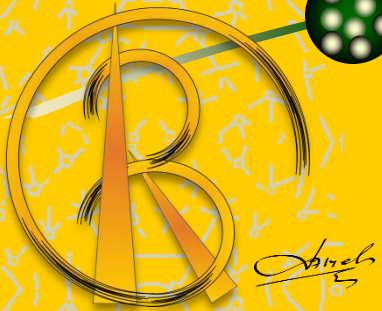
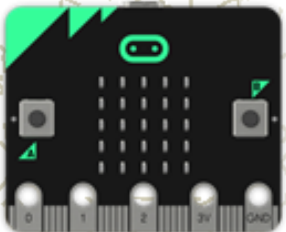


awesome
micro:bit



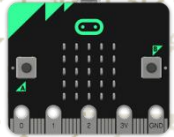
Роботика и компютърно моделиране с MicroBit

DFRobot – Micro IO box

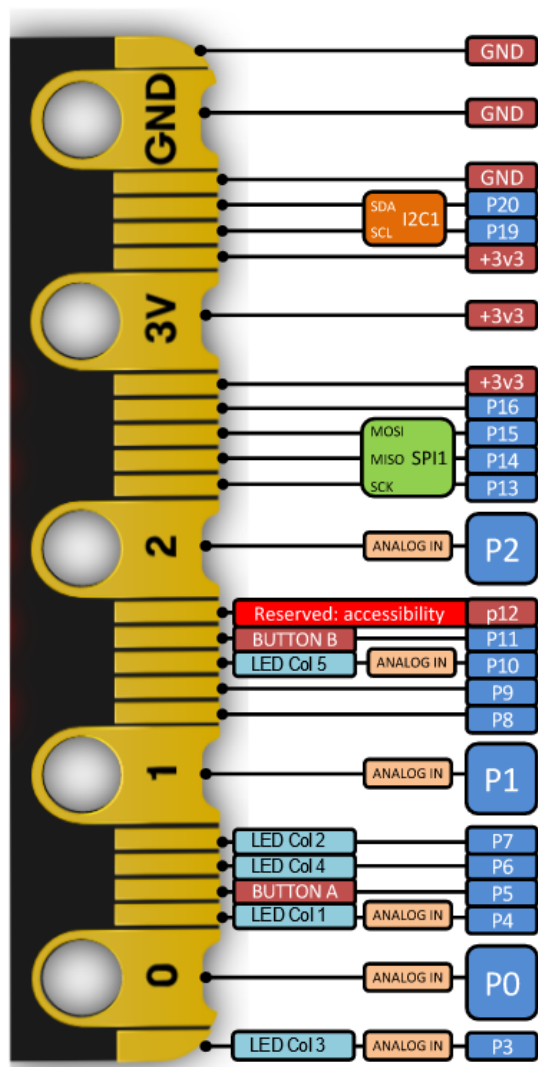
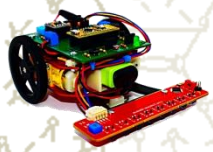


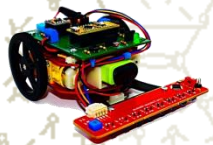


Карта на пиновете



**SOCIETY
ROBOTIC**







Функция MAP

- Функцията **map()** пренасочва число от един диапазон към друг;
- Не ограничава стойностите в диапазона, тъй като стойностите извън диапазона понякога са предвидени и полезни;

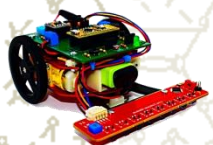


променлива

от интервал

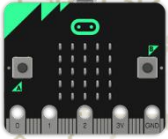
в интервал

SOCIETY
ROBOTIC-

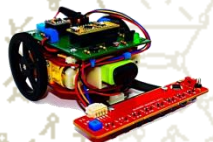




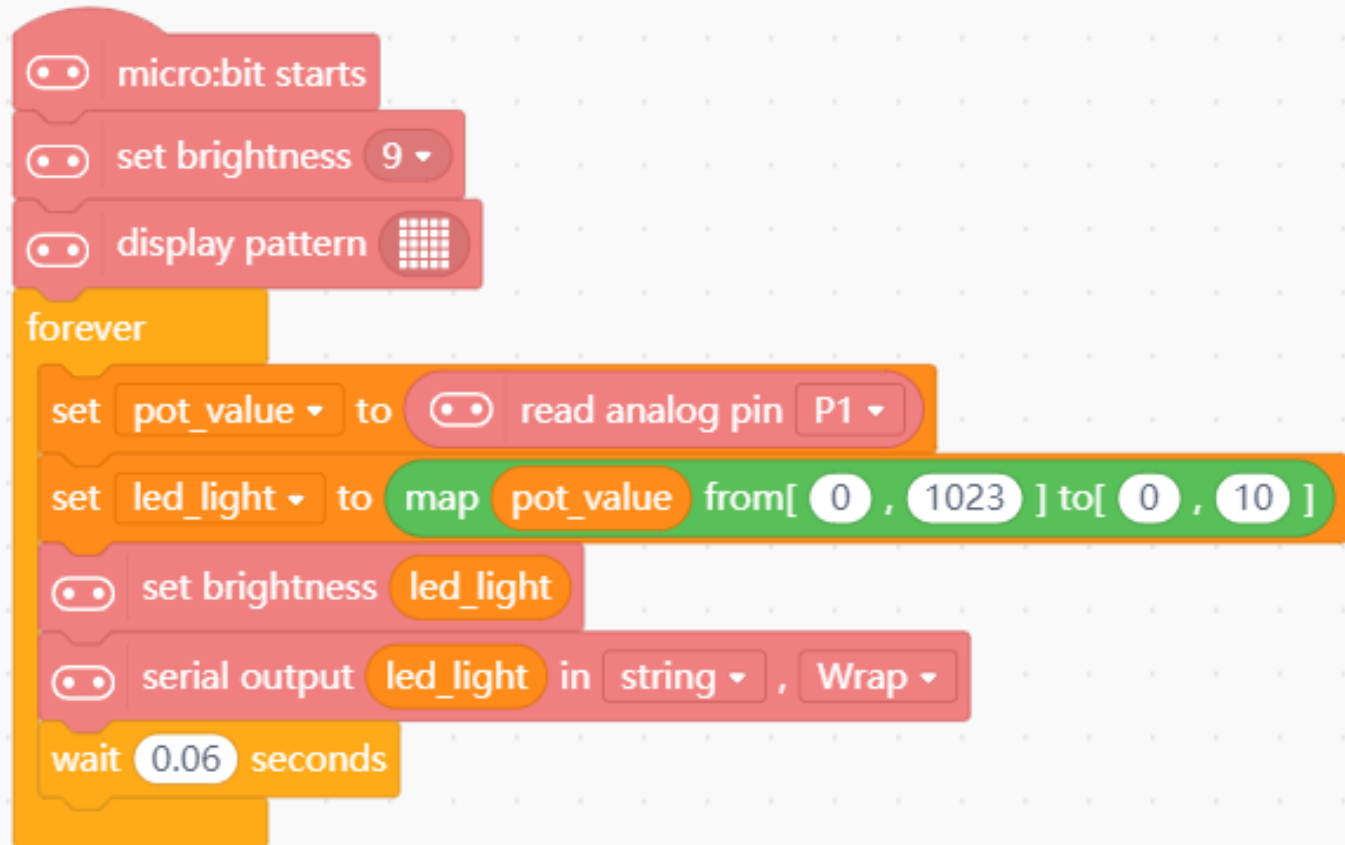
Society for Robotics



SOCIETY
ROBOTIC--



map_pot_matrix_1





Функция Constrain

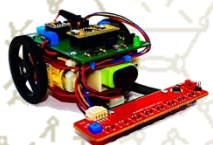
Функцията `constrain()` се използва, за да се ограничи дадена стойност в определен диапазон;



стойност

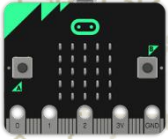
Долна граница
на диапазона

Горна граница
на диапазона

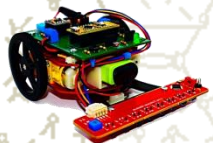




Society for Robotics



SOCIETY
ROBOTIC



constrain_pot_2

micro:bit starts

forever

set pot_value to read analog pin P1

set old_value to map pot_value from [0 , 1023] to [0 , 500]

