



Robotika se stavebnicemi VEX

PŘÍPRAVA NA VÝUKU
PRACOVNÍ LISTY

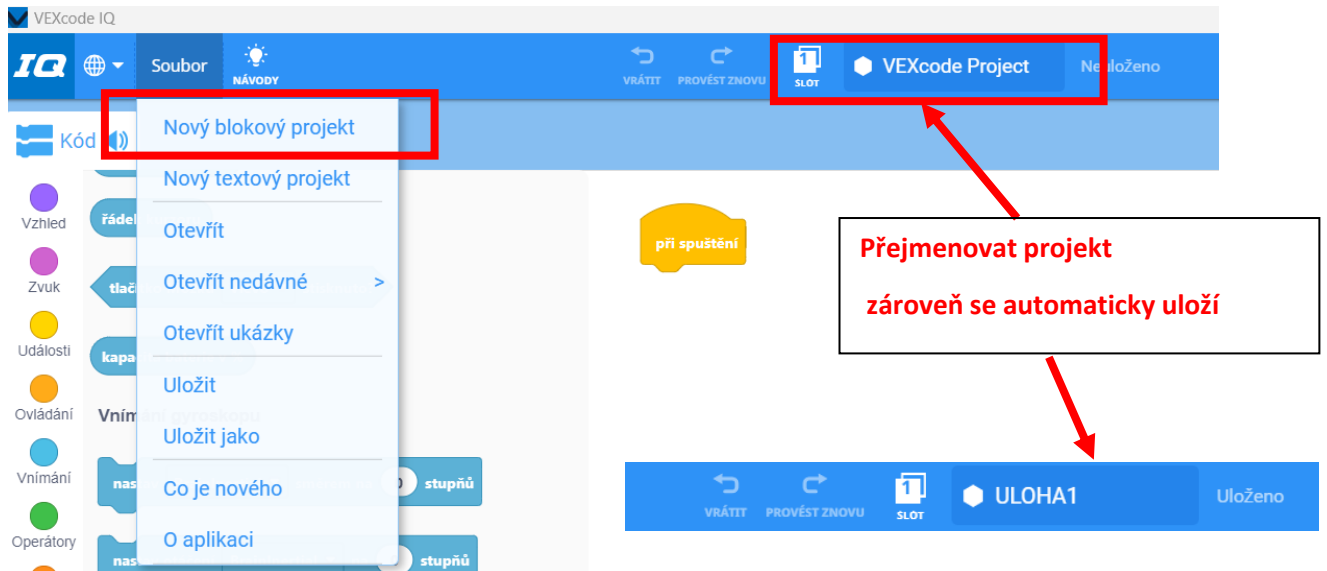
ÚLOHA 1	2
ÚLOHA 2	8
ÚLOHA 3	11
ÚLOHA 4	14
ÚLOHA 5	17
ÚLOHA 6	19
ÚLOHA 7	21
ÚLOHA 8	23

ÚLOHA 1**Cíl:**

Seznámení s aplikací VEXcode IQ, nahrání a spuštění prvního programu

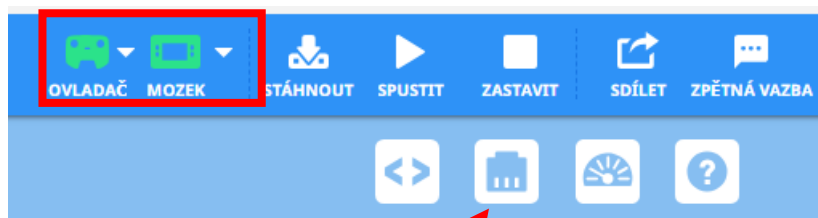
Úkol:

Robot vykoná pohyb

NÁVOD - APLIKACE

1. krok – připojení ovladače a řídicí jednotky
 - ovladač připojit přes usb k PC nebo tabletu
 - řídicí jednotku (mozek) zapnout

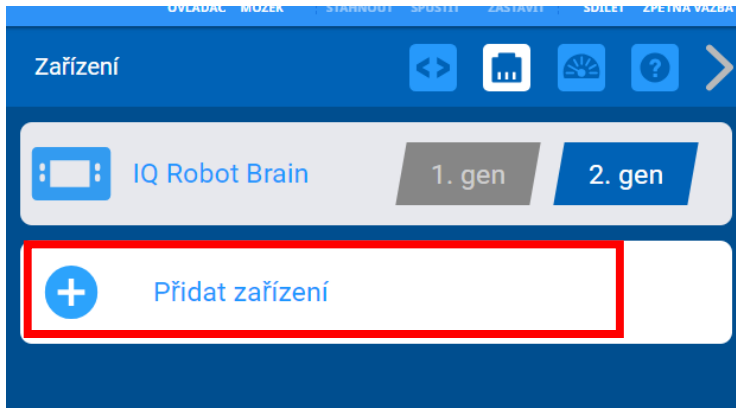
ikony ovladače a mozku v aplikaci změni barvu na zelenou



