**Mini-Sumo Robots**

**Състезание**

**Екип:**

**Силвия Войнова Искра Тенева**

**GITHUB:** [**https://github.com/SVoynova/Mini-Sumo-Robot-Competiton**](https://github.com/SVoynova/Mini-Sumo-Robot-Competiton)

Април 2021

**Съдържание:**

[1. Описание 3](#_Toc69611706)

[**1.1.** **Идея** 3](#_Toc69611707)

[**1.2.** **Фокус** 3](#_Toc69611708)

[1.3. Повече за сумо роботите и мини-сумо състезанията 3](#_Toc69611709)

[1.4. Изпълнение 4](#_Toc69611710)

[**1.5.** **Роботи** 5](#_Toc69611711)

[2. Блокова схема 7](#_Toc69611712)

[3. Електрическа схема 8](#_Toc69611713)

[4. Списък компоненти 9](#_Toc69611714)

[4.1. Списък компоненти – Марсоход 9](#_Toc69611715)

[4.2. Списък компоненти – Луноход 9](#_Toc69611716)

[4.3. Конструкция в снимки 10](#_Toc69611717)

[5. Описание на функционалността 11](#_Toc69611718)

[5.1. Марсоход 11](#_Toc69611719)

[5.2. Луноход 13](#_Toc69611720)

[5.3. Общи тактики 14](#_Toc69611721)

[6. Провеждане на тестово състезание 15](#_Toc69611722)

[7. Заключение 15](#_Toc69611723)

[8. Възможности за бъдещо надграждане 15](#_Toc69611724)

## **Описание**

Настоящият проект разглежда дизайна, разработката и тестването на два автономни робота, с които да се проведе мини-сумо състезание.

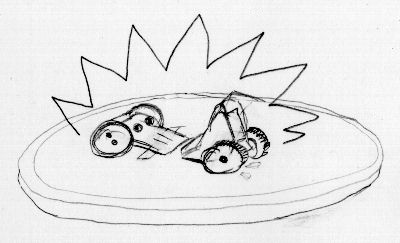
### **Идея**

Разработване на два сумо робота, които да се използват за целите на провеждането на мини-сумо състезание.

### **Фокус**

Настоящият проект се стреми да се фокусира върху вградени системи, по-специфично Ардуино микроконтролери, сензори, тяхната разработка и употреба.

## **Повече за сумо роботите и мини-сумо състезанията**



Сумо роботът представлява автономен робот, който може да бъде направен от всякакъв материал, да има двигател от всякакъв тип, мощност и напрежение. Роботът може да има всякакви процесори, датчици и друга електроника. По време на състезанието сумо роботите се борят помежду си подобно на сумо борците в Япония като целта им е да се избутат един друг от предварително начертан кръг, като за загубил се приема този, който първи премине очертаната с различен цвят граница на ринга. Роботите са самоконтрюлируеми. След поставяне на ринга не се позволява използване на дистанционни управления, позициониране или каквато и да е била друга подкрепа. Роботът процедира без външна помощ до края на рунда.

## **Изпълнение**

За целта на успешното създаване на сумо робот ще се използват достъпни материали.

Настоящият проект се базира на Pololu Zumo Robot for Arduino v1.2. (верижна робо-платформа, предназначена да се контролира от Ардуино микроконтролерни платки). С компактните си размери 98мм х 98мм и тегло 165гр (около 300гр с батерии и платка Ардуино Уно), роботът покрива критериите за участие в мини-сумо състезания.



За пълноценна реализация на идеята са използвани Arduino Uno микроконтролерни платки, за ориентация са оборудвани с инфрачервени сензори, чиято цел е да позволи на робота да се задържи на ринга, ултразвуков сензор за разстояние (при единия робот) или акселометър (при другия робот), с помощта на които да се оптимизира атакуването на засечения противник.