set new_value to constrain old_value between (min) 0 and (max) 255

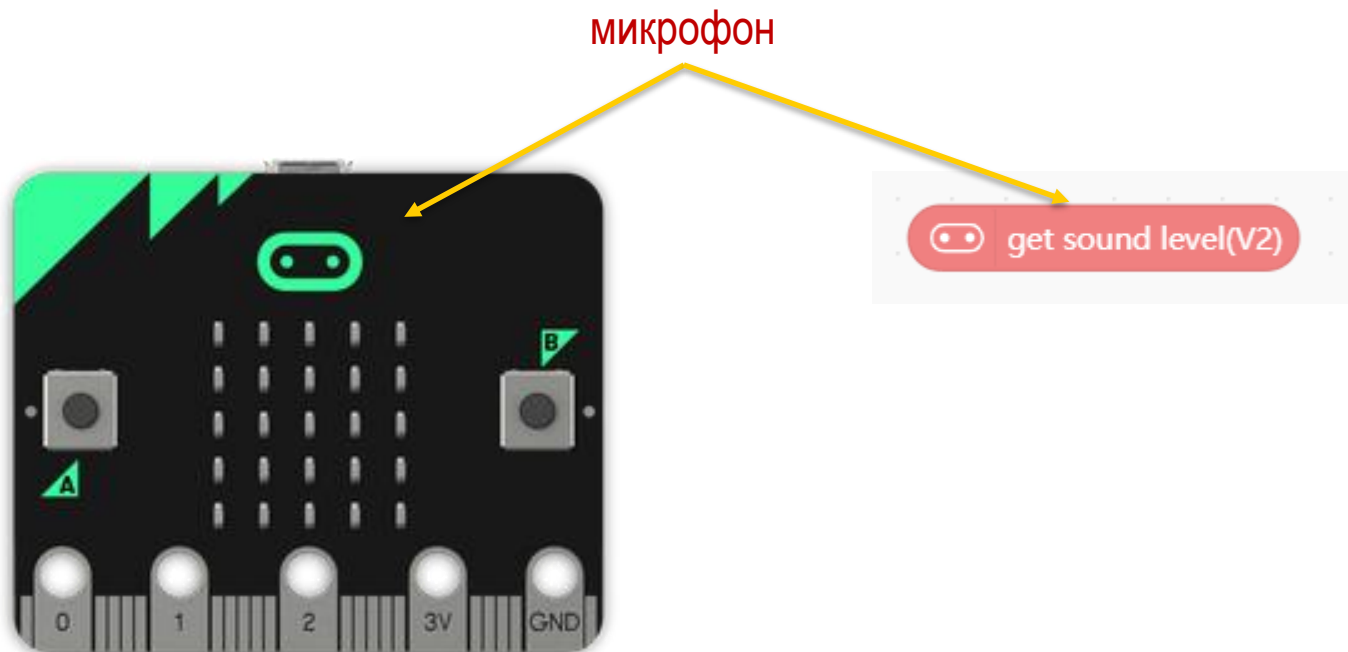
serial output new_value in string , Wrap

wait 0.5 seconds



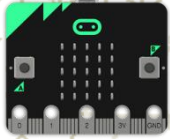
Микрофон

Microbit разполага с вграден микрофон за приемане на звукови сигнали. Той е свързан със специално обозначен пин;

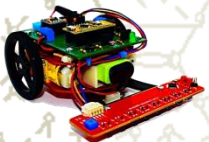




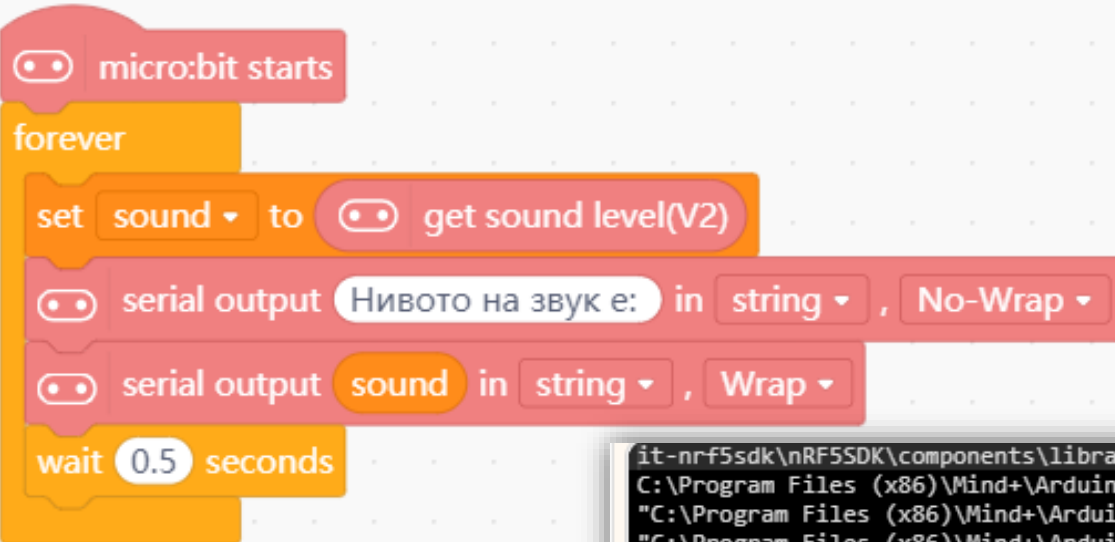
Society of Robotics



SOCIETY OF ROBOTICS



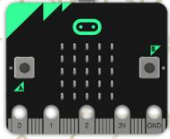
mic_sound_3



```
it-nrf5sdk\nRF5SDK\components\libraries\util -I C:\Program Files (x86)\Mind+Arduino\hardware\tools\nRF5\gcc-a
C:\Program Files (x86)\Mind+Arduino\hardware\tools\nRF5\gcc-a
"C:\Program Files (x86)\Mind+Arduino\hardware\tools\nRF5\gcc-a
"C:\Program Files (x86)\Mind+Arduino\hardware\tools\nRF5\gcc-a
The project uses 91984 bytes, occupies (18%) program memory sp
Global variables use 8896 bytes, (7%) of dynamic memory, leavi
upload success
Нивото на звук е: 7.00
Нивото на звук е: 33.00
Нивото на звук е: 26.00
Нивото на звук е: 11.00
Нивото на звук е: 15.00
Нивото на звук е: 26.00
```



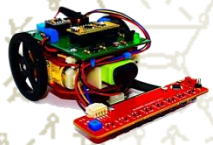
Управление на DC мотор

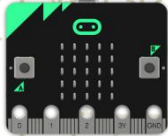


⚡ DC моторът е устройството, което преобразува постоянния ток в механична работа.

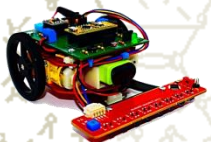
⚡ Използва се принципа на закона на Лоренц, който гласи, че "Токопроводимият проводник, поставен в магнитно и електрическо поле, изпитва сила". Тази сила се нарича сила на Лоренц.


⚡ Лявото правило на Флеминг дава посоката на силата.





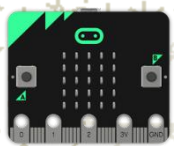
SOCIETY
ROBOTIC—



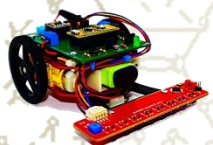
 **Правило от дясната ръка на Fleming** - ако палеца, показалеца и средния пръст от дясната ръка се изместват един от друг под ъгъл от 90° , палецът сочи посоката на тока, показалецът сочи посоката на магнитното поле, а посоката на средния пръст указва посоката на електродвижещата сила, действаща върху проводника.



Abriel

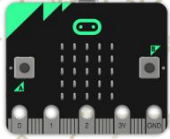


SOCIETY
ROBOTIC

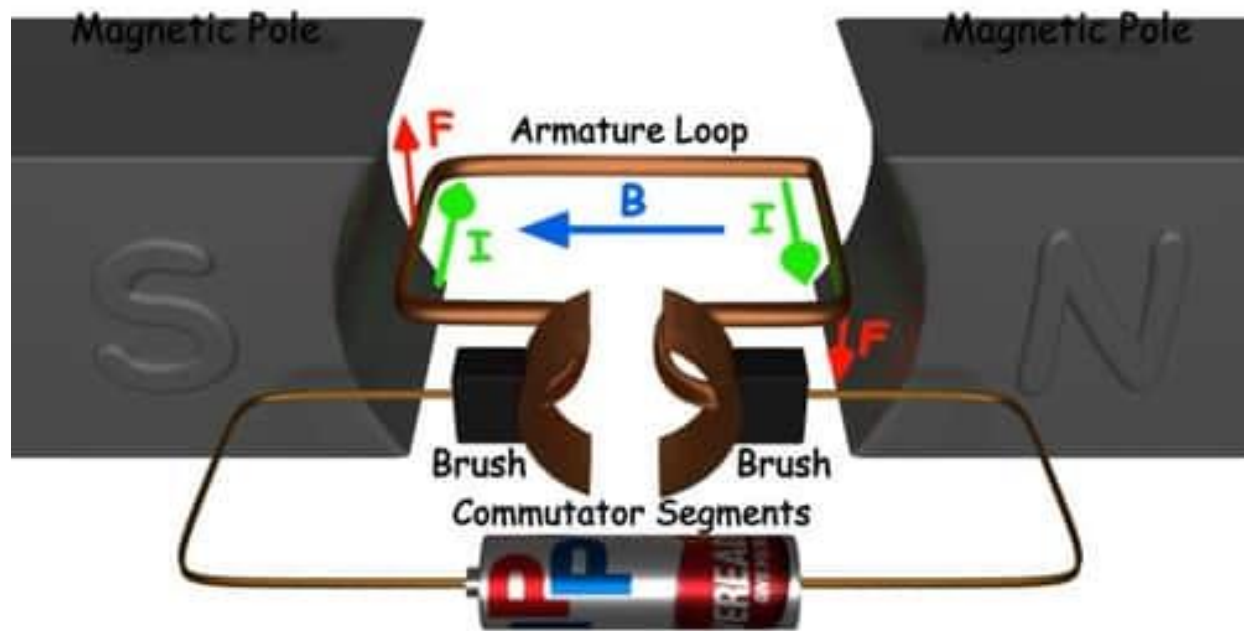
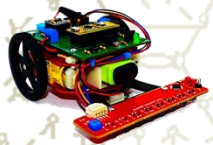


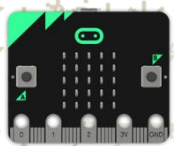


Society of Robotics

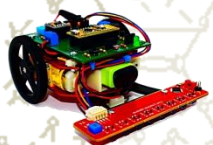


SOCIETY OF ROBOTICS





SOCIETY
ROBOTIC



motor M1 rotate CW at speed 200

✓ M1
M2
ALL(-1)

Мотори

motor M1 rotate CW at speed 200

✓ CW
CCW

Посока на въртене

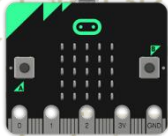
motor M1 stops

✓ M1
M2
ALL(-1)