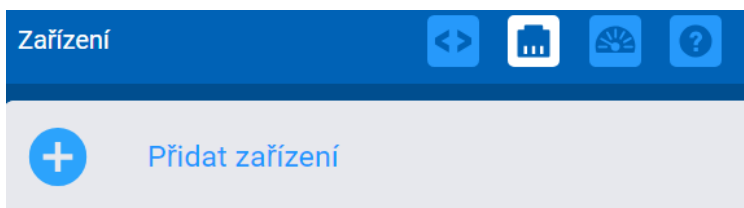
2. krok - nastavení připojených prvků

V základním modelu robota byly zatím použity pouze motory zapojené do portů 1 a 6

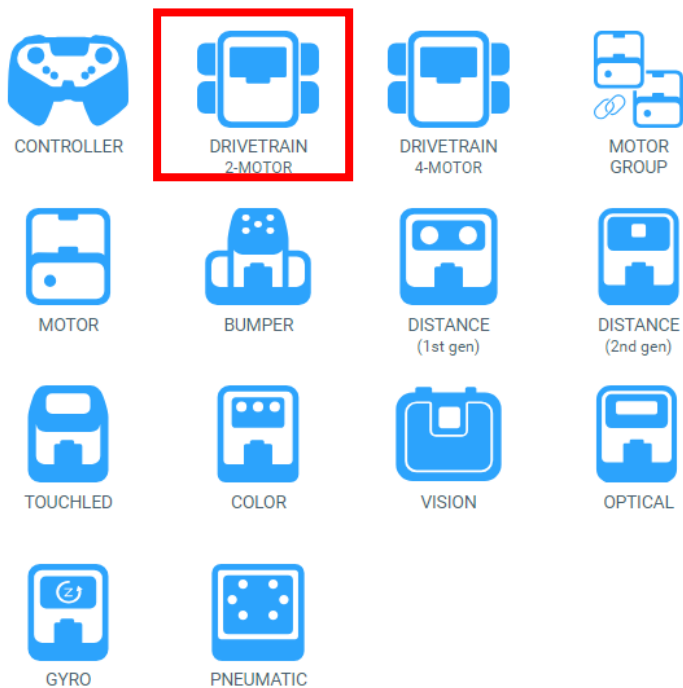
Pozn. Při zapojení prvků do řídicí jednotky nezáleží na tom, jaký port je vybrán, při nastavení portů je třeba se řídit reálnou situací



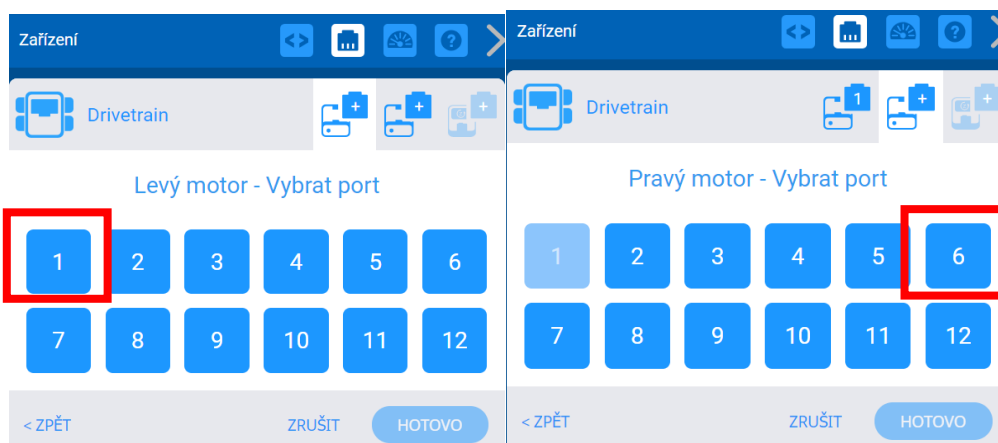
Výběr zařízení je závislý na typu a počtu reálně připojených prvků, v tomto případě pohon dvou kol, dvěma motory



