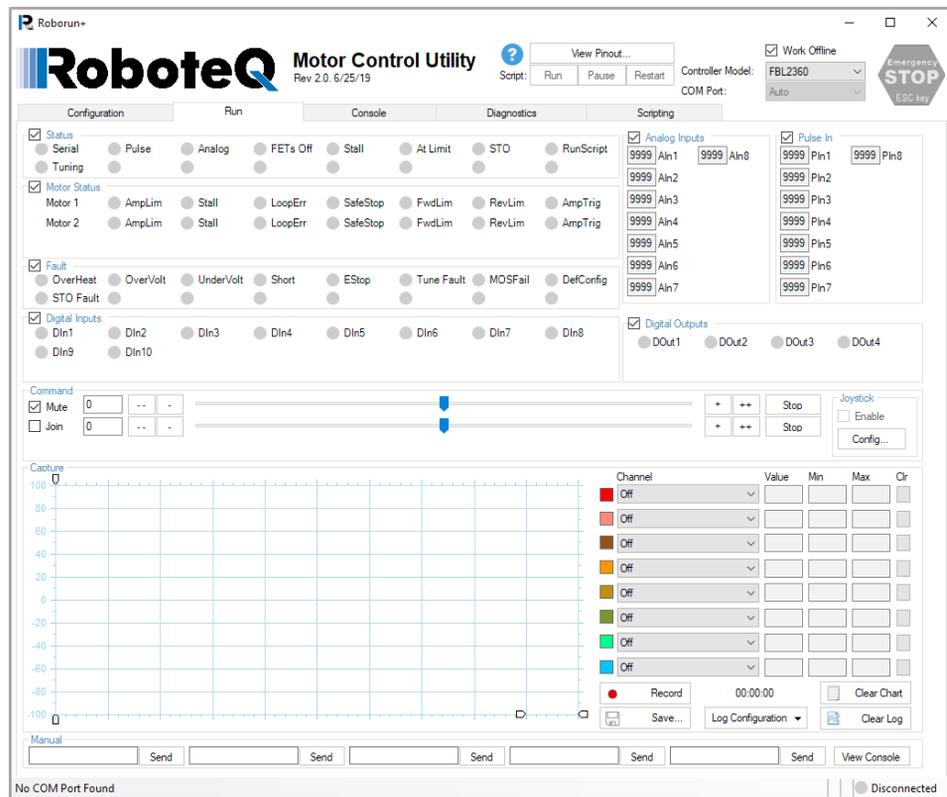


# RoboRun+ PC Utility



## User Manual

V2.0, July 8, 2019

visit [www.roboteq.com](http://www.roboteq.com) to download the latest revision of this manual

©Copyright 2016-2019 Roboteq, Inc

---

## Revision History

Date	Version	Changes
June 27, 2019	1.0	Extracted from main User Manual

The information contained in this manual is believed to be accurate and reliable. However, it may contain errors that were not noticed at the time of publication. Users are expected to perform their own product validation and not rely solely on data contained in this manual.

Revision History .....	2
Introduction.....	3
Refer to the Datasheet for Hardware-Specific Issues.....	3
User Manual Structure and Use.....	3
Introduction.....	7
Installation.....	7
System Requirements .....	7
Installation Steps .....	8
Program Folder Structure.....	9
How to Use.....	10
Header .....	10
Status Bar .....	11
Content Tabs.....	11
Correcting Compilation Errors .....	29
Connecting to Controller .....	30
USB/Serial.....	31
TCP.....	31
Controller's Pinout .....	31
Locking Controller's Configuration .....	32
Locking the Configuration .....	32
Unlocking the Configuration .....	33
Firmware Update .....	33
Update Firmware via COM Port .....	33
Update Firmware via USB .....	34
Update Control Unit Logic (obsolete) .....	37
Update Power Unit Logic (obsolete) .....	37
Motor/Sensor Setup.....	37
Sinusoidal Mode .....	39
Sensorless Mode.....	39
Sensor Linearity Correction .....	39



# Introduction

---

## Refer to the Datasheet for Hardware-Specific Issues

This manual is the companion to your controller's datasheet. All information that is specific to a particular controller model is found in the datasheet. These include:

- Number and types of I/O
- Connectors pin-out
- Wiring diagrams
- Maximum voltage and operating voltage
- Thermal and environmental specifications
- Mechanical drawings and characteristics
- Available storage for scripting
- Battery or/and Motor Amps sensing
- Storage size of user variables to Flash or Battery-backed RAM

---

## User Manual Structure and Use

The user manual discusses issues that are common to all controllers inside a given product family. Except for a few exceptions, the information contained in the manual does not repeat the data that is provided in the datasheets.

## Using the Roborun Configuration Utility

This section describes the features and capabilities of the Roborun PC utility. The utility can be used for setting/changing configurations, operate/monitor the motors and I/O, edit, simulate and run Microbasic scripts, and perform various maintenance functions such as firmware updates.



# Using the Roborun Configuration Utility

---

## Introduction

Roborun+ is a PC utility that allows users configure their Roboteq's motor controllers. Using the utility, you can alter the controller's configuration parameters in addition to monitor the operating state of the controller.

The utility provides means of diagnostics through rich charts and logs. Through the utility you can monitor the commands/responses flow through USB, RS232, or TCP.

The utility provides means of updating the controller's firmware through the UI with few mouse clicks. In addition to that, the utility provides a way for writing, compiling, and uploading user defined script through a WYSWYG script editor integrated in the utility.

The utility is available on the website free of charge.

---

## Installation

The utility can be obtained from the Support page on Roboteq's website at [www.roboteq.com](http://www.roboteq.com). Roborun+ has a setup wizard that guides you through the installation process with few clicks. However, the setup will remove previous version of the utility on installation, it is recommended to remove previous versions before attempting to install new versions.

The controller does not need to be connected to the PC to start the Utility. For installations on older versions of Windows, it may be necessary to install .NET Framework version 3.5. On Windows 10 systems, you may need to enable .net framework version 3.5.

## System Requirements

To run the utility, the following is needed:

- PC compatible computer running Microsoft Windows 7 or higher.
- Microsoft .NET Framework 2.0.
- Microsoft .NET Framework 4.0.

- A USB connector for controllers with USB connectivity.
- If communicating with the controller via RS232, an unused serial communication port on the computer with a 9-pin, female connector for controllers using RS232 communication.

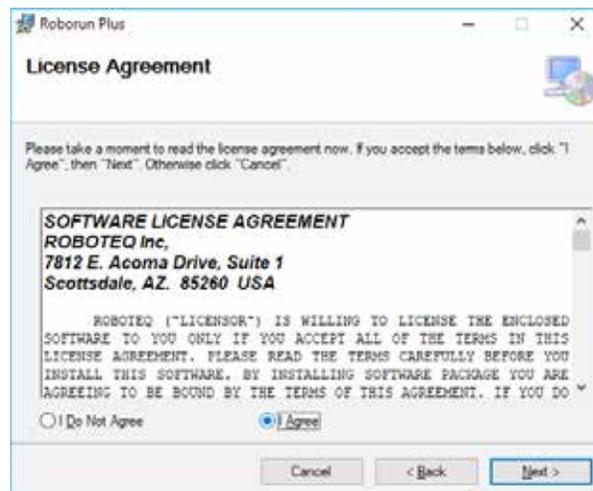
If the PC is not equipped with an RS232 serial port, one may be added using a USB to RS232 converter.

## Installation Steps

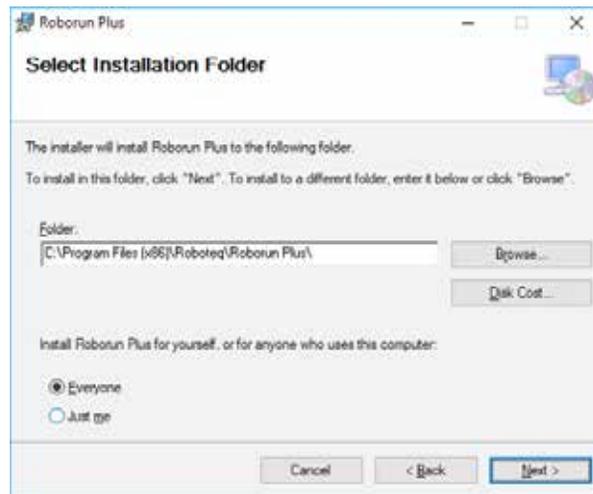
- Run **setup.exe**, on the welcome screen press **Next** button.



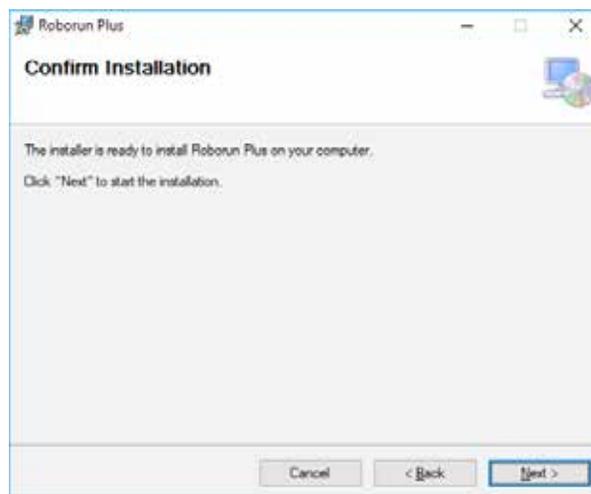
- In License Agreement screen check **"I Agree"** then press **Next** button.



- Choose where you would like to install the application then press **Next** button.



- In the Confirm Installation screen, press **Next** button to proceed with the installation process.



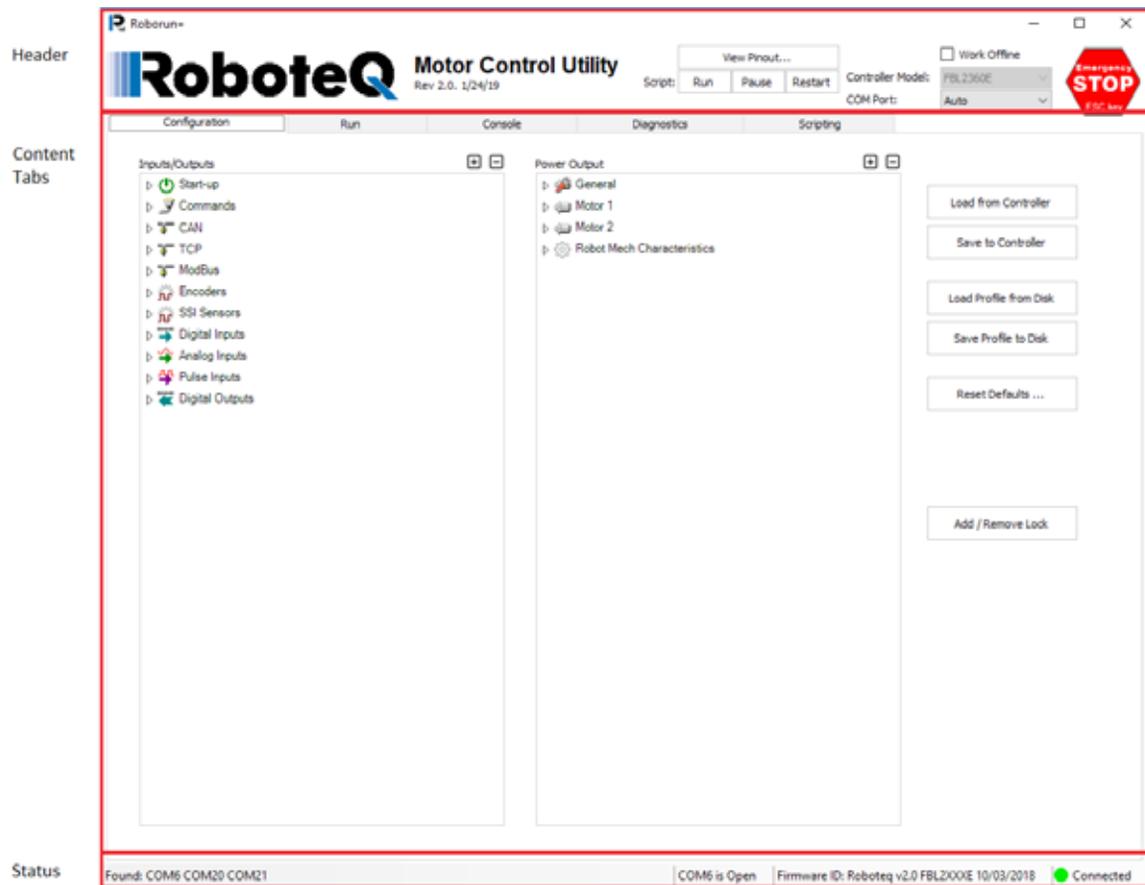
- After installation completes, press **Close** button.
- You can run the program from your Start Menu > Programs > Roboteq or from the shortcut placed on the desktop.

