# Úvod do grafického RobotC

Po spuštění programu si vytvoříme nový soubor (New File).

Pak doporučujeme **zkontrolovat**, zda je potřeba **nahrát do robota firmware** (Firmware Download). A to tehdy, pokud na displeji mozku robota vidíme něco jiného než Auto Pgms a TeleOp Pgms. Např. když tam máme programy nahrané pomocí ModKitu (na displeji jsou programy pod čísly 1 až 4), musíme firmware nahrát určitě.

**Pozn.** Někdy se stane, že mozek a senzory nebo motory nemají uvnitř stejný firmware, tuto situaci nezaměňujme s nahráním firmware z programu RobotC. Tato situace se řeší pomocí VEXOS Utility, která se nainstaluje při instalaci programu RobotC. Robot nám tuto neshodu verze firmware různých součástek sám zahlásí na displeji ve vhodné chvíli (nebo spíš nevhodné chvíli  $\odot$ ), ale dá se tomu předejít, pokud jej připojíme k počítači a zkontrolujeme přes VEXOS.

Dále můžeme přistoupit ke **konfiguraci robota** (Motor and Sensor Setup). Pokud si zatím postavíme jen podvozek, jako je na obrázku, stačí v sekci Motors nastavit, na kterém portu máme levý a na kterém pravý motor.

V prvním sloupci *Name* si motory pojmenováváme. Často se využívá původní anglický název, ale může zde být jakýkoliv. Je vohdné si vše nazvat tak, abychom se pak vyznali v našich programech.



**Ve druhém sloupci** *Type* pouze u všech motorů vybereme *VEX IQ Motor*. **Ve třetím sloupci** u jednoho motoru zatrhneme *reversed*.

		Motors a	and Sensors Setup	6	
Standard Models	lotors Devices				
Port	Name	Туре	Reversed	Drive Motor Side	
motor1		No motor 🛛 🗸			
motor2		No motor 🗸			
motor3		No motor 🗸			
motor4		No motor 👻			
motor5	Í	No motor 👻			
motor6	leftMotor	VEX IQ Motor 🗸		Left 🔻	
motor7	1	No motor 👻			
motor8		No motor 👻			
motor9	í –	No motor 🗸			
motor10	<b></b>	No motor 🗸			
motor11	1	No motor 👻			
motor12	rightMotor	VEX IQ Motor 👻		Right 👻	
	P. 0000001102.0000		·		
			Γ	OK Cancel Apply	Help

**Nastavení záložky Devices** se nás nyní netýká, ale pokud nějaké senzory zapojíme, tak je zde můžeme nazvat a nastavit. Podobně jako u motorů platí, že názvy si volíme sami, a ve sloupci Sensor Type vybíráme, o jaký senzor se jedná. Kde svítí Motor, tak na tuto položku samozřejmě nesaháme.

	Motors and Sensors Setup	×
Standard Models Motors Devices		
Port	Sensor Type	
port1	No Sensor 👻	
port2	No Sensor 👻	
port3	No Sensor 👻	
port4	No. Sensor 👻	
port5	No Sensor 👻	
port6	Motor 👻	
port7	No Sensor 🗸	
port8	No Sensor 🗸	
port9	No Sensor 🗸	
port10	No Sensor 🗸	
port11	No Sensor 👻	
port12	Motor 👻	
	OK Cancel	I <u>A</u> pply Help

## PROGRAMUJEME MINIVEXE

Příkazy vybíráme a přetahujeme myší z nabídky vlevo do očíslovaných řádků vpravo.

÷	ROBOTC	- • ×
File Edit View Robot Window Help		
New File	Motor and Sensor Setup Download Temperature Program Laboration	
Graphical Functions	VEX Start Page Graphical002.rbg*	4 Þ ×
▼ Program Flow	1 forward (1, rotations , 50);	
repeat (forever)		
repeatUntil		
while		
if if / else		
waitUntil		
> //comment		
▼ Timing		
resetTimer		
wait		
▼ Simple Behaviors	Glick and Drag	
backward	Circin alla Diag	
forward		
moveMotor		
<pre>turnLeft  </pre>		€ j
For Help, press F1	Robot VEX-IQ Graphical002.rbg*	0 Errors, 1 Warnings

#### NÁŠ PRVNÍ PROGRAM

Přetažením příkazu jsme vlastně vytvořili náš první program, nyní jej nahrajeme do robota. Připojíme mozek robota do USB počítače, zapneme jej a stiskneme Download to Robot. Počítač nás v tuto chvíli ještě poprosí o uložení našeho programu (to se zkrátka dělá vždy, než může dojít ke kompilaci a nahrání programu).



