



MINDSTORMS[®] EV3



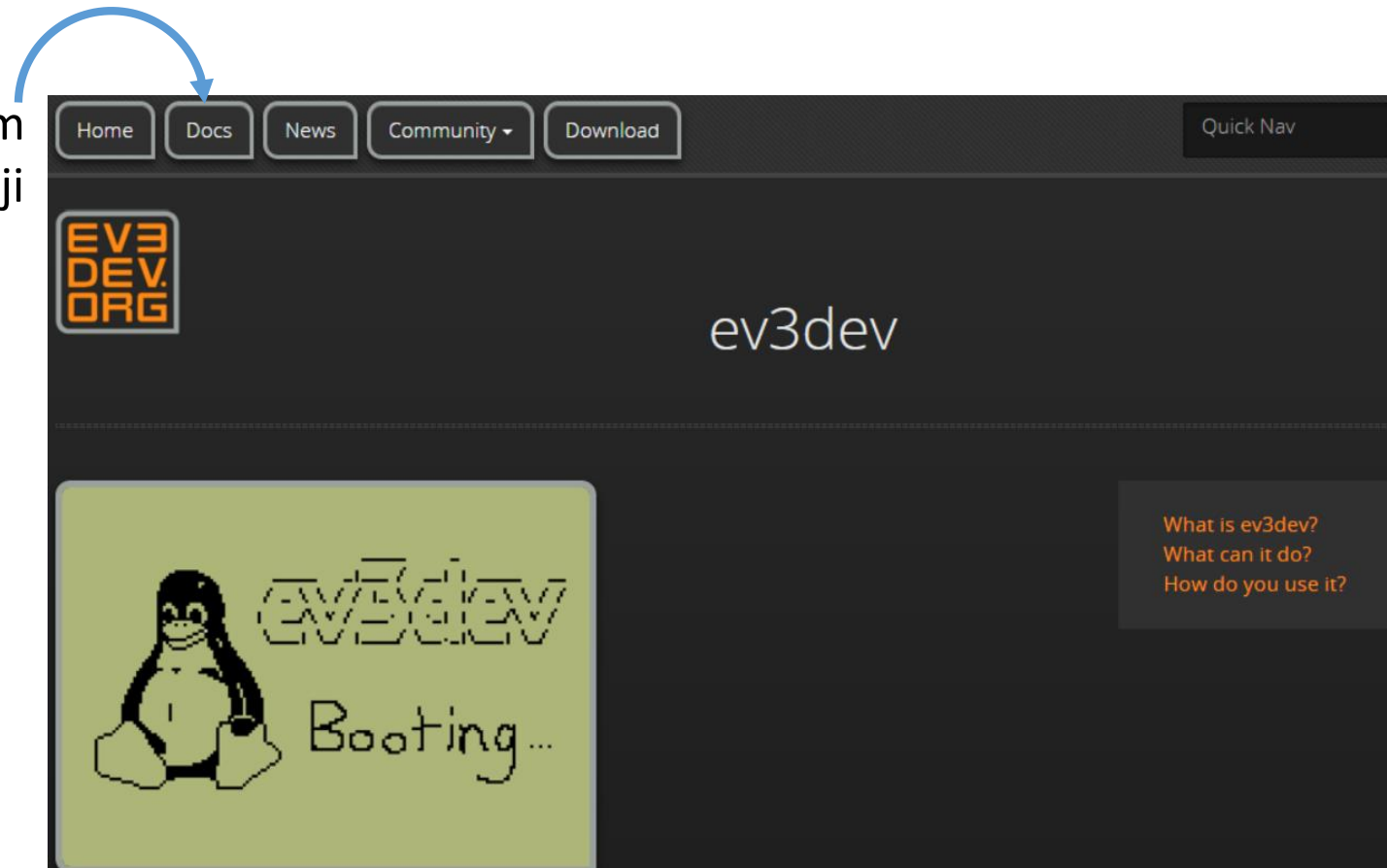
+ PYTHON

JURE ŽABKAR
UROŠ LETRIČ
FRI

EV3DEV

Namestitev okolja ev3dev (<http://www.ev3dev.org/>)

