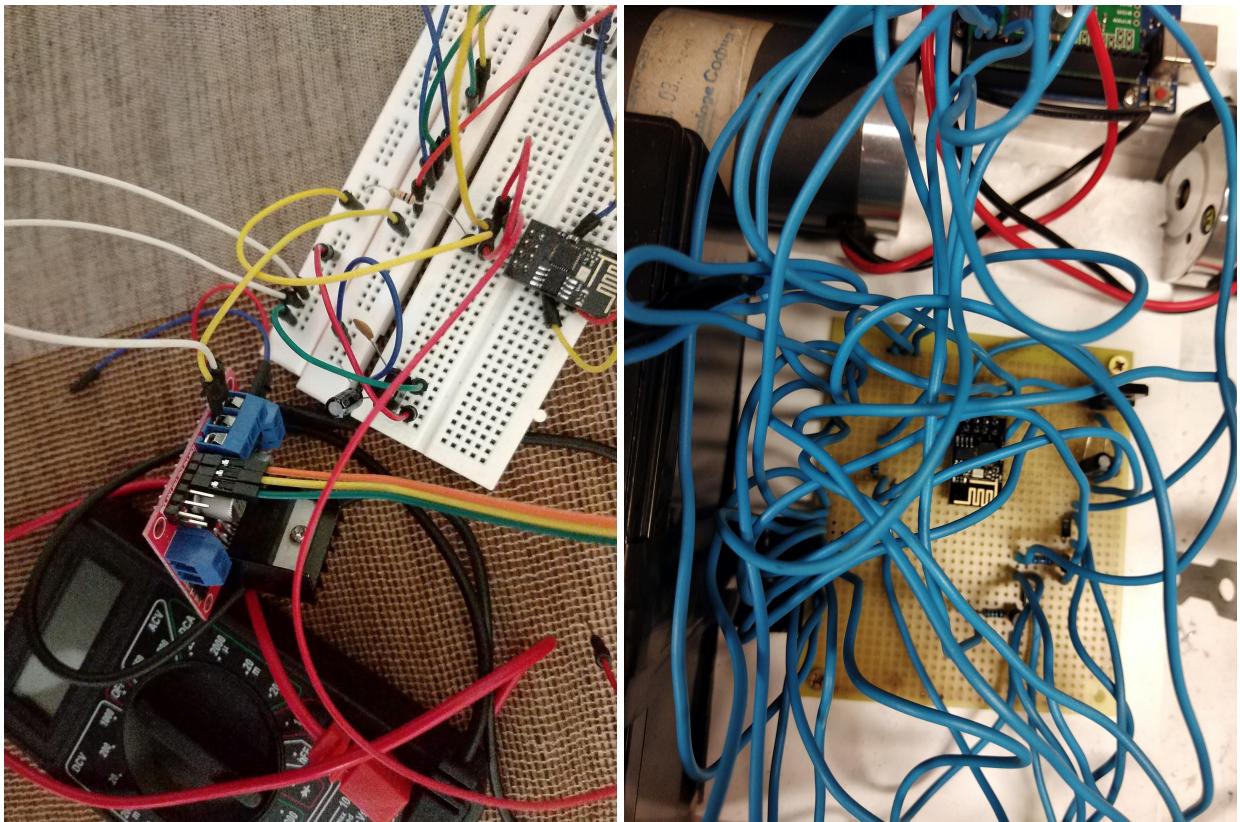
Фиг. 4.4. Дъно



Фиг. 4.5. Капак

На фиг. 4.1. - 4.5. са показани чертежи на механичните части на устройството. Корпусът на робота е изграден от 4мм неръждаема ламарина. От вътрешната си част е облицован с PVC. Стените са захванати заедно чрез заварки. Предните колела са от тип топче и са захванати за дъното на робота чрез заварки.