Pozn. při této operaci může dojít k hlášení, že nahrání se nezdařilo, nebo že selhalo propojení s robotem. Nejčastěji doporučujeme stisknout nahrávání znovu a napodruhé už to nikdy nezlobí. Pokud i nyní hlásí chybu propojení, zkontrolujeme kabel, stává se, že jej uživatel nezastrčil zcela do robota.

Po úspěšném nahrání se nám objeví buď okno s chybami nastavení robota, nebo se objeví malé okénko k ovládání programu. Zde můžeme program spustit tlačítkem Start, pokud je ještě robot připojen kabelem a můžeme zde i program hned zastavit stiskem Stop na stejném tlačítku.

	Prog	ram Debug	, ? ×	
Debug State	JS		Refresh	
Start	Suspend	Step Into	Continuous   🗸	
Clear All		Show PC	Show Datalog	

. . . .

Pozn. V případě chybových hlášení můžeme toto přeskočit, když jde o chybu čidel a program pravděpodobně bude fungovat, pokud jde o jízdu, ale skončí chybou i na displeji robota, pokud půjde o chybné nastavení motorů. V chybovém hlášení pak vidíme, že robot vidí někde námi nazvanou položku (sloupec Name), která ale není připojena (ve sloupci Type je NoSensor), nebo máme vybránu nějakou položku (ve sloupci Type je vybrán senzor), která není fyzicky zapojena.

#### CO UMÍ NÁŠ PROGRAM?

Popravdě, moc toho není. Robot popojede o jednu otáčku kol, rychlostí 50.



#### Matematicky vzato jedna otáčka kol odpovídá nějaké vzdálenosti.

Měřením můžeme zjistit, že průměr kol VEXe je 63,7 mm a tedy podle vzorce pro obvod kruhu o =  $2 \cdot \pi \cdot r$  dostaneme, že robot ujel vzdálenost  $2 \cdot 3,14 \cdot 31,85 = 200$  mm.



Jak můžeme nastavit **rychlost**? Hodnota **nabývá 1 – 100**.

Kromě hodnoty rotations můžeme volit degrees (stupně otočení kol), miliseconds, second, minutes.

( 1) forward ( 1	, rotations 🚽 ,	50 );
2	degrees rotations milliseconds seconds minutes	

#### ZATÁČENÍ ROBOTA

Určitě vás napadne použít příkaz turnLeft. Ovšem pozor, tímto způsobem se neotáčí celý robot, ale pouze kola. To jak moc se robot otočí, pak u tohoto příkazu řešíme spíše pokusem/omylem, protože to závisí na tom, jak velká kola máme a jak daleko od sebe jsou.

1 turnLeft (	1	,	rotations	<b>•</b> ,	50	);
2						

NĚKOLIK PŘÍKLADŮ K ŘEŠENÍ:

1) Kdy dojede robot nejdál? Při nastavení 2 rotations, 2 degrees nebo 2 seconds?

2) Kolik otáček kol je potřeba k ujetí vzdálenosti 45 cm?

3) Liší se vzdálenost ujetá při 1080 degrees a 3 rotations?

**4)** Dojede robot do výchozí pozice, pokud pojede *vpřed 5 rotations rychlostí 10* a pak *zpět 1800 degrees rychlostí 100*?

1	forward (	5 <b>,</b> ro	tations 🖥	,	10	);	
2	> backward <b>(</b> [	1800	degrees	•	], ]	100	);
3	>						

**5)** Otočte se s robotem čelem vzad (např. o dvě otáčky kol doleva, ale vaše hodnota může být jiná, musíte zkoušet).



Nyní popojeďte 500 mm, zde se otočte a jeďte zpět do místa, odkud jste vyjeli.

1	forward (	2.5 ,	rotations	<b>•</b> ,	50	);
2	turnLeft (	1,	rotations	• ,	50)	;
3	forward (	2.5,	rotations	<b>•</b> ,	50	);
4						

Možná vás napadlo, proč jsme jednoduše nezacouvali zpět? Záhy však zjišťujeme, že tím, jak jsme se na začátku otočili, musíme nyní po popojetí udělat znovu otáčku čelem vzad a popojet dopředu, aby byl robot opravdu exaktně ve stejné výchozí pozici.