## Program Folder Structure

- **Compiler/doc:** Contains the documentation for MicroBasic scripting.
- **Trees:** Contains configuration for the supported controller models.
- **Utility:** Contains internal utilities used by Roborun+.
- **RoborunPlus.exe:** The Roborun+ executable.
- **changelog.txt:** Contains the changes across all versions.
- **eula.rtf:** The end user license agreement file.

## How to Use

The Roborun+ utility is provided as a tool for easily configuring the Roboteq controller and running it for testing and troubleshooting purposes.



The Roborun+ main screen consists of 3 parts, header, status bar and content tabs. In the following sections we will describe each in details.

## Header

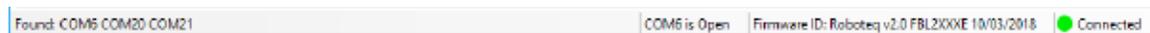


The header is always visible and contains the version number and date in addition to useful information and actions to the currently connected controller.

Item	Description
View Pinout...	Display the pinout connection for the current controller.
Script Run	Causes the controller to start run the Microbasic script previously loaded to the controller.

Item	Description
Script Pause	Causes the controller to pauses the currently running script.
Script Restart	Causes the controller to stop the currently running script and re-executes it for the beginning.
Work Offline	When checked, this causes the utility to disconnect from any connected controllers and allows user to choose the controller model from the model combo box. Working offline is useful for creating/editing configuration profiles without the need to have an actual controller attached to the PC.
Controller Model	Shows the currently connected controller model and allows the user to choose from different models in case of working offline.
COM Port	Allows the user to choose the port at which is the controller connected or make it <b>Auto</b> for the utility to search all ports for controllers. Choosing the port forces the utility to send/receive commands to the controller even if the utility unable to receive the handshake from the controller. This option allows the utility to connect to controllers through <b>TCP</b> for the controllers that supports ethernet connections (see connecting to controller section for details).
Emergency Stop	Caused the controller to enter the <b>Emergency Stop</b> state (for more information, please refer to controller user's manual). Pressing it again causes to release <b>Emergency Stop</b> state.

### Status Bar



The status bar is located at the bottom of the window and is split in 4 areas. From left to right:

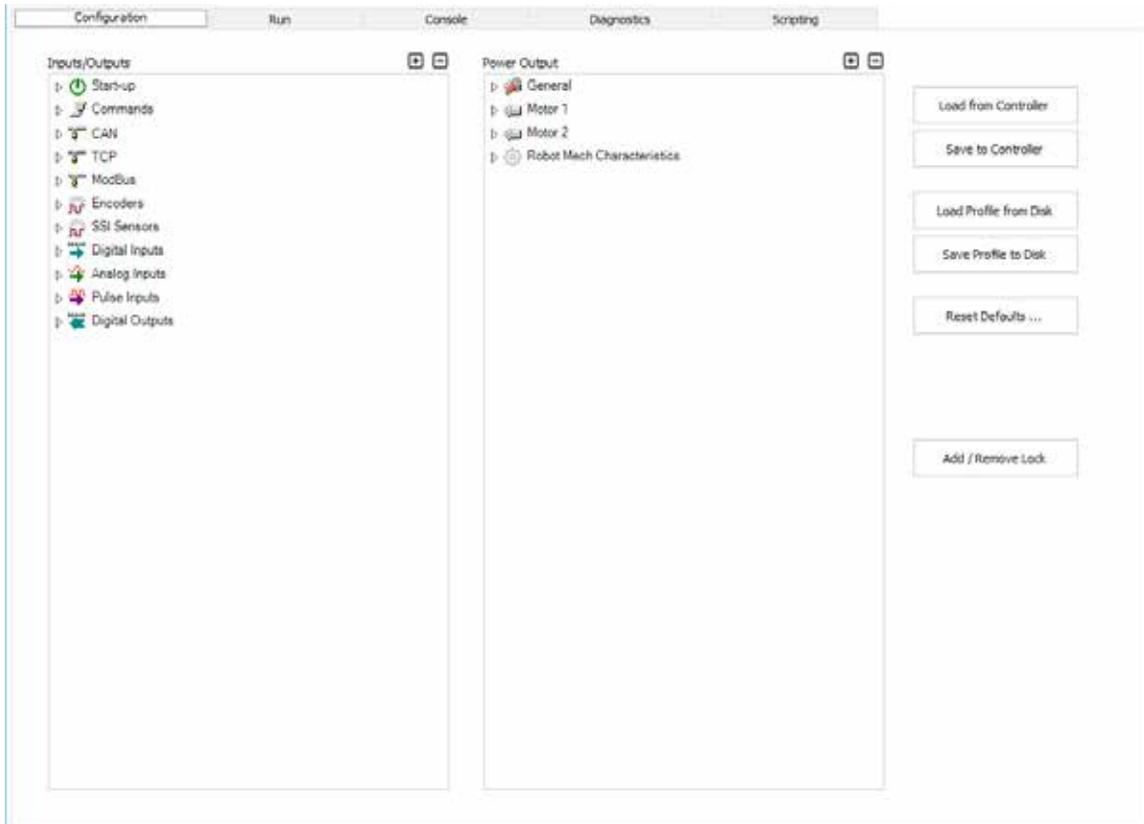
Item	Description
Ports List	Lists the available ports that utility detected on the PC. These ports are searched for the controllers when the COM Port option is <b>Auto</b> .
Port Status	Displays the status of the port being manipulated. "Port Open" indicates that communication with the controller is established.
Firmware ID	Displays the controller model and firmware version number and date as reported by the controller.
Connected/Dis-connected LED	When lit green, it indicates that the communication with the controller is OK.

### Content Tabs

The content tabs allow user to configure, monitor, and diagnose the controller through rich UI.

### 3.1.1 Configuration Tab

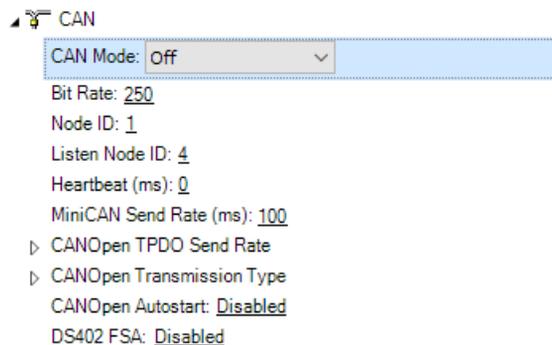
The configuration tab is used to read, modify and write the controller's many possible operating modes. It provides a user-friendly interface for viewing and editing the configuration parameters.



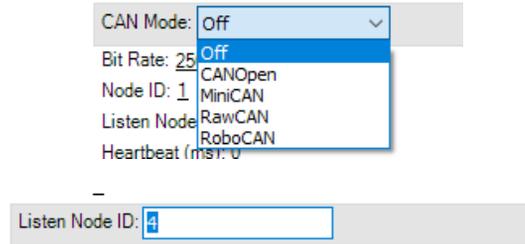
The configuration tab contains two configuration trees: the one on the left deals mostly with the I/O and control signals, while the tree on the right deals with the power output and motor parameters. The exact content and layout of a tree depends on the controller model that is detected.

The trees are, for the most part, self-explanatory and easy to follow.

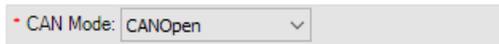
Each node will expand when clicking on the small triangle next to it. When selecting a tree item, the value of that item will show up as an underscored value. Clicking on it enables a menu list or a free-form field that you can select to enter a new configuration value.



By selecting the desired configuration, you will be able to modify it accordingly by selecting the desired value for discrete, or type in the new configuration value for wide ranged values configuration.



Selecting a new value, will cause the configuration node to display a red asterisk mark indicating that this change is not applied to the controller yet.



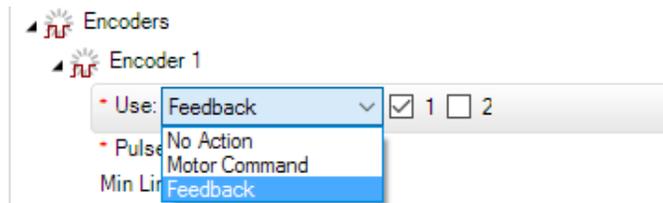
Depending on the node type, values can be entered in one of many forms:

- Numerical
- Boolean (e.g. Enable/Disable)
- Selection List
- Text String

When entering a numerical value, that value is checked against the allowed minimum and maximum range for that parameter. If the entered value is lower than the minimum, then the minimum value will be used instead, if above the maximum, then the maximum value will be used as the entered parameter.

Boolean parameters, such as Enabled/Disabled will appear as a two-state menu list.

Some parameters, like Commands or Actions have the option to apply to one or the other of the motor channels. For this type of parameters, next to the menu list are checkboxes – one for each of the channels. Checking one or the other tells the controller to which channel this input or action should apply.



## Important Notice about Decimals

**The use of Period vs Coma when entering a decimal configuration value depends on the regional settings of Windows. On USA PC's, use periods. On European PC's use comas. If unsure, load the configuration back from the controller after changing and saving. Verify that the value that was stored in the controller is the one that was entered.**

You can use  buttons to expand/collapse all nodes.

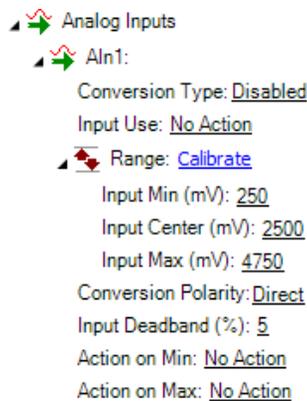
In the right side of this tab, there are actions that can be applied to the configuration:

Item	Description
Load from Controller	Fills out the configuration trees on the left by the actual values that is queried from the controller.
Save to Controller	Save the configurations you altered to the controller.
Load Profile from Disk	Allows the user to pick up a configuration XML file that is previously saved to be applied to the configuration trees.
Save Profile to Disk	Allows user to save the current configuration to an XML file.
Reset Defaults...	Resets the controller's configuration into defaults provided by the firmware.
Add/Remove Lock	Allows the user to enter a password to protect the changes from being accidentally altered or read.

### Automatic Analog and Pulse input Calibration

Analog and Pulse inputs can be configured to have a user-defined minimum, maximum and center range. These parameters can be viewed and edited manually by expanding the Range sub node.