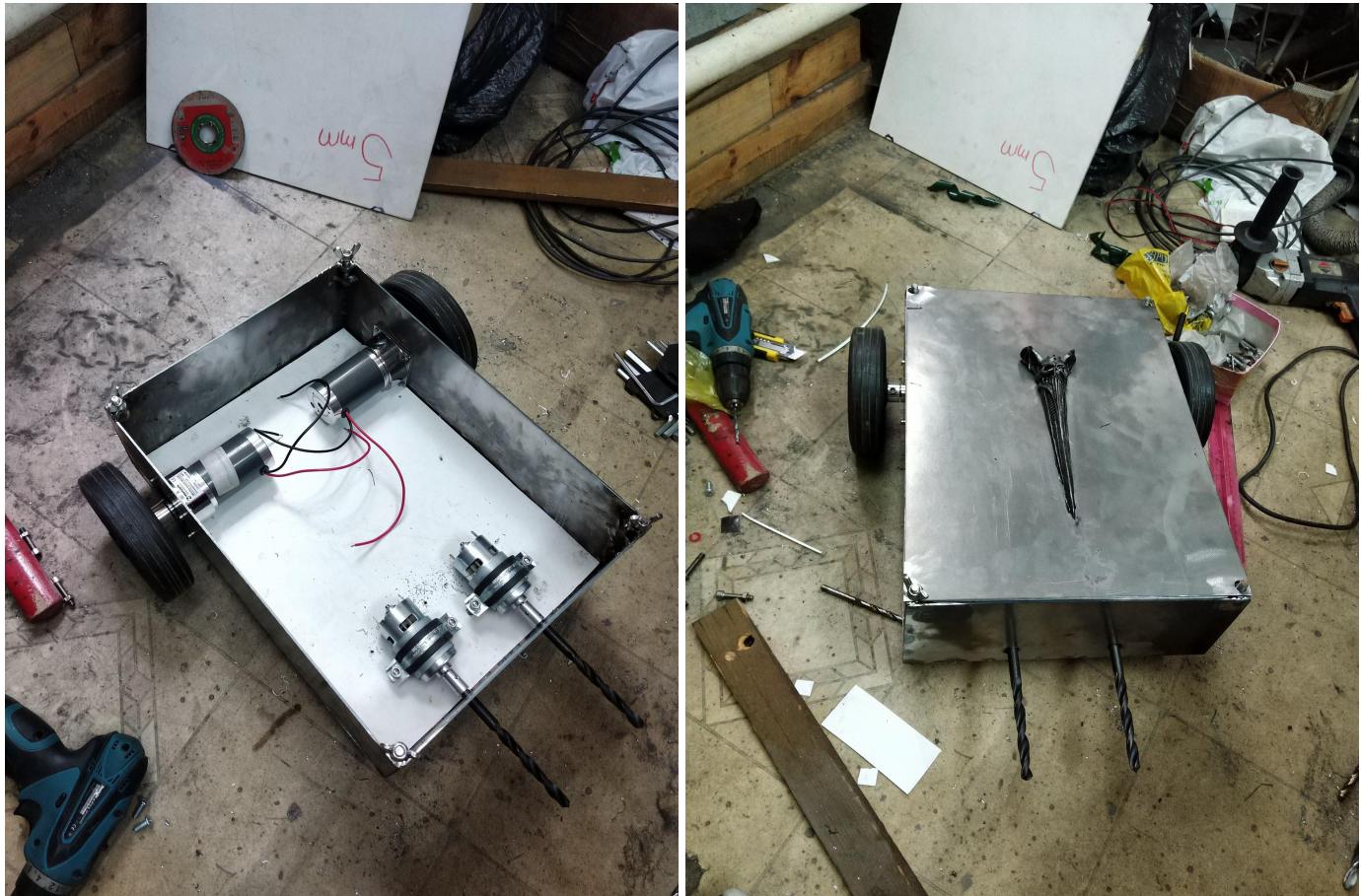
4.2. Етапи на разработката



Фиг. 4.6. Запояване на елементите върху платка



Фиг. 4.7. Първоначална версия на робота



Фиг. 4.8. Втора, подобрена версия на робота

На фиг. 4.6. - 4.8. са показани етапите на разработка на робота. На фиг. 4.6. е показано запояването на елементите на схемата върху прототипна платка. На фиг. 4.7. е показана първата версия на корпуса на робота. Тя е направена от тънка ламарина и използва колела от пазарна количка вместо топчета. На фиг. 4.8. е показана втората и последна подобрена версия на корпуса. При нея е сменена ламарината с по-дебела и са подменени предните колела на робота с топчета. Добавени са също оръжията на робота - 2 бургии за метал, закрепени към мотори за винтоверт.