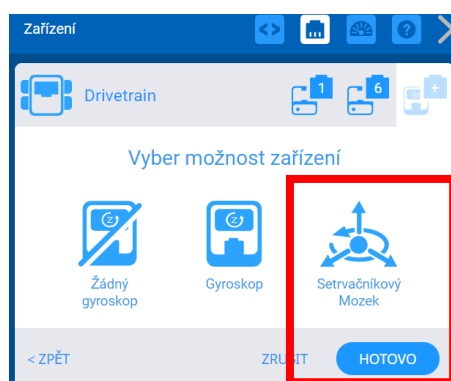
Vybrat zařízení



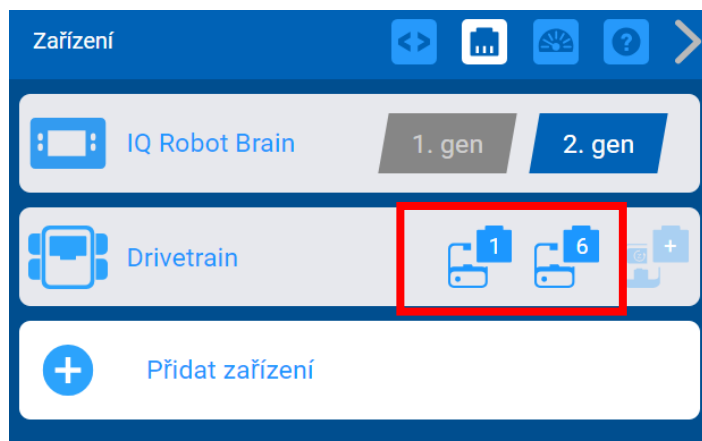
Výběr portů



Protože není připojen gyroskop je třeba vybrat setrvačnickový mozek



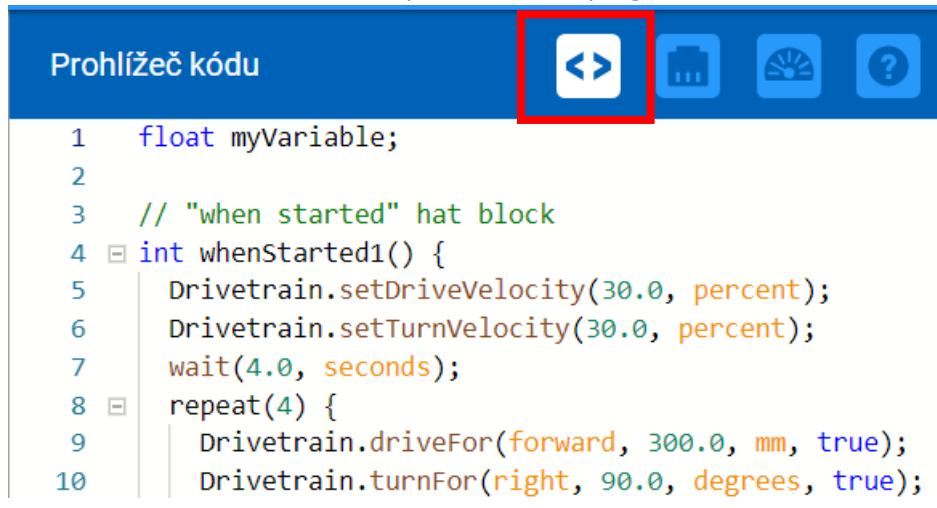
V této chvíli aplikace pracuje se dvěma motory. Levý v portu 1 a pravý v portu 6



Aplikace je připravena

PRVNÍ PROGRAM

Pozn. Při blokovém programování otevřete okno kódu v Pythonu. Naučíte se orientovat v kódu, což do budoucna usnadní psaní vlastních programů

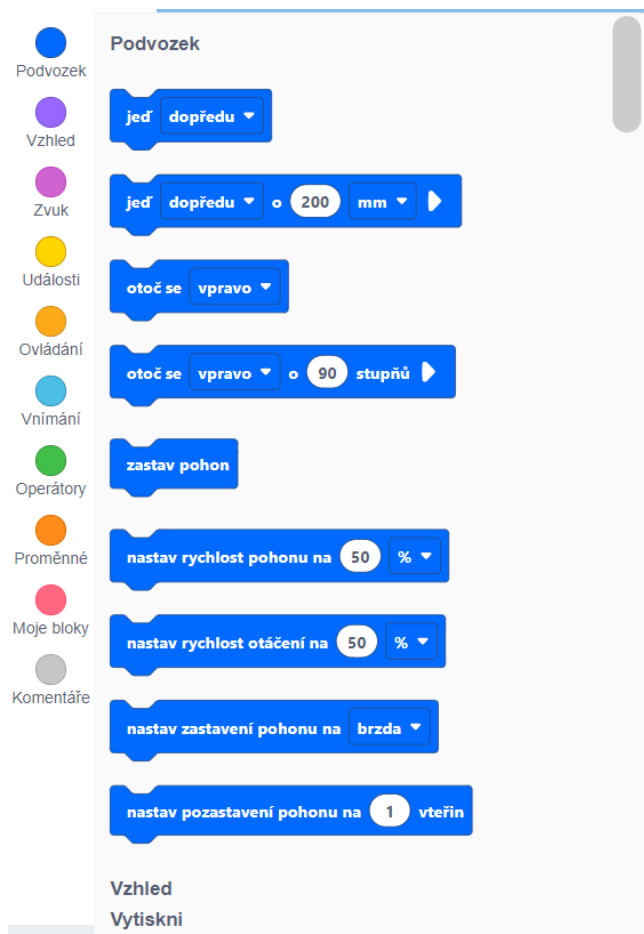


```

1 float myVariable;
2
3 // "when started" hat block
4 int whenStarted1() {
5     Drivetrain.setDriveVelocity(30.0, percent);
6     Drivetrain.setTurnVelocity(30.0, percent);
7     wait(4.0, seconds);
8     repeat(4) {
9         Drivetrain.driveFor(forward, 300.0, mm, true);
10        Drivetrain.turnFor(right, 90.0, degrees, true);