6) Nechte robota objet překážku



Možné řešení je takovéto:

1	turnLeft ( 0.5 , rotations 👻 , 50 );
2	forward (1, rotations -, 50);
3	<pre>turnRight ( .5 , rotations + , 50 );</pre>
4	forward (3, rotations -, 50);
5	<pre>turnRight ( .5 , rotations -, 50 );</pre>
6	forward (1, rotations -, 50);
7	turnLeft ( .5 , rotations -, 50 );
8	forward (1, rotations -, 50);
9	2

7) Nechte robota, aby nakreslil obrazec:



# ŘEŠENÍ PROBLÉMŮ V PROGRAMU

Jak si hrajeme, program nabobtnává a můžeme se v něm snadno ztratit. Je tedy vhodné si začít pomáhat s orientací v něm. Ukážeme si několik možností, jak na to.

1	<pre>o forward ( 1 , rotations , 50 );</pre>	
2	turnRight (0.5, rotations -, 50)	;
3	<pre>forward ( 3 , rotations -, 50 );</pre>	
4	turnRight ( 0.5 , rotations -, 50 )	;
5	forward (1, rotations , 50);	
6	turnLeft ( 0.5 , rotations -, 50 );	1
7	forward (1, rotations , 50);	
8	turnLeft ( 0.5 , rotations -, 50 );	Xint
9	*	

Program obsahuje řadu příkazů, které mají často podobný vzhled a můžeme se v něm ztratit.

#### HLÁŠENÍ NA DISPLEJI A ZVUKOVÝ DOPROVOD

Jedna ze základních možností jsou pípnutí podle toho, v které části kódu se zrovna robot nachází. Zároveň nás o tom může informovat na displeji.



## PŘIDÁVÁNÍ KOMENTÁŘŮ DO PROGRAMU

Pokud programujete, víte, že zakomentovat můžeme každý řádek, nebo určitou část programu. Nabízí nám to i grafické RobotC.



Klikáním na čísla řádků je zakomentujeme:



**Inspirace a obrázky:** Classroom Activities for the Busy Teacher: VEX IQ with ROBOTC Graphical, Damien Kee, 2016

# VYBRANÉ PROGRAMY S KOMENTÁŘI

Programy odpovídají (není-li uvedeno jinak) výchozímu robotovi Clawbotovi ze stavebnice VEX IQ.

#### Ронуву

- 1 čeká 1 sekundu po spuštění programu
- 2 otočí se o 90° (jak víme, tak reálně asi o 30°; doprava o 90° je asi 0.7 až 0.8 rotations)
- 3 popojede o jednu otáčku motoru dopředu
- 4 rozevře kleště (v portu 11 motor kleští)
- 5 zvedne držák kleští (v portu 10 motor věže / zvedáku)

1	wait ( 1 , seconds -);
2	>turnRight(90,degrees ▾, 50);
3	forward ( 1 , rotations 🗸 , 50 );
4	moveMotor ( motor11 - , 1 , rotations - , 50 );
5	<pre>moveMotor ( motor10 - , 1 , rotations - , 50 );</pre>
6	>

JÍZDA S VYUŽITÍM ČIDEL

- 1 vyčkej, dokud se uživatel nedotkne LED
- 2 rozsviť dotykovou LED zeleně
- 3 jeď dopředu o jednu otáčku kol
- 4-6 otáčej se vpravo, dokud gyrosenzor nezaznamená hodnotu 90°
- 7 zastav otáčení
- 8-9 pohni věží a klepety



#### JÍZDA, DETEKCE BARVY A PODLE NÍ ZASTAVENÍ A ROZJETÍ (POUZE UKÁZKA VYUŽITÍ ČIDLA BAREV, NEMÁ KONEC, NUTNO NA MOZKU VYPNOUT KŘÍŽKEM)

- 1 opakuje stále dokola
- 2-4 rozjede se, dokud neuvidí červenou, pak zastaví
- 5-6 jakmile vidí zelenou, zase se rozjede