4.3. Краен резултат



Фиг. 4.9. Завършиен робот

На фиг. 4.9. е показан завършеният робот. След сглобяването му са добавени декоративни елементи и корпусът е боядисан с черен и червен спрей.

КАТАЛОЖНА ИНФОРМАЦИЯ

- [1] https://www.pololu.com/docs/pdf/0J55/dual_mc33926_motor_driver_shield.pdf
- [2] <http://www.microchip.ua/wireless/esp01.pdf>
- [3] <https://store.comet.bg/download-file.php?id=2063>
- [4]
http://rohmfs.rohm.com/en/products/databook/datasheet/ic/power/linear_regulator/ba_xxbc0-e.pdf
- [5] http://www.robotstorehk.com/rs_550pcvc.pdf
- [6]
<https://vikiwat.com/elektrodvigatel-postoyannotokov-s-reduktor-12vdc-180rpm-vfga60fhh-22i-a802.html>
- [7] <https://www.onsemi.com/pub/Collateral/PN2222-D.PDF>
- [8] <https://store.comet.bg/download-file.php?id=412>
- [9] <https://www.mouser.bg/datasheet/2/427/s104z93z-46991.pdf>
- [10] https://datasheet.lcsc.com/szlcsc/LCSC-10uF-50V_C32856.pdf
- [11]
http://www.elimex.bg/userfiles/editor/file/techspecificationspdf/3/31382_AK%2012V_1.4AH.pdf
- [12]
http://www.elimex.bg/userfiles/editor/file/techspecificationspdf/2/25816_AK_12V_14AH.PDF
- [13] <https://www.diodes.com/assets/Datasheets/ds28002.pdf>

ИЗПОЛЗВАНА ЛИТЕРАТУРА

- 1) <http://remotexy.com/en/help/>
- 2) https://www.pololu.com/docs/pdf/0J55/dual_mc33926_motor_driver_shield.pdf
- 3) <https://www.hackster.io/jeffpar0721/add-wifi-to-arduino-uno-663b9e>
- 4) <https://circuitdigest.com/microcontroller-projects/arduino-relay-control>
- 5) <https://www.robotshop.com/community/blog/show/drive-motor-sizing-tool>

СЪДЪРЖАНИЕ

ЗАДАНИЕ	2
1. ПРОЕКТИРАНЕ НА БЛОКОВАТА СХЕМА НА УСТРОЙСТВОТО	2
1.1. Блокова схема на устройството	3
1.2. Описание на блоковете	4
2. ПРОЕКТИРАНЕ НА УСТРОЙСТВОТО	5
2.1. Принципна електрическа схема на Блок “Управление”	5
2.2. Принципна електрическа схема на Блок “Комуникация”	6
2.3. Принципна електрическа схема на Блок “Оръжия”	9
2.4. Принципна електрическа схема на Блок “Движение”	10
2.5. Пълна принципна електрическа схема на устройството	12
2.6. Печатна платка на устройството	13
3. ПРОГРАМЕН КОД НА УСТРОЙСТВОТО	15
3.1. Блокова схема на кода	15
3.2. Блокова схема на инициализацията	16
3.3. Блокова схема на основния цикъл	16
3.4. Програмен код на микроконтролера	19
4. МЕХАНИКА НА РОБОТА	22
4.1. Чертежи на механиката	23
4.2. Етапи на разработката	26
4.3. Краен резултат	28
КАТАЛОЖНА ИНФОРМАЦИЯ	29
ИЗПОЛЗВАНА ЛИТЕРАТУРА	29
СЪДЪРЖАНИЕ	31