```

Některé bloky v nabídce se mění podle připojených zařízení. Po nastavení připojení prvků (v této úloze jsou připojeny dva motory) se přidají bloky (v této úloze „podvozek“)



Podvozek

- Podvozek
- Vzhled
- Zvuk
- Události
- Ovládání
- Vnímání
- Operátory
- Proměnné
- Moje bloky
- Komentáře

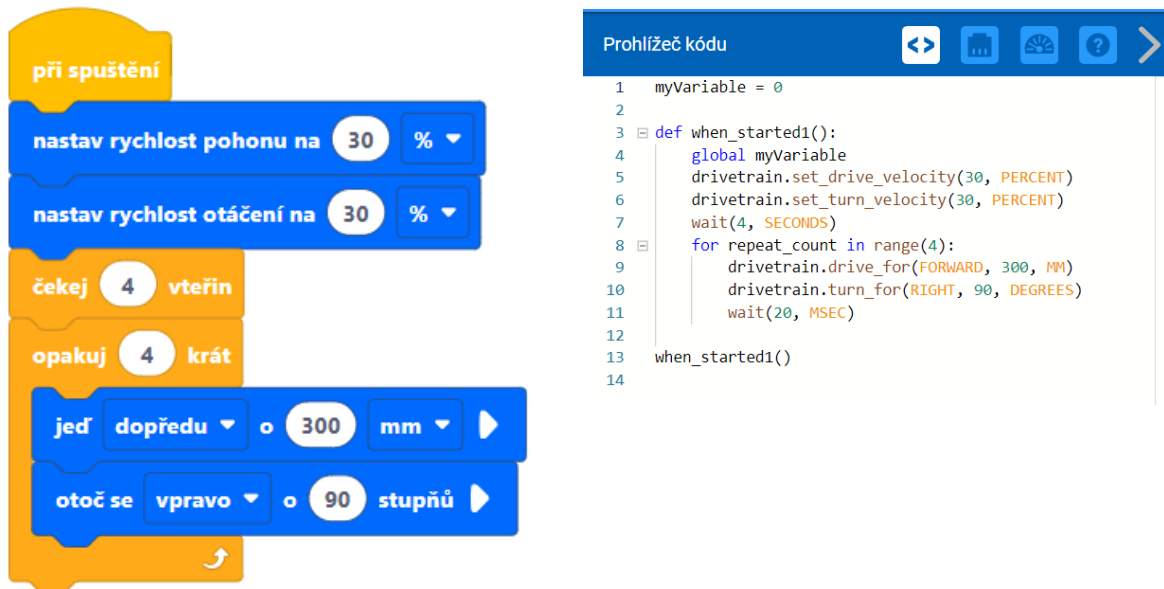
jed' dopředu
 jed' dopředu 200 mm
 otoč se vpravo
 otoč se vpravo 90 stupňů
 zastav pohon
 nastav rychlost pohonu na 50 %
 nastav rychlost otáčení na 50 %
 nastav zastavení pohonu na brzda
 nastav pozastavení pohonu na 1 vteřin

Vzhled
Vytiskni

Práce s bloky

Následující jednoduchý program nastaví rychlost pohonu na 30% výkonu, pojedje po trajektorii čtverce.

Robot pojedje v přímém směru 300mm, následně se otočí o 90°. Tento proces se opakuje 4x (čtverec)



The image shows a Scratch-style block diagram on the left and a Python code editor on the right. The block diagram starts with a 'při spuštění' (when started) block, followed by 'nastav rychlost pohonu na 30 %' and 'nastav rychlost otáčení na 30 %'. Then there is a 'čekej 4 vteřin' (wait 4 seconds) block, followed by an 'opakuji 4 krát' (repeat 4 times) loop containing 'jed' dopředu 300 mm' and 'otoč se vpravo 90 stupňů'.

The Python code in the editor is as follows:

```

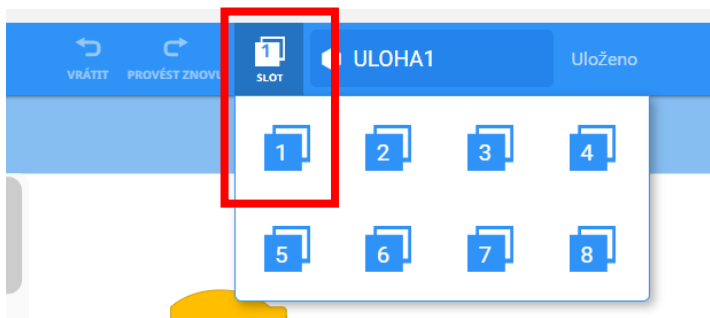
1 myVariable = 0
2
3 def when_started1():
4     global myVariable
5     drivetrain.set_drive_velocity(30, PERCENT)
6     drivetrain.set_turn_velocity(30, PERCENT)
7     wait(4, SECONDS)
8     for repeat_count in range(4):
9         drivetrain.drive_for(FORWARD, 300, MM)
10        drivetrain.turn_for(RIGHT, 90, DEGREES)
11        wait(20, MSEC)
12
13 when_started1()
14

```

Doporučení: Prostudujte kód v Pythonu tak, abyste rozuměli každému řádku kódu

Spuštění programu na robotu

Výběr slotu řídicí jednotky do kterého bude program nahrán a ze kterého bude možné jej na řídicí jednotce spustit.