The minimum, maximum and center values can also be captured automatically by clicking on the "**Calibrate**" link.



When clicking on the "**Calibrate**" link, a window pops up that displays a bar showing the live value of that analog or pulse input in real time.

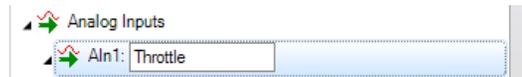
The window contains three cursors that move in relation to the input, capturing the minimum and maximum detected values. It is possible to further manually adjust further these settings by moving the sliders. The Center value will be either the value of the inputs (or the joystick position) at the time when clicking on the "**Done**" button. The Center value can also be automatically computed to be the middle between Min and Max when enabling the "**Auto Center**" checkbox. Clicking on "**Reset**" resets the Min, Max and Center sliders and lets you restart the operation.



After clicking on the “Done” button, the capture values will appear in the Min, Max and Center nodes in the tree with the orange \* next to them, indicating that they have changed but not yet be saved in the controller. At this point, they can be adjusted further manually and saved in the controller.

### Input/Output Labeling

Each analog, digital or pulse input/output, is given default label (e.g. DIN1, AIN2, ...). Alternatively, it is possible to assign or a user defined label name (e.g. Limit1, eStop, ...) to each of these signals. This label will then appear in the Run Tab next to the LED or Value box. The label will also appear in the Pin View window. Custom labels make it much easier to monitor the controller’s activity in the Run tab.



To label an Input or Output, simply select it in the tree. A text field will appear in which you can enter the label name. Beware that while it is possible to enter a long label, names with more than 8 letters will typically appear truncated in the Run tab.

### Configuration Parameters Grouping & Organization

The total number of configuration parameters is quite large. While most system will operate well using the default values, when change is necessary, viewing and editing parameters is made easy thanks to a logical graphical organization of these parameters inside collapsible tree lists.

The configuration tab contains two trees. The left tree includes all parameters that deal with the Analog, Digital, Pulse I/O, encoder and communication. The right tree includes all parameters related to the power drive section. The exact content of the trees changes according to the controller that is attached to the PC.

### Startup Parameters

These menus define the controller’s behavior immediately after startup.



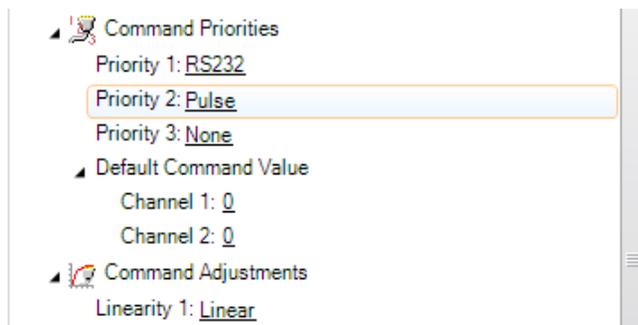
The Script Autostart enables or disables script execution. Make sure that the script is bug free before enabling.

Then a number of Command Safety parameters can also be configured. These are the Watchdog timeout when receiving Serial commands, and the safety ranges for analog commands.

The Telemetry parameter contains the string that is executed whenever controller is first powered up. This parameter is typically composed of a series of real-time queries that the controller automatically and periodically performs. Queries must be separated with the ":" colon character. The string is normally terminated with the command to repeat ("#") followed by the repeated rate in milliseconds. See "TELS - Telemetry String" on Controller's User Manual.

### Commands Parameters

In the commands menu we can set the command priorities, the linearization or exponentiation that must be performed on that input.

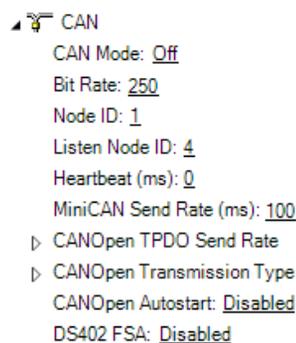


A number of Command Safety parameters can be configured. These are the Watchdog timeout when receiving Serial commands, and the safety ranges for pulse and analog commands.

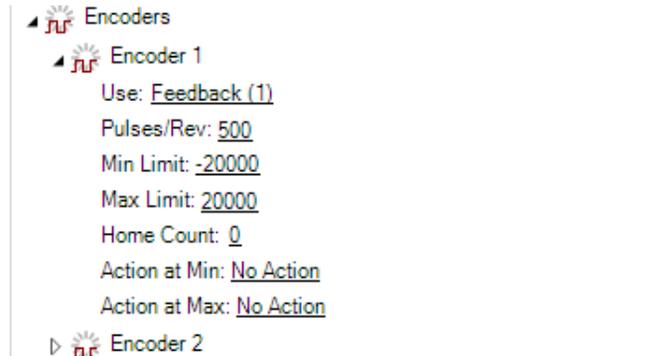


### CAN Communication Parameters

CAN bus node address, bit rate, choice of protocol and other parameter can be set from this set of menus.

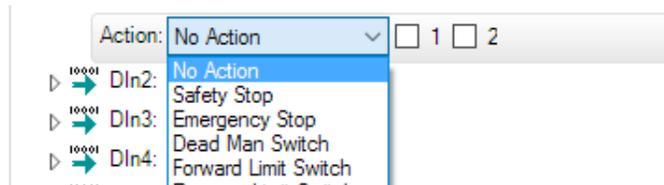


### Encoder Parameters

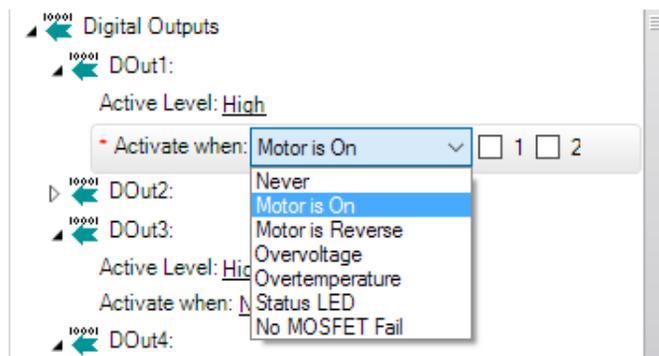


### Digital Input and Output Parameters

For Digital inputs, you can set the Active Level and select which action input should cause when it is activated and on which motor channel that action should apply.

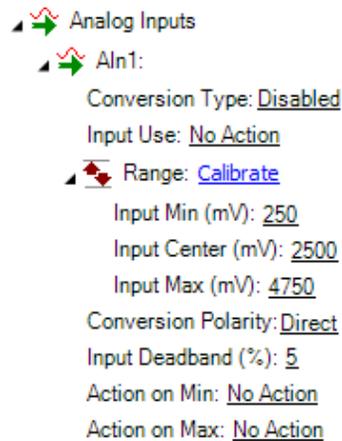


For Digital Output, you can set the Active Level and the trigger source that will activate the Output.



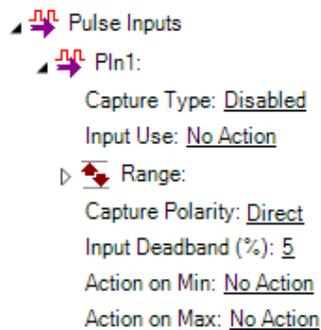
### Analog Input Parameters

For Analog inputs, all the parameters that can be selected include the enabling and conversion type what this input should be used for and for which channel the input range limits the deadband and which actions to perform when the minimum or maximum values are reached.



### Pulse Input Parameters

For Pulse inputs, the tree lets us enable that input and select what it is used for and what type of capture it is to make. The range, deadband and actions to take on when Min and Max are reached is also selectable.

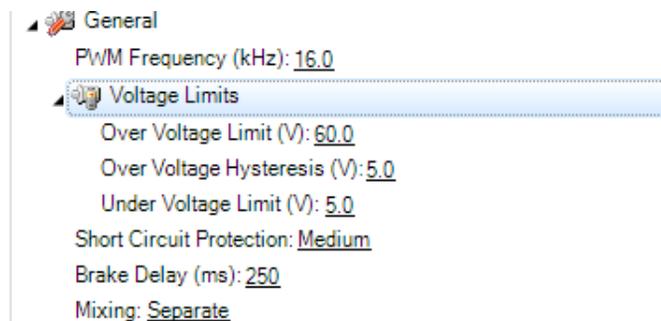


### Power Output Parameters

The on the right side of the configuration screen are the parameters that related to the motor driver and power stage of the controller.

### General Settings

There is one tree for setting parameters that apply to all channels of the controller. These are: the PWM Frequency, the low and high side Voltage Limits, the Short Circuit Protection and the mixed mode.



### Motor Parameters

The parameters for each motor are typically duplicated so that they can be set separately for each motor.

The Motor Configuration group contains menus for configuring the motor’s characteristics, and especially these of brushless motors.

The Motor Output group contains menus for setting Amps limits, Acceleration/Deceleration, operating modes, and Control Loop gains and other operating parameters

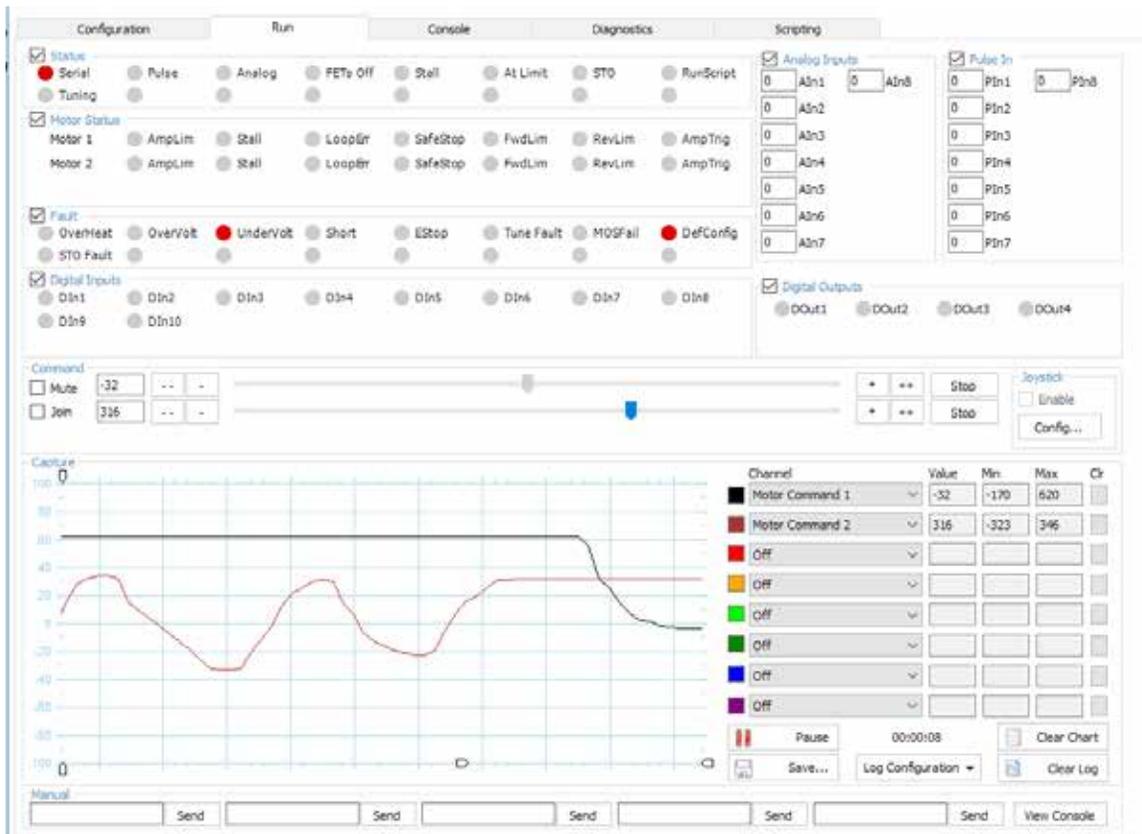
Details on each of the possible configurations can be found in the Controller’s User Manual.