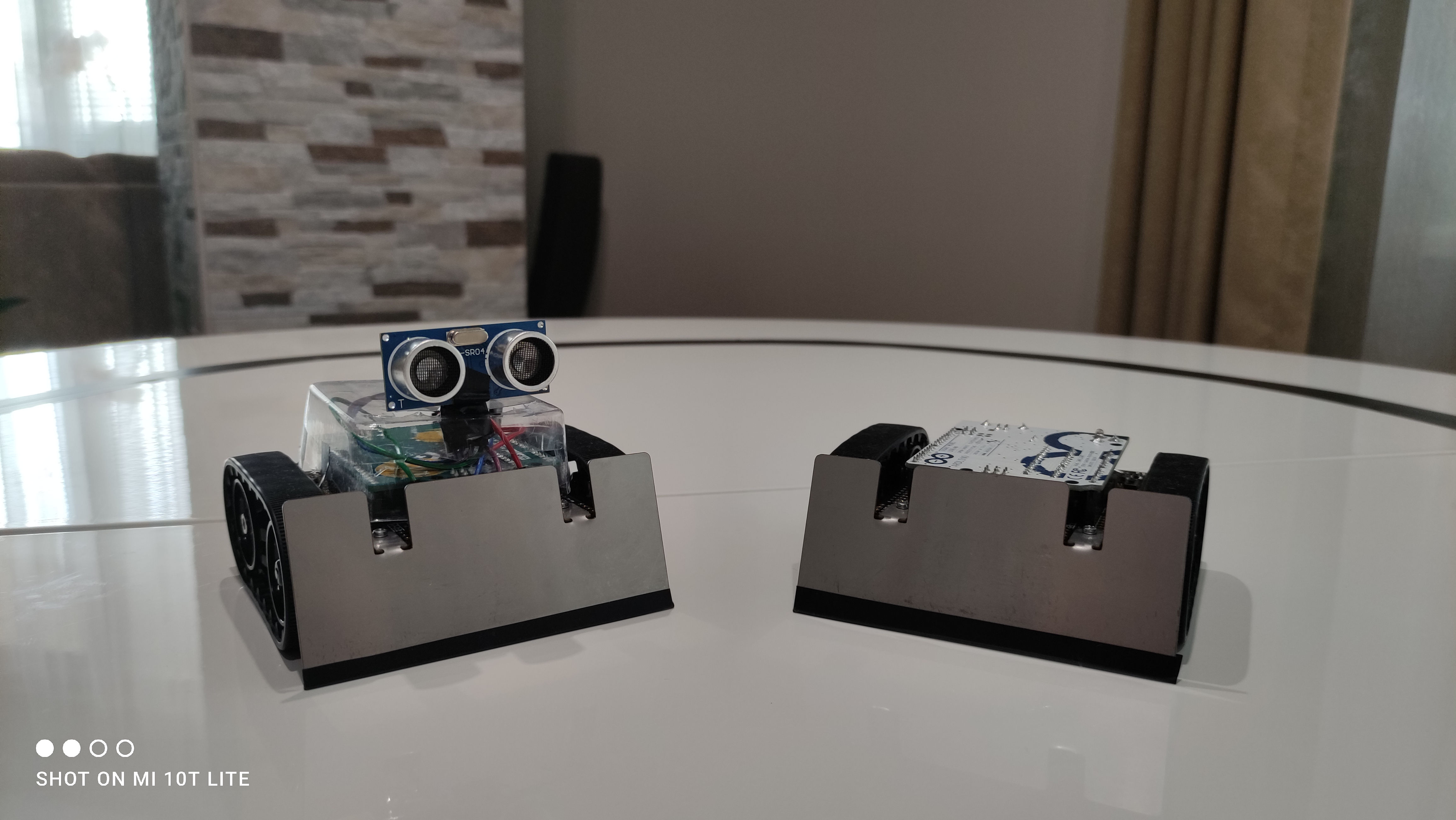
За разлика от повечето състезания по мини-сумо, игралният ринг, използван за този проект, е бял с черни граници. За целите на проекта роботите са програмирани да функционират на тази дъска. Всичко това се прави за постигане на удобство и бързина.

И при двата робота е приложена посочената тактика:

Реализира се алгоритъм за търсене на друг подвижен обект в равнина като роботът трябва да се доближи до него и в последствие да го избута от ринга преди той самият да бъде избутан от игралното поле. Същевременно инфрачервените сензорите трябва да се грижат робота да не излезе от ринга. Чрез избрани сензори трябва да се създаде способността за засичане на противника, при което да се предприеме атакуването му.