1	re	epeat (forever) {
2		setMultipleMotors ( 30 , motor1 - , motor6 - , noMotor - , noMotor - );
3		<pre>waitUntil ( getColorName(colorDetector)    ==    colorRed );</pre>
4		stopMultipleMotors ( motor1 - , motor6 - , noMotor - , noMotor - );
5		<pre>waitUntil ( getColorName(colorDetector)    ==    colorGreen );</pre>
6		setMultipleMotors ( 30 , motor1 - , motor6 - , noMotor - , noMotor - );
7	}	
8	>	

#### AUTONOMNÍ JÍZDA V PROSTŘEDÍ S PŘEKÁŽKAMI

- 1 opakuje stále dokola (pozor, program nemá konec! nutno vypnout na mozku robota křížkem)
- 2 nastaví motory 1 a 6 na couvání (mínus je dozadu) a neustále couvá
- 3 nastavil dotykovou LED na zelenou
- 4 čeká, dokud čidlo vzdálenosti nezaznamená, že méně než 150 mm od něj je překážka
- 5 v 15 cm od překážky zastaví motory
- 6 nastavil dotykovou LED na červenou
- 7 pootočí se doleva a zkouší se opět pohybovat

	> re	peat (forever) {
2	(	<pre>setMultipleMotors ( -50 , motor1 , motor6 , noMotor , no</pre>
3	(	<pre>setTouchLEDColor ( touchLED +, colorGreen +);</pre>
4		vwaitUntil ( getDistanceValue(distanceMM) ▼ < ▼ 150 );
5	(	> stopMultipleMotors ( motor1 → , motor6 → , noMotor → , noMotor → );
6	(	<pre>setTouchLEDColor ( touchLED +, colorRed +);</pre>
7	(	turnLeft ( 0.7 , rotations 🗸 , 50 );
8	}	
9	>	

#### JÍZDA, ZVEDÁNÍ DRŽÁKU S KLEPETY A DETEKCE ČÁSTI PROGRAMU POMOCÍ DOTYKOVÉ LED

1-5 jede dopředu, zvedne držák a položí ho, detekuje tuto část rozsvícením červené na LED

6-9, 10-13 opakuje pouze detekce je barvou žlutou a zelenou

14 vrátí se do výchozího bodu

(Program lze zjednodušit pomocí cyklu, při vynechání LED je to jednoduchá úloha)

1	wait ( 1 , seconds -);
2	forward ( 2.5 , rotations 🗸 , 50 );
3	moveMotor (armMotor , 1, rotations , 50);
4	moveMotor (armMotor v, -1, rotations v, 50);
5	> setTouchLEDColor ( touchLED -, colorRed -);
6	forward ( 2.5 , rotations v, 50 );
7	moveMotor (armMotor , 1, rotations , 50);
8	moveMotor ( armMotor 🗸 , -1 , rotations 🗸 , 50 );
9	<pre>setTouchLEDColor ( touchLED -, colorYellow -);</pre>
(10	forward ( 2.5 , rotations -, 50 );
(11	<pre>moveMotor ( armMotor +, 1 , rotations +, 50 );</pre>
(12	moveMotor ( armMotor v, -1 , rotations v, 50 );
(13	setTouchLEDColor(touchLED →, colorGreen →);
(14	backward ( 7.5 , rotations 🗸 , 50 );
(15	>

#### JÍZDA, DETEKCE BARVY A PRÁCE S KLEŠTĚMI

- 1 robot jede dopředu rychlostí 50 pomocí motoru v portu 1 a portu 6
- 2 testuje, zda neuvidí červený objekt
- 3 jakmile jej uvidí, zastaví všechny vybrané motory
- 4 rozevře kleště (motor v portu 11 ovládá kleště, záporná hodnota = rozevřít)
- 5 popojede ještě trochu dopředu (rychlost 20, tedy pomaleji)
- 6 sevře předmět do kleští (kladná hodnota)
- 7 couvne
- 8 rozevře kleště
- 9 znovu couvne

1	<pre>setMultipleMotors ( 50 , motor1 , motor6 , noMotor , noMotor );</pre>
2	<pre>waitUntil ( getColorName(colorDetector)    ==    colorRed );</pre>
3	<pre>stopMultipleMotors ( motor1 •, motor6 •, noMotor •, noMotor •);</pre>
4	moveMotor ( motor11 -, -0.3 , rotations -, 50 );
5	forward ( 0.4 , rotations -, 20 );
6	moveMotor ( motor11 - , 0.4 , rotations - , 50 );
7	backward ( 1 , rotations -, 50 );
8	moveMotor ( motor11 - , -0.3 , rotations - , 50 );
9	backward ( 0.5 , rotations -, 30 );
(10	$\rangle$

#### TŘÍDIČKA BAREV

Podle toho, jaký barevný předmět má před sebou jej klepety odsune vpravo nebo vlevo.