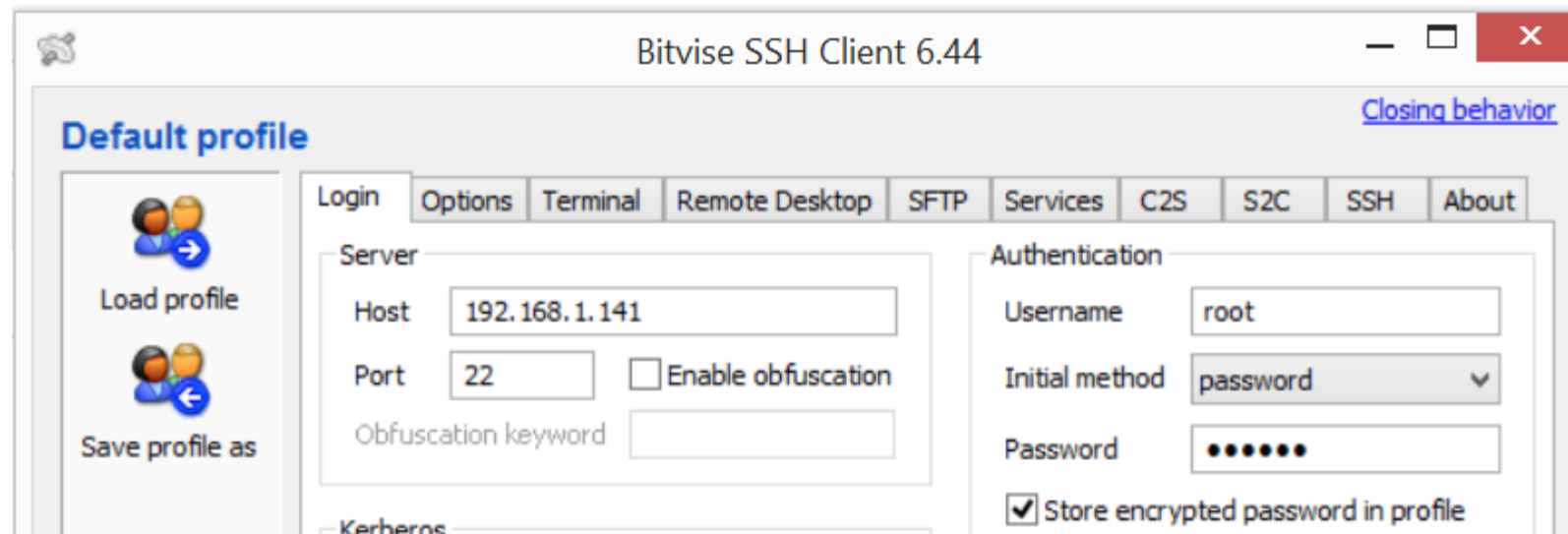
Sledi navodilom
v dokumentaciji



BITWISE SSH CLIENT

Namestitev programa za vzpostavitev SSH povezave z robotom:

<https://www.bitwise.com/ssh-client-download>



EV3DEV-LANG-PYTHON

Namestitev knjižnice za programski jezik Python:

```
root@ev3dev:~# apt-get update
```

```
root@ev3dev:~# apt-get install python-setuptools python-pil
```

```
(ali: root@ev3dev:~# python3 get-pip.py (https://bootstrap.pypa.io/get-pip.py))
```

```
(root@ev3dev:~# python3 -m pip install python-ev3dev)
```

```
root@ev3dev:~# easy_install -U python-ev3dev
```

Dokumentacija:

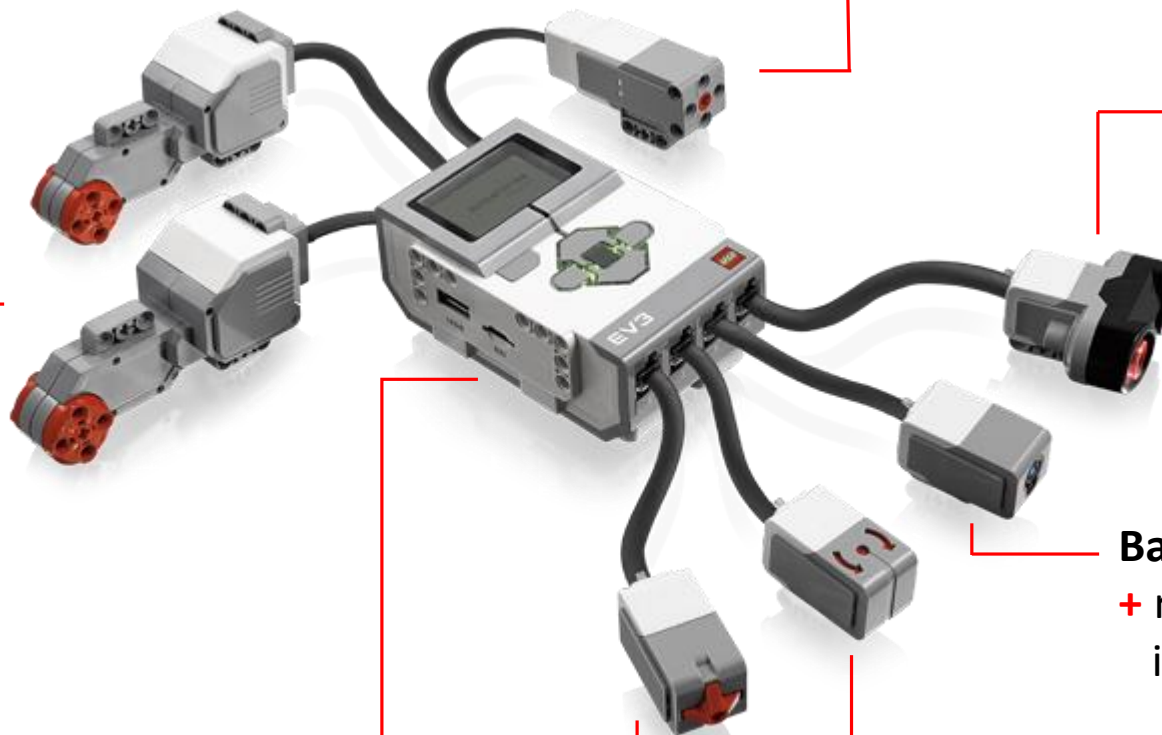
<http://python-ev3dev.readthedocs.org/en/latest/?badge=latest#>