### Run Tab

The Run tab lets you exercise the motors and visualize all the inputs and outputs of the controller.

A powerful chart recorder is provided to plot real-time controller parameters on the PC, and/or log to a file for later analysis.

This tab allows user to monitor the controller status and inputs/outputs in addition to operating parameters.



Each group of monitored parameters can be disabled with a checkbox at the upper left corner of their frame. By default, all are enabled. Disabling one or more will increase the capture resolution in the chart and log of the remaining ones.

## Status and Fault Monitoring

Status LEDs show the real-time state of key operating flags. The meaning of each LED is displayed next to it and can vary from one controller to another.

The Fault LEDs indicate all fault conditions. Any one LED that is lit will cause the controller to disable the power to all motor output channels. The meaning of each LED is displayed next to it and can vary from one controller to another.

The Def Config Fault LED indicates that an invalid configuration is read from the controller and the controller has reverted to its factory default configuration. This would be an extremely unlikely occurrence, but if it happens, reload your custom configuration and verify that the new configuration is not lost when restarting the controller a few times. If the controller loses its configuration, this means it is faulty that it should not be used.

The DefConfig Fault LED will also turn on the first time the controller is restarted after a new firmware release has been installed and default configuration first reloaded.



## Digital, Analog and Pulse Input Monitoring

The status of Digital inputs and the value Analog and Pulse can be monitored in real-time. Analog and Pulse inputs will update only if the selected channel is enabled. The labels for the digital inputs, digital outputs, analog inputs and pulse inputs can be made to take the value that has been entered in the configuration tree as described in Input/Output Labeling. Using a nickname for that signal makes it easier to monitor that information.

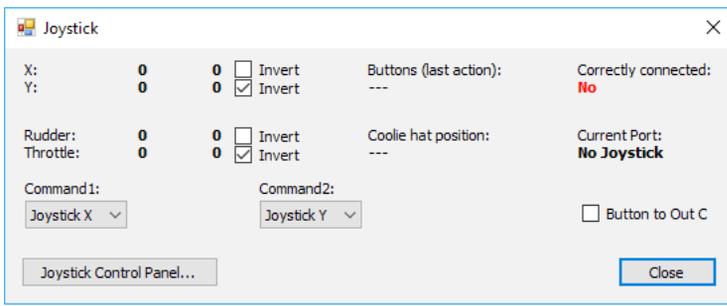
## Digital Output Activation and Monitoring

The Digital output LEDs reflect the actual state of each of the controller's Output. If an output is not changed by the controller using one of the available automatic Output Triggers, clicking on the LED will cause the selected output to toggle On and Off.

## Applying Motor Commands

The command sliders will cause the command value to be applied to the controller.

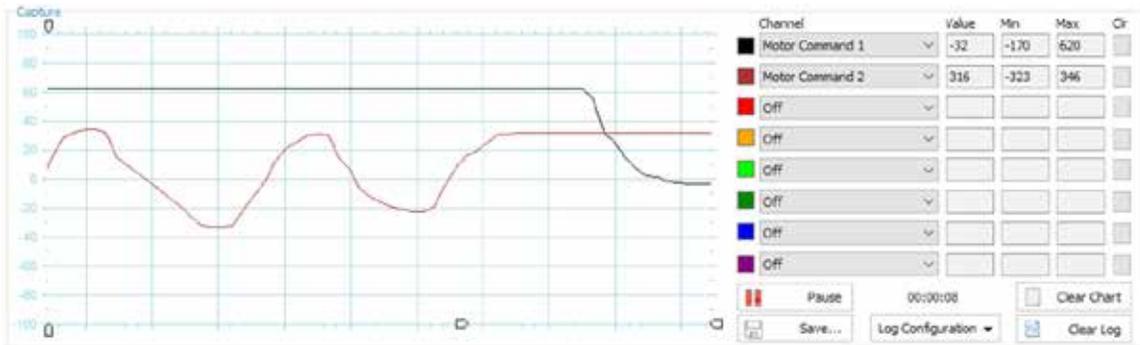


Item	Description
Mute Checkbox	Prevents the utility from sending motor commands.
Join Checkbox	In case of multi-channel controllers, the checkbox allows the user to send the same motor command value for both motors.
Textbox	Allows the user to enter the value manually. The motor commands value is ranged from -1000 to 1000.
	Decreases the motor command by 10.
	Decreases the motor command by 1.
	Increases the motor command by 1.
	Increases the motor command by 10.
Stop Button	Set the motor command to 0.
Joystick	<p>You can enable/disable using of joystick to send motor commands. You can press the <b>Configure</b> button to configure the joystick.</p> 

### Using the Chart Recorder

A powerful chart recorder is provided for real-time capture and plotting of operating parameters. This chart can display up to eight operating parameters at the same time. Each of the chart's channels has a pull-down menu that shows all of the operating parameters that can be viewed and plotted. The colors can be changed by clicking on the color icon and selecting another color.

When selecting a parameter to display, this parameter will appear in the chart and change in real-time. The three boxes show a numerical representation of the actual value and the Min and Max value reached by this input. Clicking on the "Clear" button for that channel resets the Min and Max. The chart can be paused or it can be cleared and the recorded values can be saved in an Excel format for later analysis.



“Handles” on the left vertical axis may be used to zoom in a particular vertical range. Similar handles on the horizontal axis can be used to change the scrolling speed of the chart.

Item	Description								
Channel	Allows user to choose channels to be plotted on the chart area.								
Value	The current value of selected channel.								
Min	The minimum value captured so far.								
Max	The maximum value captures so far.								
Clr	Clear the minimum and maximum and re-evaluate.								
Record/Pause	Start/Pause chart capturing.								
Save	Save the captured values in one of the following formats: <ul style="list-style-type: none"> <li>• Tab delimited.</li> <li>• Comma Separated Values (CSV).</li> <li>• Excel file.</li> </ul>								
Clear Chart	Clear the chart and start over.								
Clear Log	Clear the values saved so far.								
Timer	Displays the duration of the capturing.								
Log Configuration	Clicking this button opens the following menu: <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <table border="0"> <tr><td>Save Config</td><td>▶</td></tr> <tr><td>Load Config</td><td>▶</td></tr> <tr><td>Export/Import</td><td>▶</td></tr> <tr><td>Delete Config</td><td>▶</td></tr> </table> </div> This menu allows user to Save/Load/Export/Delete configuration for later use.	Save Config	▶	Load Config	▶	Export/Import	▶	Delete Config	▶
Save Config	▶								
Load Config	▶								
Export/Import	▶								
Delete Config	▶								

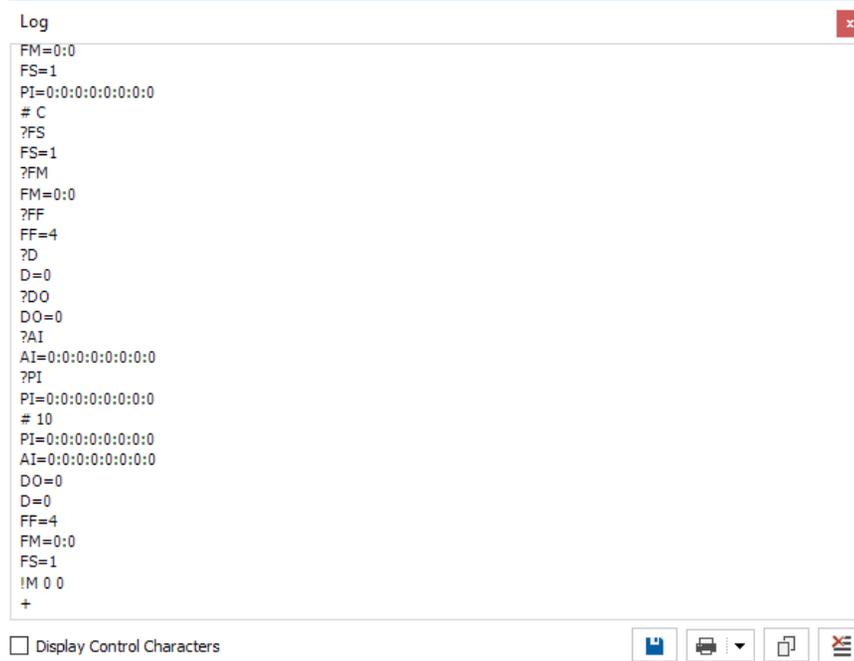
### Manual Commands

This allows the user to issue commands/queries to the controller in addition to view the controller responses.



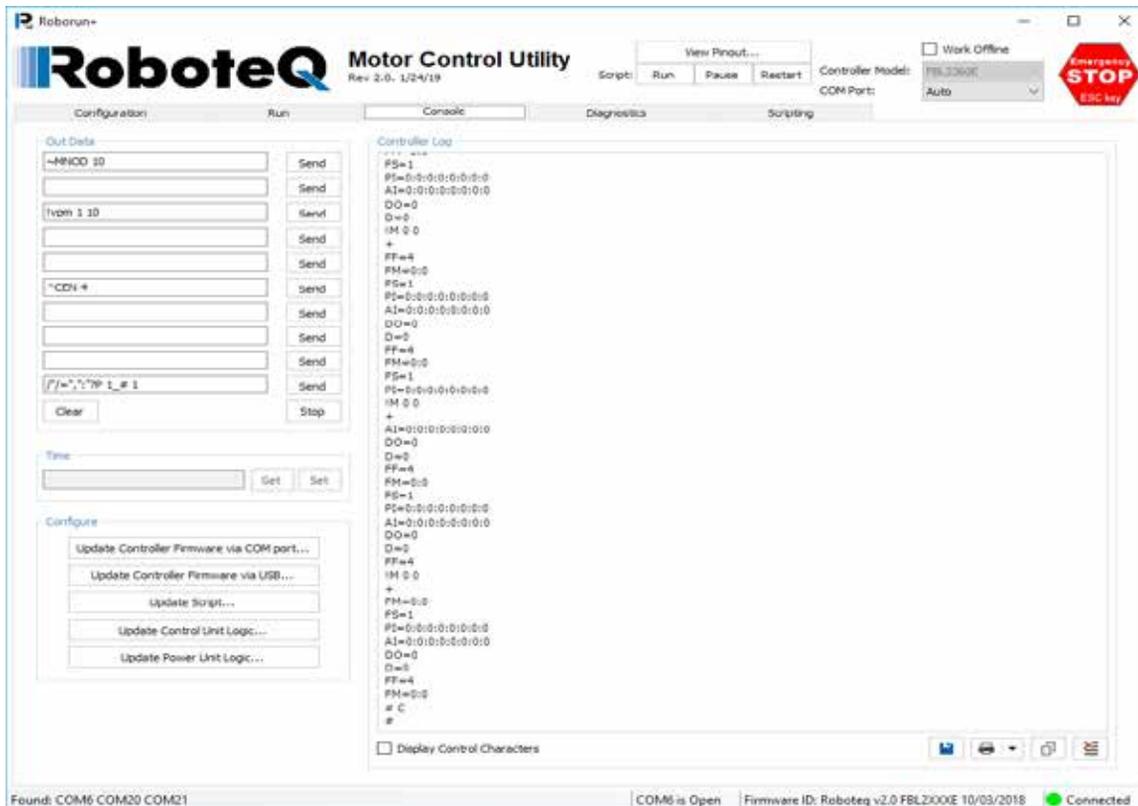
The user can enter the command/query in one of the five textboxes and press **Enter** key or **Send** button to issue the command/query to the controller.