Řídicí jednotku lze připojit přes USB přímo do PC. Lepší variantou je připojit přes USB ovladač (spárovaný se řídicí jednotkou), to umožní přenos programu do řídicí jednotky bezdrátově (řídicí jednotka musí být zapnuta).

Po výběru ikony „stáhnout“ se program nahraje do řídicí jednotky. Spustit lze ikonou „spustit“ nebo přímo na řídicí jednotce.



A CO DÁL...

- zkuste změnit na robotu porty pro připojení
 - změňte porty v aplikaci a spusťte program znovu

- vytvořte vlastní program pro jízdu
 - vyzkoušejte různé možnosti nastavení výkonu motorů, různé trajektorie jízdy, počet opakování atd.

ÚLOHA 2

Cíl:

Připojení a programování Bumper Switch



Úkol:

Robot se rozjede pozpátku, v okamžiku kdy narazí na překážku rozjede se dopředu, ujede 300mm, zastaví a napíše text „ narazil jsem na překážku“

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

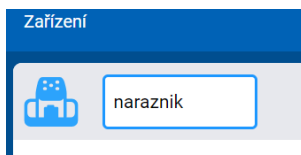
Bumper Switch – port 8

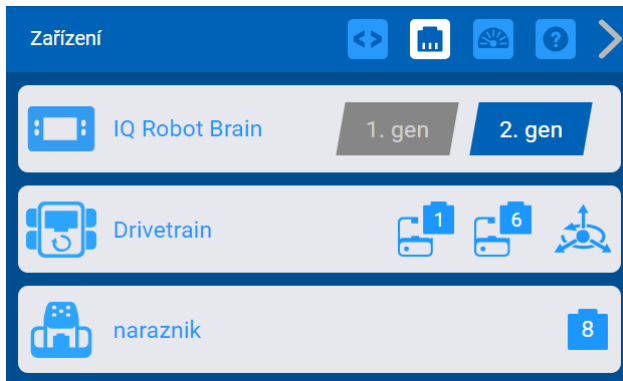
NÁVOD – APLIKACE

Připojení – port 8



Pozn. Název Bumepr8 lze změnit. Použije název bez diakritiky





VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(REVERSE)
6      while not naraznik.pressing():
7          wait(20, MSEC)
8      drivetrain.drive_for(FORWARD, 300, MM)
9
10 # Calibrate the Drivetrain Gyro
11 calibrate_drivetrain()
12
13 when_started1()
14

```

A CO DÁL...

- Proč nefunguje následující program?



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(REVERSE)
6      if naraznik.pressing():
7          wait(1, SECONDS)
8          drivetrain.drive_for(FORWARD, 300, MM)
9
10 # Calibrate the Drivetrain Gyro
11 calibrate_drivetrain()
12
13 when_started1()
14

```

- a proč tento funguje?



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(FORWARD)
6
7  def onevent_naraznik_pressed_0():
8      global myVariable
9      wait(1, SECONDS)
10     drivetrain.drive_for(FORWARD, 200, MM)
11
12 # Calibrate the Drivetrain Gyro
13 calibrate_drivetrain()
14
15 # system event handlers
16 naraznik.pressed(onevent_naraznik_pressed_0)
17 # add 15ms delay to make sure events are registered
18 wait(15, MSEC)
19
20 when_started1()

```

- zkuste si další modifikace programu

ÚLOHA 3

Cíl:

Připojení a programování Bumper Switch

**Úkol:**

Robot se rozjede pozpátku, v okamžiku kdy narazí na překážku rozjede se dopředu, ujede 300mm, zastaví a napíše text „ narazil jsem na překážku“

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

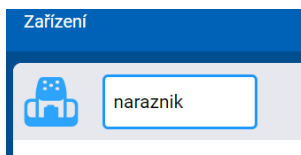
Bumper Switch – port 8

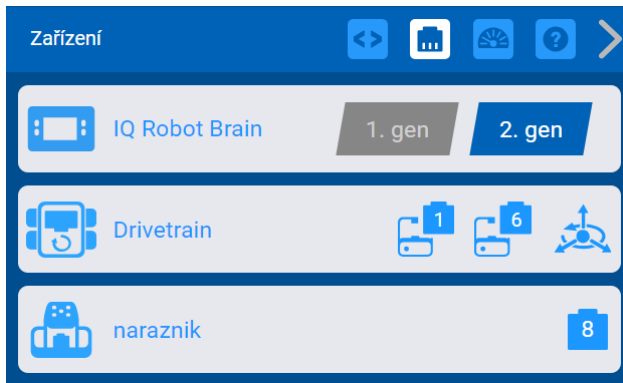
NÁVOD – APLIKACE

Připojení – port 8



Pozn. Název Bumepr8 lze změnit. Použije název bez diakritiky





VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(REVERSE)
6      while not naraznik.pressing():
7          wait(20, MSEC)
8      drivetrain.drive_for(FORWARD, 300, MM)
9
10 # Calibrate the Drivetrain Gyro
11 calibrate_drivetrain()
12
13 when_started1()
14

```

A CO DÁL...