Večja motorja

+ omogočata programiranje natančnih robotskih gibov

Srednji motor

+ manjši, šibkejši, odzivnejši motor



Ultrazvočni senzor

+ uporablja odbit zvok za merjenje razdalje od sensorja do objekta pred njim

Barvni senzor

+ razpozna 7 različnih barv in meri jakost svetlobe

Žiroskop

+ meri kot in kotno hitrost

EV3 kocka

+ osrednji del robota, procesor, spomin, napajanje

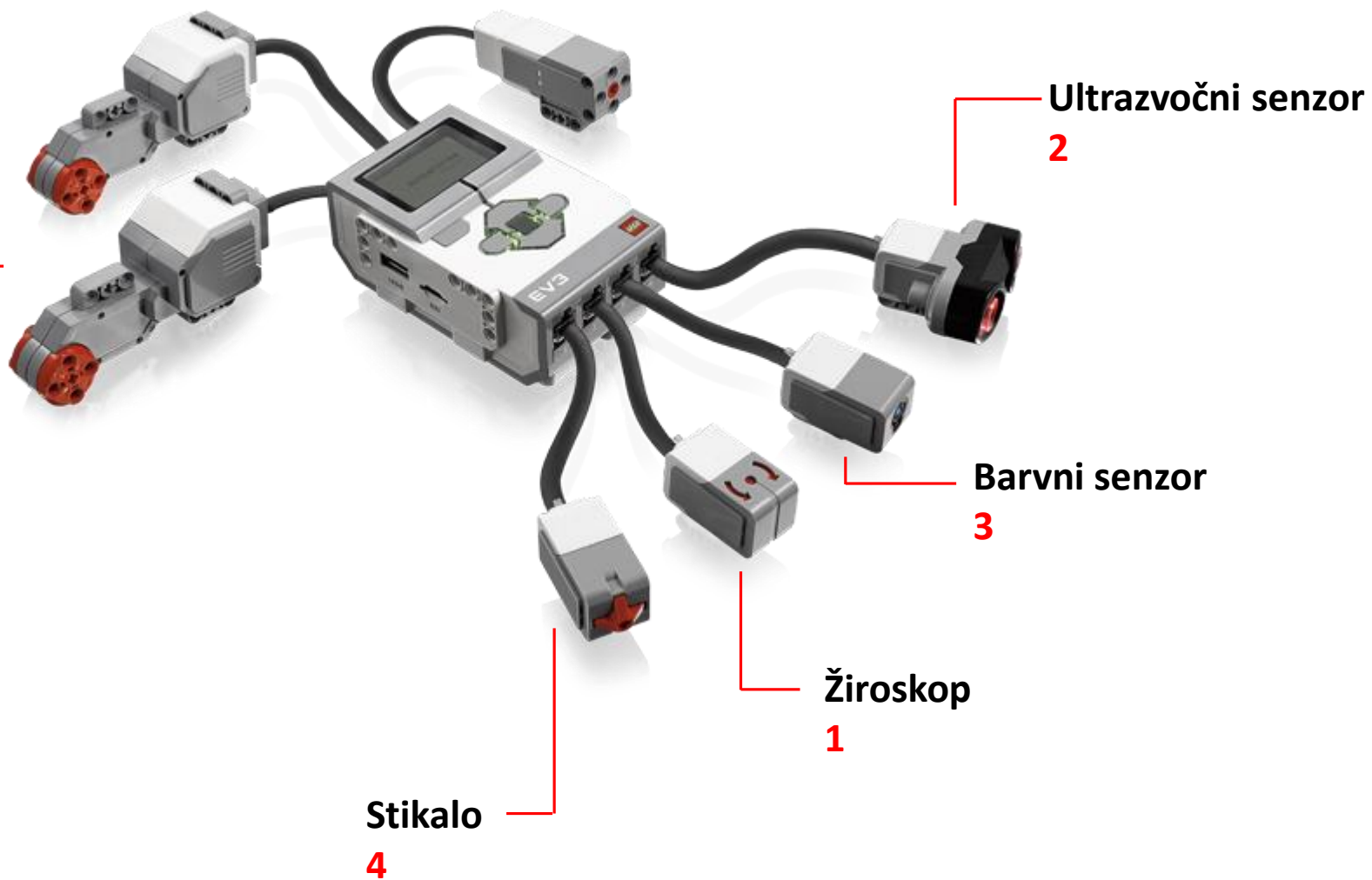
Stikalo

+ prepozna 3 stanja: dotik, pritisk in spust

Večja motorja

B ... levi motor

C ... desni motor



PYTHON

Reševanje nalog na
www.codeq.si

The screenshot displays the CodeQ Python IDE interface. At the top, there is a dark header with the CodeQ logo, navigation tabs for 'Naloge' and 'Python', and a user status indicator 'Prijavljeni ste kot test'. Below the header, a row of control buttons includes 'Plan', 'Namig', 'Testiraj', 'Zaženi', and 'Ustavi'. The main workspace is divided into three sections: 'NAVODILA' (instructions) on the left, 'PROGRAM' (code editor) in the center, and 'KONZOLA' (terminal) at the bottom right. The instructions section contains the text: 'Napiši program, ki mu uporabnik vpiše temperaturo v Fahrenheitovih stopinjah, program pa jo izpiše v Celzijevih. Med temperaturama pretvarjamo po formuli $C = 5/9 (F - 32)$.' The code editor shows two empty lines. The terminal window displays the prompt 'CodeQ Python terminal proxy' and '>>>'. The bottom status bar shows 'NAMIGI' on the left and 'KONZOLA' on the right.