The user can also press **View Console** button at any time to display the controller’s log.



### Console Tab

The console tab is useful for practicing low-level commands and viewing the raw data exchanged by the controller and the PC. The Console tab also contains the buttons for performing field updates of the controller.



## Out Data

The console tab allows you to send low-level commands and view the raw controller responses. Ten text fields are provided in which you can type commands and send them in any sequence by clicking on the respective "Send" button or pressing **Enter** key. All the traffic that is exchanged by the controller and the PC is logged in the console box on the right. It is then possible to copy that information and paste it into a word processor or an Excel spreadsheet for further analysis.

The **Stop** button sends the "#" command to the controller and will stop the automatic query updating if it is currently active.

The **Clear** button clears all the textboxes at once.

## Time

In some controller models, there is onboard clock. Pressing the **Get** button, the utility reads the time from the controller's onboard clock while pressing the **Set** button stores the current PC time into the controller.

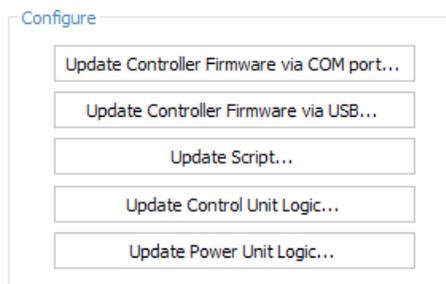
## Controller's Log

Displays the response from the controller this includes the prints from the MicroBasic script running.

Item	Description
Display Control Characters	Checking this option causes the utility to display control characters (non-printable characters) in a readable form.
	Saves the controller's log to a text file.
	Pressing the printer image, causes to print the controller's log to the default printer with the default configuration. Pressing the dropdown allows user to choose the printer and configure the page setup.
	Copies the controller's log into clipboard.
	Clears the current log and start over.

## Configure

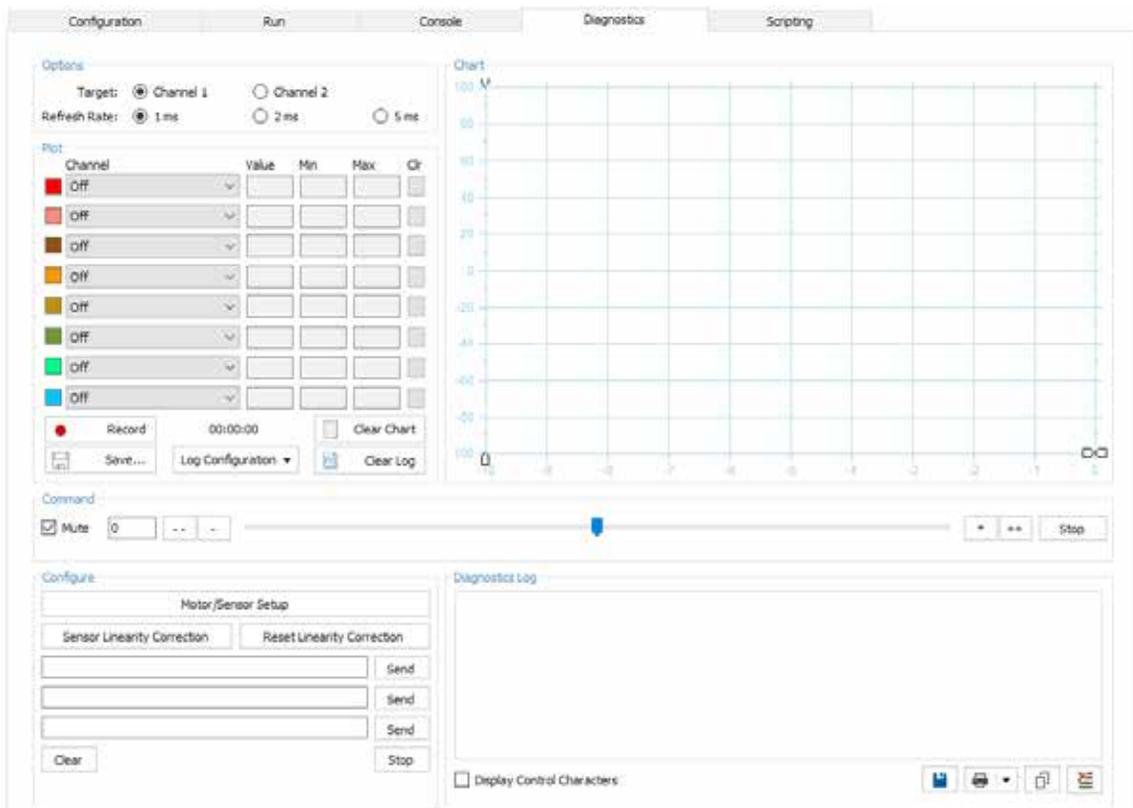
This allows user to update the controller's firmware and logic in addition to uploading a compiled Microbasic script file.



Item	Description
Update Controller Firmware via COM port	This allows user to update the controller's firmware while connected through serial communication port. See <b>Firmware Update</b> section for details.
Update Controller Firmware via USB.	This allows user to update the controller's firmware while connected through USB. See <b>Firmware Update</b> section for details.
Update Script	This allows user to update a compiled MicroBasic script file previously generated by the utility. See <b>Scripting Tab</b> section for details.
Update Control Unit Logic	This allows user to update the control unit logic. See <b>Firmware Update</b> section for details.
Update Power Unit Logic	This allows user to update the power unit logic. See <b>Firmware Update</b> section for details.

### Diagnostics Tab

In some controller versions with firmware v2.0+ the utility has additional diagnostics tab to diagnose the controller with high speed data transfer rate up to 1 millisecond refresh rate.



### Options

This allows user to choose which channel to diagnose also to configure how often the controller feeds information to the utility.

Options

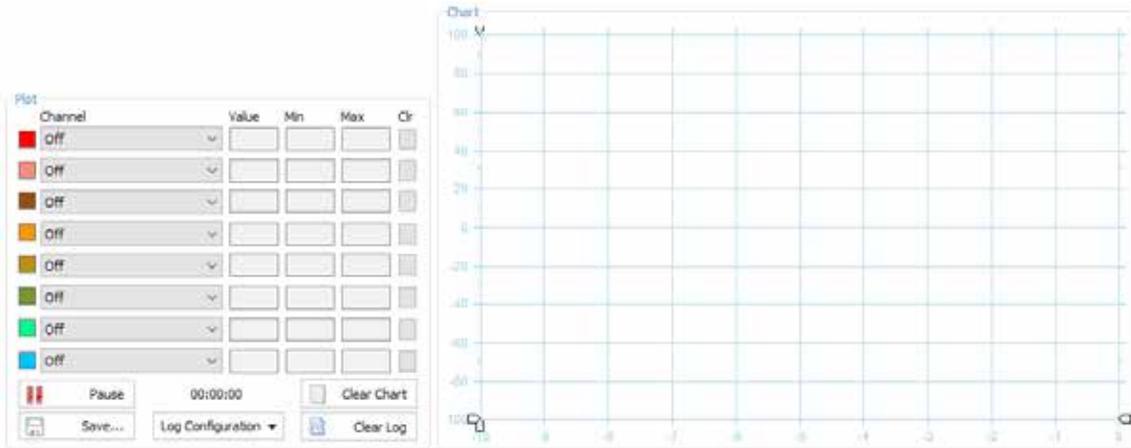
Target:  Channel 1     Channel 2

Refresh Rate:  1 ms     2 ms     5 ms

It is better to diagnose the controller with a refresh rate of 1 millisecond, however using 2 and 5 milliseconds option helps in diagnose controllers on slower connections.

### Plot/Chart

Allow user to choose the diagnose items to be captured and plotted to the chart.



Please refer to the **Capture** part under **Run Tab** for details.

### Command

Allows user to issue commands to motor at the selected channel.



Please refer to the **Motor Commands** part under **Run Tab** for details.

### Configuration

Allows user to manually configure the controller by giving the ability to issue commands/queries using the available 3 textboxes.

Configure

Motor/Sensor Setup

Sensor Linearity Correction    Reset Linearity Correction

Send

Send

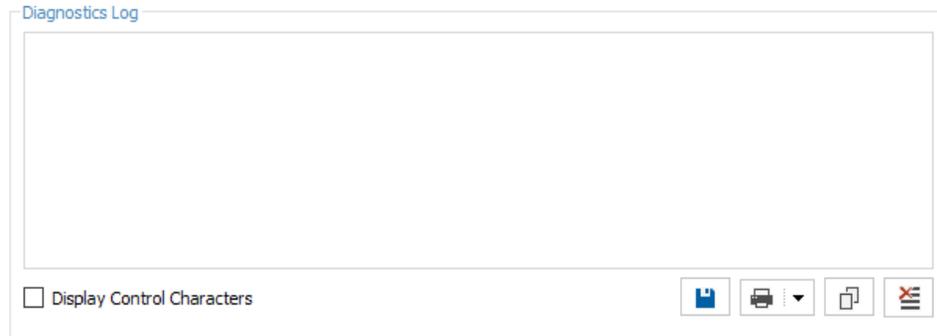
Send

Clear    Stop

User can also make motor/sensor setup in addition to linearity correction. Please refer to the **Motor/Sensor Setup** section for details.

### Diagnostics Log

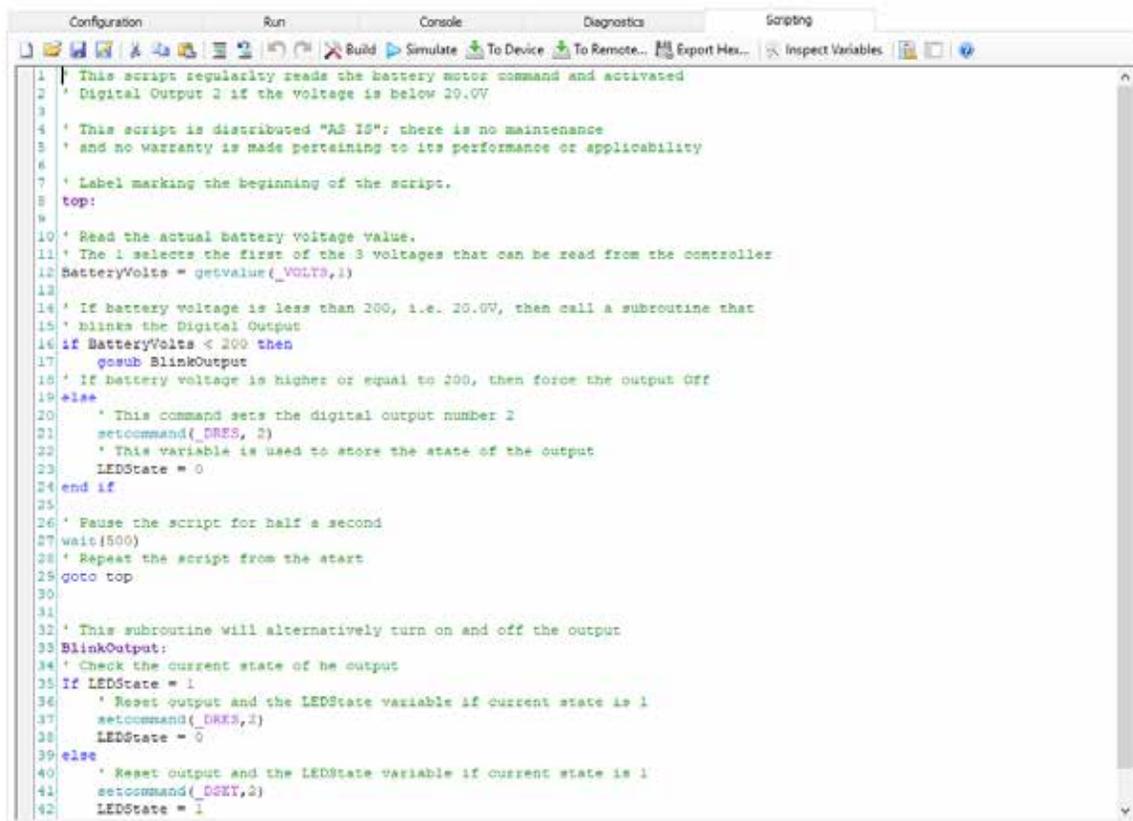
Allows user to monitor the log issued by the controller without the need to go to **Console Tab**.