- Proč nefunguje následující program?

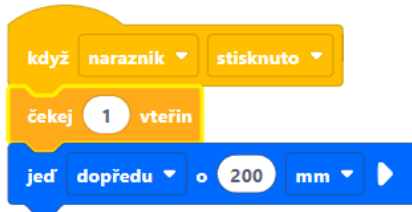


```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(REVERSE)
6      if naraznik.pressing():
7          wait(1, SECONDS)
8          drivetrain.drive_for(FORWARD, 300, MM)
9
10 # Calibrate the Drivetrain Gyro
11 calibrate_drivetrain()
12
13 when_started1()
14

```

- a proč tento funguje?



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drivetrain.drive(FORWARD)
6
7  def onevent_naraznik_pressed_0():
8      global myVariable
9      wait(1, SECONDS)
10     drivetrain.drive_for(FORWARD, 200, MM)
11
12 # Calibrate the Drivetrain Gyro
13 calibrate_drivetrain()
14
15 # system event handlers
16 naraznik.pressed(onevent_naraznik_pressed_0)
17 # add 15ms delay to make sure events are registered
18 wait(15, MSEC)
19
20 when_started1()

```

- zkuste vytvořit program, kdy robot při detekování překážky (např. 25mm před překážkou) změni směr pohybu o 90°. Po několika minutách jízdy (zvolte), robot zastaví

ÚLOHA 4**Cíl:**

Připojení a programování Distance Sensor

**Úkol:**

Robot se rozjede pozpátku, v okamžiku kdy narazí na překážku rozjede se dopředu. V okamžiku kdy bude 20mm od překážky zastaví a napíše text „ HROZI NARAZ“

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

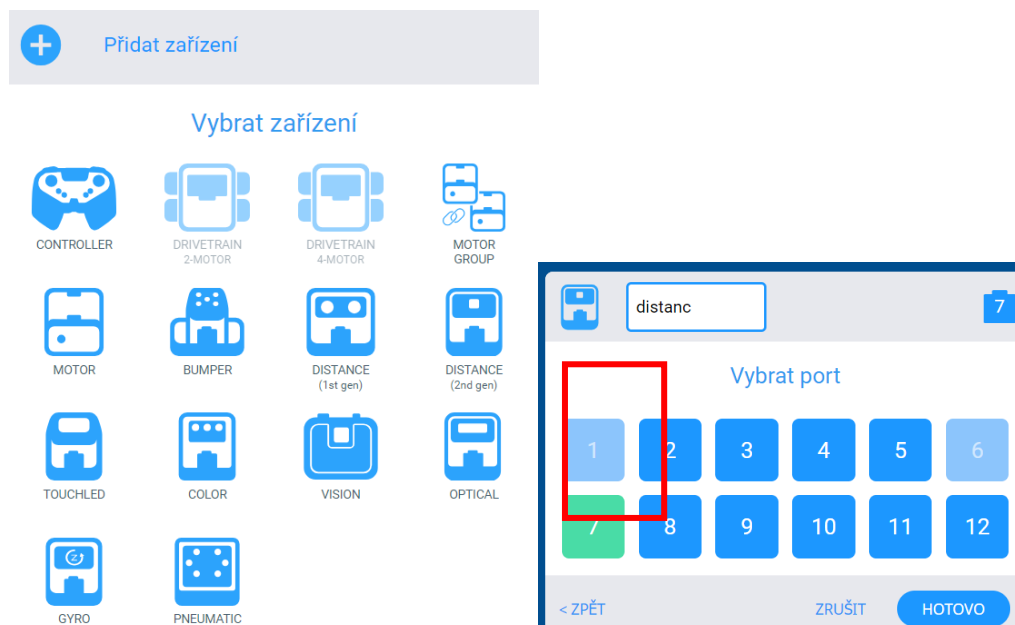
Bumper Switch – port 8

Distance senzor – port 7

NÁVOD – APLIKACE

Připojení – port 7

Distanc senzor 2.generace



The screenshot shows the application interface for adding a device. At the top, there is a button labeled "Přidat zařízení" with a plus sign icon. Below this, the text "Vybrat zařízení" is displayed. A grid of device icons is shown, including CONTROLLER, DRIVETRAIN 2-MOTOR, DRIVETRAIN 4-MOTOR, MOTOR GROUP, MOTOR, BUMPER, DISTANCE (1st gen), DISTANCE (2nd gen), TOUCHLED, COLOR, VISION, OPTICAL, GYRO, and PNEUMATIC. To the right, a dialog box titled "Vybrat port" is open, showing a search bar with "distanc" and a numeric keypad. The number 7 is highlighted in green, and a red box is drawn around the 7 and 8 keys. At the bottom of the dialog, there are buttons for "< ZPĚT", "ZRUŠIT", and "HOTOVO".

VZOROVÝ PROGRAM