### **Роботи**

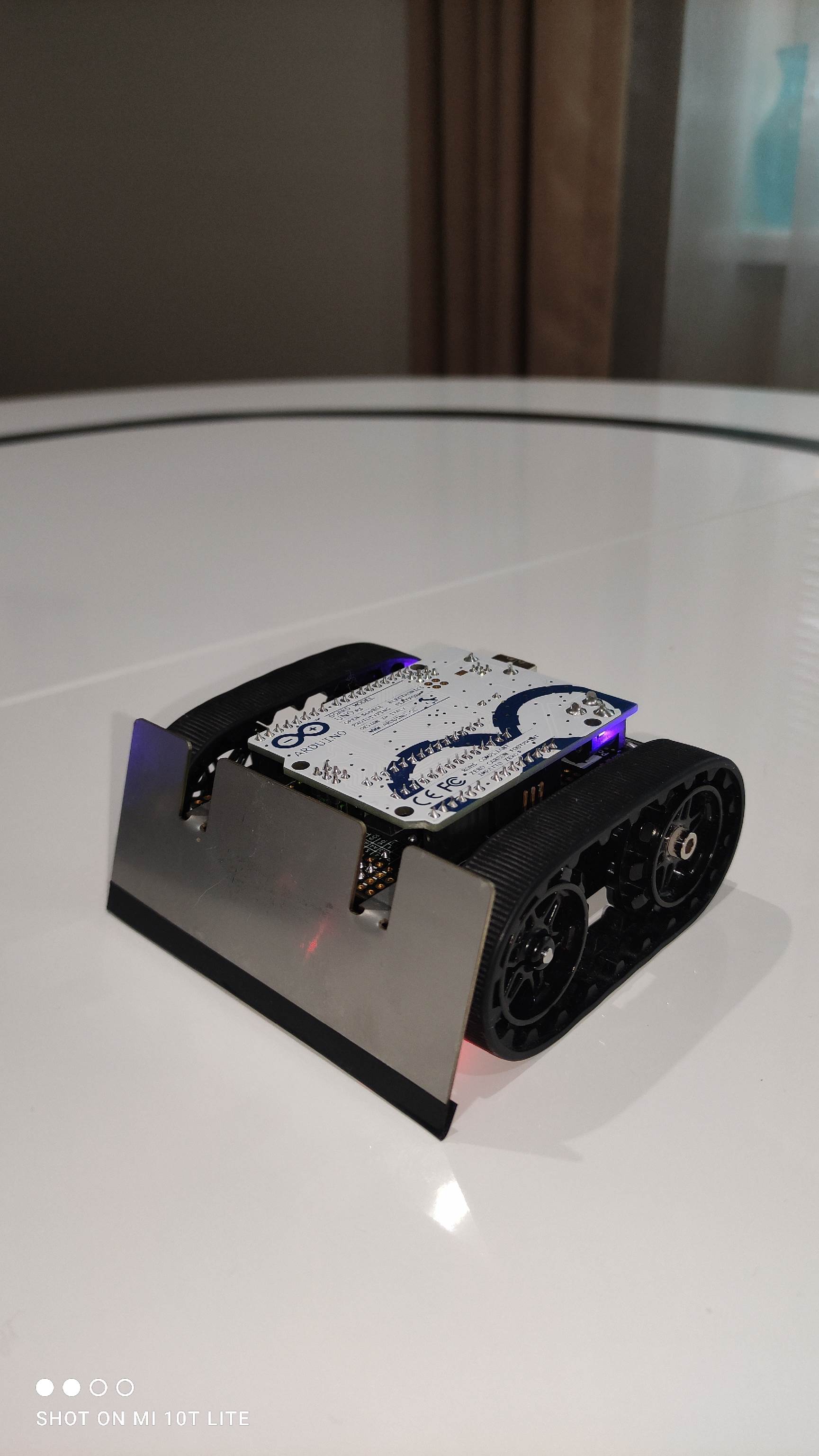
Този проект представя два Pololu Zumo робота за Arduino.



