# ROBOTC 3.0 Improvements from ROBOTC 2.x to ROBOTC 3.0

## General Improvements:

- Updated Licensing System - one installation of ROBOTC now supports all ROBOTC platforms and features!
- Robot Model Chooser - pick the default model you're using and ROBOTC sets up all motors and sensors for you.
- Natural Language Libraries now integrated – users can use words like “forward, backward, trackline, and untilEncoderCount” to control their robots. This was done to improve first time user experience for new programmers. In order to access this library of commands ROBOTC has a new platform type called “Natural Language”.
- ROBOTC3.0 Updater – users that are connected to the internet are automatically alerted when updates are available to ROBOTC3.0. ROBOTC will automatically download and begin installation of these updates if you want it to.
- New Toolbar Command - "Format Whole File" to fix the indentation of your code. This is like magic, select the magic wand tool and now ROBOTC formats the whole body of code.
- New Toolbar Command - "Toggle Block Comment Tool” to automatically 'comment out' a block of code. More magic! Select a whole block of code with your mouse and then select the Toggle Block Comment Tool and that whole block of code is commented out.
- Updated Debugger Windows - now only see the sensors you're using on your robot instead of every port available.
- Compiler improvements - detection for missed semicolons and accidental semicolons on control statements. Initially this may be confusing when you are using older ROBOTC programs because you will receive warning with the new ROBOTC compiler asking if you really meant to do this.
- Compiler Improvements - added support for C-language utility commands atoi and atol.
- Multitasking Improvements - added support for Semaphores for tasks. See help documentation for more information.
- ROBOTC Emulator - you can now run your ROBOTC program on a PC-based emulator. Great for debugging/testing programs without a robot connected.
- Compiler Improvement - new "conditional compile" statements allow you to create blocks of code in your program that will only run on a specific platform or debugger target
- "Motor and Sensors Setup" can get confused when multiple user programs are open. Change behavior so it will only open when user program is the selected window; otherwise command is disabled.
- Updated Help Documentation
- Many New and Updated Sample Programs
- Numerous other fixes and enhancements.

## NXT/TETRIX Specific Improvements

- Integrated FTC Support into ROBOTC 3.0 for all license types.
- Improved FTC Joystick Driver - prevents motors from running when communication is lost. Motors will shut down after 3 seconds to prevent damaged motors.
- Updated 3rd Party Sensor Driver Libraries
- Improved TETRIX Motor/Sensor Controller communication - prevented issues where communication would shut down on a specific battery level.
- Improved NXT Motor Compatibility - moving to specific targets, speed ramping and PID functionality has been improved.
- NXT Color Sensor Support
- 'sensorAnalogActive' and 'sensorAnalogInactive' were not properly setting up 9V control. This has been fixed.

## VEX Cortex Specific Improvements

- Improved Cortex to PC Communication support. VEX Cortex will now act as a Serial device rather than a HID device, improving compatibility on some computers.
- Easier to Choose Communication Ports - new "View - Preferences - Communications Port Selection" makes it easier to pick how to communicate to your Cortex.
- VEX Advanced Sensor Support: VEX Gyro
- VEX Cortex now uses 32-bit values for timers, instead of 16-bit values.
- Improvements and Support for VEX Cortex Serial Ports.
- VEX Cortex digital sensor 12 was not working properly as output sensor. Fixed.

## Robot Virtual World (RVW) Integration

ROBOTC is the only educational software solution that has fully integrated virtual worlds that allow users without robots to program LEGO and VEX robots without a robot. The RVW builds automatically install into ROBOTC when they are downloaded. There are over 6 worlds that will be released over the next 6 months that have already been developed. Our development team will make improvements to the worlds based on user feedback.

- New feature to allow multiple worlds to be installed. Users can now select the Virtual World they would like from the Window-Choose Virtual World Package.
- USB Joystick Functionality. Users can utilize a Logitech “Dual Action” or “F310” USB Joystick and write code to control VEX or NXT robots using a joystick.
- Full physics engine applied to Robot Virtual World models.
- Nearly all of the challenges that are found in the ROBOTC Curriculums are integrated into the new RVW.
- Popular robot competition tables are modeled in RVW and teams will be able to test their robot’s code on a virtual game table while the robot is still being built.
- Fantasy worlds are available that allow users to program robots on other planets, underwater, on tropical islands, and more. Within the next year, there will be a fully functioning ROBOTC programming game where users learn to program as they play the game.