codeq Naloge Python Prijavljeni ste kot test

Plan Namig Testiraj Zaženi Ustavi

Pretvarjanje iz Fahrenheitov v Celzije

Napiši program, ki mu uporabnik vpiše temperaturo v Fahrenheitovih stopinjah, program pa jo izpiše v Celzijevih. Med temperaturama pretvarjamo po formuli $C = 5/9 (F - 32)$.

NAVODILA line 1, column 1 PROGRAM

```
CodeQ Python terminal proxy
>>>
```

NAMIGI KONZOLA

PYTHON ZA EV3

Uporabljali bomo knjižnico
Mindstorms-widgets.

Dostopno na učilnici:

<https://ucilnica.fri.uni-lj.si/legopy>

```
40 # Mindstorms uses the following coefficients to compute speed_sp
41 # based on a power input in range 0-100 i.e. speed_sp = x * power
42 # medium motor: x = 16.4
43 # large motor : x = 10.3
44 # e.g. power of 10 sets medium motor speed_sp to 160 (deg/sec)
45
46 MEDIUM_MOTOR_SPEED_COEFFICIENT = 16.4
47 LARGE_MOTOR_SPEED_COEFFICIENT  = 10.3
48
49 class mindstorms_widgets:
50     def __init__(self):
51         self.motor = {
52             'medium':None,
53             'left'  :None,
54             'right' :None
55         }
56
57     def set_motor_attribs( self, size, motors, powers,
58                           invert, regulated, brake_at_end ):
59         # For regulated speed, speed_sp is tacho counts per
60         # second which for lego EV3 motors is also degrees per second.
61         # To provide a mindstorms 0-100 power value for regulated
62         # speed we need to convert it.
63         speed_x = LARGE_MOTOR_SPEED_COEFFICIENT
64         if size == 'medium':
65             speed_x = MEDIUM_MOTOR_SPEED_COEFFICIENT
66
```

EVE KOCKA

Processor

+ ARM 9 (Linux OS)

Zaslon

+ 178 x 128 točk

Spomin

+ 16 MB Flash in 64 MB of RAM

Uporabniški vmesnik

+ 6 osvetljenih, trobarvnih tipk



EV3 KOCKA

Čitalec SD kartic

+ 32 GB dodatnega spomina

USB 2.0

+ omogoča zaporedno povezavo z drugimi kockami in priključek na Wi-Fi USB vtič

Napajanje

+ 6 AA baterij ali 2050 mAh Li-ion
EV3 polnilna baterija



EVE KOCKA

Zvočnik

- + predvajanje vgrajenih zvočnih efektov

Vhodi

- + vhodi 1, 2, 3 in 4 namenjeni priklopu senzorjev

Izhodi

- + izhodi A, B, C in D namenjeni priklopu motorjev



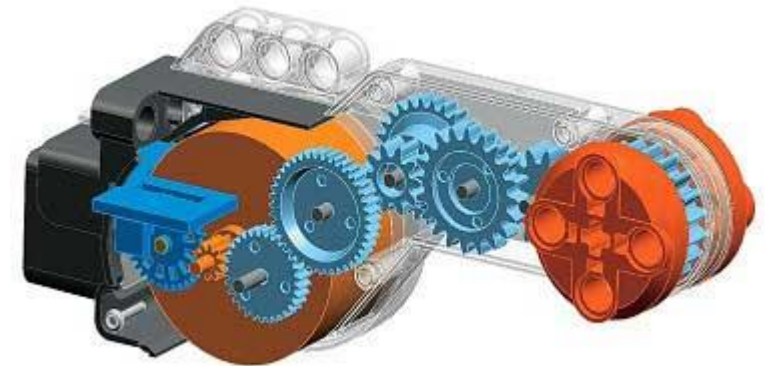
MOTORJI



- + vgrajen optični rotacijski enkoder
- + **hitrejši, šibkejši, odzivnejši motor**
hitrost 240-250 rpm
navor med tekom 8 Ncm in v mirovanju 12 Ncm



- + vgrajen optični rotacijski enkoder
- + **počasnejši, a močnejši motor**
hitrost 160-170 rpm
navor med tekom 20 Ncm in v mirovanju 40 Ncm



MINDSTORMS_WIDGETS

```
from mindstorms_widgets import mindstorms_widgets
```


ime datoteke
s končnico .py


ime razreda v datoteki
mindstorms_widgets.py

POVEZAVA Z ROBOTOM

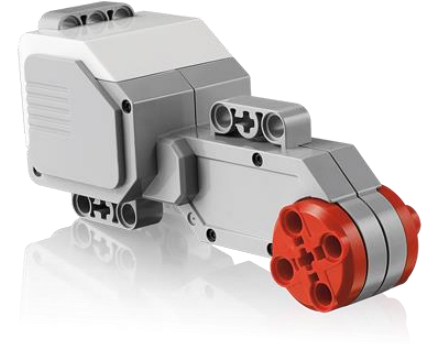
```
from mindstorms_widgets import mindstorms_widgets
```

```
# naredimo objekt mindstorms_widgets; klic metode __init__  
robot = mindstorms_widgets()
```

```
# connect_motor je metoda razreda mindstorms_widgets  
# robotu priključimo levi motor, ki mora biti na vtičnici B  
robot.connect_motor( 'left' )
```

```
# robotu priključimo desni motor, ki mora biti na vtičnici C  
robot.connect_motor( 'right' )
```