```

1 #region VEXcode Generated Robot Configuration...
45 vexcode_brain_precision = 0
46 vexcode_console_precision = 0
47 myVariable = 0
48
49 def when_started1():
50     global myVariable, vexcode_brain_precision, vexcode_console_precision
51     drivetrain.drive(FORWARD)
52     while not distanc.object_distance(MM) < 20:
53         wait(20, MSEC)
54     drivetrain.stop()
55     brain.screen.print("HROZI NARAZ")
56
57 # Calibrate the Drivetrain Gyro
58 calibrate_drivetrain()
59
60 when_started1()
61

```

A CO DÁL...

- Proč nefunguje následující program?



```

1 #region VEXcode Generated Robot Configuration...
45 vexcode_brain_precision = 0
46 vexcode_console_precision = 0
47 myVariable = 0
48
49 def when_started1():
50     global myVariable, vexcode_brain_precision, vexcode_console_precision
51     if distanc.object_distance(MM) > 20:
52         drivetrain.drive(FORWARD)
53     else:
54         drivetrain.stop()
55         brain.screen.print("HROZI NARAZ")
56
57 # Calibrate the Drivetrain Gyro
58 calibrate_drivetrain()
59
60 when_started1()
61

```

- a proč tento funguje?



```

1 #region VEXcode Generated Robot Configuration...
45 vexcode_brain_precision = 0
46 vexcode_console_precision = 0
47 myVariable = 0
48
49 def when_started1():
50     global myVariable, vexcode_brain_precision, vexcode_console_precision
51     while True:
52         if distanc.object_distance(MM) > 20:
53             drivetrain.drive(FORWARD)
54         else:
55             drivetrain.stop()
56             brain.screen.print("HROZI NARAZ")
57             wait(20, MSEC)
58
59 # Calibrate the Drivetrain Gyro
60 calibrate_drivetrain()
61
62 when_started1()
63

```

- Jaký je rozdíl mezi předchozím programem a vzorovým programem? Co se stane po odstranění překážky?

ÚLOHA 5**Cíl:**

Připojení a programování Touch LED

**Úkol:**

Dotyková sonda svítí červeně. V okamžiku kdy je stisknuta, rozsvítí se zeleně a robot ujede 250mm.

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

Bumper Switch – port 8

Distance senzor – port 7

Touch LED – port 9

NÁVOD – APLIKACE

Připojení – port 9

+ Přidat zařízení

Vybrat zařízení



CONTROLLER



DRIVETRAIN
2-MOTOR



DRIVETRAIN
4-MOTOR



MOTOR
GROUP



MOTOR



BUMPER



DISTANCE
(1st gen)



DISTANCE
(2nd gen)



TOUCHLED



COLOR



VISION



OPTICAL



GYRO



PNEUMATIC

Zařízení
<>
🏠
🌞
?
>

📁

9

Vybrat port

1	2	3	4	5	6
7	8	9	10	11	12

< ZPĚT
ZRUŠIT
HOTOVO

VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      dotyk.set_color(Color.RED)
6      while not dotyk.pressing():
7          wait(20, MSEC)
8      dotyk.set_color(Color.GREEN)
9      drivetrain.drive_for(FORWARD, 250, MM)
10
11  when_started1()
12

```

A CO DÁL...

- Co stane po spuštění následujícího programu?



```

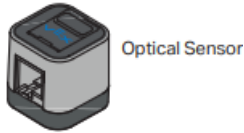
1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      dotyk.set_color(Color.RED)
6      while True:
7          while not dotyk.pressing():
8              wait(20, MSEC)
9              dotyk.set_color(Color.GREEN)
10             drivetrain.drive_for(FORWARD, 250, MM)
11             dotyk.set_color(Color.ORANGE)
12             wait(20, MSEC)
13
14  when_started1()
15

```

- Modifikujte předchozí programy

ÚLOHA 6**Cíl:**

Připojení a programování Optical Sensor

**Úkol:**

Robot jede vpřed. Při přiblížení k modré kostce ve vzdálenosti 25mm se zastaví.