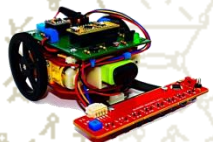
Спиране на мотори



PWM сигнал

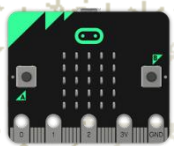


- Широчинно-импулсната модулация (PWM) е метод за намаляване на средната мощност, доставяна от електрически сигнал, чрез ефективното му нарязване на отделни части;
- Средната стойност на напрежението (и тока), подадена към товара, се контролира чрез бързо включване и изключване на превключвателя между захранване и натоварване;

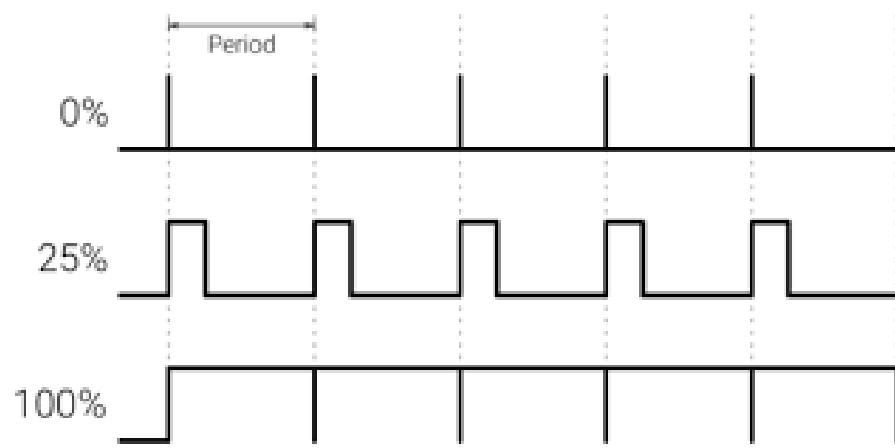




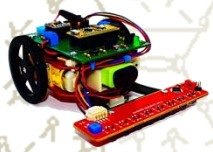
Abir



Честотата на превключване на PWM трябва да бъде достатъчно висока, за да не повлияе на товара, което означава, че резултантната форма на вълната, възприемана от товара, трябва да бъде възможно най-плавна;

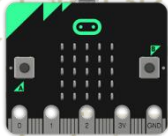


SOCIETY
ROBOTIC-





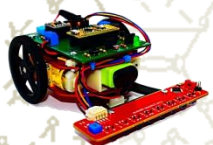
Компас

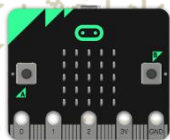


☀ Магнитният компас е магнитен навигационен уред за ориентиране в местност, чрез определяне на посоките на света;

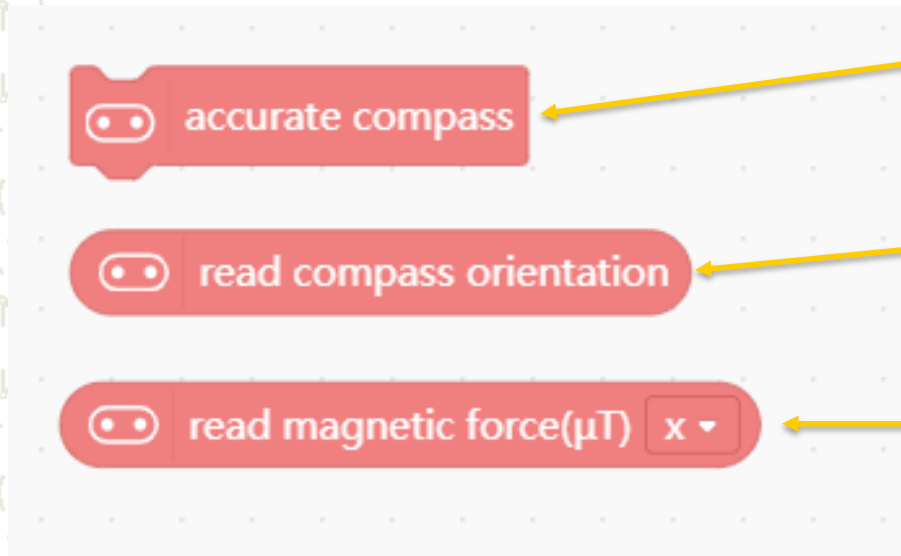
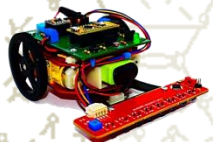
☀ Показания

0 - 22	North (север)
23 - 68	NE (североизток)
69 - 113	East (изток)
114 - 158	SE (югоизток)
159 - 201	South (юг)
202 - 248	SW (югозапад)
249 - 291	West (запад)
292 - 338	NW (северозапад)





**SOCIETY
ROBOTIC**



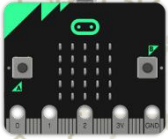
Калибриране на компаса

Прочитане на показанията
от компаса

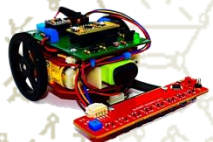
Прочитане на показанията
на магнитната сила от компаса



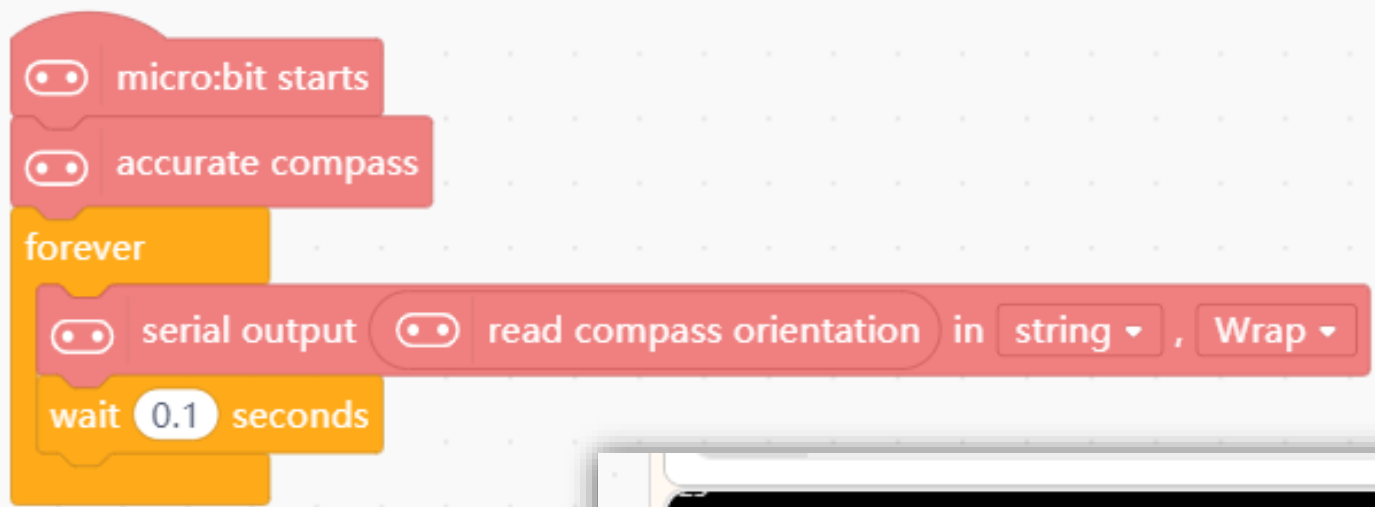
Abir



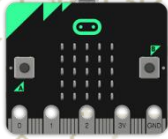
**SOCIETY
ROBOTIC**



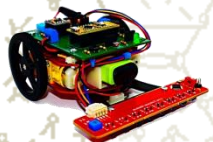
read_compass_4



```
23
11
26
25
24
341
48
27
69
68
24
36
358
348
50
15
```



SOCIETY
ROBOTIC



read_compass_4a

micro:bit starts

accurate compass

forever

set compass to read compass orientation

if compass > 0 and compass < 23 then

display pattern

else if compass > 69 and compass < 114 then

display pattern

else if compass > 159 and compass < 202 then

display pattern

else if compass > 249 and compass < 292 then

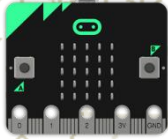
display pattern

serial output compass in string, Wrap

wait 0.1 seconds



Преобразуване на типове данни



- Можем да преобразуваме число в текст със следната функция:

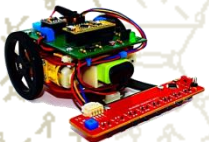
convert number 132 to string

- Можем да преобразуваме текст в цяло или реално число със следната функция:

convert string "123" to Integer ▾

✓ Integer
Decimal

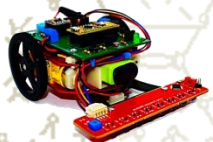
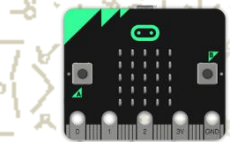
SOCIETY
ROBOTICS

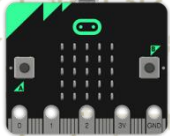




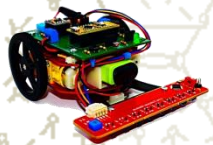
Neopixel

- 💡 **Neopixel** са модули от индивидуално адресируеми RGB светодиоди, управлявани с драйвер WS2812, работещи с напрежение от 3 до 5V;
- 💡 **Neopixel** позволяват последователно добавяне на светодиоди, които се управляват от едни канал (пин);






