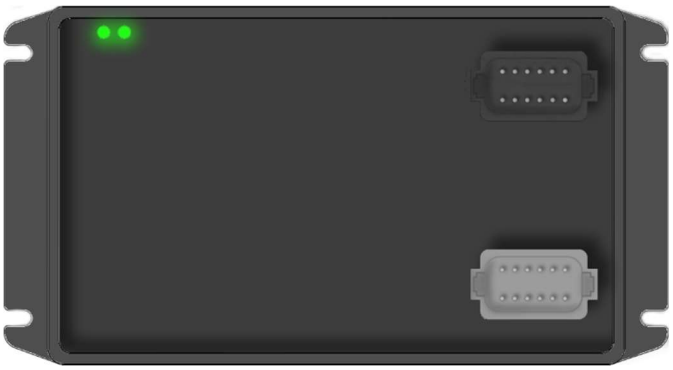
505030 M-Flex 8-Input/ 8-Output CAN Module User Guide



CREATED:	M. PETZKE	DATE:	04/19/2022
CHECKED:	J. COOPER	DATE:	06/27/2023
APPROVED:	J. COOPER	DATE:	06/27/2023
ECN:	13944E	DATE:	12/17/2021

Table of Contents

1. Overview 3
2. Software Installation 5
3. Configuring and Programming the Module 5
3.1 Connecting to the Module with the Programming Tool 8
3.2 Updating the Base Software 8
3.3 Configuring the Module 9
3.3.1 XML Mode 14
3.3.1.1 Configuration 15
3.3.1.2 Logic 17
3.3.1.2.1 Operators 17
3.3.1.2.2 Module Parameters and Variables 21
3.3.1.2.3 Example 26
3.3.2 CAN I/O Mode 27
3.3.2.1 Mode Description 27
3.3.2.2 Configuring for CAN I/O Mode 31
3.3.2.3 CAN I/O Configuration Example 33
3.3.3 Hardware I/O Mode 34
3.3.3.1 Hardware I/O Mode Description 34
3.3.3.2 Configuring for Hardware I/O Mode 34
3.3.3.3 Hardware I/O Configuration Example 36
3.3.4 Keypad Mode 37
3.3.4.1 Keypad Mode Description 37
3.3.4.2 Configuring for Keypad Mode 40
3.3.4.3 Keypad Mode Configuration Example 42

 ©2023 Marlin Technologies, Inc

3.4 Loading the User Program 43
4. LED Module 45

 ©2023 Marlin Technologies, Inc

1. Overview

The 505030 Module is a configurable CAN I/O controller with eight inputs, eight outputs, and programmable logic. The module also has optional built-in support for external Marlin devices including other I/O controllers, keypads, and LED module. The module has two LEDs (see Figure 1) that indicates the module's power and comm status.

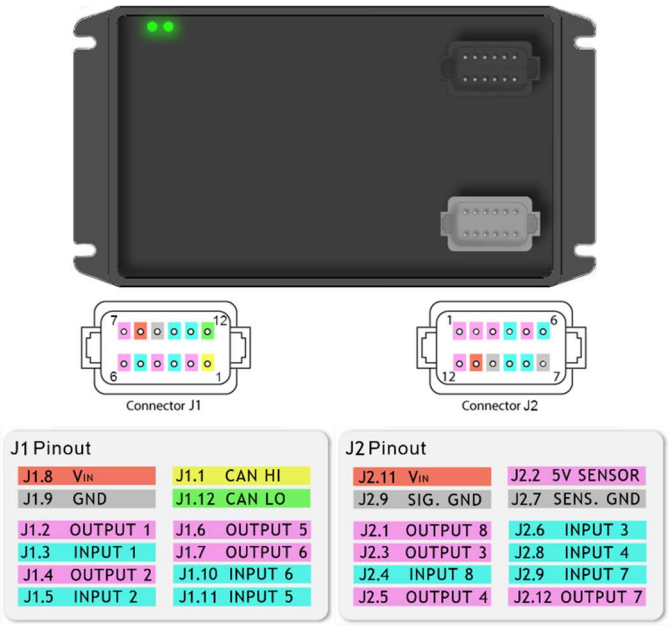


Figure 1

The input types supported by each of the input pins shown in Figure 1 are as follows:

	INPUT 1	INPUT 2	INPUT 3	INPUT 4
Digital – Active High	✓	✓	✓	✓
Digital – Active Low	✓	✓	✓	✓
Analog (0-18V)	✓	✓	✓	✓
Frequency	✓	✓	✓	✓
PWM	✓	✓	✓	✓
Resistance	✓	✓	✓	✓
Current (0-20mA)	✓	✓	✓	✓
Quadrature X1	✓	✓	✓	✓
Quadrature X2	✓	✓	✓	✓
Quadrature X4	✓	✓	✓	✓
Quadrature B	✗	✓	✓	✓

 ©2023 Marlin Technologies, Inc

	INPUT 5	INPUT 6	INPUT 7	INPUT 8
Digital – Active High	✓	✓	✓	✓
Digital – Active Low	✓	✓	✓	✓
Analog (0-18V)	✓	✓	✓	✓
Frequency	✗	✗	✗	✗
PWM	✗	✗	✗	✗
Resistance	✗	✗	✗	✗
Current (0-20mA)	✗	✗	✗	✗
Quadrature X1	✓	✓	✓	✗
Quadrature X2	✓	✓	✓	✗
Quadrature X4	✓	✓	✓	✗
Quadrature B	✓	✓	✓	✓

The output types supported by each output pin shown in Figure 1 are as follows:

	OUTPUT 1	OUTPUT 2	OUTPUT 3	OUTPUT 4
Digital Active High	✓	✓	✓	✓
Digital Active Low	✓	✓	✓	✓
PWM Active High Closed Loop	✓	✓	✓	✓
PWM Active High Open Loop	✓	✓	✓	✓
PWM Active Low Closed Loop	✓	✓	✓	✓
PWM Active Low Open Loop	✓	✓	✓	✓

	OUTPUT 5	OUTPUT 6	OUTPUT 7	OUTPUT 8
Digital Active High	✓	✓	✓	✓
Digital Active Low	✓	✓	✓	✓
PWM Active High Closed Loop	✓	✓	✓	✓
PWM Active High Open Loop	✓	✓	✓	✓
PWM Active Low Closed Loop	✓	✓	✓	✓
PWM Active Low Open Loop	✓	✓	✓	✓

 ©2023 Marlin Technologies, Inc

The module can be configured to run in one of four modes:

● **XML Mode**	The module executes user-defined logic, allowing for highly customizable functionality.
● **CAN I/O Mode**	The 505030 module is commanded by another Marlin controller via CAN communication.
● **Hardware I/O Mode**	The module functions like a relay module, in which input signals drive corresponding output signals.
● **Keypad Mode**	The module's outputs are driven by button presses on a Marlin keypad.

2. Software Installation

In order to program the 505030 module, the Marlin Programming Tool application, as well as the driver for the chosen USB-CAN dongle must be installed on a PC. Run the setup.exe file included with the Programming Tool installation files and follow the directions on the screen to complete installation. When installation is complete, open the Programming Tool's user guide located at

C:\Program Files (x86)\Marlin Technologies\MarlinProgTool\UserGuide.pdf

Note: The exact file path and file name may vary, depending on the version of the Programming Tool and where it was installed. Follow the directions in the Programming Tool user guide to install the driver for the appropriate USB-CAN dongle.

3. Configuring and Programming the Module

The 505030 module runs on two pieces of embedded software, both of which can be loaded onto the module using the Programming Tool:

- **Base Software.** This software serves as an interface between the user program and the module's hardware. It carries out the user's instructions, handling the more technical details of low-level software and hardware interaction. Base software comes pre-loaded on the 505030 module. Updates, if available, are provided by Marlin Technologies in the form of an .S19 file.

- **User Program.** This software is what gives the 505030 module its customized functionality. The user program is created within the Configurator window of the Programming Tool application. This is where I/O is configured and functional logic is written. The Configurator saves the user program to an .XML file. Data from the .XML file is then loaded to the module via the EEPROM Read/Write window of the Programming tool.

 ©2023 Marlin Technologies, Inc

To program the module with either type of software, the setup shown in Figure 2 is required. Connect the USB-CAN dongle to the USB port of the PC. Then connect the CAN lines between the dongle and the 505030 module. See the pinout in Figure 1 for the CAN HI and CAN LO pins on the 505030 module. The configuration and programming process using the Programming Tool is summarized in Figure 2. Details of the process are covered in the sections that follow.