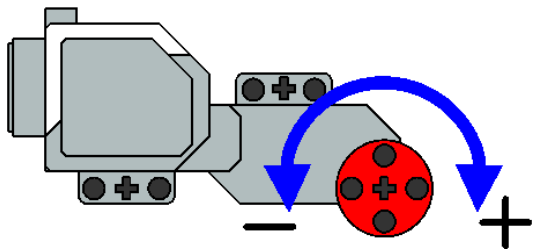
MOTOR



`connect_motor(motor, port=None)`

motor ... 'medium', 'left', 'right'

port ... 'A', 'B', 'C', 'D'



MOVE STEERING

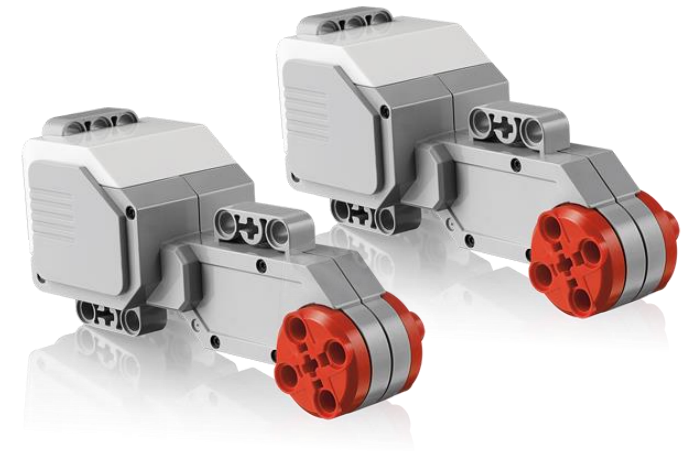
Move Steering

- + Nadzira in **regulira** dva motorja hkrati
- + Nadzor prek dodeljevanja moči enemu/drugemu motorju

```
move_steering( mode, direction=0, power=50,  
               seconds=0, degrees=0, rotations=0,  
               invert=False, regulated=True,  
               brake_at_end=True )
```

mode:	direction [-100, 100]:
'on'	-100 = pivot left (right = power, left = -power)
'off'	-50 = sharp left (right = power, left = 0)
'on_for_seconds'	-25 = turn left (right = power, left = 1/2 power)
'on_for_degrees'	0 = straight
'on_for_rotations'	25 = turn right
	50 = sharp right
	100 = pivot right

power (-100, 100)
invert (True, False)
regulated (True, False)
brake_at_end (True, False)



MOVE TANK

Move Tank

- + Nadzira oba motorja hkrati, a dovoljuje ločeno podajanje moči za vsak motor
- + Dovoljuje tudi obrat na mestu (en motor se vrti naprej, drugi nazaj)

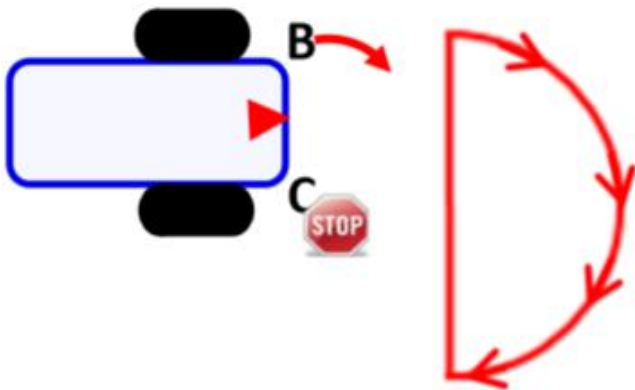
```
move_tank( mode, lr_power=None,  
           seconds=0, degrees=0, rotations=0,  
           invert=False, regulated=True,  
           brake_at_end=True )
```

```
mode:  
'on'  
'off'  
'on_for_seconds'  
'on_for_degrees'  
'on_for_rotations'
```

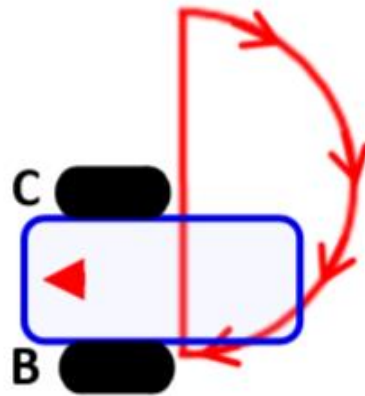
```
lr_power ([L, R], kjer L, R v [-100, 100])  
invert (True, False)  
regulated (True, False)  
brake_at_end (True, False)
```