С цел осъществяване на реалистично мини-сумо състезание, софтуерът и хардуерът на роботите сe различават по следния начин (за по-лесно идентифициране на разликите между двата робота, те получават посочениете по-долу имена):

* ***MarsWalker / Марсоход***

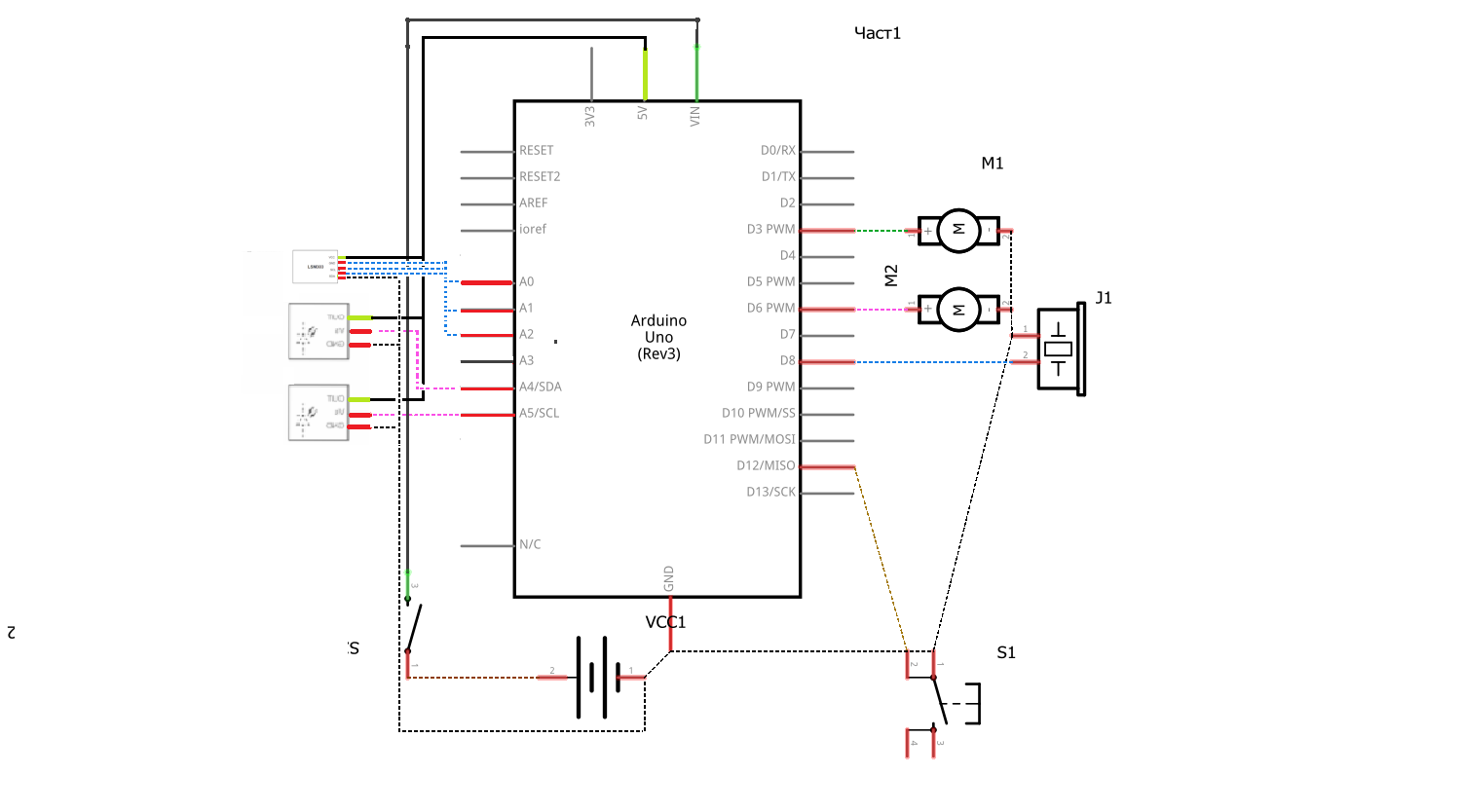
Способностите му са разширени с въвеждането на ултразвуков сензор. Чрез него Марсоходът ще засича обектите пред него. Това ще се отрази значително на поведението му по време на битка. Той ще е способен да атакува противника, дори когато е на разстояние от него.



* ***MoonWalker / Луноход***

Уменията му са оптимизирани с въвеждането на акселометър. Чрез него Луноходът ще засича обектите пред него. Това ще се отрази значително на поведението му по време на битка. Той ще е способен да реагира подобаващо при контакт/сблъсък с противника.