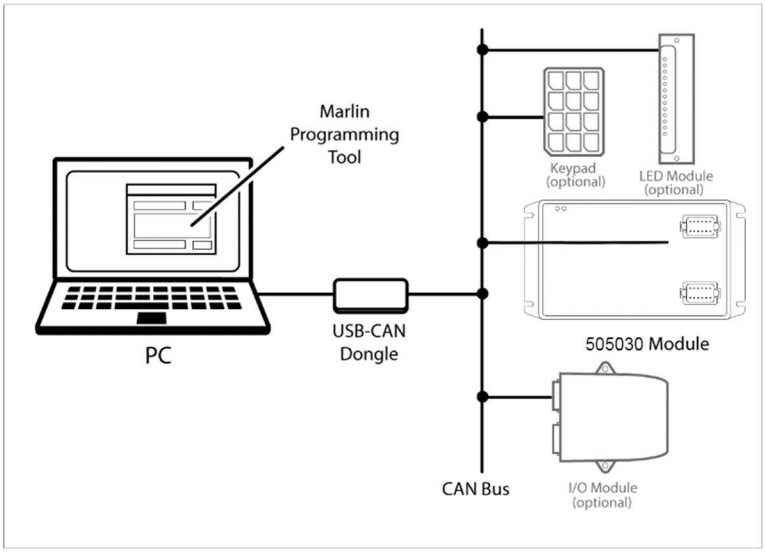


Figure 1

 ©2023 Marlin Technologies, Inc

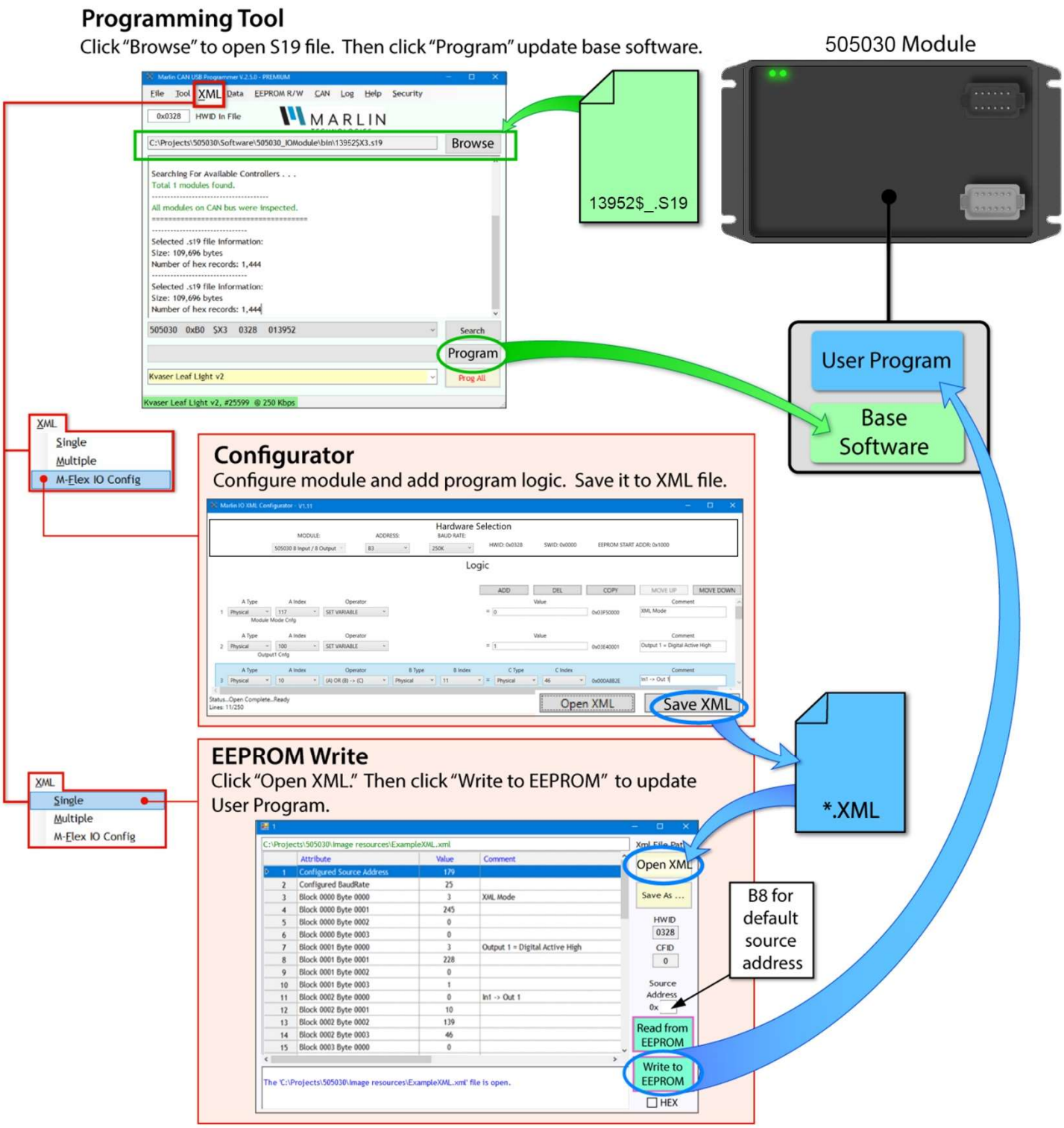


Figure 2

 ©2023 Marlin Technologies, Inc

3.1 Connecting to the Module with the Programming Tool

After installing the Programming Tool, there should be an icon for it on the PC desktop. Double-click the icon to launch the Programming Tool app. If all devices are set up properly, as shown in Figure 1, information regarding the USB-CAN dongle and the 505030 module should automatically appear near the bottom of the Programming Tool window (see the circled text in *Figure* 3). If the module information does not appear, click on the "Search" button to prompt the Programming Tool to search again for the module. If the module or dongle information fail to appear, or if there are any other problems encountered with the Programming Tool, see the Programming Tool user guide mentioned in section 2. Software Installation.

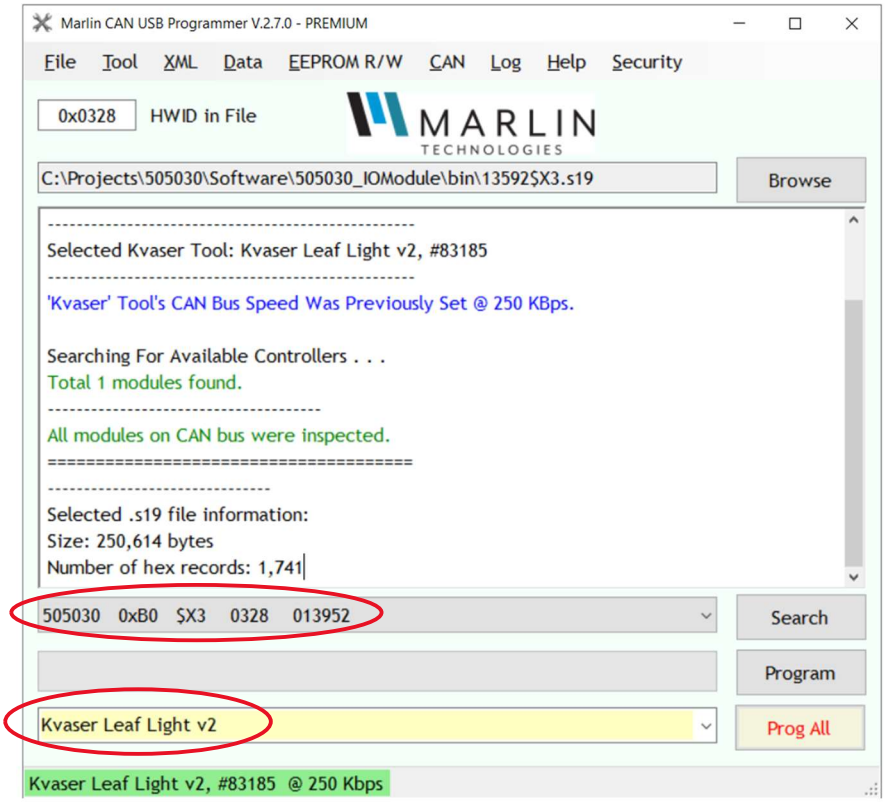


Figure 3

3.2 Updating the Base Software

Since the 505030 module comes with the base software pre-loaded, it should not be necessary to load the base Software unless a new revision of the software has been released. In that case, an .S19 file containing the updated software can be obtained from Marlin Technologies.

With the Program Tool running and connected to the module (see section *3.1 Connecting to the Module with the Programming Tool* for details), click on the "Browse" button, as seen circled in Figure 4.

 ©2023 Marlin Technologies, Inc

Figure 4

In the window that pops up, browse for the appropriate .S19 file and select it. Then press the "Program" button, shown circled in Figure 5.

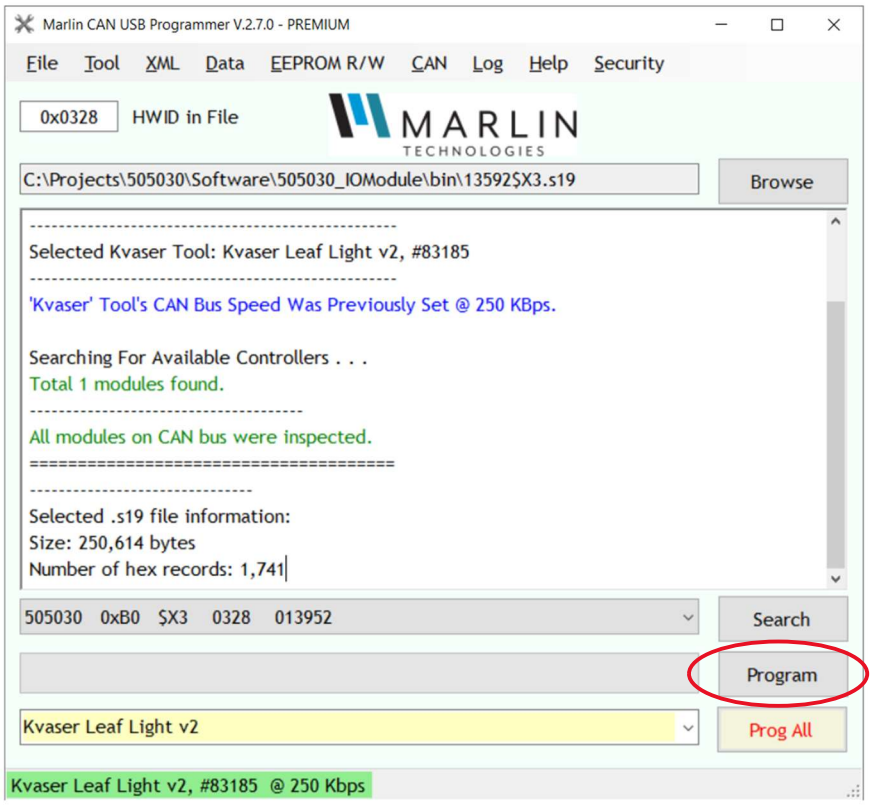


Figure 5

While programming, a green progress bar will appear to the left of the "Program" button. Programming is complete once the progress bar reaches 100% and a message indicating "successful programming" is displayed. If programming is not successful, see the Programming Tool user guide mentioned in section *2.* Software Installation to troubleshoot.

3.3 Configuring the Module

The Configurator window of the Programming Tool allows the user to configure the 505030 module's mode, its I/O, and (if applicable) create its functional logic. Configuration instructions

 ©2023 Marlin Technologies, Inc

created in the Configurator can be saved to an .XML file, which can be later opened by the Programming Tool's EEPROM-writing utility and used to program the module (see Figure 2 for an overview of this process).

To open a Configurator window, launch the Programming Tool app, and then select the "XML->M-Flex IO Config" menu option, shown circled in Figure 7.

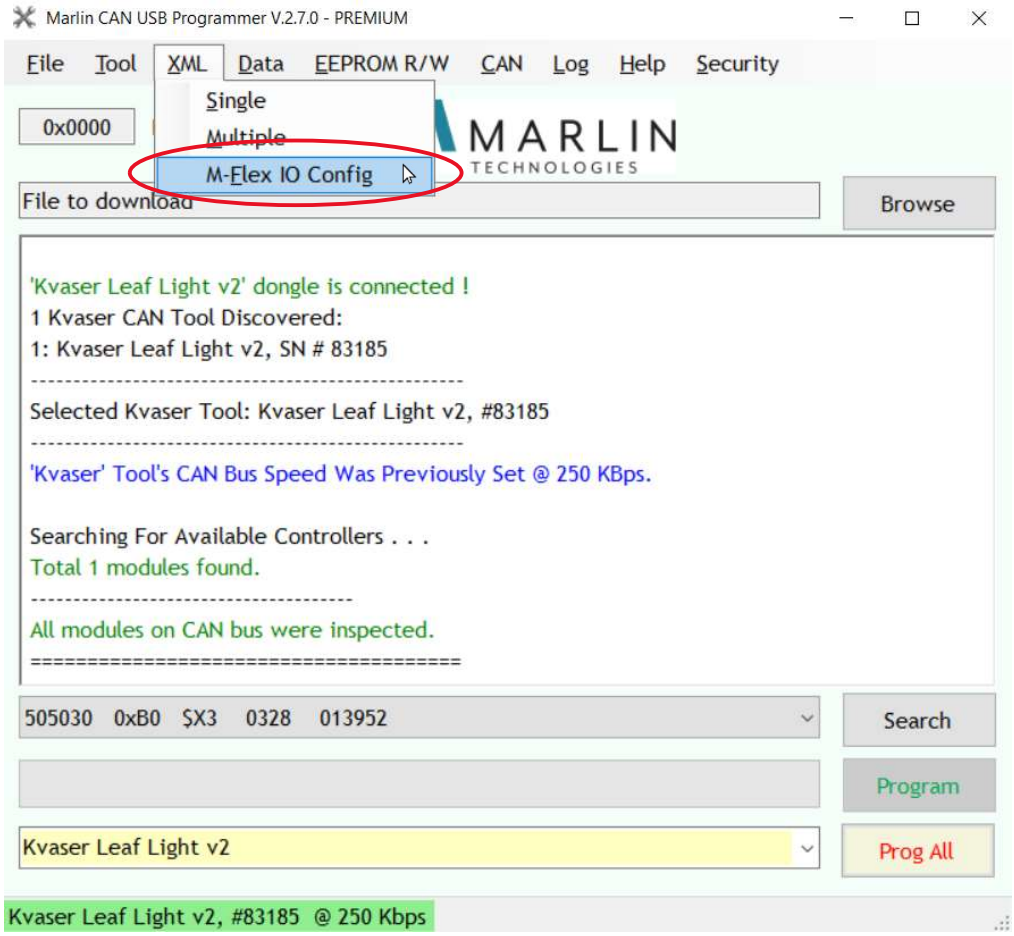


Figure 6

This will open a Configurator window, separate from the existing main Programming Tool window. In the Configurator window, using the drop-down menus circled in Figure 7, select "505030 8I/8O" for the module type, the desired source address for CAN messages, and the desired baud rate for CAN messages.