#### OVLADAČ A CLAWBOT

Program signalizuje, že běží svítící LED (zeleně). Pohyb paže a klepet se děje na tlačítkách vpředu vlevo a vpravo. Jízda dopředu a dozadu zajišťují proměnné *cislo1* a *cislo2*, které jsme si sami zavedli.



### JÍZDA PO ČÁŘE, ZÁKLADNÍ

Nutno zjisti práh detekce černé a bílé (někde mezi 50 a 150), nutno **nastavit barevný senzor na Greyscale!** 

	re	epeat (fo	rever) {
2		while (	getColorGrayscale(colorDetector) 🗸 < 🖌 120 ) {
3		> setM	otor ( leftMotor 🗸 , 50 );
4		> setM	otor ( rightMotor 🗸 , 0 );
5		}	
6		while (	getColorGrayscale(colorDetector)  → >=   120 ) {
7		> setM	otor ( rightMotor 🗸 , 50 );
8		> setM	otor ( leftMotor 🗸 , 0 );
9		}	
(10	}		
(11	>		

JÍZDA PO ČÁŘE S VYUŽITÍM VHODNÉHO PŘEDPŘIPRAVENÉHO PŘÍKAZU

Příkaz lineTrackLeft (nebo Right) umožňuje sledovat okraj čáry a plní stejnou funkci

1	> re	peat (forever) {
2		lineTrackLeft ( colorDetector 🗸 , 120 , 50 , 0 );
3		displaySensorValues ( line1 •, colorDetector •);
4	}	
5	>	

PŘÍKLAD PROGRAMU S VLOŽENÝMI KOMENTÁŘI, OVLÁDÁNÍ VSTUPU DO PROGRAMU POMOCÍ DOTYKOVÉ LED

	Otaceni porad dokola						
2	<pre>2 repeat (forever) {</pre>						
	Kdyz TouchLED stisknuta (hodn 1)						
4	if ( getTouchLEDValue(touchLED)    ==    1 ) {						
	Zapni motor klesti na rychlost 25						
6	setMotor ( clawMotor 🗸 , 25 );						
	Resetuj GyroSensor						
8	resetGyro (gyroSensor 👻 );						
	Delej dokud je otocka robota pod 180						
(10	while ( getGyroDegrees(gyroSensor) ▼ < ▼ 180 ) {						
	Pomalu otacime robota doleva						
(12	setMotor ( leftMotor 🗸 , -25 );						
(13	setMotor ( rightMotor 🗸 , 25 );						
(14	setMotor ( clawMotor 🗸 , 50 );						
(15	}						
	Jed dopredu 5 sekund rychlosti 50						
(17	forward ( 5 , seconds 🗸 , 50 );						
	Otevri klepeta na 3 sekundy rychlosti 50						
(19	moveMotor ( clawMotor 🗸 , 3 , seconds 🗸 , -50 );						
(20	}						
(21	}						
(22)							

#### PROGRAMOVÁNÍ OVLADAČE

Tento program je pro robota Stretch s úpravami s blinkry, tzv. Paraplíčko :-)

Tlačítka vpředu vlevo a vpravo ovládají paži (motor 4) a klepeta (motor 10) robota Stretch.

Jestliže hodnota naklonění kolíku A je větší, než 50, dotyková led vlevo se rozsvítí zeleně (na našem Stretchi tato TouchLED přidaná jako tzv. levý blinkr) a levý motor se rozjede rychlostí 50. Podobně při náklonu dolů (do -50) se motor rozjede dozadu rychlostí -50 (pro jednoduchost zde není programován i motor vpravo).

Pokud je kolík v nulové poloze, dotyková LED svítí červeně.





Robot Paraplíčko, upravený Stretch s blinkry. Dotyková LED je zde celkem třikrát. Jednou se spouští a signalizují programy a dvě jsou po stránách.



# **Remote Control - Names**