## **Блокова схема**

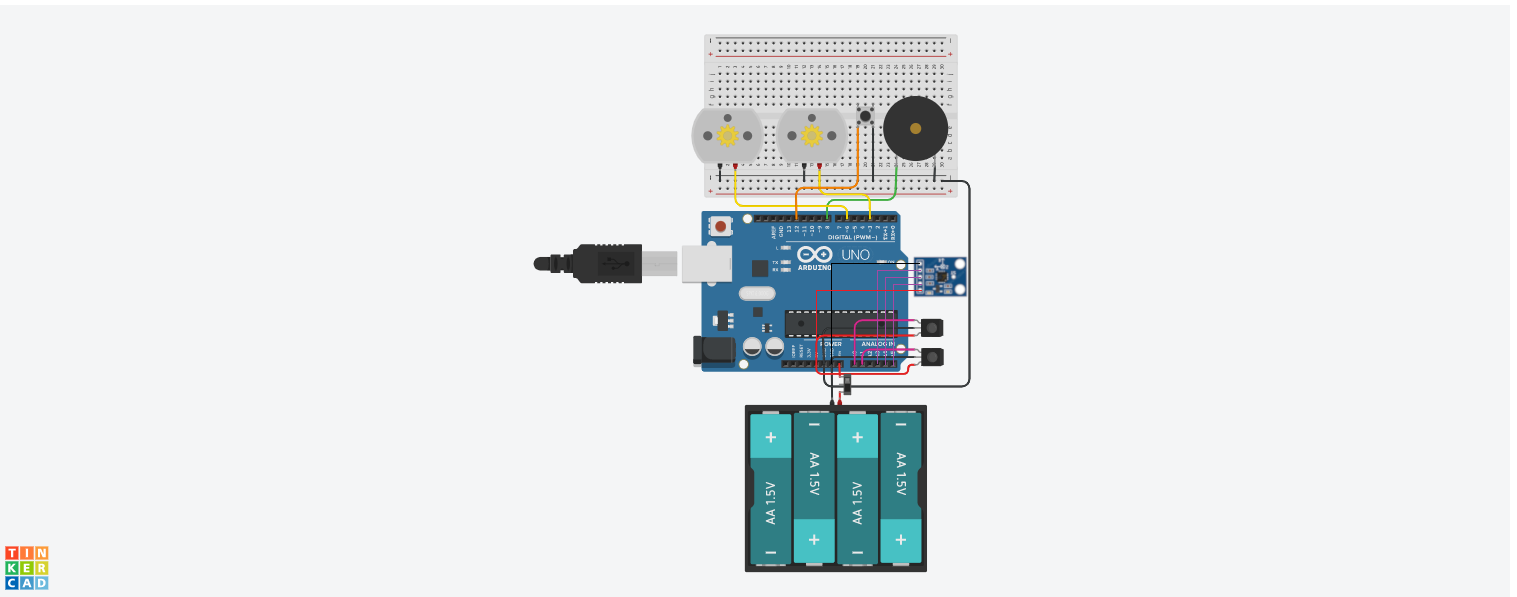


* **Луноход**
* **Марсоход**

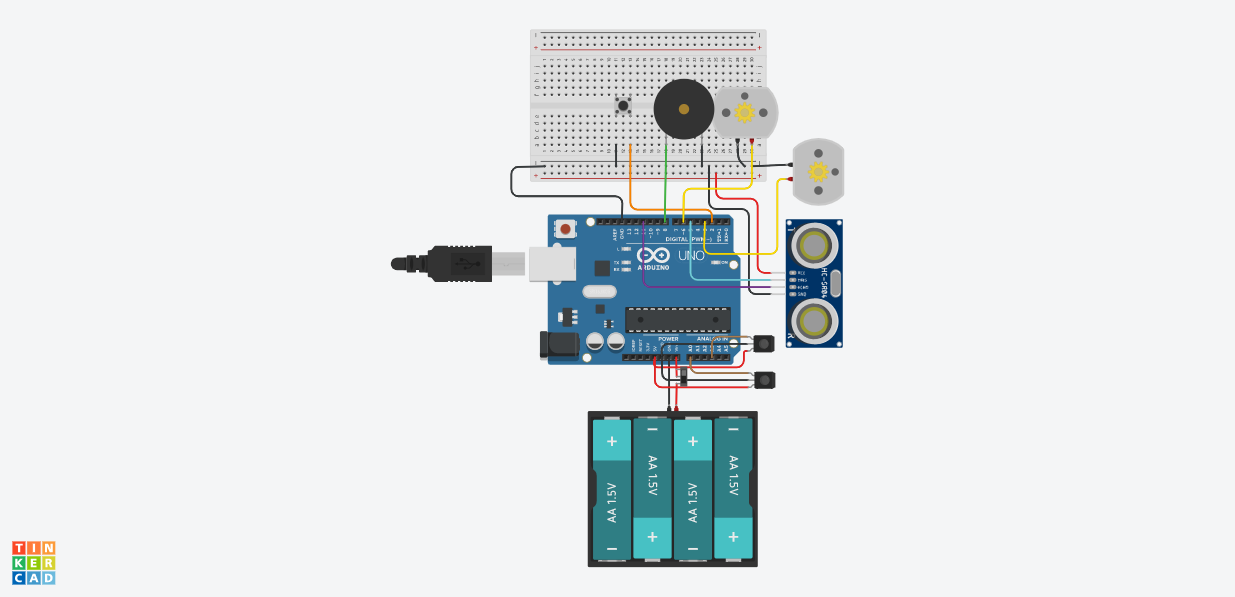


## **Електрическа схема**

* **Луноход**



* **Марсоход**



## **Списък компоненти**

## **Списък компоненти – Марсоход**

* Arduino Uno
* Pololu Zumo Shield
* 2 micro metal gearmotors
* dual motor drivers
* 2 silicone tracks
* stainless steel blade