SOCIETY
ROBOTICS



MicroBit IO shield има **4 бр.** индивидуално адресируеми RGB светодиоди, които се управляват от **пин 15** на MicroBit;

pin P15 the **0** LED display color 

Задава цвят на светодиодите

red **255** green **255** blue **255**

Задава цвят на светодиодите

pin P15 rotate pixels by **0**

Разбърква през определена стойност определени светодиоди

pin P15 LED brightness **255**

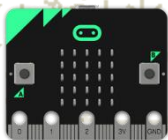
Задава яркост на светодиоди

pin P15 clear all LEDs

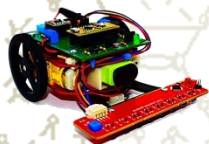
Изгася светодиоди



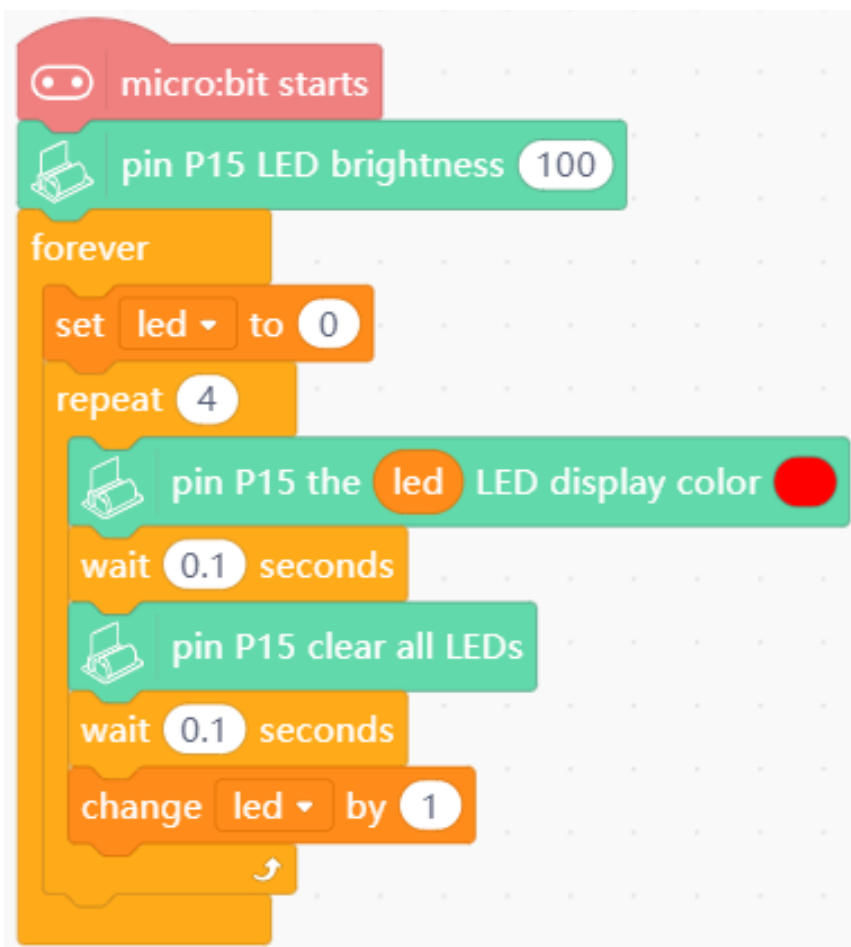
Abirah



SOCIETY
ROBOTIC



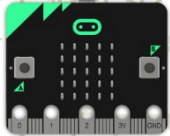
led_5



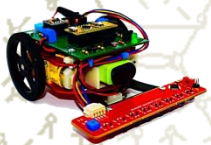
Бягаци светлини



Abir



SOCIETY
ROBOTIC



led_button_5a

micro:bit starts

pin P15 LED brightness 100

when button A pressed

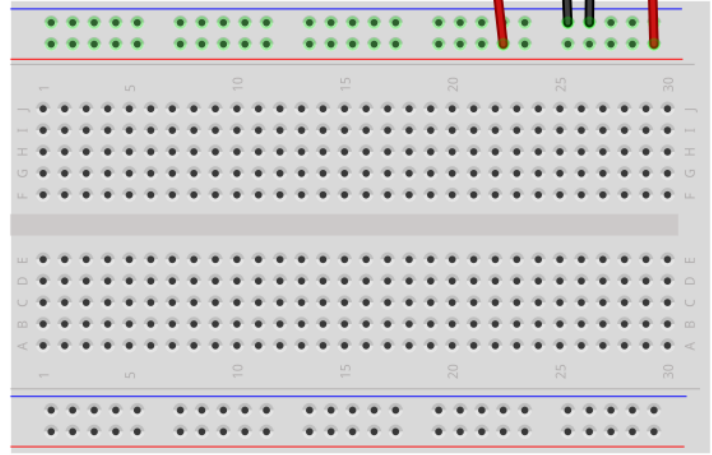
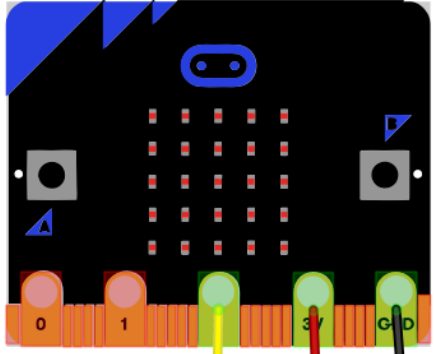
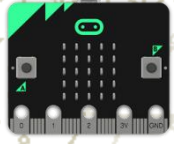
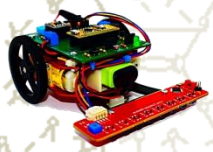
pin P15 the ALL(-1) LED display color red 100 green 50 blue 255

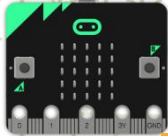
when button B pressed

pin P15 clear all LEDs

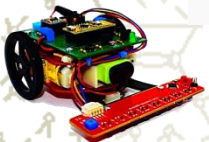


**SOCIETY
ROBOTIC**





**SOCIETY
ROBOTIC**



← Back

Select Display

Board

Kit

Shield

Sensor

Actuator

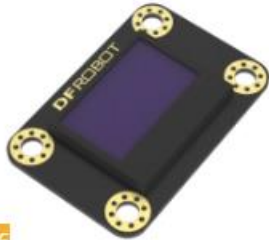
Communication

Display

Fun

Can't find what you want? [Click here](#) to find more

DFR0486



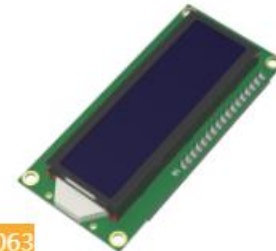
OLED-12864 Display
I2C OLED-12864 Display
Module

DFR0352



WS2812 RGB LED Strip
Control WS2812-based LED
strip modules

DFR0063

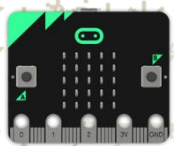


LCD1602 Display
LCD module which can show
2 lines and 16 characters in
each line

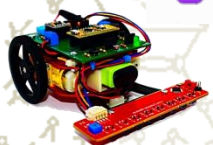
DFR0021



Digital LED I/O
White, red, green
LED module



SOCIETY
ROBOTIC



► (FIT0352)WS2812 RGB LED Strip

pin P2 7 RGB LEDs

pin P2 LED brightness 255

pin P2 clear all LEDs

pin P2 RGB 0 to 4 show color

red 255 green 255 blue 255

pin P2 0 to 4 RGBs show gradient color from 1 to 360

pin P2 shift pixels by 0

pin P2 rotate pixels by 0

pin P2 0 to 4 RGB LEDs show histogram current value: 7 max value: 7

Задава брой светодиоди

Задава яркост на светодиоди

Изгася светодиоди

Задава цвят на всички светодиоди

Задава определен цвят на всички светодиоди

Задава градиент на всички светодиоди

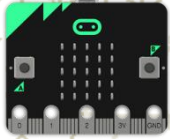
Изгася светодиодите със стъпка

Разбърква през определена стойност определени светодиоди

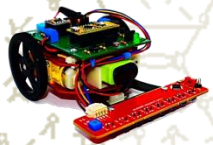
Светва зададен брой светодиоди като хистограма



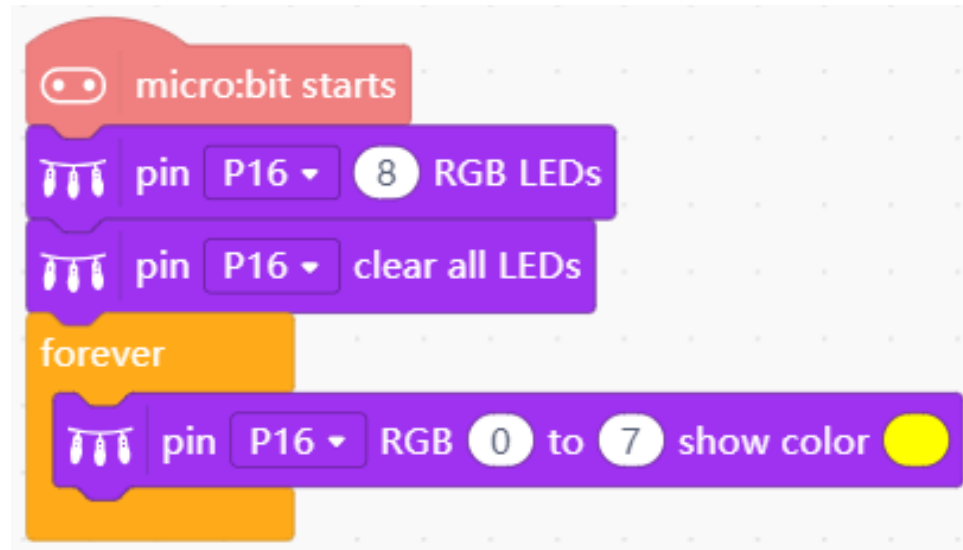
Society of Robotics



**SOCIETY
ROBOTIC**

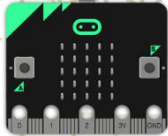


rgb_leds_6

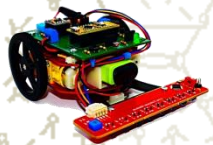




Society of Robotics



SOCIETY
ROBOTIC



random_leds_6a

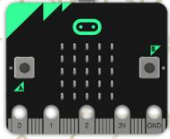
```
micro:bit starts
pin P16 8 RGB LEDs
pin P16 clear all LEDs
forever
  set RED to pick random 1 to 255
  set GREEN to pick random 1 to 255
  set BLUE to pick random 1 to 255
  pin P16 RGB 0 to 7 show color red RED green GREEN blue BLUE
  wait 1 seconds
```