 ©2023 Marlin Technologies, Inc

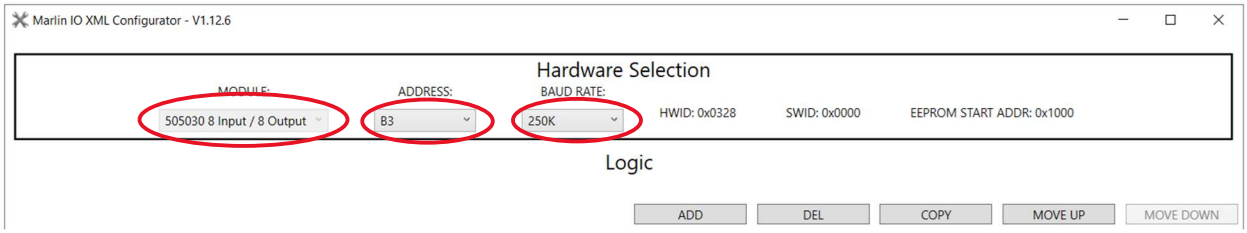


Figure 7

Now instructions for configuring the module can be added. Click the “ADD” button to add an instruction. The default instruction that appears is shown Figure 8. The type of operation that instruction performs is indicated by the “Operator” field. For configuration, only *SET VARIABLE* instructions need be used.

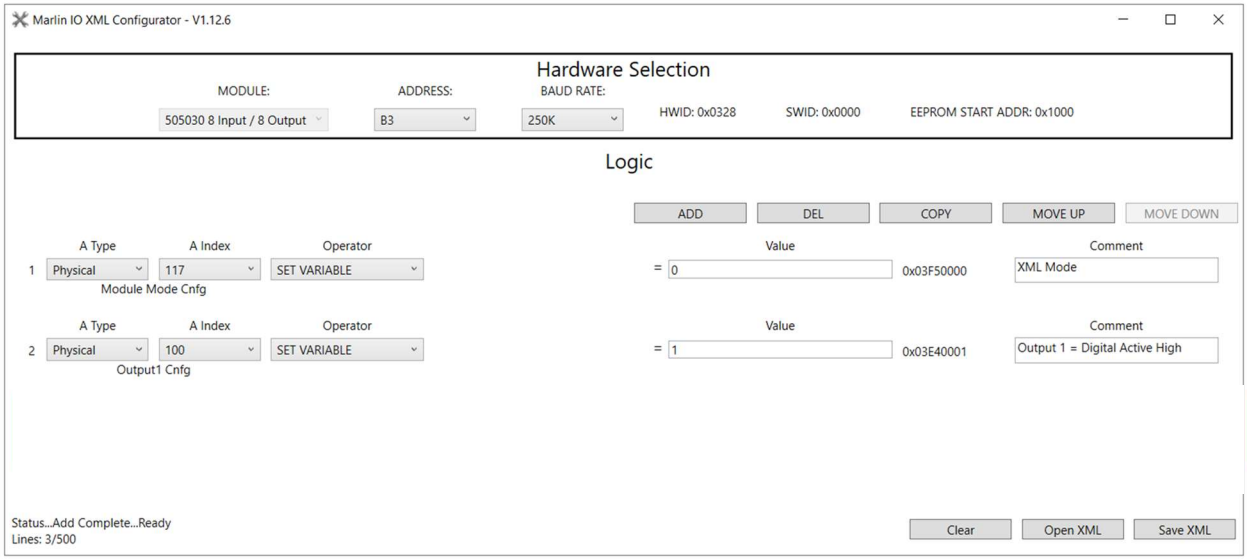


Figure 8

 ©2023 Marlin Technologies, Inc

To change the recently added instruction to a *SET VARIABLE* instruction, select “SET VARIABLE” operand from the drop-down menu of the Operator field, as shown in Figure 9.

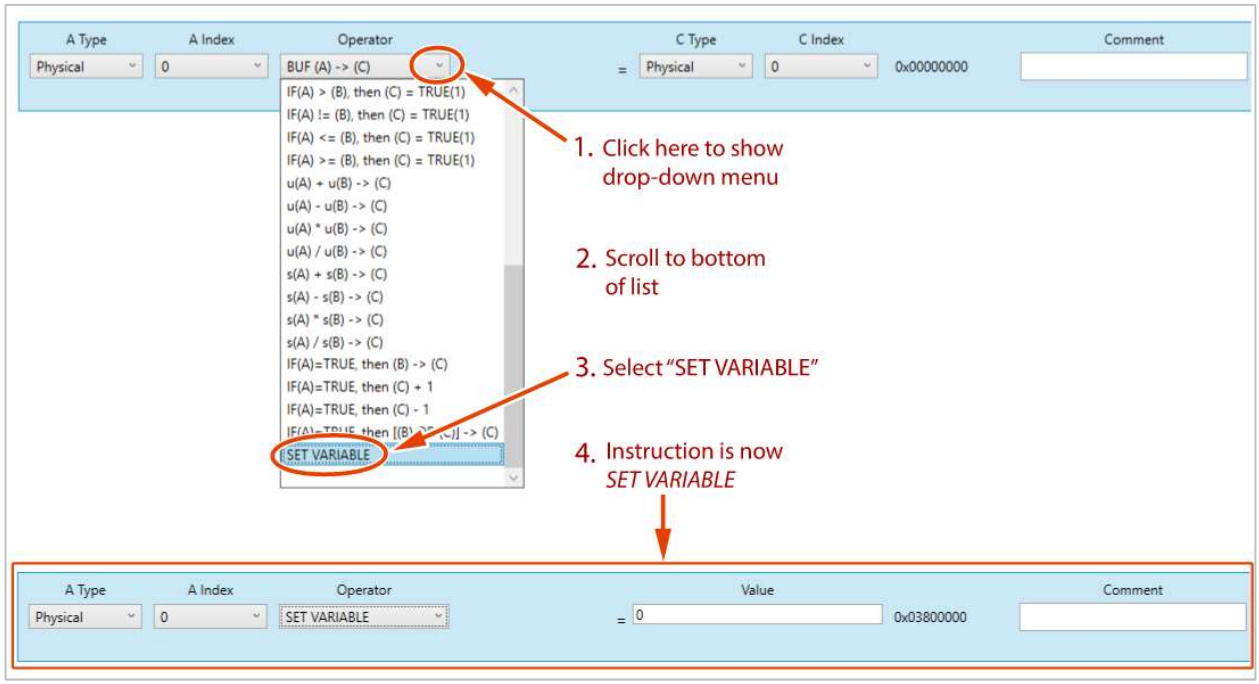


Figure 9

The instructions available in the Configurator consist of an operator that takes inputs, performs an operation, and puts the result in an output variable. Inputs may be a fixed value or a variable. The variables used are A, B, and C. *SET VARIABLE* is a simple instruction that takes a fixed value and assigns it to variable A, hence the two fields: “*A* Type” and “*A* Index.” In this case, variable A represents the module parameter that the instruction will configure.

Since the module mode dictates the other parameters that need to be configured, setting the module mode is a good choice for the first instruction. To select the module mode as the parameter to be configured, use the drop-down menus to set the Type to “Physical” and Index to “117”:

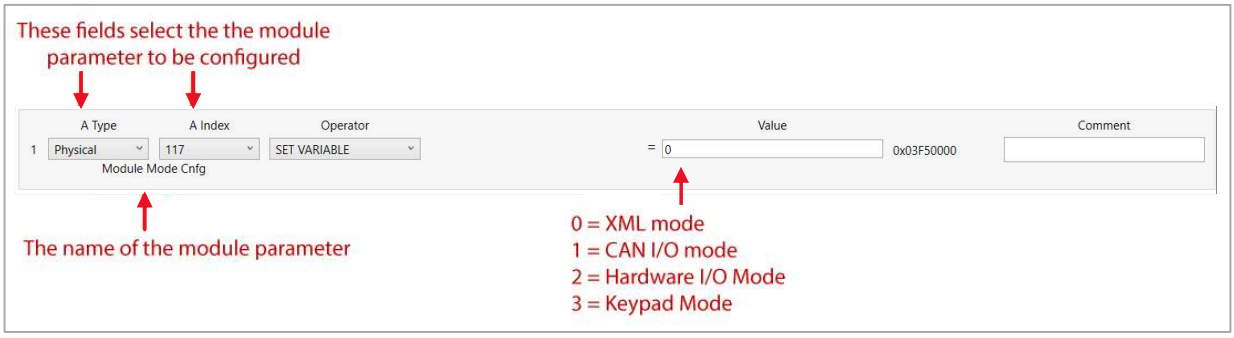


Figure 10

 ©2023 Marlin Technologies, Inc

Notice that when A Type and A Index is changed, the name of the module parameter is updated automatically.

Now additional instructions can be added pertaining to the module mode that has been selected:

- For **XML Mode**, see section 3.3.1 XML Mode. Use XML Mode for highly customized functionality. In XML Mode, the module executes user-defined logic.
- For **CAN I/O Mode**, see section 3.3.2 CAN I/O Mode. Use CAN I/O mode if the 505030 is to be commanded by another controller via CAN communication.
- For **Hardware I/O Mode**, see section 3.3.3 Hardware I/O Mode. Use Hardware I/O Mode if the module is to function as a relay module, in which input signals drive corresponding output signals.
- For **Keypad Mode,** see section
- 3.3.4 Keypad Mode. Use Keypad Mode if the module's outputs are to be driven by button presses on a keypad.

 ©2023 Marlin Technologies, Inc

3.3.1 XML Mode

In XML Mode, the 505030 module executes user-defined logic that allows for highly customizable functionality. With the Programming Tool's Configurator utility, the user creates a program that consists of two main groups of instructions: the initialization block and the logic block. When the module powers up, the initialization block is executed once, then the logic block is executed, repeatedly in an infinite loop, every 10ms.

Instructions are executed in the order they appear in the Configurator, from top to bottom. Initialization instructions utilize only the *SET VARIABLE* operator, and must be grouped together in the top rows of the program. The first instruction that does *not* use the *SET VARIABLE* operator represents the beginning of the logic block. It is this instruction that program execution loops back to after the last logic instruction is executed. The path of program execution is visualized in Figure 11.

