

## INITIALIZE

The screenshot shows the ROBOTC software interface. The 'Robot' menu is circled in red. Below the menu, the configuration is as follows:

- Robot Platform Type: VEX Robotics, VEX 2.0 Cortex, Natural Language 2.0
- Robot VEXCortex Communication Mode: VEXNet or USB
- Robot Compiler target: Physical Robot

A red box on the right contains the following text:

If you want to get default back on the robot  
[VEX.com/firmware](http://VEX.com/firmware)  
For upgrading joystick or robot firmware

## START A NEW FILE

### MOTORS and SENSORS

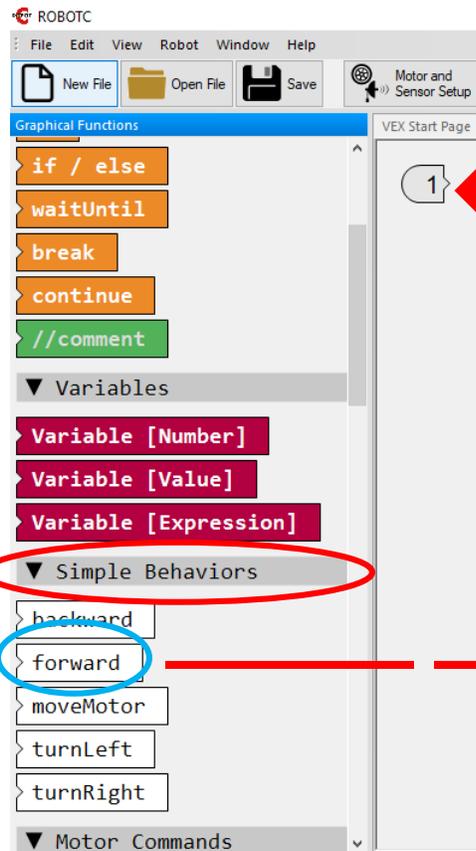
The screenshot shows the 'Motors and Sensors Setup' window in ROBOTC. The 'Motor and Sensor Setup' button in the menu is circled in red. The window displays a table of motor configurations:

Port	Name	Type	Reversed	Encoder Port	PID Control	Drive Motor Side
port1		No motor				
port2	rightMotor	VEX 393 Motor	<input checked="" type="checkbox"/>	I2C_1	<input checked="" type="checkbox"/>	Right
port3	leftMotor	VEX 393 Motor	<input type="checkbox"/>	I2C_2	<input checked="" type="checkbox"/>	Left
port4		No motor				
port5		No motor				
port6		No motor				
port7		No motor				
port8		No motor				
port9		No motor				
port10		No motor				

Annotations with red arrows point to specific elements:

- 'Select the correct' points to the 'port3' row.
- 'Name the motor' points to the 'leftMotor' name field.
- 'One motor needs to be reverse from the' points to the 'Reversed' checkbox for port3.
- 'Indicate which side of the robot the motor is on' points to the 'Left' dropdown in the 'Drive Motor Side' column.

## BUILD A PROGRAM:



Drag and drop your command

Choose: degrees & rotations require an encoder; better to choose time, milliseconds, \*seconds or minutes

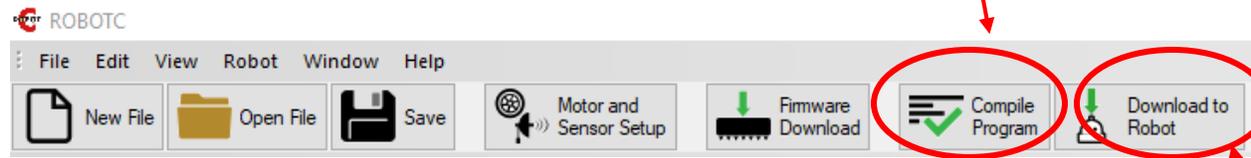
```
1 forward ( 1 , seconds , 50 );  
2
```

# of rotations

Speed; 0-100

Select to check your program for errors

or



**MUST DO THIS EVERY TIME YOU CHANGE THE CODE**  
You should get a debugger window.  
Click **START** to test program

You can create wider turns using the following to make each motor run at different speeds:

```
1 setMotor ( rightMotor , 100 );
2 setMotor ( leftMotor , 50 );
3 }
```

You must also add timing and stop motors

▼ Timing

- resetTimer
- wait

▼ Motor Commands

- moveMotorTarget
- resetMotorEncoder
- setMotor
- setMotorTarget
- setMultipleMotors
- setPIDforMotor
- stopAllMotors
- stopMotor
- stopMultipleMotors

```
1 setMotor ( rightMotor , 100 );
2 setMotor ( leftMotor , 50 );
3 wait ( 1 , seconds );
4 stopAllMotors ( );
5 }
```