Случайни
цветове

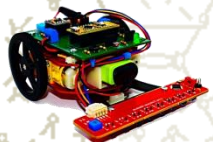




Society for Robotics



SOCIETY
ROBOTIC

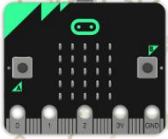


leds_low_step_6b

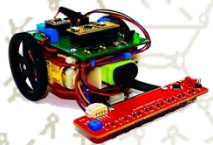




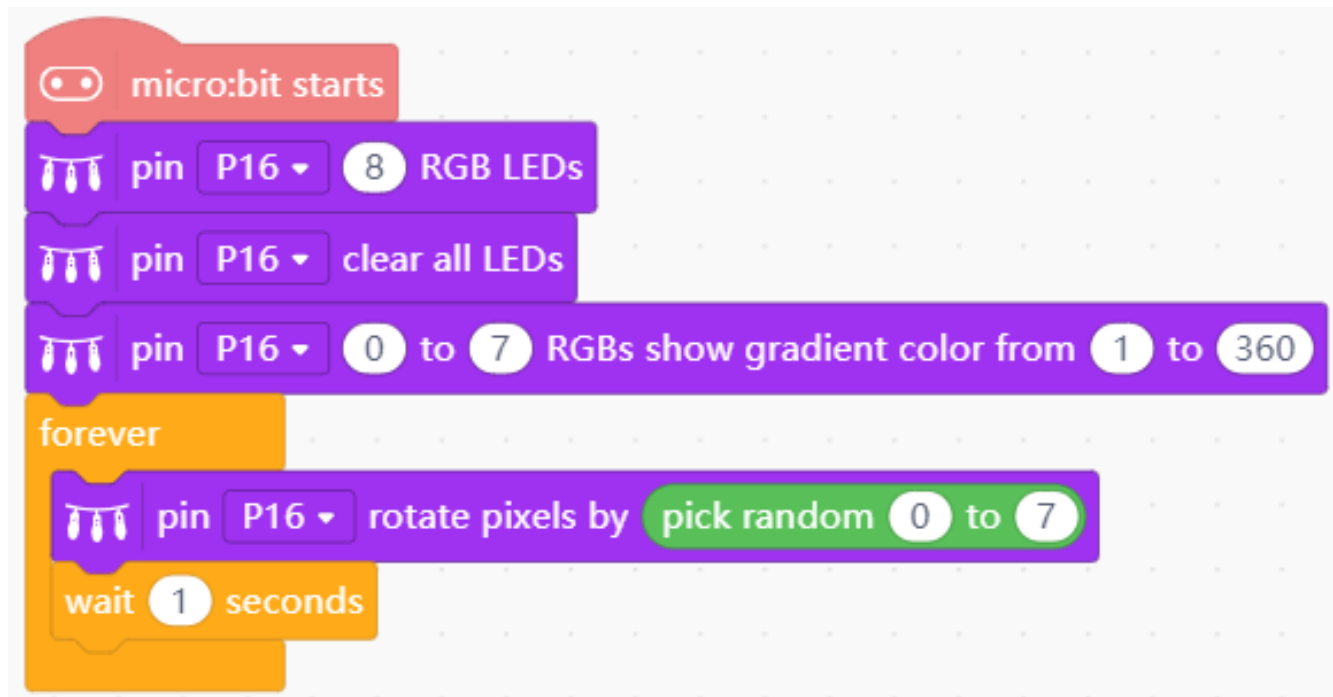
Abirah



SOCIETY
ROBOTIC

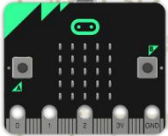


random_leds_gradient_6c

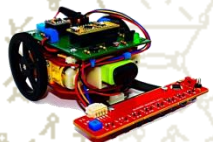




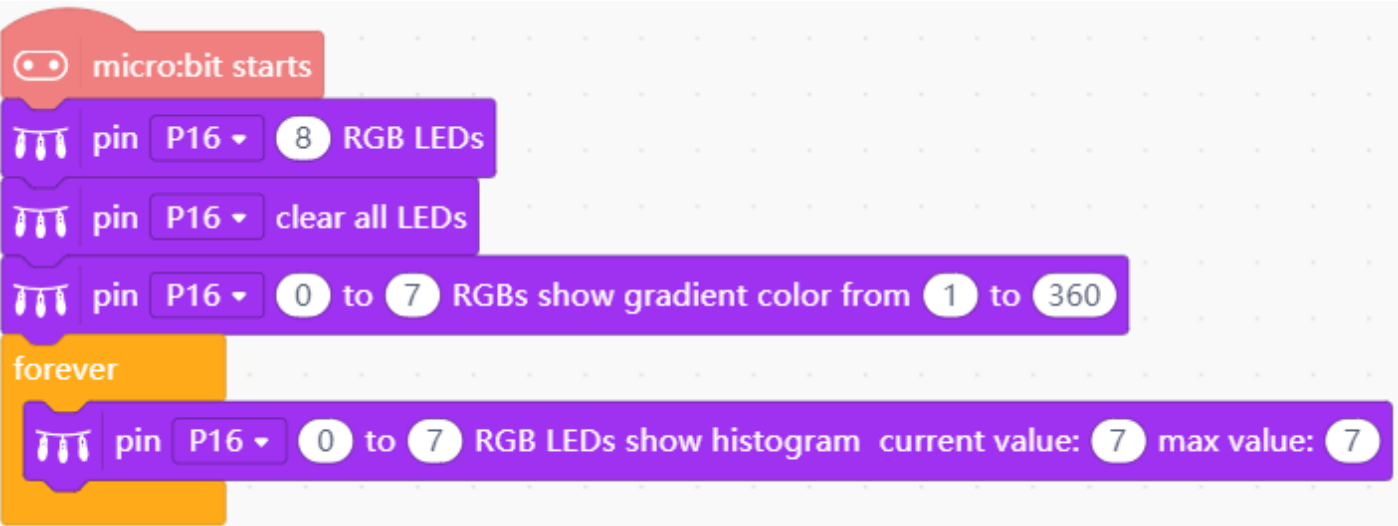
Abirah

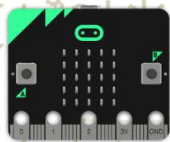


SOCIETY
ROBOTIC

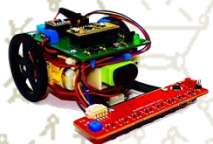


leds_histogram_6d





SOCIETY
ROBOTIC



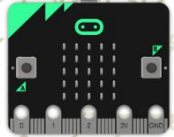
rgb_leds_6e

```
micro:bit starts
pin P16 8 RGB LEDs
pin P16 clear all LEDs
forever
  set index to 0
  repeat until index >= 8
    pin P16 RGB 0 to index show color green
    wait 0.5 seconds
    change index by 1
  pin P16 clear all LEDs
```

Бягащи светлини

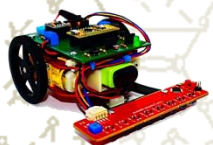


WiFi комуникация



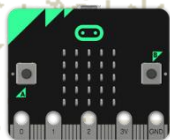
- 💡 Централният процесор (CPU) на microbit v2 е **Nordic Semiconductor nRF52833**. Освен компютърен процесор с общо предназначение, този чип съдържа и вграден 2.4GHz радио модул.
- 💡 Това радио може да бъде конфигурирано по различни начини и е предназначено основно да работи с **Bluetooth Low Energy (BLE)** протокол. Въпреки това, той може да бъде поставен и в много по-опростен режим на работа, който позволява проста, директна комуникация от microbit към microbit.

SOCIETY
ROBOTICS



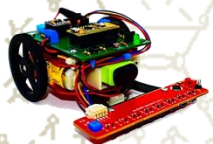


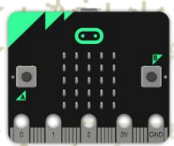
Abriel



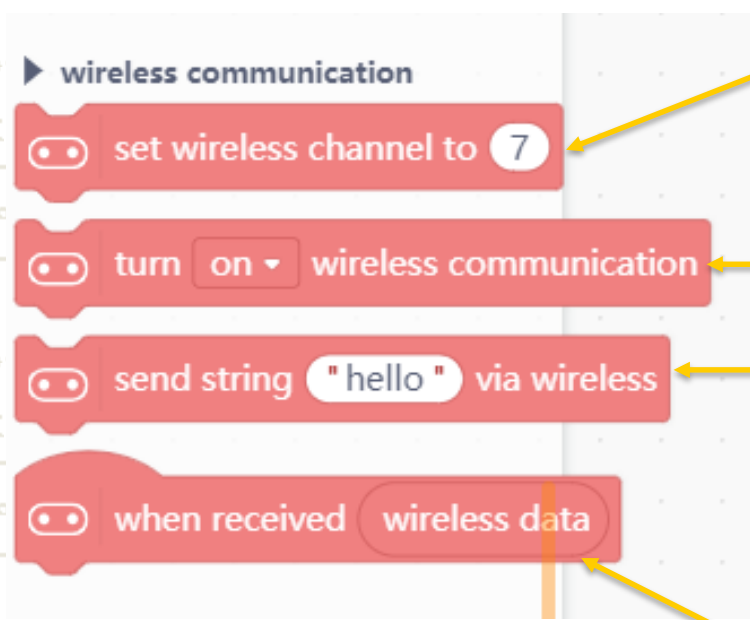
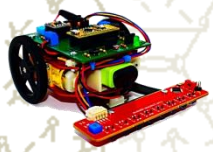
- Компонентът MicroBitRadio се състои от три класа - **MicroBitRadio**, **MicroBitRadioEvent** и **MicroBitRadioDatagram**;
- Заедно те предоставят възможност за изпращане на пакети с данни с общо предназначение от един microbit към друг и за разширяване на шина за съобщения, за да обхване множество microbits. Така ако се случи събитие на един microbit, можете да го получите на друг с помощта на нормалния механизъм за слушане.