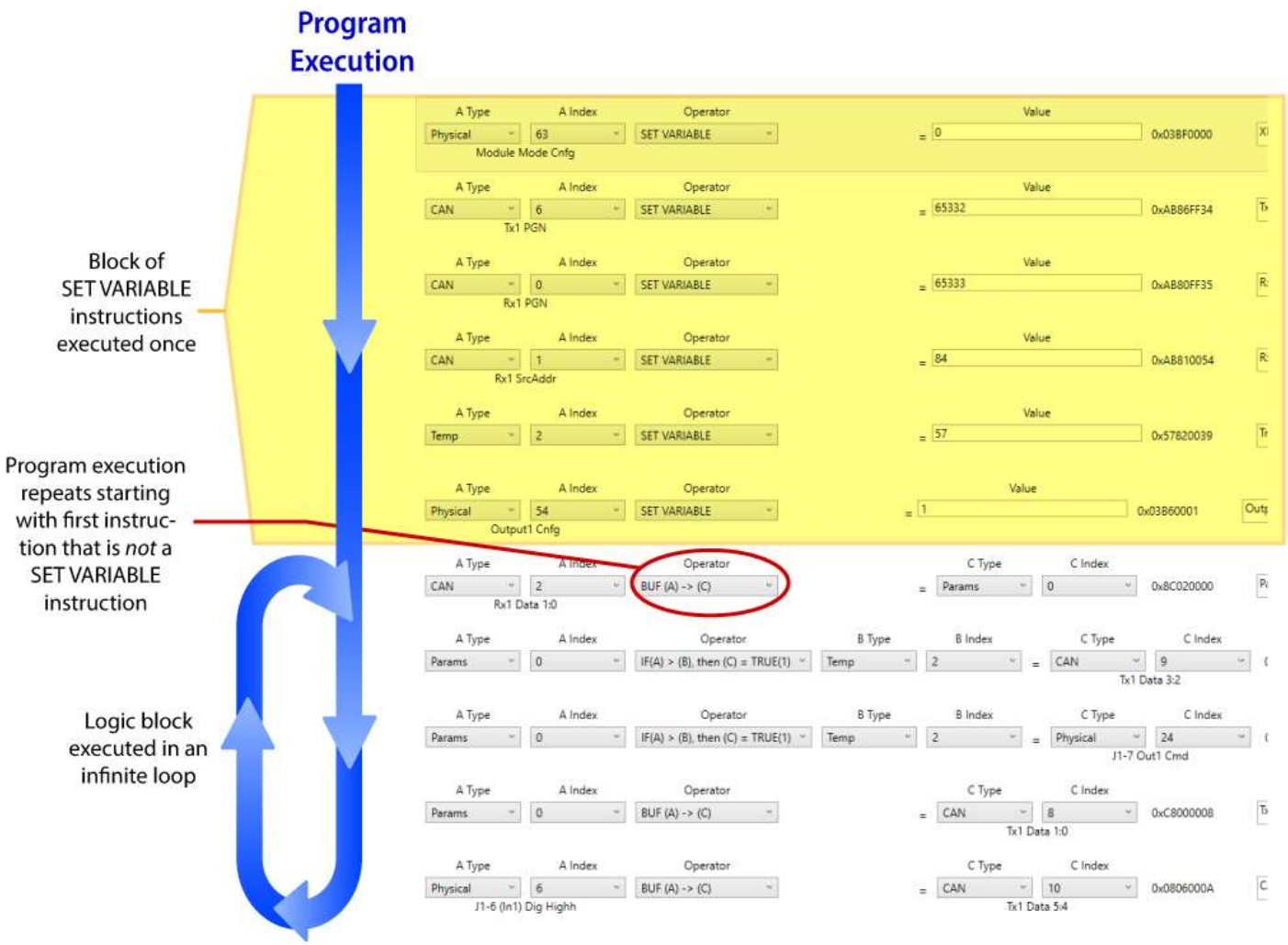


Figure 11

 ©2023 Marlin Technologies, Inc

The initialization block consists of instructions that configure module parameters such as output types and current limits. It may also set up variables that store gains, timer reload values, CAN masks, or values to compare to. It might also be where the PGN value of a transmitted CAN message is set, if the PGN value is not expected to change.

3.3.1.1 Configuration

Use the ADD button to add *SET VARIABLE* instructions for each parameter listed in the *List of Configuration Parameters Used in XML Mode* below. Refer to Figure 12 as a guide for filling out the *SET VARIABLE* instructions. If necessary, see section *3.3 Configuring the Module* for more details about adding *SET VARIABLE* instructions.

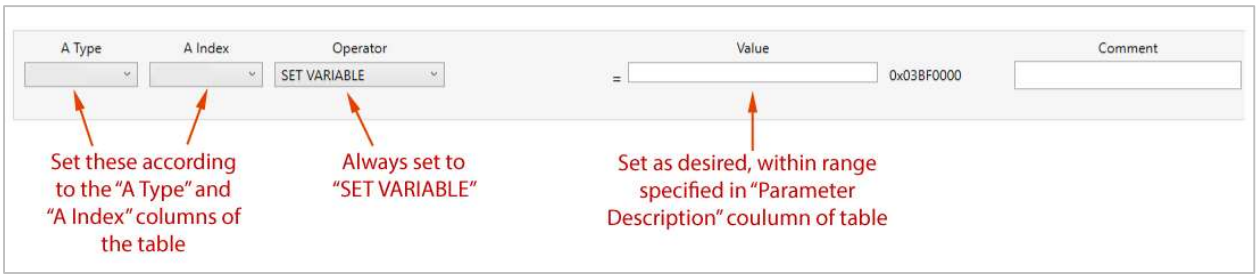


Figure 12

List of Configuration Parameters Used in XML Mode

A Type	A Index	Parameter Description
Physical	117	Module Mode. Must be 0 for XML Mode.
Physical	100	Output 1 Type [1]
Physical	101	Output 2 Type [1]
Physical	102	Output 3 Type [1]
Physical	103	Output 4 Type [1]
Physical	104	Output 5 Type [1]
Physical	105	Output 6 Type [1]
Physical	106	Output 7 Type [1]
Physical	107	Output 8 Type [1]
Physical	99	Frequency for all PWM Outputs (30-1000Hz)
Physical	78	Threshold Voltage for Digital inputs to be set to Active [65.535mV, 1mV/bit]
Physical	79	Hysteresis Voltage for Digital inputs to return to Inactive [65.535mV, 1mV/bit]
Physical	80	P-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	81	I-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	82	D-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	62	Input 1 Type[2]
Physical	63	Input 2 Type[2]
Physical	64	Input 3 Type[2]
Physical	65	Input 4 Type[2]
Physical	66	Input 5 Type[2]
Physical	67	Input 6 Type[2]
Physical	68	Input 7 Type[2]
Physical	69	Input 8 Type[2]
Physical	70	Output 1 Current Limit [1 mA/bit, 0-3000mA]

 ©2023 Marlin Technologies, Inc

A Type	A Index	Parameter Description
Physical	71	Output 2 Current Limit [1 mA/bit, 0-3000mA]
Physical	72	Output 3 Current Limit [1 mA/bit, 0-3000mA]
Physical	73	Output 4 Current Limit [1 mA/bit, 0-3000mA]
Physical	74	Output 5 Current Limit [1 mA/bit, 0-3000mA]
Physical	75	Output 6 Current Limit [1 mA/bit, 0-3000mA]
Physical	76	Output 7 Current Limit [1 mA/bit, 0-3000mA]
Physical	77	Output 8 Current Limit [1 mA/bit, 0-3000mA]
Physical	108	Output 1 Keypad Config
Physical	109	Output 2 Keypad Config
Physical	110	Output 3 Keypad Config
Physical	111	Output 4 Keypad Config
Physical	112	Output 5 Keypad Config
Physical	113	Output 6 Keypad Config
Physical	114	Output 7 Keypad Config
Physical	115	Output 8 Keypad Config
Physical	116	Keypad Source Address
CAN	79	Cmd Message Timeouts [10mS/bit, 0=NoTimeout]

[1] 0 = Off [Default]
1 = High Side Digital (Active High)
2 = Low Side Digital (Active Low)
3 = High Side Current (mA)
4 = High Side PWM (Duty Cycle)
5 = Low Side Current (mA)
6 = Low Side PWM (Duty Cycle)
7 = Bi-Directional (Half-Bridge) *All 505030 outputs are half-bridge by default
8 = Voltage

[2] 0 = Digital/Analog/PWM [Default]
1 = Current
2 = Resistance
3 = Quadrature X1
4 = Quadrature X2
5 = Quadrature X4

Configuration Example for XML Mode

Configurator Instruction				
Operator	**A Type**	**A Index**	**Value**	**Result**
SET VARIABLE	Physical	117	0	Module Mode = XML Mode
SET VARIABLE	Physical	100	1	Output 1 is a digital active high output
SET VARIABLE	Physical	101	2	Output 2 is a digital active low output
SET VARIABLE	Physical	102	3	Output 3 is an active high PWM closed loop output
SET VARIABLE	Physical	103	6	Output 4 is an active low PWM open loop output
SET VARIABLE	Physical	99	100	Output PWM Frequency is 100Hz

 ©2023 Marlin Technologies, Inc

Configurator Instruction				
Operator	**A Type**	**A Index**	**Value**	**Result**
SET VARIABLE	Physical	62	1	Input 1 is a current input
SET VARIABLE	Physical	78	2000	Input digital Thresholds = Vbatt-2V & GND+2V
SET VARIABLE	Physical	79	1000	Input digital hysteresis = 1V
SET VARIABLE	Physical	80	2500	P-Term = 2500
SET VARIABLE	Physical	81	750	I-Term = 750
SET VARIABLE	Physical	82	0	D-Term = 0

3.3.1.2 Logic

Logic instructions are to be added *below* the block of initialization instructions that use the *SET VARIABLE* instruction. Unlike initialization instructions, logic instructions can utilize a wide variety of operators. The various operators are discussed in section *3.3.1.2.1 Operators*, and the parameters and variables that the operators act upon are discussed in section *3.3.1.2.2 Module Parameters and Variables*.

3.3.1.2.1 Operators

Each instruction in the Configurator performs a single operation. An operation consists of an Operator, which acts upon variables A, B, and C. The number of variables involved varies from operator to operator. Variables A and B typically serve as inputs to an operation, while variable C is mainly used to store the operation's output. (The only exception is the *SET VARIABLE* operator, which uses variable A, instead of C, for its output.)

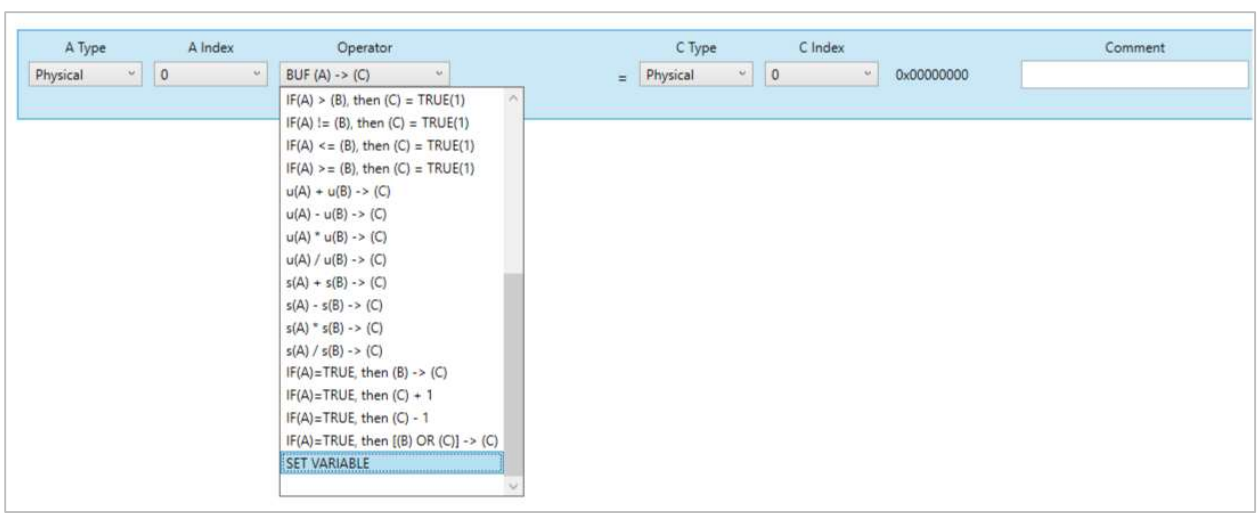


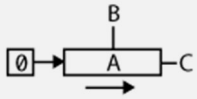
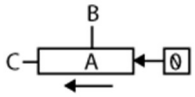
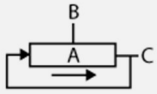
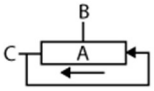
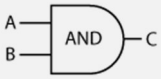
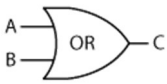
Figure 13

 ©2023 Marlin Technologies, Inc

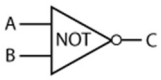
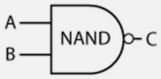
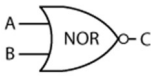
ASSIGNMENT OPERATORS

Name	Menu Item	Comment
Set Variable	SET VARIABLE	Sets variable A to a given fixed value.
Copy	BUF (A) -> (C)	Copies variable A's value to variable C.

