ROBOTC Natural Language - VEX Cortex Quick Reference:

Set Servo Set a servo to a desired position. *Default servo and position: port6, 0.*	`setServo();`	`setServo(port7, 95);`

Start Motor
Set a specific motor to a speed.
Default motor and speed: port6, 95.
	`startMotor();`
	`wait();`
	`stopMotor();`
		`startMotor(port8, -32);`
		`wait(0.5);`
		`stopMotor(port8);`

Stop Motor
Stop a specific motor.
Default motor: port6.
	`startMotor();`
	`wait();`
	`stopMotor();`
		`startMotor(port8, -32);`
		`wait(0.5);`
		`stopMotor(port8);`

Wait
Wait an amount of time measured in seconds.
Default time: 1.0.
	`startMotor();`
	`wait();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`wait(2.7);`
		`stopMotor(port8);`

Wait in Milliseconds
Wait an amount of time measured in milliseconds.
Default time: 1000.
	`startMotor();`
	`waitInMilliseconds();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`waitInMilliseconds(2700);`
		`stopMotor(port8);`

Until Touch
The robot waits for the Touch Sensor to be pressed.
Default sensor port: dgtl6.
	`startMotor();`
	`untilTouch();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilTouch(dgtl10);`
		`stopMotor(port8);`

Until Release
The robot waits for the Touch Sensor to be released.
Default sensor port: dgtl6.
	`startMotor();`
	`untilRelease();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilRelease(dgtl10);`
		`stopMotor(port8);`

Until Bump
The robot waits for the Touch Sensor to be pressed in and then released out.
Default sensor port and delay time: dgtl6, 10.
	`startMotor();`
	`untilBump();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilBump(dgtl10, 100);`
		`stopMotor(port8);`

Until Button Press
The robot waits for a button on the VEX LCD to be pressed.
Default button: centerBtnVEX.
	`startMotor();`
	`untilButtonPress();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilButtonPress(rightBtnVEX);`
		`stopMotor(port8);`

Until Sonar - Less Than
The robot waits for the Sonar Sensor to read a value in cm less than the threshold.
Default threshold and sensor port: 30, dgtl8+9.
	`startMotor();`
	`untilSonarLessThan();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilSonarLessThan(45, dgtl2);`
		`stopMotor(port8);`

Until Sonar - Greater Than
The robot waits for the Sonar Sensor to read a value in cm greater than the threshold.
Default threshold and sensor port: 30, dgtl8+9.
	`startMotor();`
	`untilSonarGreaterThan();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilSonarGreaterThan(45, dgtl2);`
		`stopMotor(port8);`

ROBOTC Natural Language - VEX Cortex Quick Reference:

Until Potentiometer - Greater Than
The robot waits for the Potentiometer Sensor to read a value greater than a set position.
Default threshold and sensor port: 2048, in6.
	`startMotor();`
	`untilPotentiometerGreaterThan();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilSonarGreaterThan(4000, in4);`
		`stopMotor(port8);`

Until Potentiometer - Less Than
The robot waits for the Potentiometer Sensor to read a value less than a set position.
Default threshold and sensor port: 2048, in6.
	`startMotor();`
	`untilPotentiometerLessThan();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilSonarLessThan(40, in4);`
		`stopMotor(port8);`

Until Dark
The robot waits for the Light Sensor to read a value greater than the threshold.
Default threshold and sensor port: 1500, in2.
	`startMotor();`
	`untilDark();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilDark(1005, in4);`
		`stopMotor(port8);`

Until Light
The robot waits for the Light Sensor to read a value less than the threshold.
Default threshold and sensor port: 1500, in2.
	`startMotor();`
	`untilLight();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilLight(1005, in4);`
		`stopMotor(port8);`

Until Rotations
The robot waits for an encoder to reach a specified number of rotations.
Default rotations, encoder: 1.0, dgtl1+2
	`startMotor();`
	`untilRotations();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilRotations(2.75, dgtl3);`
		`stopMotor(port8);`

Until Encoder Counts
The robot waits for an encoder to reach a specified number of encoder counts.
Default counts, encoder: 360, dgtl1+2.
	`startMotor();`
	`untilEncoderCounts();`
	`stopMotor();`
		`startMotor(port8, 63);`
		`untilEncoderCounts(990, dgtl3);`
		`stopMotor(port8);`