SOCIETY
ROBOTIC





SOCIETY
ROBOTIC



Задава канал за комуникация
0 - 255 канала

Включва/Изключва комуникация

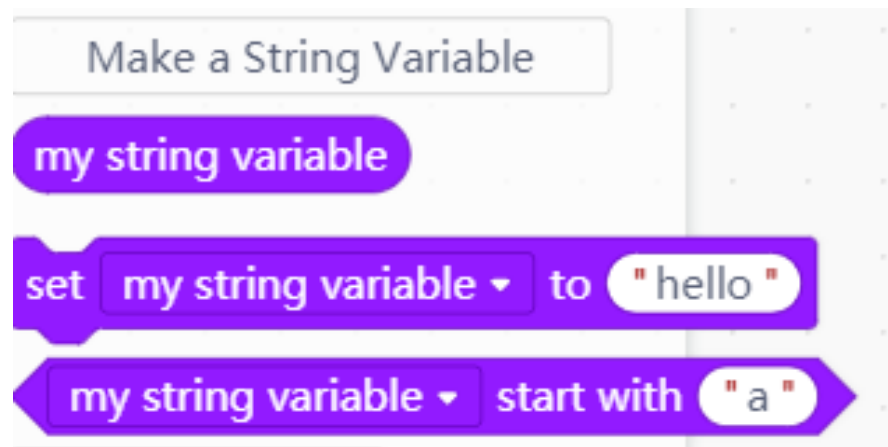
Изпраща текстово съобщение

Когато получи съобщение
го съхранява в променливата
wireless data

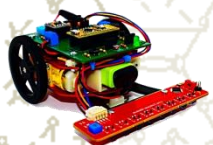


Текстова променлива

Можем да създаваме променлива от тип **STRING** като използваме следните блокове:



SOCIETY
ROBOTIC—



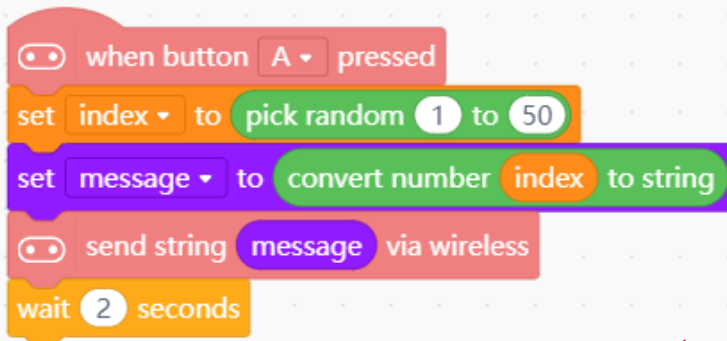
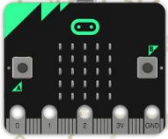


Society for Robotic Education

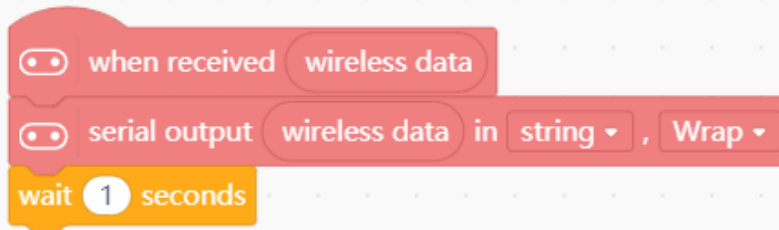
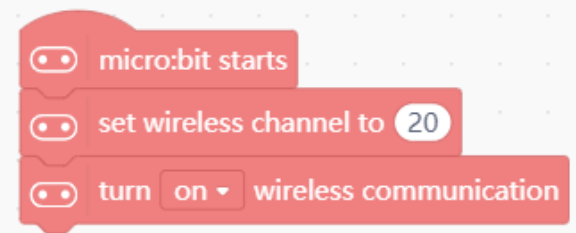
Случайно число

transmitter_7

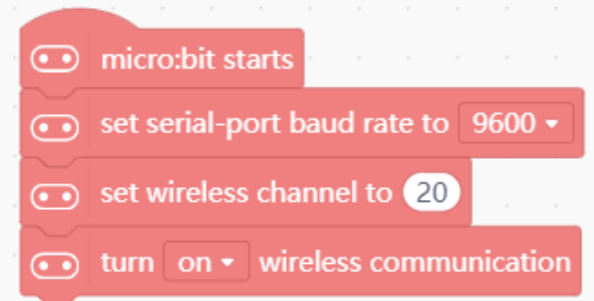
receiver_7a



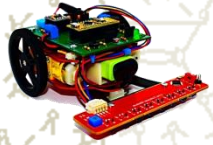
transmitter



receiver



SOCIETY
ROBOTIC



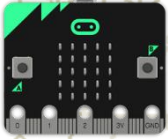


Society for Robotic Education

message_7b

Съобщение

message_7c



```
when button A pressed
  set message to "Hello"
  send string message via wireless
  wait 2 seconds
```

transmitter

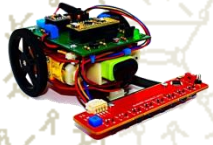
```
micro:bit starts
  set wireless channel to 20
  turn on wireless communication
```

```
when received wireless data
  set message to wireless data
  serial output message in string , Wrap
  wait 0.1 seconds
```

receiver

```
micro:bit starts
  set wireless channel to 20
  turn on wireless communication
```

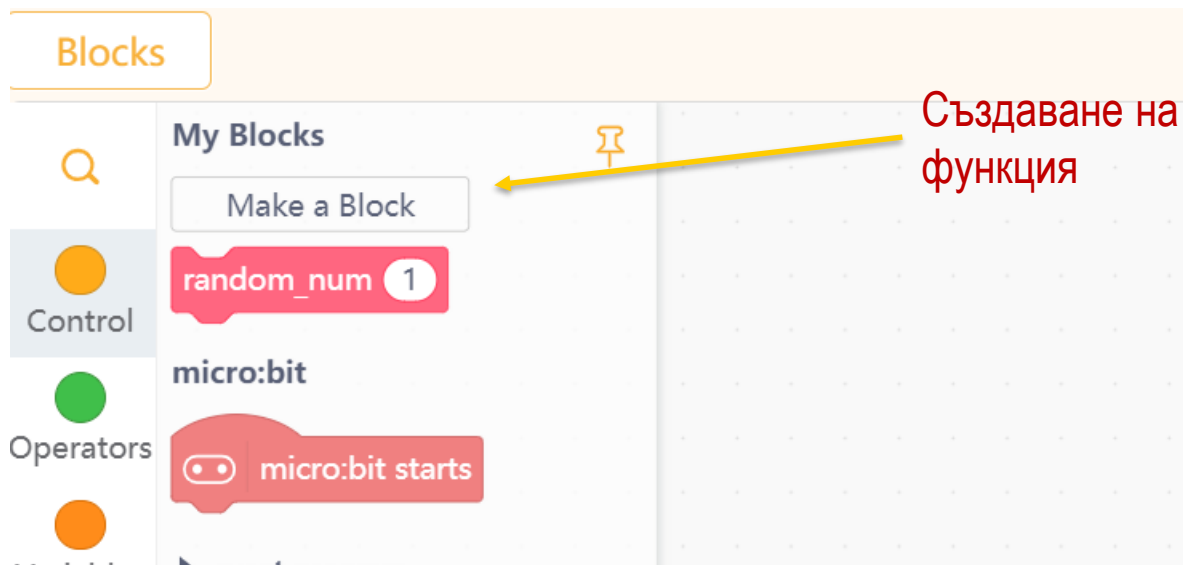
SOCIETY
ROBOTIC



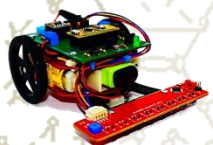


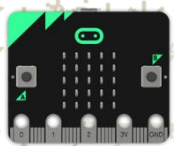
Дефиниране на функция

Можем да създадем собствена функция, като дефинираме съответните елементи на функцията и създадем код;

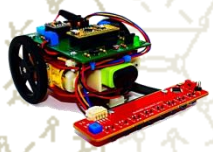


Създаване на блок за функция









**SOCIETY
ROBOTIC**





Make a Block


Add an input
text


Add an input
number

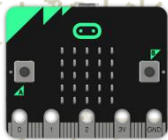

Add an input
boolean


Add a label

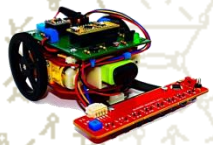
Cancel

OK

Добавяне на съответните компоненти към функцията



SOCIETY
ROBOTIC



my_function_8

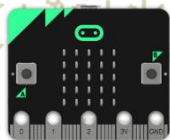
```
define random_num max_val
set rand_num to pick random 1 to max_val
serial output rand_num in string , Wrap

micro:bit starts
forever
  random_num 10
  wait 1 seconds
```