* IR reflectance sensor array
* HC-SR04 ultrasonic distance sensor
* buzzer
* 4 AA batteries
* **Jumper wires**

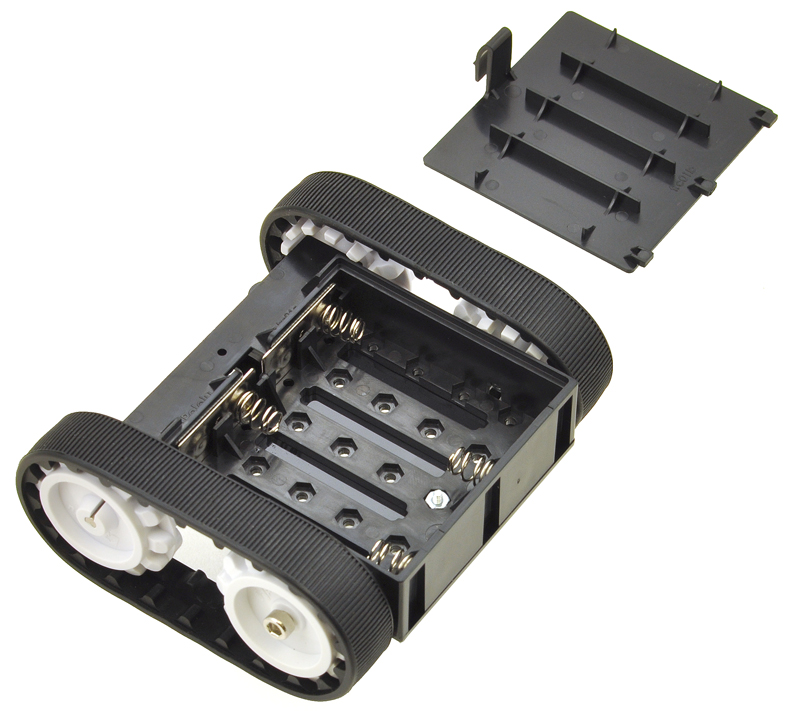
## **Списък компоненти – Луноход**

* Arduino Uno
* Pololu Zumo Shield
* 2 micro metal gearmotors
* dual motor drivers
* 2 silicone tracks
* stainless steel blade
* IR reflectance sensor array
* 3-axis accelerometer
* buzzer
* 4 AA batteries