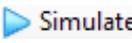
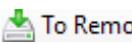


Please refer to the **Controller's Log** part under **Console Tab** for details.

### Scripting Tab

One of the controller's most powerful and innovative features is the ability for the user to write programs that are permanently saved into, and run from the controller's Flash Memory. This capability is the equivalent of combining the motor controller functionality and this of a PLC or Single Board Computer directly into the controller. The scripting tab is used to write, simulate, and download custom scripts to the controller.

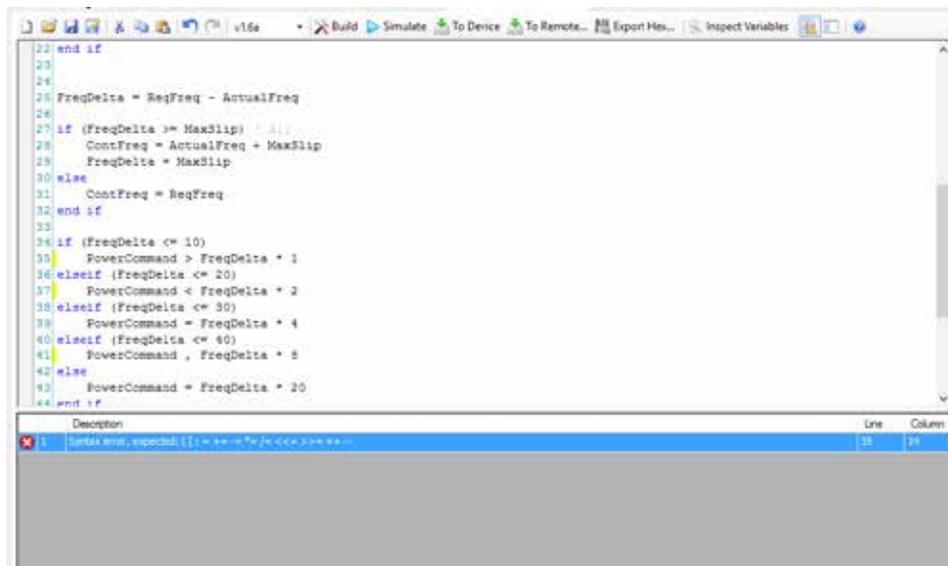


Item	Description
	Create a new script file.
	Open an existing script file.
	Save the current script to disk.
	Save the current script to a different file.
	Cut the current selection.
	Copy the current selection.
	Paste the contents of the clipboard.
	Comment out the selected code.
	Uncomment the selected code.
	Undo the previous change.
	Redo the previous undo.
 Build	This will cause the source code to be immediately interpreted in low level instructions that are understandable by the controller. A window then pops up showing the result of the translation. The code is not downloaded into the controller. This command is generally used to check whether the code you are typing is grammatically correct or not needed. It may be also used to see how many bytes will be taken by the script inside the controller's flash.
 Simulate	This will cause the source code to be interpreted and run in simulation mode on the PC. This function is useful for simplifying script development and debug. The simulator will operate identically to the real controller except for all commands that normally read or write controller configuration and operation data. For these commands, the simulated program will prompt the programmer for values to be entered manually, or output data to the console.
 To Device	This will cause the source code to be immediately interpreted in low level instructions that are understandable by the controller. If no errors are found during the translation, the code is automatically transferred in the controller's flash memory where it is then ready for execution.  <b>Note:</b> For safety, you need to make sure that the connected motors to the controller are not running during downloading scripts to device.
 To Remote...	This will cause the source code to be interpreted and downloaded to a remote controller on a RoboCAN network. After clicking a pop-up window will list in a pull-down menu all controllers found alive on the CAN network. Select the node you wish the script to be downloaded to.  For more details about RoboCAN network please refer to <b>Roboteq Controllers User's Manual</b> .  <b>Note:</b> For safety, you need to make sure that the connected motors to the controller are not running during downloading scripts to device.

Item	Description
Export Hex...	Used to save the compiled code of a given script. This way a script can be loaded in a controller without giving away the source code. The Hex file can then be loaded from the Console tab using the Update Script button.
Inspect Variables	It is a kind of debugging feature that enables the user to see real-time values of variables while the script is running. The steps for debugging are the following: Write the code and download it to controller using <b>Download To Device</b> feature. From the utility header, run the script by pressing the <b>Run</b> button. While the script is running, press <b>Inspect Variables</b> button to start variables inspection. While in variable inspection mode, you can hover on any variable to get its value.
	This button is to show/hide compilation errors/warnings.
	This button is to show/hide the syntax tree for debugging purposes.
	Shows a local version of the MicroBasic reference. For latest documentation, please refer to <b>MicroBasic</b> reference at website.

### Correcting Compilation Errors

When building or trying to download a script that contains errors, the compile errors will be listed in the bottom half of the window. Double clicking on the error will move the cursor to the place in the source code where the error was found.



Beware that not all programming errors are detected. Be especially careful with variable names. Use the option explicit directive to enforce variable declarations. Beware not to mix SetConfig() and SetCommand() when changing a configuration or a com-

mand. A faulty script can cause the controller to crash. Enable the scripting Auto Start configuration only on known working scripts.

### Executing Scripts

Scripts are not automatically executed after the transfer. To execute manually, you must Run or Restart buttons that are in the Utility's header. Alternatively, click on the Console tab and send the !r command via the console. Unless a script includes print statements, it will run silently with no visible signs in the console. Clicking on !r 0 will stop a script, !r or !r 1 will resume a stopped script. !r 2 will clear all variables and restart a script. The Run-script LED in the Run tab will be on when script is running.

Executing a script on a remote controller on a CANbus network using the RoboCAN protocol is done using the @nn!r command, where nn is the remote node address in hex format.

### Debugging Scripts

A number of techniques can be used to debug a script that is not behaving as expected. You can view the value of variable in real time during program execution by clicking on the Inspect Variables button. Then hover the mouse over a variable in the program listing. The variable value will be read and displayed at the mouse location. The variable value is read only once when first hovering over the variable. To read the value again, move the mouse away and return over that variable.

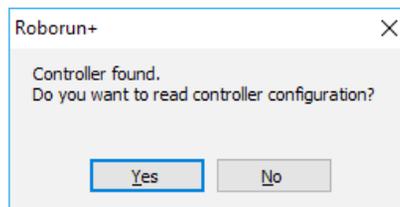
Embedding print statements is another common technique. Place print statements in specific places of the script to verify that a given part of the code gets executed. Print the value of variables you wish to see. To avoid large data dumps on the screen, add code to print conditionally, for example when a variable changes or reaches a given value range.

---

## Connecting to Controller

You can connect to the controller through serial communication (USB or RS232) or through TCP connection for models that supports TCP.

When the utility detects the controllers, it pops up asking for reading controller's configuration.



Choosing **Yes** causes the utility to query for the controller's configuration and updates the utility's configuration trees accordingly.

It is highly recommended to load the controller's configurations first before attempting to make any changes in order to make sure that the configurations are in sync with the controller.

**Note:** For safety, you need to make sure that the connected motors to the controller are not running during loading/saving of the configuration from/to device.

## USB/Serial

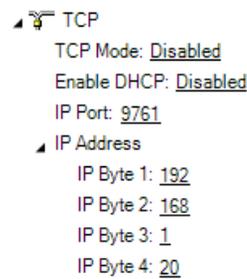
The utility checks for controllers on all available COM ports when the **COM Port** options is set to **Auto**.

If the utility cannot find the controller, you can manually choose the port from the **COM Port** list. Choosing the COM port causes the utility to not handshakes with the controller and this is suitable when there are issues with the controller.

## TCP

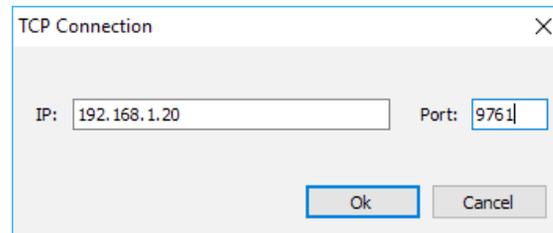
In controllers that supports Ethernet, you connect to the controller using TCP connection. In order to do that, you should know the IP address and communication port of the controller and make sure that TCP is enabled in the controller.

The TCP configuration could be set using the utility from the configuration tab:



In order to connect to the controller through TCP:

- Choose **TCP** from **COM Port** drop down list.
- Type in the IP address and port:



- Press **Ok** button.

## Controller's Pinout

In order to know the pinout of the controller, you can simply press **View Pinout** button while the controller is connected or the model is selected in **Work Offline** mode. The pinout form pops up with each connector configuration.

**Note:** Some connectors are upside down in the pinout form in order to retain pin ordering, so rely only on the pin number in the connection.

**R** FBL-Type Pinout

25 Pins 10 Pins

Pin	15	16	17	18	19	20	21	22	23	24
Digital In	DIN1	DIN2	DIN4				DIN5	DIN6		DIN7
Pulse In	RC1	RC2	RC4				RC5	RC6		RC7
Ana In	ANA1	ANA2	ANA4				ANA5	ANA6		ANA7
Sin/Cos In										ASIN2
Digital Out				DOUT1	DOUT3					
Encoder							Enc2A	Enc2B		Enc1A
Com						CANH			RS485+	

Pin	2	3	4	6	7	8	9	10	11	12
Digital In			DIN3				DIN9	DIN10		DIN8
Pulse In			RC3							RC8
Ana In			ANA3							ANA8
Sin/Cos In							ASIN1	ACOS1		ACOS2
Digital Out				DOUT2	DOUT4					
Encoder										Enc1B
Com	TxData	RxDData				CANL			RS485-	

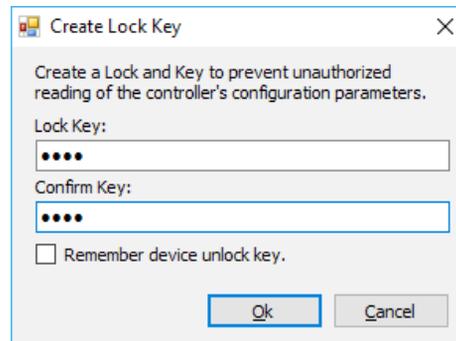
For most recent configuration for each model, please refer to product’s datasheet on our website.

## Locking Controller’s Configuration

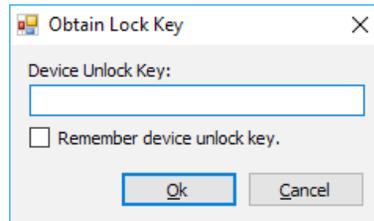
In order to make sure that the configuration is retained on the controller without accidentally changes, the controllers have a configuration locking/unlocking feature.

### Locking the Configuration

- Make sure the controller is connected to the utility.
- From utility’s configuration tab, press **Add/Remove Lock** button.
- The **Create Lock Key** dialog appears.