LOGICAL OPERATORS

Name	Menu Item	Comment
Shift Right 	(A) >> (B) -> (C)	The value in A is shifted bitwise to the right B times, with zero shifted into the most significant bit. The result is placed in output C.
Shift Left 	(A) << (B) -> (C)	The value in A is shifted bitwise to the left B times, with zero shifted into the least significant bit. The result is placed in output C.
Rotate Right 	ROT RGHT (A) BY (B)-> (C)	The value in A is rotated bitwise to the right B times, with the least significant bit rotated into the most significant bit. The result is placed in output C.
Rotate Left 	ROT LEFT (A) BY (B)-> (C)	The value in A is rotated bitwise to the right B times, with the most significant bit rotated into the least significant bit. The result is placed in output C.
Logical AND 	(A) AND (B) -> (C)	The value in A is AND'd with the value in B and placed in the output C. Example: 1100 AND 1010 = 1000
Logical OR 	(A) OR (B) -> (C)	The value in A is OR'd with the value in B and placed in the output C. Example: 1100 OR 1010 = 1110
Logical XOR	(A) XOR (B) -> (C)	The value in A is XOR'd with the value in B and placed in the output C. Example: 1100 XOR 1010 = 0110

 ©2023 Marlin Technologies, Inc

Name	Menu Item	Comment
Logical NOT 	NOT (A) -> (C)	The value in A is inverted and placed in the output C. Example: NOT 1010 = 0101
Logical NAND 	(A) NAND (B) -> (C)	The value in A is AND'd with the value in B, then inverted and placed in the output C. Example: 1100 NAND 1010 = 0111
Logical NOR 	(A) NOR (B) -> (C)	The value in A is OR'd with the value in B, then inverted and placed in the output C. Example: 1100 NOR 1010 = 0001
Logical NXOR	(A) XNOR (B) -> (C)	The value in A is XOR'd with the value in B, then inverted and placed in the output C. Example: 1100 XNOR 1010 = 1001

COMPARATIVE OPERATORS

Name	Menu Item	Comment
If Equal	IF(A) = (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.
If Less Than	IF(A) < (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.
IF Greater Than	IF(A) > (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.
IF Not Equal	IF(A) != (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.
If Less Than or Equal To	IF(A) <= (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.
If Greater Than or Equal To	IF(A) >= (B) -> (C) = TRUE(1)	Otherwise C is left unchanged.

MATHEMATICAL OPERATORS (Unsigned Values)

Name	Menu Item	Comment
Add	u(A) + u(B) -> u(C)	Value A, B, and C must fit within a range of 0 to 65535.
Subtract	u(A) - u(B) -> u(C)	Value A, B, and C must fit within a range of 0 to 65535.
Multiply	u(A) * u(B) -> u(C)	Value A, B, and C must fit within a range of 0 to 65535.

 ©2023 Marlin Technologies, Inc

Name	Menu Item	Comment
Divide	u(A) / u(B) -> u(C)	Value A, B, and C must fit within a range of 0 to 65535.

MATHEMATICAL OPERATORS (Signed Values)

Name	Menu Item	Comment
Add	s(A) + s(B) -> s(C)	Value A, B, and C must fit within a range of -32768 to 32767.
Subtract	s(A) - s(B) -> s(C)	Value A, B, and C must fit within a range of -32768 to 32767.
Multiply	s(A) * s(B) -> s(C)	Value A, B, and C must fit within a range of -32768 to 32767.
Divide	s(A) / s(B) -> s(C)	Value A, B, and C must fit within a range of -32768 to 32767.

CONDITIONAL OPERATORS

Name	Menu Item	Comment
Conditional Copy	IF(A)=TRUE, then (B) -> (C)	Otherwise C is left unchanged.
Conditional INCREMENT	IF(A)=TRUE, the (C) + 1	Otherwise C is left unchanged.
Conditional DECREMENT	IF(A)=TRUE, the (C) - 1	Otherwise C is left unchanged.
Conditional OR	IF(A)=TRUE, the [(B) OR (C)] -> C	Otherwise C is left unchanged.

 ©2023 Marlin Technologies, Inc

3.3.1.2.2 Module Parameters and Variables

In a Configurator instruction, the module parameter that Variable A, B, or C represents is specified by a Type field and an Index Number field. Thus, each variable appearing in an operation has two drop-down menus associated with it. For example, in Figure 14, the user has selected *IF (A) > (B), then (C) = TRUE(1)* as the Operator. As a result, the row was populated with drop down menus for variables A (highlighted in red), B (highlighted in blue), and C (highlighted in green).

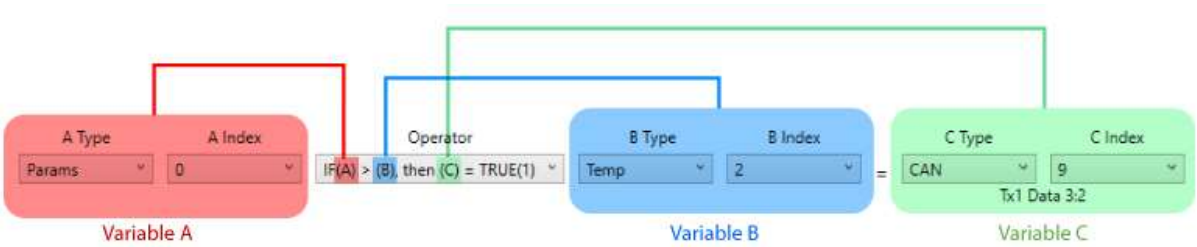


Figure 14

The options available in the Type menus are:

- **Physical** – These parameters often pertain to the module's I/O.
- **Temp** – These are variables intended for use as temporary or constant values within the initial block of *SET VARIABLE* operations that precede the program's infinite loop. For example, they might be used to store gains, timer reload values, CAN masks, or values to compare to. All Temp variables can be set to values of the user's choosing except for TRUE and FALSE, which are to remain at fixed values of 1 and 0, respectively.
- **CAN** – These parameters pertain to the settings and data of received and transmitted CAN messages
- **Params** – These are variables that are meant to be stored between module power cycles. These parameters are loaded from non-volatile memory upon power ON. Changes to a value prompt a write (see note) to non-volatile memory. *SET VARIABLE* operations for Params type are ignored.

The module parameters available through the Type and Index drop-down menus in Figure 14 are presented on the pages below. Note that each parameter value is 16 bits.

When finished adding instructions in the Configurator, click "Save XML" in the bottom-right corner of the window (as seen in Figure 9) to save the data to an .XML file. To load this data on to the 505030 module, follow the instructions in *section 3.4 Loading the* User Program.

Note: Minimum rated non-volatile memory writes before memory degradation begins is 1,000,000. Care should be taken to limit writes to the Param type as much as possible. Use the below equation to estimate the life span of a single param location:

$$1{,}000{,}000 \,/\, [Number\ of\ writes\ per\ second] \;=\; [expected\ number\ of\ seconds\ before\ earliest\ memory\ failure]$$

(To guarantee a 10 year lifespan minimum, the write frequency of a single location should be no faster than ~5 minutes).

 ©2023 Marlin Technologies, Inc

Physical Parameters

Index	Description	
0	**DO NOT USE**	
1	Input	Battery Voltage [0-42,000 mV, 1mV/bit]
2	Input	Input 1 Analog Input/Quad Count [(0-42,000 mV, 1mV/bit); (4-20mA, 1uA/bit); (0-65535Ω, 1Ω/bit); Unsigned Count]
3	Input	Input 2 Analog Input/Quad Count [(0-42,000 mV, 1mV/bit); (4-20mA, 1uA/bit); (0-65535Ω, 1Ω/bit); Unsigned Count]
4	Input	Input 3 Analog Input/Quad Count [(0-42,000 mV, 1mV/bit); (4-20mA, 1uA/bit); (0-65535Ω, 1Ω/bit); Unsigned Count]
5	Input	Input 4 Analog Input/Quad Count [(0-42,000 mV, 1mV/bit); (4-20mA, 1uA/bit); (0-65535Ω, 1Ω/bit); Unsigned Count]
6	Input	Input 5 Analog Voltage/Quad Count [(0-42,000 mV, 1mV/bit), Unsigned Count]
7	Input	Input 6 Analog Voltage/Quad Count [(0-42,000 mV, 1mV/bit), Unsigned Count]
8	Input	Input 7 Analog Voltage/Quad Count [(0-42,000 mV, 1mV/bit), Unsigned Count]
9	Input	Input 8 Analog Voltage [0-42,000 mV, 1mV/bit]
10	Input	Input 1 Digital High [1=Active High, 0=otherwise]
11	Input	Input 1 Digital Low [1=Active Low 0=otherwise]
12	Input	Input 2 Digital High [1=Active High, 0=otherwise]
13	Input	Input 2 Digital Low [1=Active Low, 0=otherwise]
14	Input	Input 3 Digital High [1=Active High, 0=otherwise]
15	Input	Input 3 Digital Low [1=Active Low, 0=otherwise]
16	Input	Input 4 Digital High [1=Active High, 0=otherwise]
17	Input	Input 4 Digital Low [1=Active Low, 0=otherwise]
18	Input	Input 5 Digital High [1=Active High, 0=otherwise]
19	Input	Input 5 Digital Low [1=Active Low, 0=otherwise]
20	Input	Input 6 Digital High [1=Active High, 0=otherwise]
21	Input	Input 6 Digital Low [1=Active Low, 0=otherwise]
22	Input	Input 7 Digital High [1=Active High, 0=otherwise]
23	Input	Input 7 Digital Low [1=Active Low, 0=otherwise]
24	Input	Input 8 Digital High [1=Active High, 0=otherwise]
25	Input	Input 8 Digital Low [1=Active Low, 0=otherwise]
26	Input	Output 1 Current [0-4200mA, 1mA/bit]
27	Input	Output 2 Current [0-4200mA , 1mA/bit]
28	Input	Output 3 Current [0-4200mA , 1mA/bit]
29	Input	Output 4 Current [0-4200mA , 1mA/bit]
30	Input	Output 5 Current [0-4200mA , 1mA/bit]
31	Input	Output 6 Current [0-4200mA , 1mA/bit]
32	Input	Output 7 Current [0-4200mA , 1mA/bit]
33	Input	Output 8 Current [0-4200mA , 1mA/bit]
34	Input	Input 1 Duty Cycle [0-100%, 0.1%/bit]
35	Input	Input 1 Period [0-65535uS, 1uS/bit]
36	Input	Input 1 Frequency [32-10,000Hz, 0.1Hz/bit]
37	Input	Input 2 Duty Cycle [0-100%, 0.1%/bit]
38	Input	Input 2 Period [0-65535uS, 1uS/bit]
39	Input	Input 2 Frequency [32-10,000Hz, 0.1Hz/bit]
40	Input	Input 3 Duty Cycle [0-100%, 0.1%/bit]
41	Input	Input 3 Period [0-65535uS, 1uS/bit]
42	Input	Input 3 Frequency [32-10,000Hz, 0.1Hz/bit]
43	Input	Input 4 Duty Cycle [0-100%, 0.1%/bit]
44	Input	Input 4 Period [0-65535uS, 1uS/bit]
45	Input	Input 4 Frequency [32-10,000Hz, 0.1Hz/bit]
46	Output	Output 1 Digital [1=On, 0=Off]