# **Конструкция в снимки**



**1**



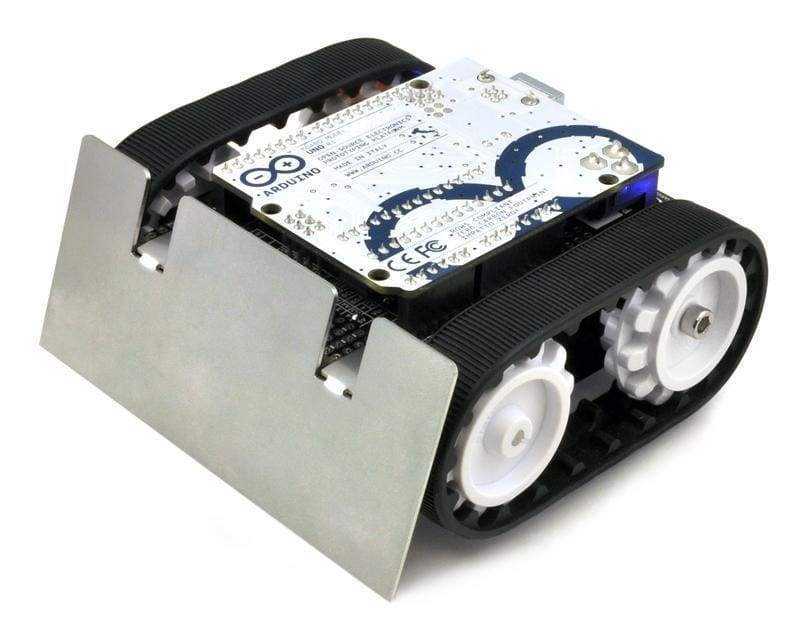
**2**

**3**



**4**

**5**



## **Описание на функционалността**

Настоящият проект проследява разработката на два самостоятелни сумо робота и за целта функционалността на всеки ще бъде разгледана поотделно.

# **Марсоход**

Софтуерът на Марсохода следва следната базисна логика:

* Подсигурява се, че роботът не прекосява границата на игралния ринг
* Задейства се процес на „атака“, ако опонентът бива засечен
* Сканира се игралния ринг в търсене на опонента

С най-голям приоритет и важност за мини-сумо състезанието е прекосяването на линията, точно това е главната насока при разработката на софтуера на Марсохода.

Единствено и само след проверка на повърхността под робота (използвайки данните, получени от модула от инфрачервени сензори), се задейства проверка за други явления и проверки на околната среда.

Това се забелязва в главната loop() функция от source code-а на Марсохода:

void loop()

{

// increase the timer with the milliseconds passed since switched on

currentTime = millis();

/\* check which proccess should be performed:

\* not crossing the border is of highest importance

\* only then we check for other circumstances

\* the attacking/moving forward process starts if there is an opponent detected

\* otherwise the robot starts searching/scanning the ring for the opponent

\*/

if (CheckForBorder())

{

currentProccess = AVOIDBORDER;

}

else

{

if (CheckIfOpponentIsDetected())

{

currentProccess = MOVEFORWARD;

}

else

{

currentProccess = SEARCHOPPONENT;

}

}

// take a correct action depending on what proccess the robot is in

switch (currentProccess)

{

case AVOIDBORDER: AvoidBorder();

break;

case MOVEFORWARD: MoveForward();

break;

case SEARCHOPPONENT: SearchForOpponent();

break;

}

}

Способностите на Марсохода се реализират благодарение на функции, които биват извикани само и единствено при нужда (след проверка за проявили се събития). Пример за това е атакуването на опонента само след като се подсигури, че той е бил засечен. Осъществява се чрез следните функции, за чиито свързване е отговорен switch..case statement в loop().

/\*\*

\* Function used to check whether the robot has stepped on a border and stores where the border is

\* if any sensor detects a black line, the function returns true, otherwise returns false

\*/

bool CheckForBorder()

{

reflectanceSensors.read(sensor\_values);

if (sensor\_values[5] > QTR\_THRESHOLD )

{

lastTurnsDirection = 'R';

return true;

}

else if (sensor\_values[0] > QTR\_THRESHOLD)

{

lastTurnsDirection = 'L';

return true;

}

return false;

}

/\*\*

\*Function used to make the robot avoid the border

\*if a black line is detected by the rightmost sensor, it calls the TurnRight() function to make the move

\*if a black line is detected by the leftmost sensor, it calls the TurnRight() function to make the move

\*/

void AvoidBorder()

{

motors.setSpeeds(0,0);

//while there is a border under the robot

while (CheckForBorder())

{

//the robot should reverse

motors.setSpeeds(REVERSE\_SPEED, REVERSE\_SPEED);