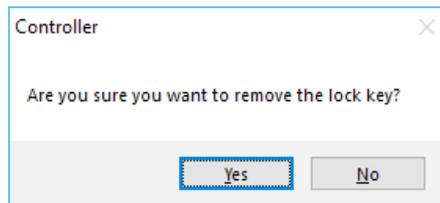


- Type in an **integer number** representing your key and the same in the **Confirm Key** box and press **Ok** button.
- Next time, when you try to load the configuration from the utility you will be asked to enter the configuration key.

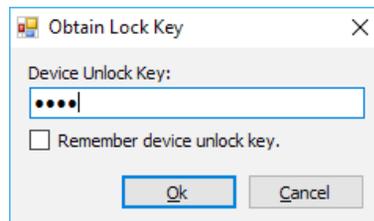


### Unlocking the Configuration

- Make sure the controller is connected to the utility.
- From utility's configuration tab, press **Add/Remove Lock** button.
- A configuration message appears, press **Yes** button.



- A dialog appears asking for the key. Type in the key and press **Ok** button.




---

## Firmware Update

It is recommended to always have the latest firmware in order to be up to date with new features and issue fixing.

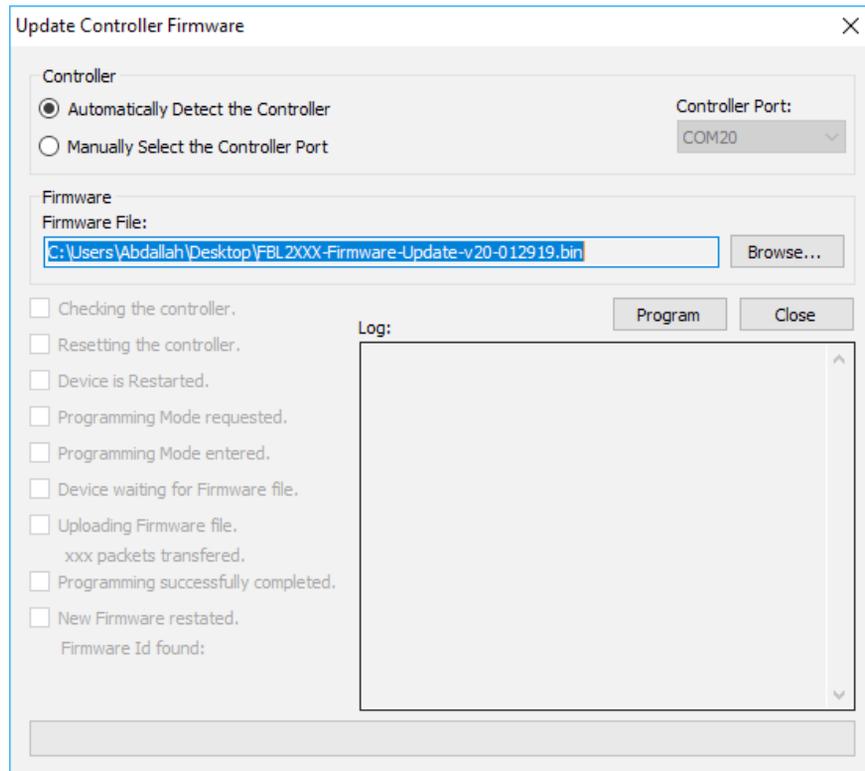
You can visit our website for latest firmware release.

### Update Firmware via COM Port

This method is used to update the controller's firmware when it is connected through RS232. You need to have the **.bin** or **.zbin** firmware file from our website.

In order to go with the update, make sure that the controller is connected through RS232 and all the connected motors is not running.

Go to the **Console Tab** and press **Update Controller Firmware via COM Port** button, the following dialog appears.



Choose your firmware file and press **Program** button.

After the controller is updated, it will be restarted. Press the **Close** button to return back to the utility.

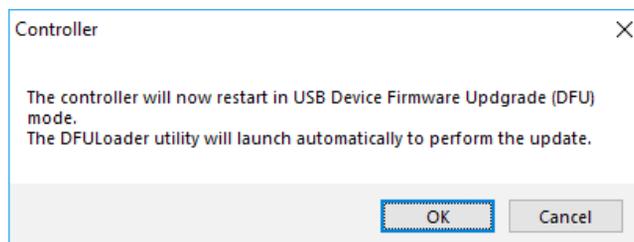
Make sure that the new firmware version is reported in the **Status Bar**.

## Update Firmware via USB

This method is used when the controller is connected through **USB**. This method uses **DFU** (Device Firmware Update) method to do the update.

You need to have the **.dfu** or **.zdfu** firmware file in order to be able to update.

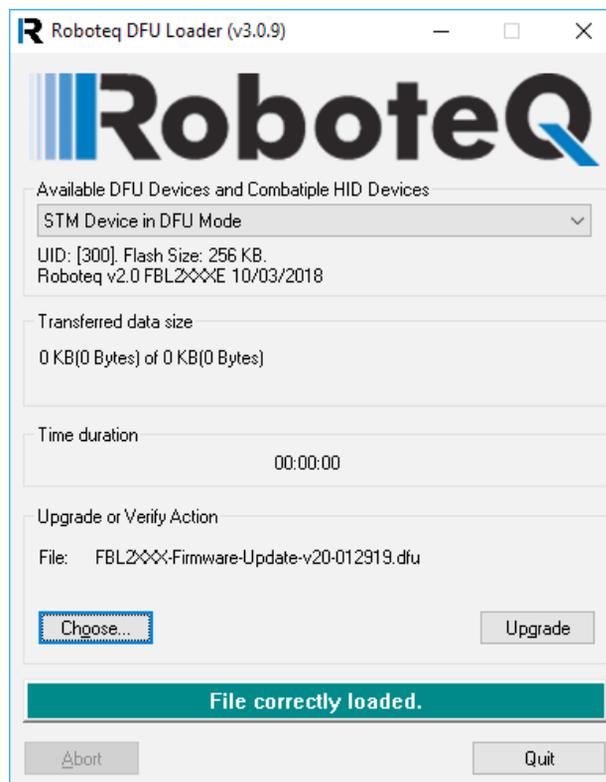
In order to do the update, go to the **Console Tab** and press **Update Controller Firmware via USB**. The following confirmation message will popup, you need to press **OK**.



The device will be restarted in **DFU** mode and the **DFU** Loader utility will open.

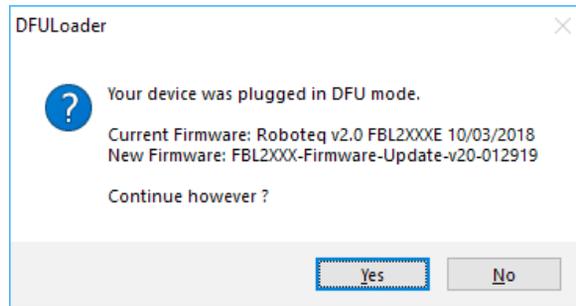


Make sure that the device is selected in the combo box. Press Choose and point to the firmware file.

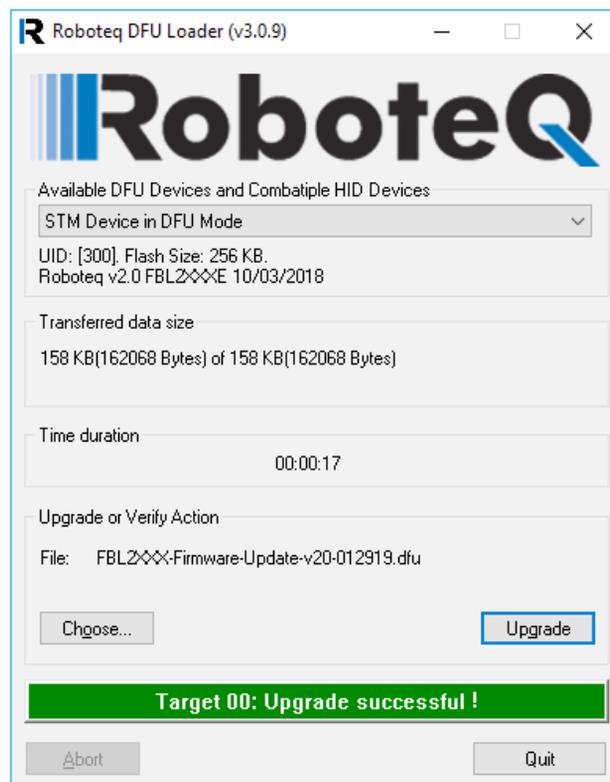


After locating the firmware file, press **Upgrade** button.

You may have a confirmation message to ensure you are using the correct firmware file, make sure you have the correct firmware file and press **Yes** button to continue.



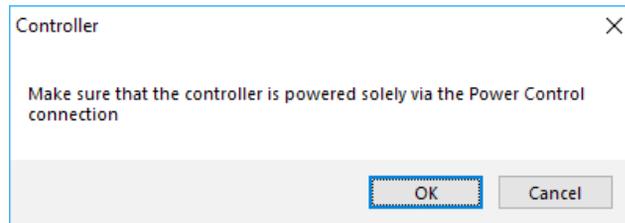
After the firmware update, press **Quit** button in order to exit the **DFU** mode.



The device will be restarted, so you need to go back to the utility and make sure the controller reports the new firmware in the **Status Bar**.

### Update Control Unit Logic (obsolete)

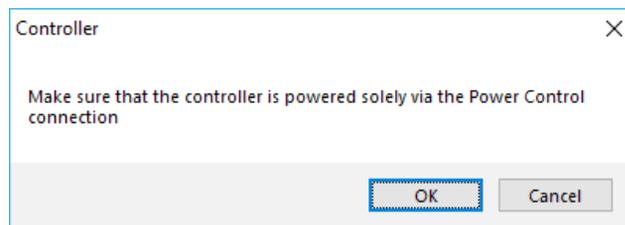
There are boards that have CPLD used for the controlling part of the motor controller. With this button one can update the binary of the CPLD.



The update should be done only via RS232, in a testbench environment, and only power control should be connected to the controller. Beware this action should be done only with the support of a Roboteq technician, since any mal-function will end up with a unresponsive board.

### Update Power Unit Logic (obsolete)

There are boards that have CPLD used for the powering part of the motor controller. With this button one can update the binary of the CPLD.