+ speeds run clockwise  
- Speeds run counterclockwise

You can also select which motors you would like to stop

### ADD OTHER MOTORS

Clicking on the numbers will turn the line green and makes them // COMMENTS

```
1 moveMotor ( armMotor , 1 , seconds , 50 );
2 setMotor ( clawMotor , 50 );
3 wait ( 1 , seconds );
4 stopAllMotors ( );
5 }
```

Make sure you use "seconds" for timing if you are not using an encoder

You can create a template for your motor set up:

- Motors and Sensors
- Standard Models
- Custom Configuration
- Save as New User Model
- Apply
- File containing User Def. Mod. Config (browse)

## REMOTE CONTROL

Graphical Functions

- stopMultipleMotors
- ▼ Remote Control
  - arcadeControl
  - armControl
  - buttonControl
  - joystickControl
  - setJoystickScale
  - tankControl**
- ▼ Timing

```
1 tankControl ( Ch3 , Ch2 , 10 );  
2
```

Buttons on the controller

threshold

armControl and buttonControl do the same thing

## CREATE A LOOP

Graphical Functions

- ▼ Program Flow
  - repeat
  - repeat (forever)**
  - repeatUntil
  - while
  - if
  - if / else
  - waitUntil
  - break
  - continue
  - //comment

```
1 repeat (forever) {  
2   tankControl ( Ch3 , Ch2 , 10 );  
3   armControl ( port10 , Btn5U , Btn5D , 75 );  
4 }  
5
```

speed

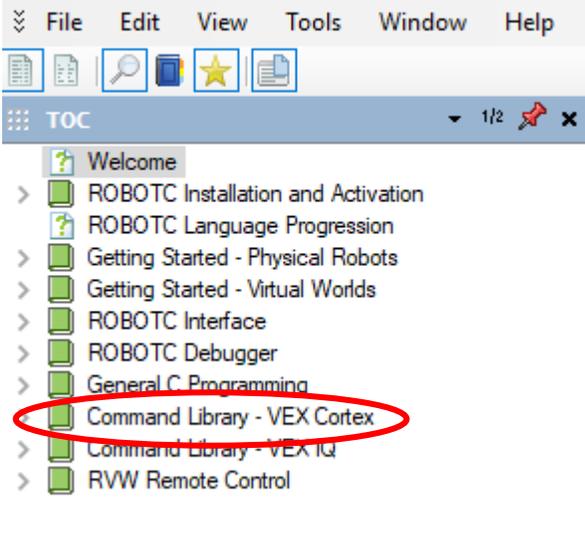
These may not default to up and down, you may have to switch them here



turn on the robot, then turn on the remote



MadCap Help Viewer V6.3



If you forget or need help you can use the help function. All functions can be found in HELP

Example:

- \* Help
- Open Help F1
- Command Library VEX Cortex
- Graphical
- Remote Control
- Tank control

**COMPETITION TEMPLATE**  
Important for autonomous



**File**  
**New**  
**Competition Template**

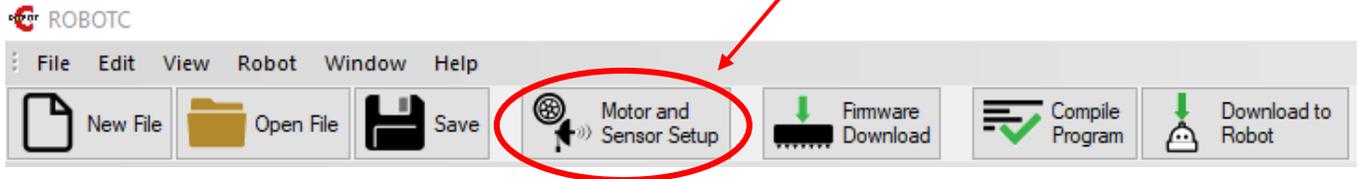
```

1 // Use Motors & Sensor Setup to Configure Your Robot!
2 enableCompetitionMode ( );
3 repeat (forever) {
4     // Place Autonomous Commands Below
5     if ( getCompetitionAutonomous() == true ) {
6     }
7 }
8     // Place Remote Control Commands Below
9     if ( getCompetitionDriverControl() == true ) {
10 }
11 }
12 }
13
    
```

## SENSORS

You can check where to plug sensors in by going to "Motors and Sensors setup"

Select your sensor.



```
1 while ( SensorValue[bumperSwitch] == 0 ) {  
2   setMultipleMotors ( 50 , leftMotor , rightMotor , noMotor , noMotor );  
3 }  
4 stopAllMotors ( );  
5 }
```

Or

```
1 if ( SensorValue[bumperSwitch] == 0 ) {  
2   setMultipleMotors ( 50 , rightMotor , leftMotor , noMotor , noMotor );  
3 } else {  
4   stopAllMotors ( );  
5 }  
6 }
```

\*To see values of sensors:  
With debugger window open

Robot  
Debugger window  
Sensors