## General ROBOTC Bug Fixes

- Corrected bug in temporary register re-use. Compile would re-use a 2-byte temporary variable when 4-byte temporary was required. Result was that intermediate calculations were being incorrectly truncated to lower 16-bits only.
- Corrected bug in "sgn" opcode for 'float' parameter was incorrectly checking the 'short' result to see if it was a 'float' NAN (Not A Number).
- "Variable Argument" list bug fix. All varArg arguments should be RAM variables. Compiler automatically converts constant values to temporary RAM variables except for constants that are "short" values (e.g. the "sizeof(xx)" function)
- Fix undefined macro variables in inactive conditional compiles (i.e. "#elif" and "#if". Compile error was incorrectly being generated when the condition was "dead".
- Add support for preprocessor command "#" (i.e. 'stringizer'). Add support for built-in preprocessor macros __DATE__, __TIME__, __FILE__ and __LINE__.
- Fix code optimization bug where two internal compiler variables of type "pointer + struct offset" would match as equal if only the "pointer" value matched and was not also matching on "struct offset" matching.
- Improved implementation of "pow" (powerOf) function to use standard C libraries.
- Fix problem with procedure return value not being assigned to temporary variable as if "if (fcn(5) < fcn(6))" then return results from "fcn(5)" should be saved in a temporary variable.
- Improve auto-indenting when editing/typing code. Was not properly indenting following "if", "else", "for" and "while".
- Support "expansion" of the same macro. e.g. #define min3(a, b, c) min(a, min(b, c))
- Add new version of "wait1MSec" that accepts a "long" variable parameter.
- Added/Improved Keyboard Shortcuts for Commands - List can be found inside of the ROBOTC menus.
- Add overloaded version of 'strcat' and 'strcpy' intrinsic for 'string' parameter that will range check result size.
- Add intrinsic functions that will concatenate a single character to a string.
- Debugger Commands - 'set next line' and 'run to cursor' improvements.
- Closing active user source file while running debugger would crash ROBOTC. This is fixed.
- Numerous other fixes and enhancements.

## 3.00 to 3.01 Change Log:

- REQUIRED: All NXT Users must update to the new NXT USB Communication Driver (Fantom.dll). ROBOTC may not communicate with the NXT unless this driver is updated.
- Added NXT + Samantha Module Wifi support. Users can now download/debug over Wifi with a Samantha Module.
- Updated JoystickDriver.c for NXT Remote Control to improve communication and fix errors when communication was lost.
- Fixed Preferences Menu where User Specified "Include" directories were not being saved.
- Fixed bug where ROBOTC would crash when opening more than 25 files. The limit is now 100.
- Fixed bug where TETRIX Servo and Motor Names were not properly being stored.
- Added new competition templates for VEX Cortex and PIC. All user spawned tasks are now stopped when switching between Autonomous and Tele-Op.
- Fixed bug with VEX Remote LCD Screen - Text was being displayed off of the visible window space sometimes.

## 3.01 to 3.02 Change Log:

- Added Error Handling and Support for all versions of Fantom.dll. ROBOTC would crash if the latest version of the NXT driver was not installed on the system. ROBOTC now prompts and informs the user of the error, rather than crashing.
- Fixed Quick Preference for NXT to allow Wireless (Wifi and BT) searching to be disabled quickly.
- Fixed an issue where '*' to indicate a file was modified was not properly working with "new files" that had never been saved. When you "CTRL-Z" to get back to an empty file the '*' did not go away. And you could get multiple '*' added to the file name.
- Fixed error message prompt "No program to compile and download". This message was generated twice when trying to compile and download without a program open.
- Fixed issue Servo names in disassembly output were being displayed as numeric indices instead of the alphanumeric enum name.
- Fixed issue where CTRL + DELETE is supposed to delete rest of word from the current cursor position. There was a bug if the cursor is currently at the end of line because there was no check for this and tries to access invalid characters beyond the end of line looking for "rest of word".
- Small JoystickDriver.c fix to allow NXT-only communication to use the "WaitForStart" function.

## ROBOTC 3.0 Future Improvements

These features are coming in ROBOTC 3.0, but did not make the cut for the first release. Our plan is to release these features once they have been full tested and implemented. These major updates will be free upgrades for ROBOTC 3.0 users.

- Multi-Robot Tools and Libraries using Xbee Radios for all of ROBOTC's supported platforms.
- VEX I2C Sensors and Integrated Motor Encoder support