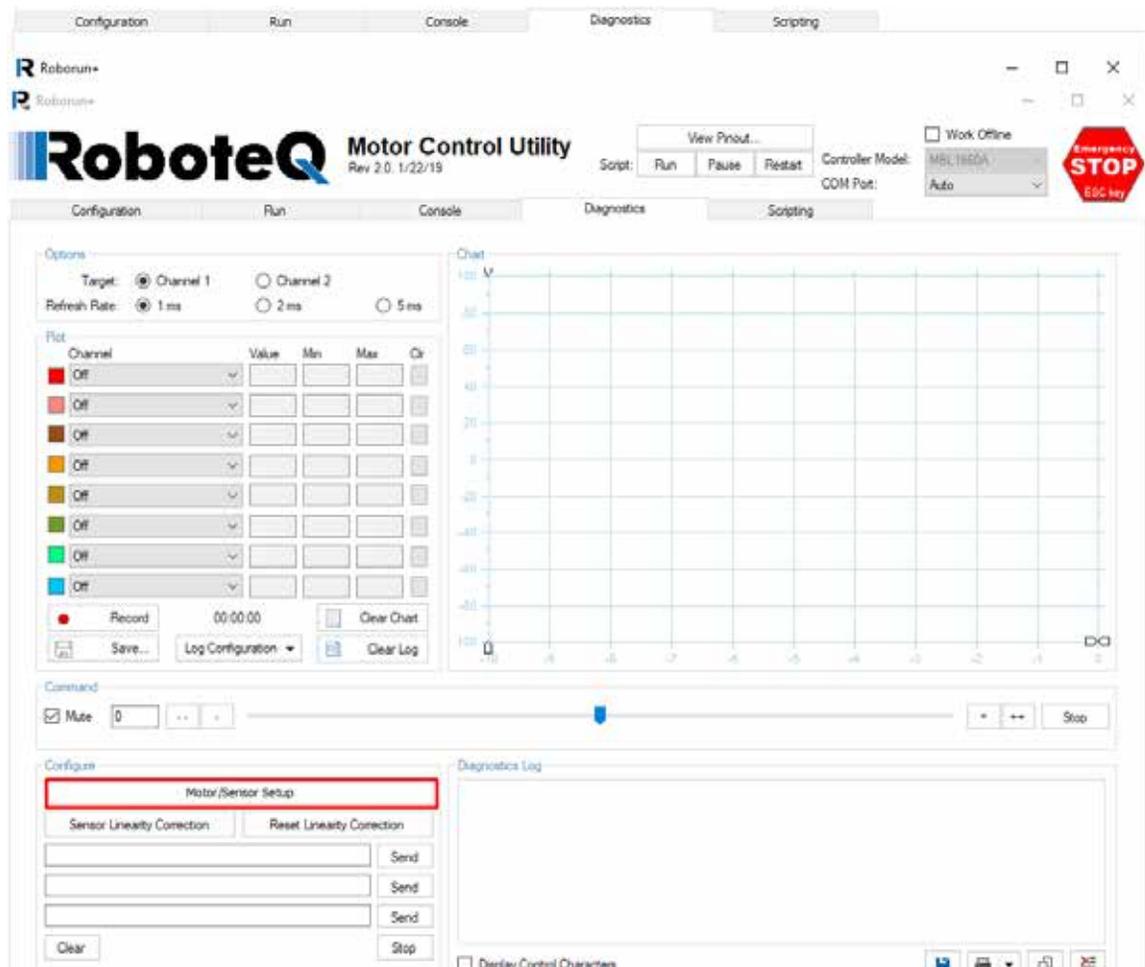


The update should be done only via RS232, in a testbench environment, and only power control should be connected to the controller. Beware this action should be done only with the support of a Roboteq technician, since any mal-function will end up with a unresponsive board.

---

## Motor/Sensor Setup

Brushless motor controllers with firmware v2.0 and higher have the ability to automatically setup the motor and/or the sensor (used for angle estimation) in order to perform optimally in sinusoidal mode. The respective button "Motor/Sensor Setup" in diagnostics tab will initiate the abovementioned process.



This process supports the following commutation modes:

- Hall Sinusoidal Mode
- Hall+Encoder Sinusoidal Mode.
- Encoder Sinusoidal Mode.
- SPI/SSI Sinusoidal Mode.
- Sin/Cos Sinusoidal Mode.
- Resolver Sinusoidal Mode.
- Sensorless Trapezoidal Mode.

Before pressing this button, the following configuration fields need to be set:

- Number of Pole Pairs (according to motor manufacturer).
- Switching Mode (Sinusoidal or Sensorless).
- Reference Seek Power (usually half of the motor's nominal current is enough).
- SinCos/SSI Sensor Poles (according to SSI or Sin/Cos or Resolver sensor manufacturer).
- Sensor Phase Shift Angle (according to Sin/Cos or Resolver sensor manufacturer).
- Sinusoidal Angle Sensor (according to the sensor used).
- Encoder Pulses/Rev (in case of Encoder or Hall+Encoder Sinusoidal mode).
- SPI/SSI Sensor Resolution (in case of SPI/SSI Sinusoidal mode).

## Sinusoidal Mode

After pressing the button, the motor will move back and forth from 1 electrical revolution up to 1 mechanical revolution, depending the sensor configured. If the motor is not moving smoothly try to increase the Reference Seek Power.

The main tasks of the process are:

- To determine the hall sensor alignment and adjust.
- To determine the Sin/Cos or Resolver sensor range and adjust.
- To determine where the zero angle is.
- To align motor and sensor directions.

During the process, the following configuration fields will be modified in order to achieve optimal performance:

- Motor Pole Pairs (only the sign).
- Swap Windings.
- Angle Zero Adjust.
- Hall Sensor Angle Table.
- Sine/Cosine Calibration Values.
- Encoder Pulses/Rev (only the sign).
- SPI/SSI Sensor Resolution (only the sign).

## Sensorless Mode

In case of Sensorless Mode only the Number of Pole Pairs and the Switching Mode configuration fields need to be configured before performing the motor tuning. After pressing the button, the motor will perform 3 tasks:

1. Calibrate the board's sensors, which are used for the sensorless commutation. This is done only once per board and is irrelevant to the motor type.
2. Try to move at 20% of the duty cycle in order to calculate low speed data and
3. Try to move at 100% of the duty cycle in order to calculate high speed data.

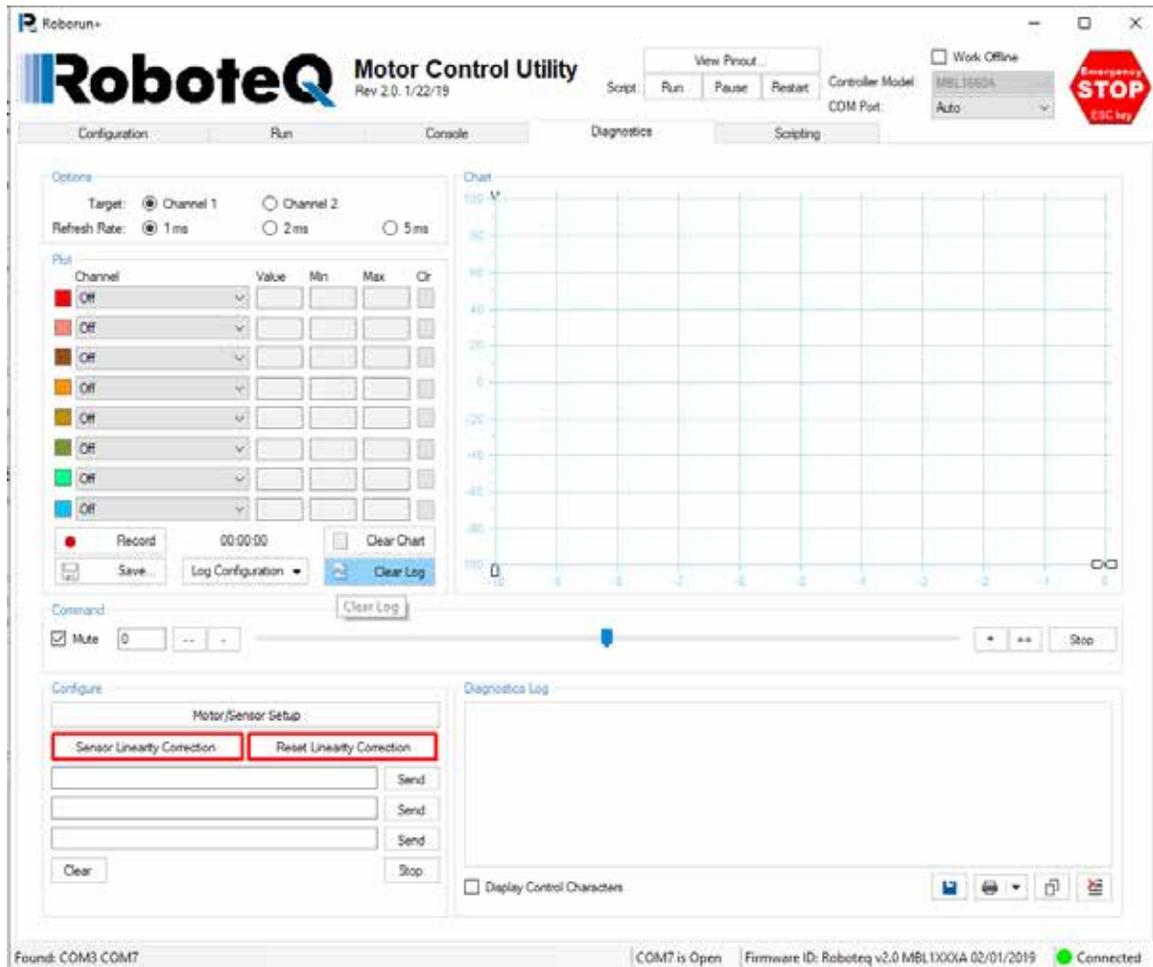
With step 1 some hardware characteristics of the board will be determined in order to make correct calculations during sensorless commutation. The calibration value ZSRM is configured. In case of error during the calibration, in order to re-do the calibration, the command `^ZSRM -1` needs to be sent to the controller. With steps 2 and 3 the proper BEMF Integrator Limit and the BEMF Coupling Constant configuration fields are calculated, in order for the sensorless commutation to be optimal and symmetrical for the specific motor.

In cases of incorrect computations, the process will fail and the message "Bemf Integration Limit Detection FAILED!" will be printed on the console. In that case make sure the motor is properly connected to the controller and retry.

---

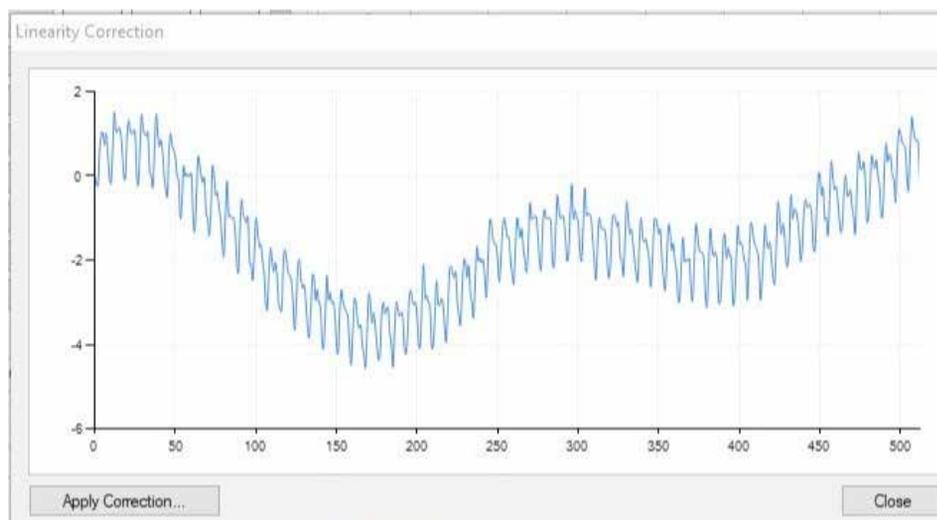
## Sensor Linearity Correction

Brushless motor controllers with firmware v2.0 and higher have the ability to automatically determine whether a sensor has linearity issues and perform correction on it. The respective button "Sensor Linearity Correction" in diagnostics tab will initiate the above-mentioned process, while the button "Reset Linearity Correction" will erase any kind of correction that has been applied.



This process is applicable only for the following sensors:

- SPI/SSI Sensor.
- Sin/Cos Sensor.
- Resolver Sensor.



After pressing the "Sensor "Linearity Correction" the motor will move for a couple of revolutions until it determines the behavior of the sensor. Upon done a pop-up window like the one above will be displayed. If the shape of the graph is sinusoidal-like then the sensor can be considered as non-linear. Repeat the process a couple of times in order to make sure that the behavior is recorded identically and until the graph has not spikes. Then press the "Apply Correction" button in order to apply the correction to controller and save to flash.

If the behavior of the motor after the correction is not better than before, then you can remove the correction by pressing the "Reset Linearity Correction" button.