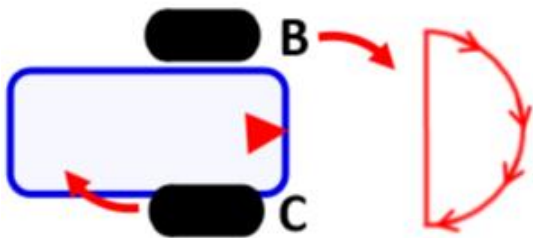
Start Position



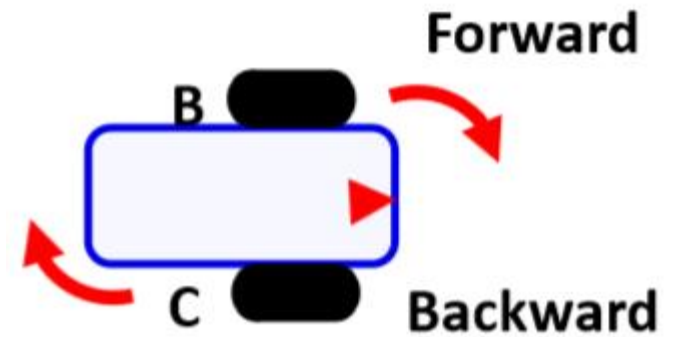
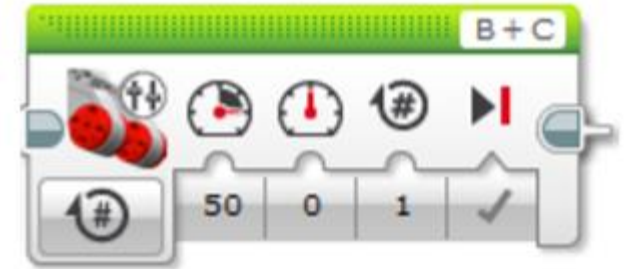
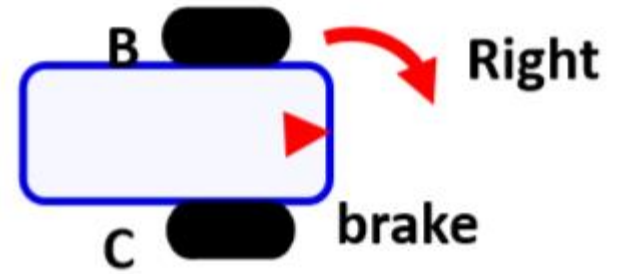
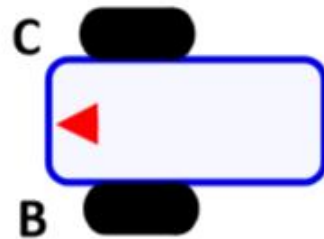
End Position



