

# ROBOTC Natural Language - TETRIX Quick Reference:

<p><b>Set Servo</b> Set a servo to a desired position. <i>Default motor and speed: srvo_S1_C1_1, 0.</i></p>	<pre>setServo();</pre>	<pre>setServo(srvo_S1_C1_6, 200);</pre>
<p><b>Start Motor</b> Set a specific motor to a speed. <i>Default motor and speed: motorA, 75.</i></p>	<pre>startMotor(); wait(); stopMotor();</pre>	<pre>startMotor(motorF, -25); wait(0.5); stopMotor(motorF);</pre>
<p><b>Stop Motor</b> Stop a specific motor. <i>Default motor: motorA.</i></p>	<pre>startMotor(); wait(); stopMotor();</pre>	<pre>startMotor(motorC, -25); wait(0.5); stopMotor(motorC);</pre>
<p><b>Wait</b> Wait an amount of time measured in seconds. <i>Default time: 1.0.</i></p>	<pre>startMotor(); wait(); stopMotor();</pre>	<pre>startMotor(motorC, 50); wait(2.7); stopMotor(motorC);</pre>
<p><b>Wait in Milliseconds</b> Wait an amount of time measured in milliseconds. <i>Default time: 1000.</i></p>	<pre>startMotor(); waitInMilliseconds(); stopMotor();</pre>	<pre>startMotor(motorC, 50); waitInMilliseconds(2700); stopMotor(motorC);</pre>
<p><b>Until Touch</b> The robot waits for the Touch Sensor to be pressed. <i>Default sensor port: S1.</i></p>	<pre>startMotor(); untilTouch(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilTouch(S4); stopMotor(motorC);</pre>
<p><b>Until Release</b> The robot waits for the Touch Sensor to be released. <i>Default sensor port: S1.</i></p>	<pre>startMotor(); untilRelease(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilRelease(S4); stopMotor(motorC);</pre>
<p><b>Until Bump</b> The robot waits for the Touch Sensor to be pressed in and then released out. <i>Default sensor port and delay time: S1, 10.</i></p>	<pre>startMotor(); untilBump(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilBump(S4, 100); stopMotor(motorC);</pre>
<p><b>Until Button Press</b> The robot waits for a button on the NXT to be pressed. <i>Default button: centerBtnNXT.</i></p>	<pre>startMotor(); untilButtonPress(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilButtonPress(rightBtnNXT); stopMotor(motorC);</pre>
<p><b>Until Sonar - Less Than</b> The robot waits for the Sonar Sensor to read a value in cm less than the threshold. <i>Default threshold and sensor port: 30, S4.</i></p>	<pre>startMotor(); untilSonarLessThan(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilSonarLessThan(45, S1); stopMotor(motorC);</pre>
<p><b>Until Sonar - Greater Than</b> The robot waits for the Sonar Sensor to read a value in cm greater than the threshold. <i>Default threshold and sensor port: 30, S4.</i></p>	<pre>startMotor(); untilSonarGreaterThan(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilSonarGreaterThan(45, S1); stopMotor(motorC);</pre>