 ©2023 Marlin Technologies, Inc

Index	Description	
47	Output	Output 2 Digital [1=On, 0=Off]
48	Output	Output 3 Digital [1=On, 0=Off]
49	Output	Output 4 Digital [1=On, 0=Off]
50	Output	Output 5 Digital [1=On, 0=Off]
51	Output	Output 6 Digital [1=On, 0=Off]
52	Output	Output 7 Digital [1=On, 0=Off]
53	Output	Output 8 Digital [1=On, 0=Off]

 ©2023 Marlin Technologies, Inc

CAN Parameters (XML Mode)

Index #				
0	Rx1 PGN		Rx CAN Msg #1	PGN of CAN Message to be received (*)
1	Rx1 SrcAddr			Source Address of CAN Message to be received (**)
2	Rx1 Data 1:0			Data Bytes 1[MSB] & 0[LSB] of message Received
3	Rx1 Data 3:2			Data Bytes 3[MSB] & 2[LSB] of message Received
4	Rx1 Data 5:4			Data Bytes 5[MSB] & 4[LSB] of message Received
5	Rx1 Data 7:6			Data Bytes 7[MSB] & 6[LSB] of message Received
6	Tx1 PGN		Tx CAN Msg #1	PGN of CAN Message to be Transmitted (*)
7	Tx1 MsgRate			Std/Ext ID Flag : Message Transmit Rate [10mS/bit, 0-32,000mS] (***)
8	Tx1 Data 1:0			Data Bytes 1[MSB] & 0[LSB] to send
9	Tx1 Data 3:2			Data Bytes 3[MSB] & 2[LSB] to send
10	Tx1 Data 5:4			Data Bytes 5[MSB] & 4[LSB] to send
11	Tx1 Data 7:6			Data Bytes 7[MSB] & 6[LSB] to send
12	Rx2 PGN		Rx CAN Msg #2	PGN of CAN Message to be received (*)
13	Rx2 SrcAddr			Source Address of CAN Message to be received (**)
14	Rx2 Data 1:0			Data Bytes 1[MSB] & 0[LSB] of message Received
15	Rx2 Data 3:2			Data Bytes 3[MSB] & 2[LSB] of message Received
16	Rx2 Data 5:4			Data Bytes 5[MSB] & 4[LSB] of message Received
17	Rx2 Data 7:6			Data Bytes 7[MSB] & 6[LSB] of message Received
18	Tx2 PGN		Tx CAN Msg #2	PGN of CAN Message to be Transmitted (*)
19	Tx2 MsgRate			Std/Ext ID Flag : Message Transmit Rate [10mS/bit, 0-32,000mS] (***)
20	Tx2 Data 1:0			Data Bytes 1[MSB] & 0[LSB] to send
21	Tx2 Data 3:2			Data Bytes 3[MSB] & 2[LSB] to send
22	Tx2 Data 5:4			Data Bytes 5[MSB] & 4[LSB] to send
23	Tx2 Data 7:6			Data Bytes 7[MSB] & 6[LSB] to send
24	Rx3 PGN		Rx CAN Msg #3	PGN of CAN Message to be received (*)
25	Rx3 SrcAdr			Source Address of CAN Message to be received (**)
26	Rx3 Data 1:0			Data Bytes 1[MSB] & 0[LSB] of message Received
27	Rx3 Data 3:2			Data Bytes 3[MSB] & 2[LSB] of message Received
28	Rx3 Data 5:4			Data Bytes 5[MSB] & 4[LSB] of message Received
29	Rx3 Data 7:6			Data Bytes 7[MSB] & 6[LSB] of message Received
30	Tx3 PGN		Tx CAN Msg #3	PGN of CAN Message to be Transmitted (*)
31	Tx3 MsgRate			Std/Ext ID Flag : Message Transmit Rate [10mS/bit, 0-32,000mS] (***)
32	Tx3 Data 1:0			Data Bytes 1[MSB] & 0[LSB] to send
33	Tx3 Data 3:2			Data Bytes 3[MSB] & 2[LSB] to send
34	Tx3 Data 5:4			Data Bytes 5[MSB] & 4[LSB] to send
35	Tx3 Data 7:6			Data Bytes 7[MSB] & 6[LSB] to send
79	Cnfg	TimeOut	Time	Cmd Message Timeouts [10mS/bit, 0=NoTimeout]

(*) NOTE: 11-bit Std_ID: PGN is Message ID
29-bit Ext_ID: PGN is PDU:Format and PDU:Specific of Message ID
Priority is ignored on Received messages, and default of 6 for Transmitted messages.

(**) NOTE: 11-bit Std_ID: Not Applicable, set as 65535.
29-bit Ext_ID: SourceAddress of Message ID

(***) NOTE: Add 32768 to the desired transmit rate to send the PGN value as a Standard ID.

 ©2023 Marlin Technologies, Inc

CAN Parameters (XML Mode) cont.

63	Kypd	Input	Digital	Keypad Input 1 [0=Up,1=Down,2=Err,3=No Key]
64	Kypd	Input	Digital	Keypad Input 2 [0=Up,1=Down,2=Err,3=No Key]
65	Kypd	Input	Digital	Keypad Input 3 [0=Up,1=Down,2=Err,3=No Key]
66	Kypd	Input	Digital	Keypad Input 4 [0=Up,1=Down,2=Err,3=No Key]
67	Kypd	Input	Digital	Keypad Input 5 [0=Up,1=Down,2=Err,3=No Key]
68	Kypd	Input	Digital	Keypad Input 6 [0=Up,1=Down,2=Err,3=No Key]
69	Kypd	Input	Digital	Keypad Input 7 [0=Up,1=Down,2=Err,3=No Key]
70	Kypd	Input	Digital	Keypad Input 8 [0=Up,1=Down,2=Err,3=No Key]
71	Kypd	Output	6-bit	Keypad Out1 LEDs [0=Off,1=On,2=Blink,3=NoChg]
72	Kypd	Output	6-bit	Keypad Out2 LEDs [0=Off,1=On,2=Blink,3=NoChg]
73	Kypd	Output	6-bit	Keypad Out3 LEDs [0=Off,1=On,2=Blink,3=NoChg]
74	Kypd	Output	6-bit	Keypad Out4 LEDs [0=Off,1=On,2=Blink,3=NoChg]
75	Kypd	Output	6-bit	Keypad Out5 LEDs [0=Off,1=On,2=Blink,3=NoChg]
76	Kypd	Output	6-bit	Keypad Out6 LEDs [0=Off,1=On,2=Blink,3=NoChg]
77	Kypd	Output	6-bit	Keypad Out7 LEDs [0=Off,1=On,2=Blink,3=NoChg]
78	Kypd	Output	6-bit	Keypad Out8 LEDs [0=Off,1=On,2=Blink,3=NoChg]

Keypad functionality can be used in XML mode in a similar way to Keypad I/O Mode. Keypad configuration is identical to Keypad I/O Mode. The difference being that in XML mode the buttons do not directly control the outputs. The keypad button state is instead stored in CAN 63-70. The state is dependent on the button configuration and the keypad LEDs can be controlled either automatically or by writing to CAN 71-78. See section 3.3.4 Keypad Mode for more information on how to initialize keypad values.

Temporary Variables

Index	Description	
0	FALSE	0 value
1	TRUE	1 value
2	Variables available to user to store values	
3		
.		
.		
.		
.		
126		
127		
128		

 ©2023 Marlin Technologies, Inc

Params Variables

Index	Description
0	Variables available to user to store values
1	
2	
3	
.	
.	
.	
.	
126	
127	
128	

3.3.1.2.3 Example

Figure 15 shows an example of a simple vehicle application using logic symbols. Following that is the same implementation using Configurator instructions. The blue numbers in Figure 15 correlate to the blue numbers in the Configurator implementation, to show equivalent operations between the two implementations. The application checks if there are any E-Stop signals (emergency stop) from various sources and then, if necessary, activates an E-Stop indicator light, sends a CAN message commanding the TCU (traction control unit) to neutral, sends a CAN message commanding the ECU (engine control unit) to stop the engine, and kills the throttle signal.

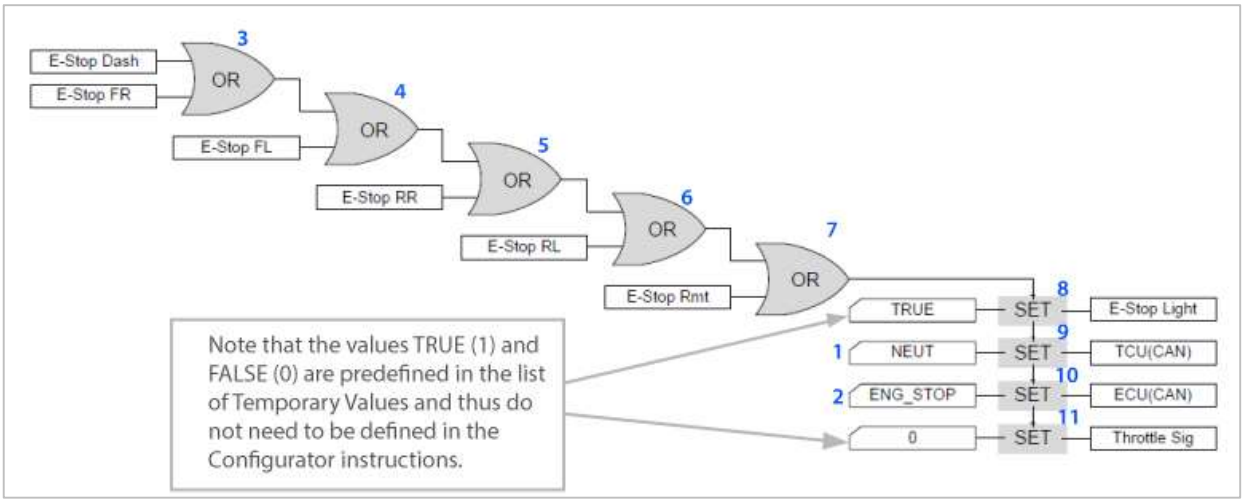


Figure 15

 ©2023 Marlin Technologies, Inc

Configurator Implementation

	Variable A	Operator	Variable B		Variable C (or fixed value)	Comment
	Physical #63	SET VARIABLE		=	0	XML mode
1	Temp #2	SET VARIABLE		=	NEUT	Neutral value
2	Temp #3	SET VARIABLE		=	ENG_STOP	Engine stop
3	E-Stop Dash	Logical OR	E-Stop FR	=	Temp #4	Check e-stops.
4	Temp #4	Logical OR	E-Stop FL	=	Temp #4	
5	Temp #4	Logical OR	E-Stop RR	=	Temp #4	
6	Temp #4	Logical OR	E-Stop RL	=	Temp #4	
7	Temp #4	Logical OR	E-Stop Rmt	=	Temp #4	
8	Temp #4	IF(A)=TRUE, then (B) -> (C)	Temp #1 (TRUE/1)	=	E-Stop Light	Activate E-stop indicator light.
9	Temp #4	IF(A)=TRUE, then (B) -> (C)	Temp #2 (NEUT)	=	TCU(CAN)	Command TCU to neutral.
10	Temp #4	IF(A)=TRUE, then (B) -> (C)	Temp #3 (ENG_STOP)	=	ECU(CAN)	Command ECU to stop engine.
11	Temp #4	IF(A)=TRUE, then (B) -> (C)	Temp #0 (FALSE/0)	=	ThrottleSig	Kill throttle signal.