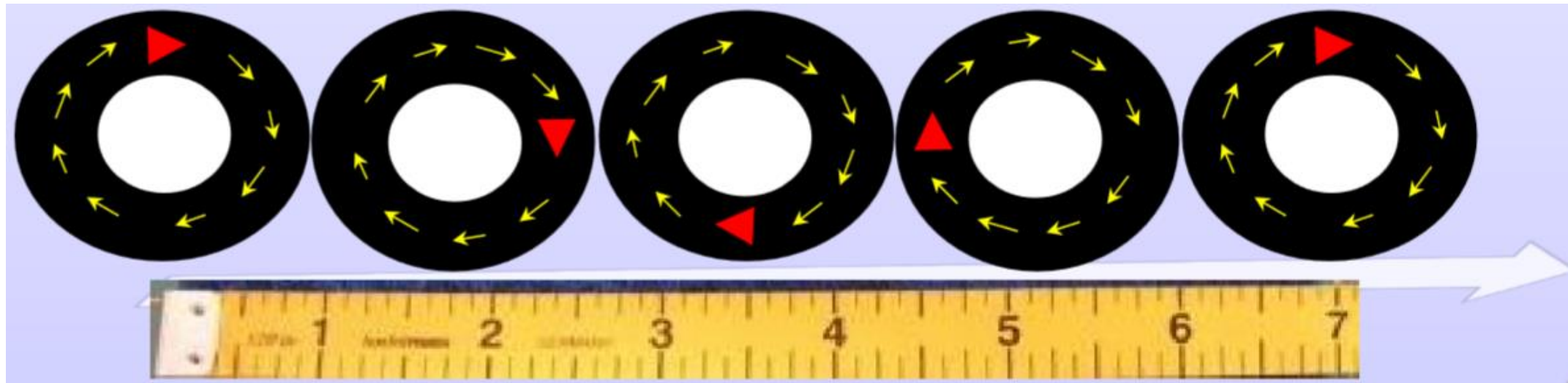
Start Position



End Position



SE

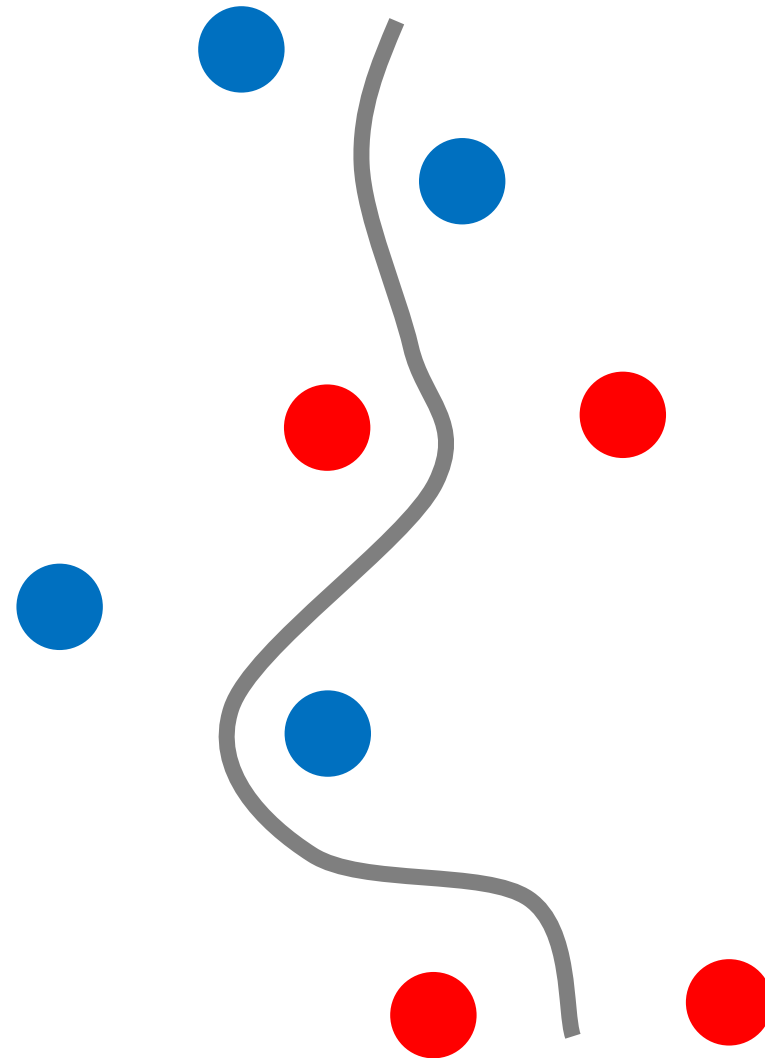


obseg = $\pi \cdot \text{premer} = 3,14 \cdot 1.6 = 5 \text{ inch}$

pot za 1 inch = $360^\circ / 5 = 72^\circ$

SLALOM

Napiši program, s katerim bo robot odpeljal slalom po postavljeni progi.



SENZORJI




Ultrazvočni

- 2 načina delovanja:
- + merjenje razdalje
cm (ali inch)
3 – 250 cm (+/- 1 cm)
 - + komunikacija z drugim
UZ senzorjem



Barvni

- 3 načini delovanja:
- + prepoznavanje barv

 - + jakost odbite svetlobe
posveti z rdečo svetlobo
in meri jakost odboja
(temno 0 – svetlo 100)
 - + jakost ambientne svetlobe
jakost svetlobe v okolju
(temno 0 – svetlo 100)



1-osni žiroskop

- + merjenje kota in
kotne hitrosti
(največ 440°/s)



Stikalo oz. senzor dotika

- + prepozna pritisk na
gumb, spuščen gumb
ter pritisk in spust,
neposredno eno za drugim

SENZORJI

```
from mindstorms_widgets import mindstorms_widgets
robot = mindstorms_widgets()
robot.connect_motor( 'left' )
robot.connect_motor( 'right' )
```

```
robot.connect_sensor( 'gyro' )
robot.connect_sensor( 'touch' )
robot.connect_sensor( 'infrared' )
robot.connect_sensor( 'color' )
robot.connect_sensor( 'ultrasonic' )
```

BARVNI SENZOR



`color_sensor_measure(mode)`

mode:

'color'

'reflected_light_intensity'

'ambient_light_intensity'

metoda glede na *mode* vrne številsko vrednost.

IZPISI BARVO

Robot naj se pelje čez barvne črte; pri tem naj na zaslon izpisuje barvo, ki jo trenutno zaznava.



PRESTEJ ČRTE

Robot naj se pelje čez barvne črte in jih prešteje.



ZIROSKOP



reset_gyro()

gyro_set_mode(mode)

mode:

'angle' stopinje v (-32768, 32767)

'rate' kotna hitrost (deg/s) v (-440, 440)

gyro_sensor_measure()

vrne številsko vrednost glede na *mode*

Pomembno

+ priklop senzorja (preden vtaknemo vanj kabel, mora mirovati)

obrati za 90°

Robot naj se obrne na mestu za 90°;
pri tem naj si pomaga z žiroskopom.

Pomembno

- + moč motorjev (nižja moč, bolj natančen obrat; ne prešibko, ker potem motorji ne delajo dobro)

VRNI ME NAZAJ

- Napišite program, ki bo robota vedno vrnil v začetno smer
dovoljena napaka je 1 kotna stopinja.
- Uporabite žiroskop.
- Robot naj na zaslonu izpisuje trenutni kot.
- Ko stisnete senzor dotika, naj se žiroskop postavi na nič.
- Večji kot je kot, hitreje naj se robot giblje proti ničelni legi
najnižja hitrost je 10
najvišja hitrost je 100
- Uporabni funkciji
abs – vrne absolutno vrednost
sgn – vrne predznak

ULTRAZVOČNI SENZOR



`ultrasonic_sensor_measure(mode)`

trenutno podpira samo mode = 'distance-cm'

vrne razdaljo v cm (0 - 255)

PROTI STENI

- + Robot naj se vozi naravnost z 80% močjo
- + Ko se steni približa na manj kot 50 cm, naj se začne gibati počasneje, z 20% močjo.
- + Ko se steni približa na manj kot 10 cm, naj se ustavi.