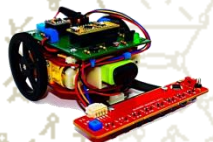
Генериране на случайно число в диапазон зададен от потребителя



Abriel



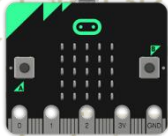
SOCIETY
ROBOTIC—



УПРАЖНЕНИЕ

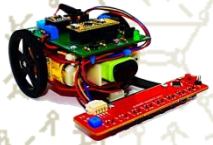


Задача 1



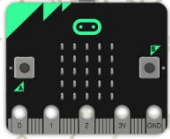
☀ Да се направи проект, който използва две платки microbit, keypad shield и светофарна система;

☀ При натискане на определен бутон от keypad shield да се изпраща код с WiFi комуникация от едната платка към другата и да се визуализира на светофарната система;

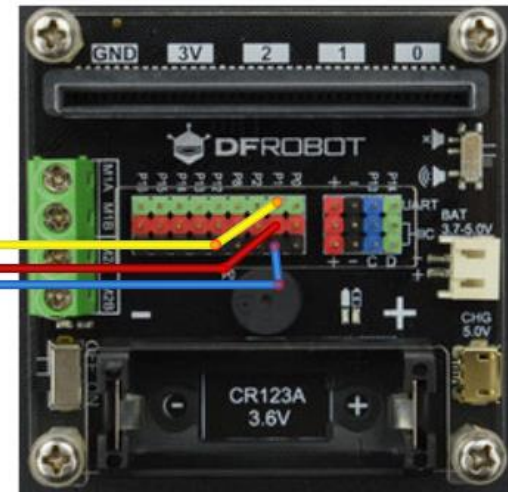
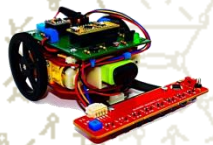




Society of Robotics



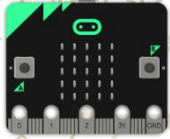
SOCIETY OF ROBOTICS



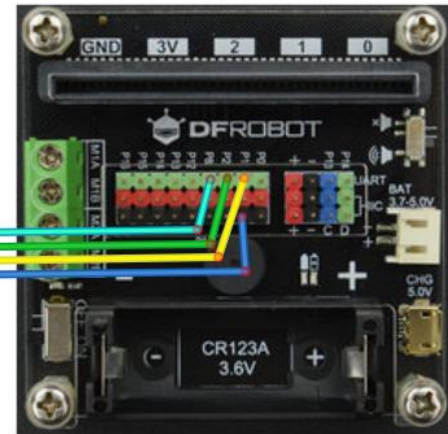
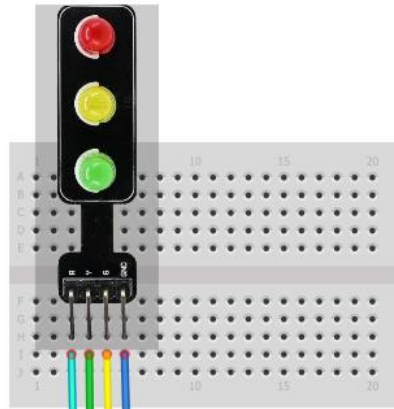
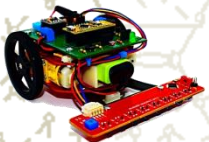
Transmitter



Society of Robotics



SOCIETY OF ROBOTICS



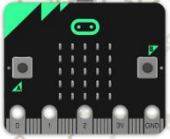
Receiver



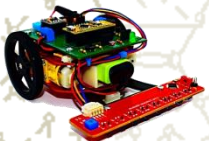
Задача 2

pot_value_map_servo_angle

- Да се направи програма, която използва потенциометър за задаване на ъгъл на завъртане на сервомотор;
- Да се използва създадена функция, която получава стойността на потенциометъра като входен параметър и използва функцията **map()** за преобразуване в ъгъл;

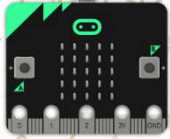


SOCIETY
ROBOTICS

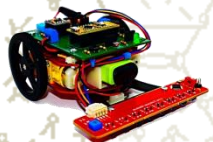




Society of Robotics



SOCIETY
ROBOTIC



define servo_angle val1

set angle to map val1 from[0 , 1023] to[0 , 180]

set pin P13 servo to angle degree

micro:bit starts

forever

set pot_value to read analog pin P1

servo_angle pot_value

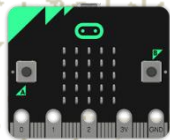
serial output pot_value in string , Wrap

wait 0.04 seconds

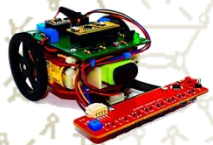


Задача 3

- Да се направи проект, който използва два микроконтролера MicroBit и един сензор за измерване на температура DS18B20;
- Сензорът за измерване на температура DS18B20 е свързан с единия микроконтролер и изпраща по WiFi данните за измерената температура на втория микроконтролер;
- Вторият микроконтролер сравнява получената температура с граничната стойност 30°C и ако е по-голяма от нея издава прекъснат звуков сигнал със зумера;