For the sake of simplicity, parameters in the orange cells are not shown to be referenced using a variable type and index number, nor is the configuration related to such parameters (CAN messages, output type, etc) shown in the configuration portion of the program. In an actual implementation, the E-Stop parameters may be read from bytes of a received CAN message. The E-Stop light may driven by a digital or PWM output. And so on.

3.3.2 CAN I/O Mode

3.3.2.1 Mode Description

In CAN I/O mode, the 505030 module is commanded by another Marlin 5050xx controller via CAN communication. The 505030 module reports its input statuses, while the commanding module commands the 505030 module to turn its outputs on or off. The CAN messages involved in this functionality are included below for reference. While the 505030 module automatically handles these CAN messages, some CAN functionality is configurable (e.g. the transmit rate for each message, and whether they are transmitted at all).

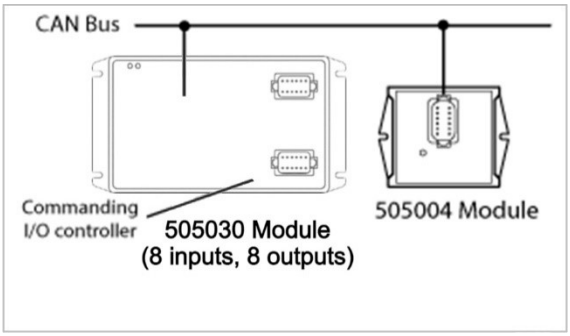


Figure 16

 ©2023 Marlin Technologies, Inc

Transmitted CAN messages:

Transmitted by 505030

	PGN	LSB							MSB
Digital Status Msg	18FF40sa	In1-4(Hi)	In1-4(Lo)	In5-8(Hi)	In5-8(Lo)	0x00	0x00	0x00	0x00
		2-bit Status Flags[00=False, 01=True, 10=Error, 11=Not Applicable]							
Analog Status Msg	18FF41sa	Input Analog 1		Input Analog 2		Input Analog 3		Input Analog 4	
		Voltage/Amps/Ohms/Count (Mode Based) [1mV,uA,Ω/bit]							
Analog Status Msg	18FF42sa	Input Analog 5		Input Analog 6		Input Analog 7		Input Analog 8	
		Voltage/Amps/Ohms/Count (Mode Based) [1mV,uA,Ω/bit]							
Analog Status Msg	18FF43sa	Input 1 Frequency		Input 2 Frequency		Input 1 Duty Cycle		Input 2 Duty Cycle	
		Frequency: 32.0-10,000.0Hz [0.1Hz/bit]				Duty Cycle: 0.0-100.0% [0.1%/bit]			
Analog Status Msg	18FF43sa	Input 3 Frequency		Input 4 Frequency		Input 3 Duty Cycle		Input 4 Duty Cycle	
		Frequency: 32.0-10,000.0Hz [0.1Hz/bit]				Duty Cycle: 0.0-100.0% [0.1%/bit]			
Module Status Msg	18FF60sa	Supply Voltage		System Temperature		Sys. Internal Ref V		Module HWID	
Output Fault Msg	18FF61sa	Out1Err	Out2Err	Out3Err	Out4Err	Out5Err	Out6Err	Out7Err	Out8Err
Output Setpoint Msg	18FF63sa	Out 1 Cmd (16-bit)		Out 2 Cmd (16-bit)		Out 3 Cmd (16-bit)		Out 4 Cmd (16-bit)	
		Current: 0-4200 [0-4,200mA, 1mA/bit]							
Output Feedback Msg	18FF64sa	Out 5 Cmd (16-bit)		Out 6 Cmd (16-bit)		Out 7 Cmd (16-bit)		Out 8 Cmd (16-bit)	
		Current: 0-4200 [0-4,200mA, 1mA/bit]							
Output Setpoint Msg	18FF65sa	Output 1 Current		Output 2 Current		Output 3 Current		Output 4 Current	
		Current: 0-4200 [0-4,200mA, 1mA/bit]							
Output Feedback Msg	18FF66sa	Output 5 Current		Output 6 Current		Output 7 Current		Output 8 Current	
		Current: 0-4200 [0-4,200mA, 1mA/bit]							

 ©2023 Marlin Technologies, Inc

Fault Status – Detailed View

Out1Err	Out2Err	Out3Err	Out4Err	Out5Err	Out6Err	Out7Err	Out8Err

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Timeout	Fault Flag	Output Over-voltage Fault	Output Under-voltage Fault	Power Under Voltage Fault	Over Temperature Fault	Over Temperature Warning	Over-Current/Short Circuit Fault

*All 1's (0xFF) means output is inactive (not configured)

**If Bit 6 is a 1, then Bit 0 is a short circuit fault. If Bit 6 is 0, then Bit 0 is an over-current fault.

Binary Active Hi/Lo Status – Detailed View

In1-4(Hi)	In1-4(Lo)	In5-8(Hi)	In5-8(Lo)	0x00	0x00	0x00	0x00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input 4		Input 3		Input 2		Input 1	

Note: Each input has a 2-bit flag: 00 – Input is Digital Low
01 – Input is Digital High
1X – Error

 ©2023 Marlin Technologies, Inc

Received CAN messages:

Received by 505030

	PGN	LSB							MSB
Cmd Msg: Outputs	18EFsaxx	0x60	0xFF	0x00	0x00	Out 1-4	Out 5-8	Reserved	Reserved
		0x61	0xFF	0x00	0x00	Out1 Cmd	Out2 Cmd	Out3 Cmd	Out4 Cmd
					0x01	Out5 Cmd	Out6 Cmd	Out7 Cmd	Out8 Cmd
						DC: 0-250 [0-100.0%, 0.4%/bit]			
		0x62	0xFF	0x00	0x00	Out 1 Cmd (16-bit)*		Out 2 Cmd (16-bit)*	
					0x01	Out 3 Cmd (16-bit)*		Out 4 Cmd (16-bit)*	
					0x02	Out 5 Cmd (16-bit)*		Out 6 Cmd (16-bit)*	
					0x03	Out 7 Cmd (16-bit)*		Out 8 Cmd (16-bit)*	
						DC: 0-1000 [0-100.0%, 0.1%/bit] mA: 0-3000 [0-3000mA, 1mA/bit]			
		0x70	0xFF	0x00	0x00	In 1 Quad Cnt Cmd*		In 2 Quad Cnt Cmd*	
					0x01	In 3 Quad Cnt Cmd*		In 4 Quad Cnt Cmd*	
					0x02	In 5 Quad Cnt Cmd*		In 6 Quad Cnt Cmd*	
					0x03	In 7 Quad Cnt Cmd*		Reserved (leave as 0x00)	
						Count Setpoint [0-65535]			
Cmd Msg: Beacon On	18EFsaF9	0x15	0xFF	0xF9	0xB0	0x44	0x55	0x66	0x77
Cmd Msg: Beacon Off	18EFsaF9	0x16	0xFF	0xF9	0xB0	0x44	0x55	0x66	0x77

Binary Active Hi/Lo Control – Detailed View

0x60	0xFF	0x00	0x00	Out 1-4	Out 5-8	Reserved	Reserved
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Output 4		Output 3		Output 2		Output 1	

Note: Each output cmd is a 2-bit flag: 00 – OFF
01 – ON
10 – Reserved
11 – Reserved

Notes:
- Status messages are sent every 100mS
- sa = Source Address of the 505030 Module
- xx = Source Address of the controller commanding the 505030 Module
- Beacon: Flashes the status LED so that the commanded module can be identified
- All multi-byte data is little endian

Outputs (and inputs in certain circumstances) are commanded via the PGN 0xEF00 when the module is configured in CAN I/O mode. The command operation is specified by the first data byte in the message.

The command operation 0x70 acts as a means to reset or initialize the quadrature count for inputs configured as quadrature. This command will have no effect on inputs not configured as quadrature, nor inputs acting as Quadrature B. This command does affect the associated interval count. At the next 0xFF41 or 0xFF42 transmission, the interval count will reflect any change in total count caused by using this command.

 ©2023 Marlin Technologies, Inc

3.3.2.2 Configuring for CAN I/O Mode

To configure the various module parameters involved in CAN I/O Mode, open a Configurator window as described in section *3.3 Configuring the Module*. Use the ADD button to add *SET VARIABLE* instructions for each parameter listed in the *List of CAN I/O Mode Parameters* below. Refer to Figure 17 as a guide for filling out the *SET VARIABLE* instructions. If necessary, see section *3.3 Configuring the Module* for more details about adding *SET VARIABLE* instructions.

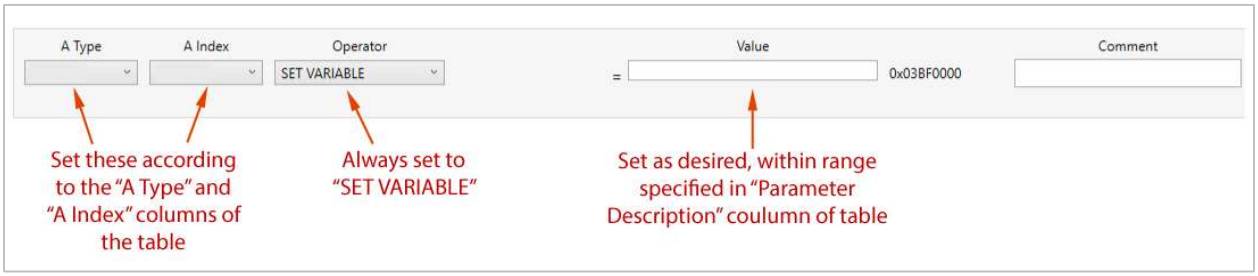


Figure 17

List of CAN I/O Mode Parameters

A Type	A Index	Parameter Description
Physical	117	Module Mode. Must be 1 for CAN I/O Mode.
Physical	78	Threshold Voltage for Digital inputs to be set to Active [65.535mV, 1mV/bit]
Physical	79	Hysteresis Voltage for Digital inputs to return to Inactive [65.535mV, 1mV/bit]
Physical	80	P-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	81	I-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	82	D-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	99	Frequency for all PWM Outputs (30-1000Hz)
Physical	100	Output 1 Type [1]
Physical	101	Output 2 Type [1]
Physical	102	Output 3 Type [1]
Physical	103	Output 4 Type [1]
Physical	104	Output 5 Type [1]
Physical	105	Output 6 Type [1]
Physical	106	Output 7 Type [1]
Physical	107	Output 8 Type [1]
Physical	62	Input 1 Type[2]
Physical	63	Input 2 Type[2]
Physical	64	Input 3 Type[2]
Physical	65	Input 4 Type[2]
Physical	66	Input 5 Type[2]
Physical	67	Input 6 Type[2]
Physical	68	Input 7 Type[2]
Physical	69	Input 8 Type[2]
Physical	70	Output 1 Current Limit [1 mA/bit, 0-3,000mA]
Physical	71	Output 2 Current Limit [1 mA/bit, 0-3,000mA]
Physical	72	Output 3 Current Limit [1 mA/bit, 0-3,000mA]
Physical	73	Output 4 Current Limit [1 mA/bit, 0-3,000mA]
Physical	74	Output 5 Current Limit [1 mA/bit, 0-3,000mA]
Physical	75	Output 6 Current Limit [1 mA/bit, 0-3,000mA]
Physical	76	Output 7 Current Limit [1 mA/bit, 0-3,000mA]