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

Bumper Switch – port 8


Distance senzor – port 7

Touch LED – port 9















Optical Sensor – port 10

NÁVOD – APLIKACE

Připojení – port 10

 Přidat zařízení

Vybrat zařízení

 CONTROLLER	 DRIVETRAIN 2-MOTOR	 DRIVETRAIN 4-MOTOR	 MOTOR GROUP
 MOTOR	 BUMPER	 DISTANCE (1st gen)	 DISTANCE (2nd gen)
 TOUCHLED	 COLOR	 VISION	 OPTICAL
 GYRO	 PNEUMATIC		

oko 10

Vybrat port

1	2	3	4	5	6
7	8	9	10	11	12

< ZPĚT ZRUŠIT HOTOVO

VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      oko.gesture_disable()
6      oko.set_light_power(30, PERCENT)
7      drivetrain.drive(FORWARD)
8      while not oko.color() == Color.GREEN:
9          wait(20, MSEC)
10     drivetrain.turn_for(LEFT, 90, DEGREES)
11     drivetrain.drive(FORWARD)
12     while not oko.color() == Color.BLUE:
13         wait(20, MSEC)
14     drivetrain.stop()
15
16     when_started1()
17

```

A CO DÁL...

- Modifikujte předchozí program

ÚLOHA 7**Cíl:**

Připojení a programování drapáku

**Úkol:**

Robot rozevře drapák, vyčká pět sekund, zavře drapák a popojede o 100mm.

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 2

Motor drapáku – port 4

NÁVOD – APLIKACE

Připojení – port 4

The image shows two screenshots from the LEGO Mindstorms software interface. The left screenshot shows the 'Vybrat zařízení' (Select device) screen with a grid of device icons. The 'MOTOR' icon is highlighted with a red box. The right screenshot shows the 'Zařízení' (Device) screen for the gripper, with the name 'drapak' and a '4' in a blue box. Below the name are buttons for 'dopředu' (forward) and 'dozadu' (backward), and a toggle switch for 'Normální' (Normal) and 'Couvát' (Retract).

VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drapak.set_position(0, DEGREES)
6      drapak.set_velocity(30, PERCENT)
7      drapak.spin_for(REVERSE, 30, DEGREES)
8      wait(3, SECONDS)
9      drapak.spin_for(FORWARD, 35, DEGREES)
10     drivetrain.set_drive_velocity(30, PERCENT)
11     drivetrain.drive_for(FORWARD, 100, MM)
12
13     when_started1()
14

```

A CO DÁL...

- Modifikujte předchozí program
- Připojte optický senzor a pokuste vytvořit program využívající různých barev kostek
 - např. u modrých se otočí a jede dál, červené uchopí apod.

ÚLOHA 8**Cíl:**

Připojení a programování drapáku a paže

**Úkol:**

Robot rozevře drpák, jede vpřed 150mm, zavře drpák (uchopí předmět), paže se zvedne, robot se otočí vlevo o 90° jede vpřed 150, spustí paži, otevře drpák a jede vzad 200mm.

VZOROVÉ ZAPOJENÍ

Levý motor – port 1

Pravý motor – port 6

Motor drpáku – port 9















Motor paže – port 10

NÁVOD – APLIKACE


Připojení – port 4

+ Přidat zařízení

Vybrat zařízení

 CONTROLLER	 DRIVETRAIN 2-MOTOR	 DRIVETRAIN 4-MOTOR	 MOTOR GROUP
 MOTOR	 BUMPER	 DISTANCE (1st gen)	 DISTANCE (2nd gen)
 TOUCHLED	 COLOR	 VISION	 OPTICAL
 GYRO	 PNEUMATIC		

Zařízení





paze
10

Název a pokyny pro změnu

dopředu

dozadu





Normální Couvat

VYMAZAT
ZRUŠIT
HOTOVO

VZOROVÝ PROGRAM



```

1  myVariable = 0
2
3  def when_started1():
4      global myVariable
5      drapak.set_position(0, DEGREES)
6      paze.set_position(0, DEGREES)
7      drapak.set_velocity(30, PERCENT)
8      drapak.set_velocity(30, PERCENT)
9
10 def when_started2():
11     global myVariable
12     drapak.spin_for(FORWARD, 15, DEGREES)
13     drivetrain.drive_for(FORWARD, 150, MM)
14     drapak.spin_for(REVERSE, 20, DEGREES)
15     drapak.set_stopping(HOLD)
16     paze.spin_for(REVERSE, 90, DEGREES)
17     drapak.set_stopping(HOLD)
18     drivetrain.turn_for(LEFT, 90, DEGREES)
19     drivetrain.drive_for(FORWARD, 150, MM)
20     paze.spin_for(FORWARD, 90, DEGREES)
21     drapak.spin_for(FORWARD, 15, DEGREES)
22     drivetrain.drive_for(REVERSE, 200, MM)
23     drapak.spin_to_position(0, DEGREES)
24
25 ws2 = Thread( when_started2 )
26 when_started1()

```

A CO DÁL...

- modifikujte vzorový program
- zapojte optický senzor a nechte robota zvedat kostky jedné barvy
- zapojte Touch LED. V případě, že robot koná pohyb nechte LED svítit oranžově, v případě, že robot drží předmět nechte LED svítit zeleně
- vymyslete vlastní možnosti využití