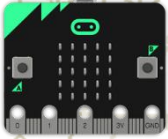


SOCIETY
ROBOTICS

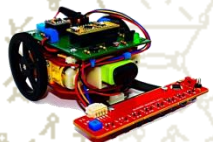




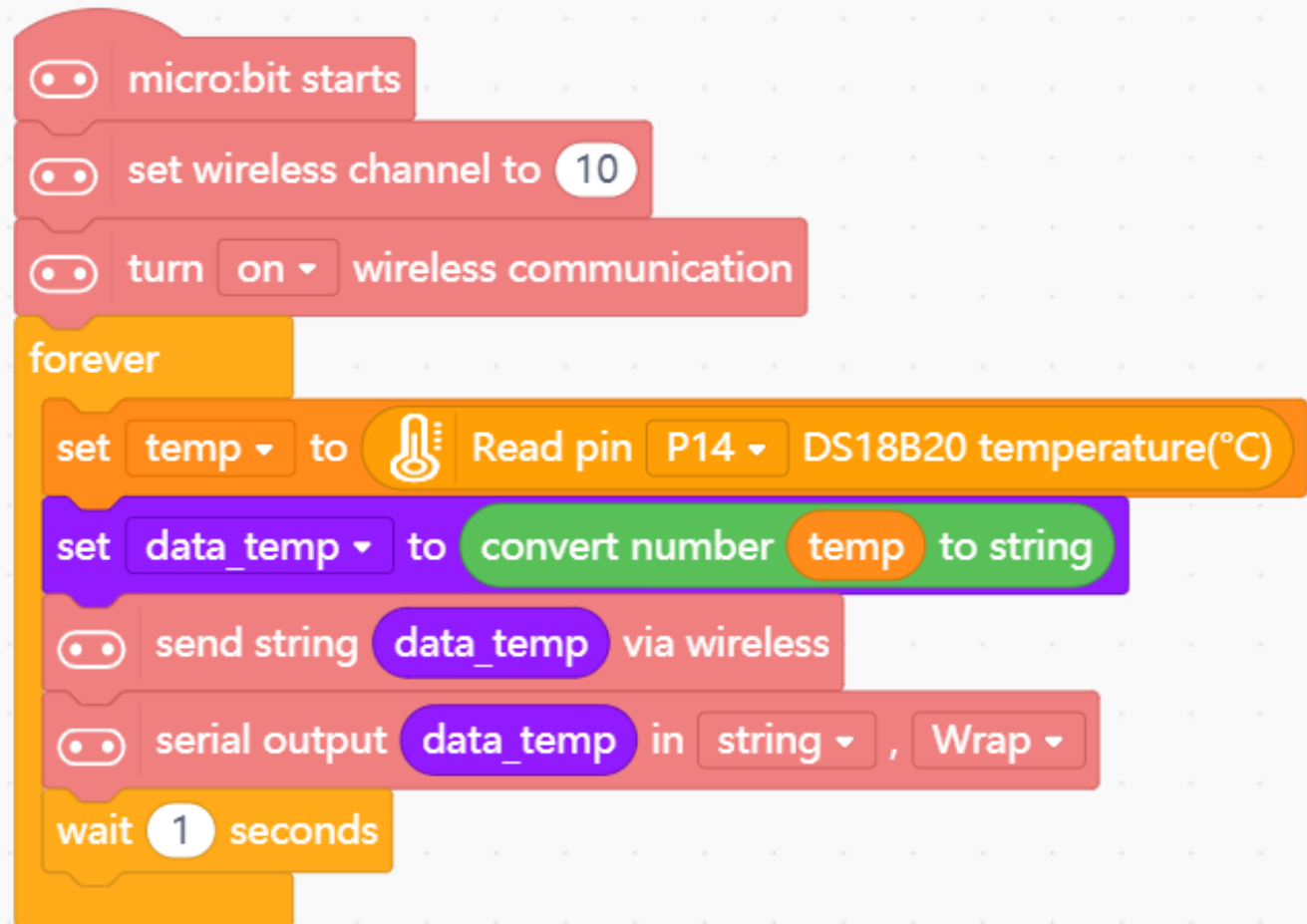
Society of Robotics



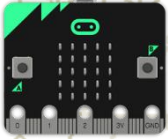
SOCIETY
ROBOTIC



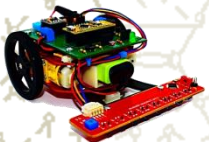
WiFi_DS18B20_buzzer_master



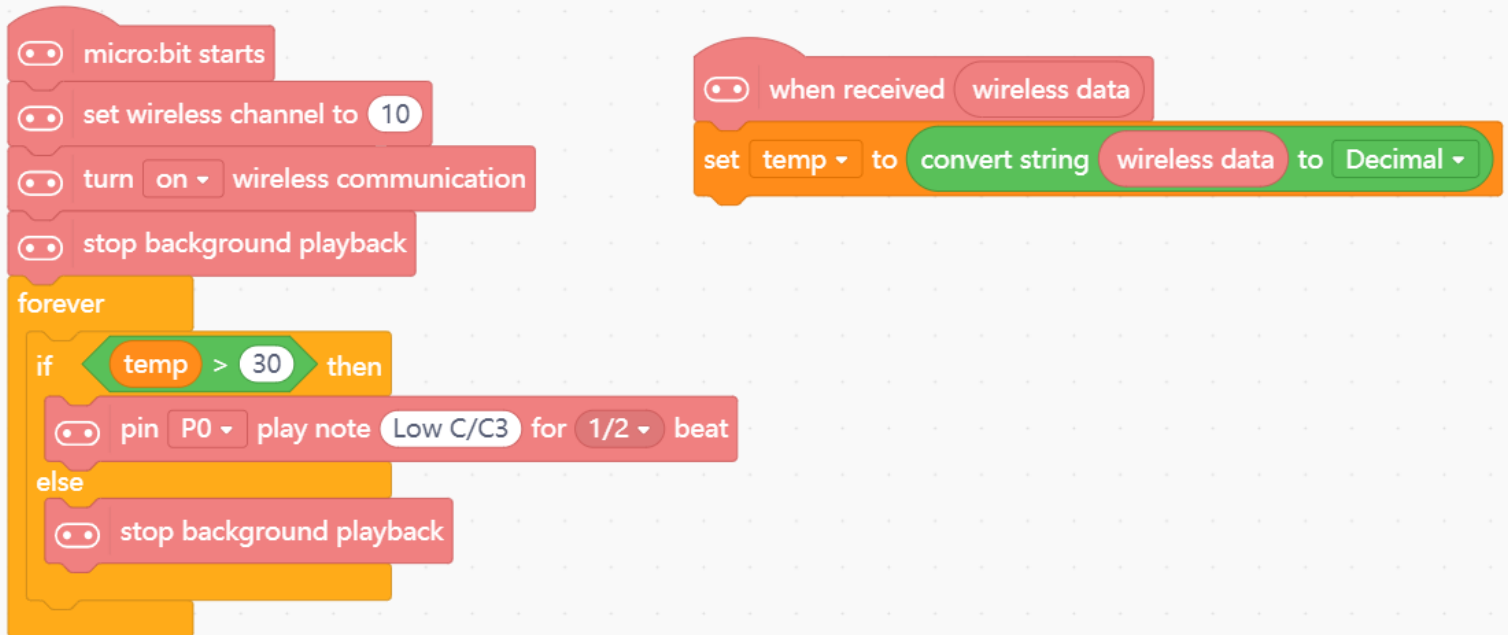
Master



**SOCIETY
ROBOTIC**



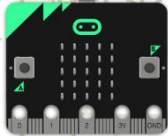
WiFi_DS18B20_buzzer_slave



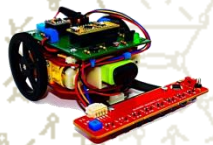
Slave



Задача 4

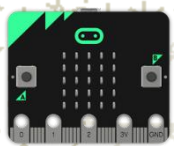


- ☛ Да се направи проект за управление на светофарни уредби за кръстовище, който използва два микроконтролера MicroBit и два светофара;
- ☛ Единият от микроконтролерите да бъде главен, а другият да бъде подчинен;
- ☛ Да се използва показаната схема на взаимодействие на светофарите;

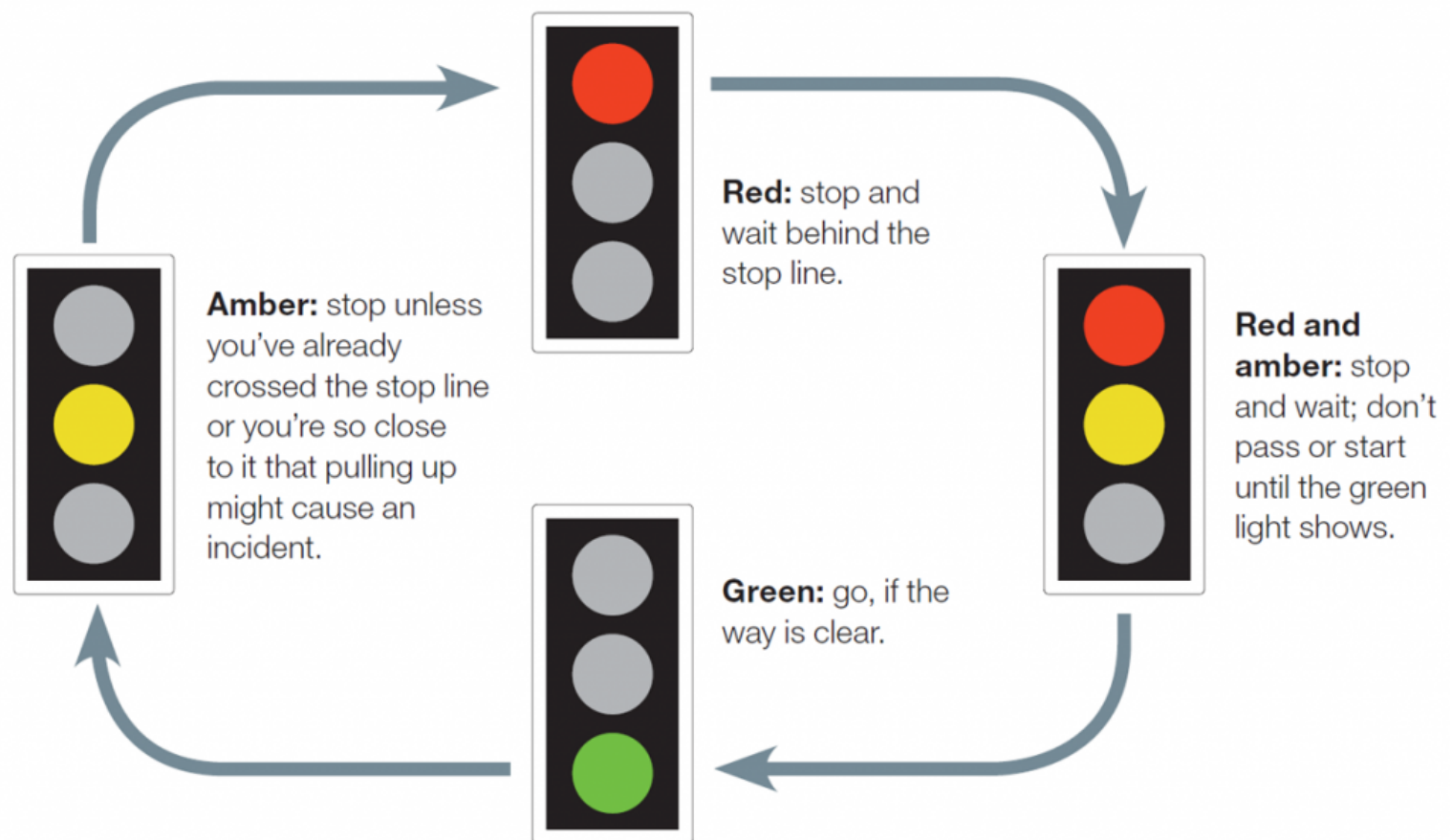
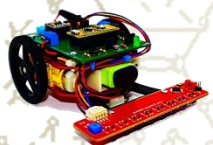




Abir



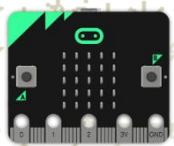
SOCIETY
ROBOTIC



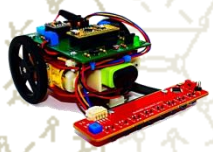


Abirah

Traffic_lights_master



SOCIETY
ROBOTIC



Главное устройство

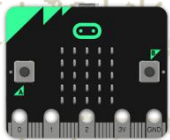
```
micro:bit starts
set wireless channel to 2
turn on wireless communication

forever
  display pattern R
  send string "RED" via wireless
  digital pin P0 output LOW
  digital pin P1 output LOW
  digital pin P2 output HIGH
  wait 10 seconds
  digital pin P0 output LOW
  digital pin P1 output HIGH
  digital pin P2 output HIGH
  display pattern T
  send string "RED_YELLOW" via wireless
  wait 4 seconds
  digital pin P0 output HIGH
  digital pin P1 output LOW
  digital pin P2 output LOW
  display pattern C
  send string "GREEN" via wireless
  wait 10 seconds
  digital pin P0 output LOW
  digital pin P1 output HIGH
  digital pin P2 output LOW
  display pattern Y
  send string "YELLOW" via wireless
  wait 4 seconds
```



Abir

Traffic_lights_slave



SOCIETY
ROBOTIC

Подчинено
устройство

```
micro:bit starts
set wireless channel to 2
turn on wireless communication

when received wireless data
set light to wireless data

forever
  if light = RED then
    digital pin P0 output HIGH
    digital pin P1 output LOW
    digital pin P2 output LOW
    display pattern [diagonal line of dots]
    wait 10 seconds
  if light = RED_YELLOW then
    digital pin P0 output LOW
    digital pin P1 output HIGH
    digital pin P2 output HIGH
    display pattern [diagonal line of dots]
    wait 4 seconds
  if light = GREEN then
    digital pin P0 output LOW
    digital pin P1 output LOW
    digital pin P2 output HIGH
    display pattern [diagonal line of dots]
    wait 10 seconds
  if light = YELLOW then
    digital pin P0 output LOW
    digital pin P1 output HIGH
    digital pin P2 output LOW
    display pattern [diagonal line of dots]
    wait 4 seconds
```