STENI



Robot naj se 20 sekund vozi ob steni, na razdalji 20 cm.

Na koncu naj izračuna, kako dobro mu je šlo – na kakšni razdalji od stene je vozil v povprečju.



`touch_sensor_measure()`

vrne 0 (sproščen) ali 1 (pritisnjen)

SLEDENJE ČRTI

Napišite program za sledenje črti

+ Prvi pristop

če je robot na beli podlagi, naj zavije levo
če je robot na temni podlagi, naj zavije desno

+ Drugi pristop

bolj kot je svetlo, bolj naj robot zavije levo
bolj kot je temno, bolj naj robot zavije desno

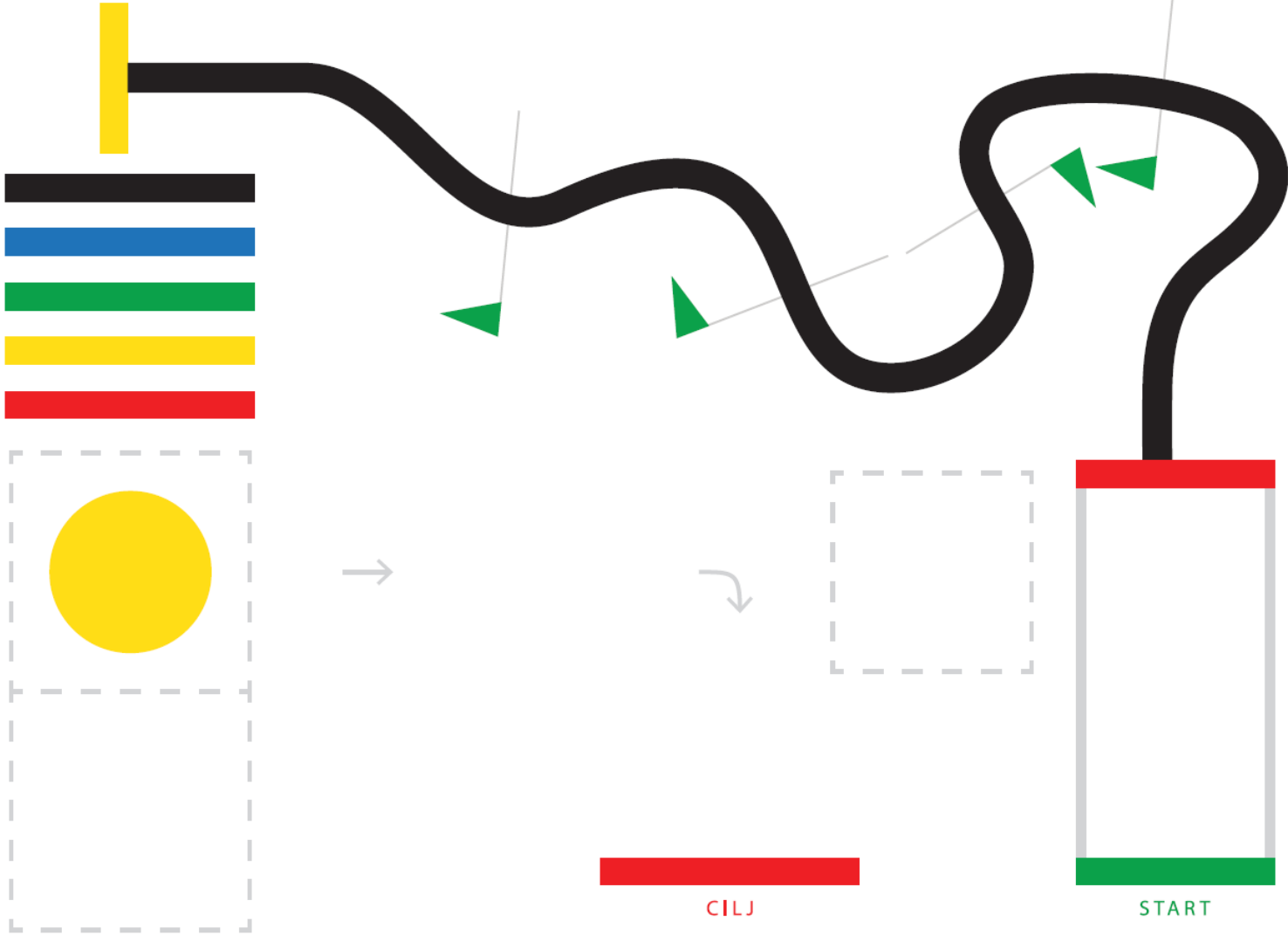
+ Tretji pristop

regulator PID

proporcionalni člen (odziv na trenutek)
integralni člen (odziv na preteklo obnašanje)
diferencialni člen (predvidevanje prihodnosti)

NALOGA 7

Vožnja po poligonu



LABIRINT

Robot naj najde pot iz labirinta.