}

//if a border is detected on the left

if (lastTurnsDirection == 'L')

{

//the robot should turn to the right

TurnRight();

}

//otherwise if a border is detected on the right

else if (lastTurnsDirection == 'R')

{

//the robot should turn to the left

TurnLeft();

}

// pause for some time after a turn

delay(turnDuration);

// after making a turn the robot should start searching its opponent

currentProccess = SEARCHOPPONENT;

}

# **Луноход**

Софтуерът на Лунохода следва следната базисна логика:

* Подсигурява се, че роботът не прекосява границата на игралния ринг
* Задейства се процес на „атака“, ако контакт с опонента бива засечен
* Сканира се игралния ринг в търсене на опонента

С най-голям приоритет и важност за мини-сумо състезанието е прекосяването на линията, точно това е главната насока при разработката на софтуера на Лунохода.

Единствено и само след проверка на повърхността под робота (използвайки данните, получени от модула от инфрачервени сензори), се задейства проверка за други явления и проверки на околната среда.

Това се забелязва в главната loop() функция от source code-а на Лунохода:

void loop()

{

if (button.isPressed())

{

// if button is pressed, stop and wait for another press to go again

motors.setSpeeds(0, 0);

button.waitForRelease();

CountDownAndResetLoop(true);

}

startTime = millis();

acc.readAcceleration(startTime);

sensors.read(sensor\_values);

if ((startTime - fullSpeedTime > FULL\_SPEED\_DURATION\_LIMIT))

{

motors.setSpeeds(FULL\_SPEED,FULL\_SPEED);

}

BlackBorderDetection();

}

Способностите на Лунохода се реализират благодарение на функции, които биват извикани само и единствено при нужда (след проверка за проявили се събития). Пример за това е атакуването на опонента само след като се подсигури, че е засечен контакт с него (използвайки акселометъра).

if (CheckForContact())

{

ContactMade();

}

else

{

motors.setSpeeds(FORWORD\_SPEED, FORWORD\_SPEED);

}

## **Общи тактики**

Тъй като мини-сумо състезанието не е игра с много сложен или дълъг правилник, тактиките на Марсохода и на Лунохода се припокриват от части. Това е така най-вече заради главаната цел на всеки играч в сумо състезание – „Не излизай от игралния ринг!“

За подобряване поведението и на двата робота се дефинират различни скорости на движението, различни видове процеси, използвани за задействане на робота, няколко таймера, отсичащи времето за изпълняване на даден процес и определящи последващо действие.

Също така, поради високата им оптимизация, и двата робота използват предефинирани обекти и функции от библиотеката “Popolu Zumo Shield” и “Wire”.

## **Провеждане на тестово състезание**

За да гледате мини-сумо състезанието между *Луноходеца и Марсоходеца*, моля натиснете [ТУК](https://github.com/SVoynova/Mini-Sumo-Robot-Competiton/blob/main/README.md)!

## **Заключение**

Настоящият проект представи разработването на два сумо робота, базирани на микроконтролерна платка Arduino. За реализацията на проекта са прегледани различни характеристики и особености на сензорите и компонентите, както и основните технологии за изграждане на такъв тип роботи. След процеса на проучване, планиране и изследване са реализирани два реални сумо робота. За потвърждаване функционалността и надежността на изразботките са упоменати и документирани няколко „битки“ между двата робота.

# **Възможности за бъдещо надграждане**

Arduino е добра платформа за обучение и разработка на проекти с различни приложения. Богатото разнообразие от компоненти за микроконтролерната платка предлага възможност за реализиране на широка гама от идеи за разработка. Също така и голям потенциал за осъвършенстване на вече създаден проект.

Основно направление за надграждане е оптимизирането на софтуерната част на текущия проект.

Друго направление за развитие е подобряване и развиване на хардуерната част. Възможно е да се добавят повече сензори. Добра идея би било да се интегрират различни видове сензори за сканиране на заобикалящата среда. Закачането на ултазвуковия сензор за допълнително поставен серво мотор би увеличило значително възможността за победа на Марсохода.

За подсигуряване наличието на захранваща енергия би могло да се добави LCD екран както и да се разработи проверка за състоянието на батерията.

Микроконтролерите могат да бъдат удобни за разработка в много случаи, така че в един такъв проект винаги има място за растеж.