# ROBOTC Natural Language - TETRIS Quick Reference:

<p><b>Until Dark</b> The robot waits for the Light Sensor to read a value less than the threshold. <i>Default threshold and sensor port: 45, S3.</i></p>	<pre>startMotor(); untilDark(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilDark(15, S2); stopMotor(motorC);</pre>
<p><b>Until Light</b> The robot waits for the Light Sensor to read a value greater than the threshold. <i>Default threshold and sensor port: 45, in2.</i></p>	<pre>startMotor(); untilLight(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilLight(85, S2); stopMotor(motorC);</pre>
<p><b>Until Sound - Less Than</b> The robot waits for the Sound Sensor to read a value less than the threshold. <i>Default threshold and sensor port: 50, S2.</i></p>	<pre>startMotor(); untilSoundLessThan(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilSoundLessThan(15, S3); stopMotor(motorC);</pre>
<p><b>Until Sound - Greater Than</b> The robot waits for the Sound Sensor to read a value greater than the threshold. <i>Default threshold and sensor port: 50, S2.</i></p>	<pre>startMotor(); untilSoundGreaterThan(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilSoundGreaterThan(85, S3); stopMotor(motorC);</pre>
<p><b>Until Rotations</b> The robot waits for a motor-encoder to reach a specified number of rotations. <i>Default rotations, encoder: 1.0, motorB</i></p>	<pre>startMotor(); untilRotations(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilRotations(2.75, motorA); stopMotor(motorC);</pre>
<p><b>Until Encoder Counts</b> The robot waits for a motor-encoder to reach a specified number of encoder counts. <i>Default counts, encoder: 360, motorB.</i></p>	<pre>startMotor(); untilEncoderCounts(); stopMotor();</pre>	<pre>startMotor(motorC, 50); untilEncoderCounts(990, motorA); stopMotor(motorC);</pre>
<p><b>Forward</b> The robot drives straight forward. <i>Default speed: 75.</i></p>	<pre>forward(); wait(); stop();</pre>	<pre>forward(50); wait(2.0); stop();</pre>
<p><b>Backward</b> The robot drives straight backward. <i>Default speed: -75.</i></p>	<pre>backward(); wait(); stop();</pre>	<pre>backward(50); wait(2.0); stop();</pre>
<p><b>Point Turn</b> The robot makes a sharp turn in place. <i>Default direction and speed: right, 75.</i></p>	<pre>pointTurn(); wait(); stop();</pre>	<pre>pointTurn(left, 50); wait(0.4); stop();</pre>
<p><b>Swing Turn</b> The robot makes a wide turn, activating only one drive motor. <i>Default direction and speed: right, 75.</i></p>	<pre>swingTurn(); wait(); stop();</pre>	<pre>swingTurn(left, 50); wait(0.75); stop();</pre>
<p><b>Stop</b> The robot halts both driving motors, coming to a stop.</p>	<pre>forward(); wait(); stop();</pre>	<pre>forward(50); wait(2.0); stop();</pre>

# ROBOTC Natural Language - TETRIS Quick Reference:

<p><b>Line Track - for Time</b> The robot tracks a dark line on a light surface for a specified time in seconds. <i>Default time, threshold, sensors: 5.0, 45, S3.</i></p>	<pre>lineTrackForTime (); stop ();</pre>	<pre>lineTrackForTime (7.5, 75, S2); stop ();</pre>
<p><b>Line Track - for Rotations</b> The robot tracks a dark line on a light surface for a specified distance in rotations. <i>Default time, threshold, sensors: 3.0, 45, S3.</i></p>	<pre>lineTrackForRotations (); stop ();</pre>	<pre>lineTrackForRotations (4.75, 75, S3); stop ();</pre>
<p><b>Move Straight - for Time</b> The robot will use encoders to maintain a straight path for a specified time in seconds. <i>Default time, rightEncoder, leftEncoder: 5.0, motorB, motorC.</i></p>	<pre>moveStraightForTime (); stop ();</pre>	<pre>moveStraightForRotations (4.75, motorC, motorA); stop ();</pre>
<p><b>Move Straight - for Rotations</b> The robot will use encoders to maintain a straight path for a specified distance in encoder rotations. <i>Default rotations, rightEncoder, leftEncoder: 1.0, motorB, motorC.</i></p>	<pre>moveStraightForRotations (); stop ();</pre>	<pre>moveStraightForRotations (4.75, motorC, motorA); stop ();</pre>
<p><b>Tank Control</b> The robot is remote controlled with the right motor mapped to the right joystick and the left motor mapped to the left joystick. <i>Default right and left and joystick: joy1_y2, joy1_y1.</i></p>	<pre>while (true) {     tankControl (); }</pre>	<pre>while (true) {     tankControl (joystick.joy1_x2, joystick.joy1_x1); }</pre>
<p><b>Arcade Control</b> The robot is remote controlled with both motors mapped to a single joystick. <i>Default vertical and horizontal joysticks: joy1_y2, joy1_y1.</i></p>	<pre>while (true) {     arcadeControl (); }</pre>	<pre>while (true) {     arcadeControl (joystick.joy1_y1, joystick.joy1_x1); }</pre>
<p><b>Robot Type</b> Choose which robot you are using. <i>Default bot: none.</i></p>	<pre>robotType ();</pre>	<pre>robotType (mantis);</pre>