LED ON
Turn an LED in a specified digital port ON.
Default sensor port: dgtl2.
	`turnLEDOn();`
	`wait();`
	`turnLEDOff();`
		`turnLEDOn(dgtl7);`
		`wait(0.5);`
		`turnLEDOff(dgtl7);`

LED OFF
Turn an LED in a specified digital port OFF.
Default sensor port: dgtl2.
	`turnLEDOn();`
	`wait();`
	`turnLEDOff();`
		`turnLEDOn(dgtl7);`
		`wait(0.5);`
		`turnLEDOff(dgtl7);`

Flashlight ON
Turn a VEX Flashlight in a specified motor port ON at a specified brightness.
Default motor port and brightness: port4, 63.
	`turnFlashlightOn();`
	`wait();`
	`turnFlashlightOff();`
		`turnFlashlightOn(port10, 127);`
		`wait(0.5);`
		`turnFlashlightOff(port10);`

Flashlight OFF
Turn a VEX Flashlight in a specified motor port OFF.
Default motor port: port4.
	`turnFlashlightOn();`
	`wait();`
	`turnFlashlightOff();`
		`turnFlashlightOn(port10, 127);`
		`wait(0.5);`
		`turnFlashlightOff(port10);`

Robot Type Choose which robot you are using (Recbot or Swervebot). *Default bot:* none.	`robotType();`	`robotType(swervebot);`

ROBOTC Natural Language - VEX Cortex Quick Reference:

Forward
The robot drives straight forward.
Default speed: 95.
	`forward();`
	`wait();`
	`stop();`
		`forward(63);`
		`wait(2.0);`
		`stop();`

Backward
The robot drives straight backward.
Default speed: -95.
	`backward();`
	`wait();`
	`stop();`
		`backward(63);`
		`wait(2.0);`
		`stop();`

Point Turn
The robot makes a sharp turn in place.
Default direction and speed: right, 95.
	`pointTurn();`
	`wait();`
	`stop();`
		`pointTurn(left, 63);`
		`wait(0.4);`
		`stop();`

Swing Turn
The robot makes a wide turn, activating only one drive motor.
Default direction and speed: right, 95.
	`swingTurn();`
	`wait();`
	`stop();`
		`swingTurn(left, 63);`
		`wait(0.75);`
		`stop();`

Stop
The robot halts both driving motors, coming to a stop.
	`forward();`
	`wait();`
	`stop();`
		`forward(63);`
		`wait(2.0);`
		`stop();`

Line Track - for Time
The robot tracks a dark line on a light surface for a specified time in seconds.
Default time, threshold, sensors: 5.0, 505, in1, in2, in3 (Left, Center, Right).
	`lineTrackForTime();`
	`stop();`
		`lineTrackForTime(7.5, 99, in6, in7, in8);`
		`stop();`

Line Track - for Rotations
The robot tracks a dark line on a light surface for a specified distance in rotations.
Default time, threshold, sensors: 3.0, 505, in1, in2, in3 (Left, Center, Right).
	`lineTrackForRotations();`
	`stop();`
		`lineTrackForRotations(4.75, 99, in6, in7, in8);`
		`stop();`

Move Straight - for Time
The robot will use encoders to maintain a straight path for a specified time in seconds.
Default time, rightEncoder, leftEncoder: 5.0, dgtl1+2, dgtl3+4.
	`moveStraightForTime();`
	`stop();`
		`moveStraightForTime(7.5, dgtl5, dgtl3);`
		`stop();`

Move Straight - for Rotations
The robot will use encoders to maintain a straight path for a specified distance in encoder rotations.
Default rotations, rightEncoder, leftEncoder: 1.0, dgtl1+2, dgtl3+4.
	`moveStraightForRotations();`
	`stop();`
		`moveStraightForRotations(4.75, dgtl5, dgtl3);`
		`stop();`

Tank Control
The robot is remote controlled with the right motor mapped to the right joystick and the left motor mapped to the left joystick.
Default right and left joystick, threshold: Ch2, Ch3, 10.
	`while(true)`
	`{`
	`  tankControl();`
	`}`
		`while(true)`
		`{`
		`  tankControl(Ch1, Ch4, 5);`
		`}`

Arcade Control
The robot is remote controlled with both motors mapped to a single joystick.
Default vertical, horizontal joysticks and threshold: Ch2, Ch1, 10.
	`while(true)`
	`{`
	`  arcadeControl();`
	`}`
		`while(true)`
		`{`
		`  arcadeControl(Ch1, Ch4, 5);`
		`}`