 ©2023 Marlin Technologies, Inc

A Type	A Index	Parameter Description
Physical	77	Output 8 Current Limit [1 mA/bit, 0-3,000mA]
Physical	54	Output 1 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	55	Output 2 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	56	Output 3 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	57	Output 4 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	58	Output 5 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	59	Output 6 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	60	Output 7 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
Physical	61	Output 8 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [3]
CAN	48	Tx Rate of Msg FF40 [10mS/bit, 0=Off]
CAN	49	Tx Rate of Msg FF41 [10mS/bit, 0=Off]
CAN	50	Tx Rate of Msg FF60 [10mS/bit, 0=Off]
CAN	51	Tx Rate of Msg FF61 [10mS/bit, 0=Off]
CAN	52	Tx Rate of Msg FF62 [10mS/bit, 0=Off]
CAN	53	Tx Rate of Msg FF63 [10mS/bit, 0=Off]
CAN	54	Tx Rate of Msg FF64 [10mS/bit, 0=Off]
CAN	55	Tx Rate of Msg FF65 [10mS/bit, 0=Off]
CAN	56	Tx Rate of Msg FF66 [10mS/bit, 0=Off]
CAN	57	Tx Rate of Msg FF67 [10mS/bit, 0=Off]
CAN	58	Tx Rate of Msg FF68 [10mS/bit, 0=Off]
CAN	79	Cmd Message Timeouts [10mS/bit, 0=No Timeout]

1 0 = Off [Default]
1 = High Side Digital (Active High)
2 = Low Side Digital (Active Low)
3 = High Side Current (mA)
4 = High Side PWM (Duty Cycle)
5 = Low Side Current (mA)
6 = Low Side PWM (Duty Cycle)
7 = Bi-Directional (Half-Bridge) *All 505030 outputs are half-bridge by default
8 = Voltage

2 0 = Digital/Analog [Default]
1 = Current
2 = Resistance
3 = Quadrature X1
4 = Quadrature X2
5 = Quadrature X4

3 This parameter only applies if the corresponding output is configured as Current or PWM, per Physical 100-107.

When finished adding instructions in the Configurator, click “Save XML” in the bottom-right corner of the window (as seen in Figure 8) to save the data to an .XML file. To load this data on to the 505030 module, follow the instructions in section *3.4 Loading the User Program*.

 ©2023 Marlin Technologies, Inc

3.3.2.3 CAN I/O Configuration Example

Configuration Example for CAN I/O Mode

Configurator Instruction				Result
Operator	**A Type**	**A Index**	**Value**	
SET VARIABLE	Physical	117	1	Module Mode = CAN I/O Mode
SET VARIABLE	Physical	100	1	Output 1 is a digital active high output
SET VARIABLE	Physical	101	2	Output 2 is a digital active low output
SET VARIABLE	Physical	102	3	Output 3 is an active high PWM closed loop output
SET VARIABLE	Physical	103	6	Output 4 is an active low PWM open loop output
SET VARIABLE	Physical	99	100	Output PWM Frequency = 100Hz
SET VARIABLE	Physical	56	250	Output 3 current setpoint is 250mA
SET VARIABLE	Physical	57	750	Output 4 Duty cycle is 75.0%
SET VARIABLE	Physical	62	1	Input 1 is a current input
SET VARIABLE	Physical	78	2000	Input digital Thresholds = Vbatt-2V & GND+2V
SET VARIABLE	Physical	79	1000	Input digital hysteresis = 1V
SET VARIABLE	Physical	80	2500	P-Term = 2500
SET VARIABLE	Physical	81	750	I-Term = 750
SET VARIABLE	Physical	82	0	D-Term = 0
SET VARIABLE	CAN	79	100	CAN Timeout = 1s

 ©2023 Marlin Technologies, Inc

3.3.3 Hardware I/O Mode

3.3.3.1 Hardware I/O Mode Description

In Hardware I/O Mode, the module functions like a relay module, in which input signals drive corresponding output signals. When a given input is connected to power or ground, the input state is set to TRUE. Then, after the user-specified On-Delay time (Td_On in Figure 18), the state of the corresponding output changes to TRUE (for digital outputs, that means the output turns on; for PWM outputs or current outputs, that means the output is set to the specified output Duty Cycle or Output Current, respectively (see Physical parameters 54-61 in *List of Hardware* I/O *Mode Parameters* below). When power or ground is removed from the input, the input floats at

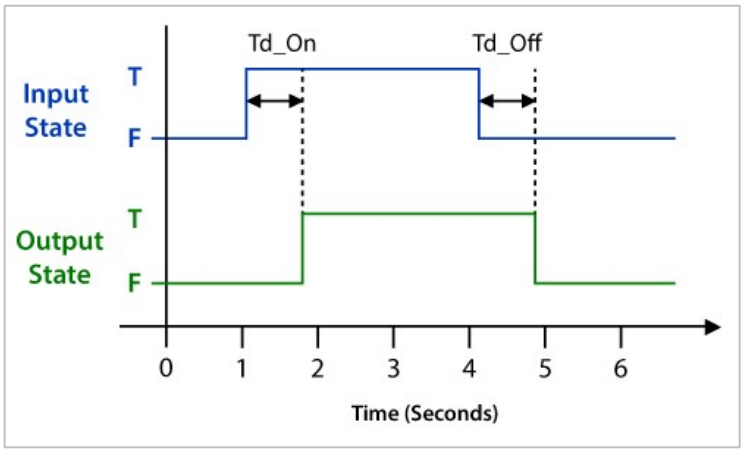


Figure 18

half the system voltage, which sets the input state to FALSE. After the user-specified Off_Delay time (Td_Off in Figure 18), the output state then changes to FALSE.

Configuration of inputs is disabled in this mode and all inputs behave as digital inputs.

3.3.3.2 Configuring for Hardware I/O Mode

To configure the various module parameters involved in Hardware I/O Mode, open a Configurator window as described in section *3.3 Configuring the Module*. Use the ADD button to add *SET VARIABLE* instructions for each parameter listed in the *List of Hardware* I/O *Mode Parameters* below. Refer to Figure 19 as a guide for filling out the *SET VARIABLE* instructions. If

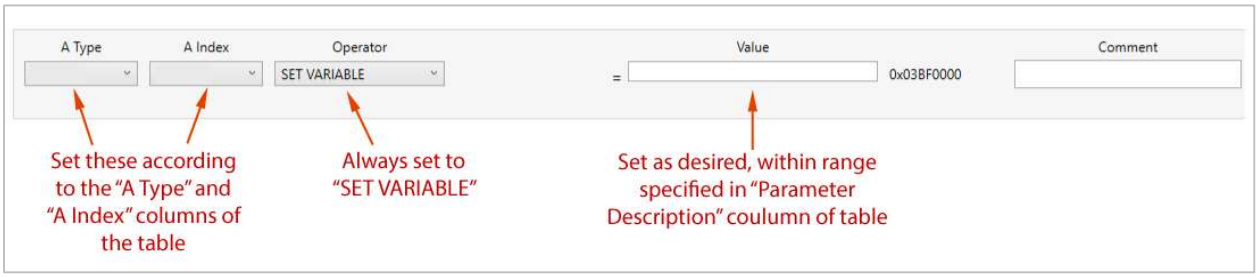


Figure 19

 ©2023 Marlin Technologies, Inc

necessary, see section *3.3 Configuring the Module* for more details about adding *SET VARIABLE* instructions.

List of Hardware I/O Mode Parameters

A Type	A Index	Parameter Description
Physical	117	Module Mode. Must be 2 for Hardware I/O Mode.
Physical	100	Output 1 Type [1]
Physical	101	Output 2 Type [1]
Physical	102	Output 3 Type [1]
Physical	103	Output 4 Type [1]
Physical	104	Output 5 Type [1]
Physical	105	Output 6 Type [1]
Physical	106	Output 7 Type [1]
Physical	107	Output 8 Type [1]
Physical	99	Frequency for all PWM Outputs (30-1000Hz)
Physical	78	Threshold Voltage for Digital inputs to be set to Active [65.535mV, 1mV/bit]
Physical	79	Hysteresis Voltage for Digital inputs to return to Inactive [65.535mV, 1mV/bit]
Physical	80	P-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	81	I-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	82	D-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	70	Output 1 Current Limit [1 mA/bit, 0-3000mA]
Physical	71	Output 2 Current Limit [1 mA/bit, 0-3000mA]
Physical	72	Output 3 Current Limit [1 mA/bit, 0-3000mA]
Physical	73	Output 4 Current Limit [1 mA/bit, 0-3000mA]
Physical	74	Output 5 Current Limit [1 mA/bit, 0-3000mA]
Physical	75	Output 6 Current Limit [1 mA/bit, 0-3000mA]
Physical	76	Output 7 Current Limit [1 mA/bit, 0-3000mA]
Physical	77	Output 8 Current Limit [1 mA/bit, 0-3000mA]
Physical	54	Output 1 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	55	Output 2 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	56	Output 3 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	57	Output 4 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	58	Output 5 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	59	Output 6 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	60	Output 7 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	61	Output 8 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	83	Relay-Style On-Delay for Output 1 [0-655,350mS, 10mS/bit]
Physical	84	Relay-Style On-Delay for Output 2 [0-655,350mS, 10mS/bit]
Physical	85	Relay-Style On-Delay for Output 3 [0-655,350mS, 10mS/bit]
Physical	86	Relay-Style On-Delay for Output 4 [0-655,350mS, 10mS/bit]
Physical	87	Relay-Style On-Delay for Output 5 [0-655,350mS, 10mS/bit]
Physical	88	Relay-Style On-Delay for Output 6 [0-655,350mS, 10mS/bit]
Physical	89	Relay-Style On-Delay for Output 7 [0-655,350mS, 10mS/bit]
Physical	90	Relay-Style On-Delay for Output 8 [0-655,350mS, 10mS/bit]
Physical	91	Relay-Style Off-Delay for Output 1 [0-655,350mS, 10mS/bit]
Physical	92	Relay-Style Off -Delay for Output 2 [0-655,350mS, 10mS/bit]
Physical	93	Relay-Style Off -Delay for Output 3 [0-655,350mS, 10mS/bit]
Physical	94	Relay-Style Off -Delay for Output 4 [0-655,350mS, 10mS/bit]
Physical	95	Relay-Style Off -Delay for Output 5 [0-655,350mS, 10mS/bit]
Physical	96	Relay-Style Off -Delay for Output 6 [0-655,350mS, 10mS/bit]

 ©2023 Marlin Technologies, Inc

A Type	A Index	Parameter Description
Physical	97	Relay-Style Off -Delay for Output 7 [0-655,350mS, 10mS/bit]
Physical	98	Relay-Style Off -Delay for Output 8 [0-655,350mS, 10mS/bit]

[1] 0 = Off [Default]
1 = High Side Digital (Active High)
2 = Low Side Digital (Active Low)
3 = High Side Current (mA)
4 = High Side PWM (Duty Cycle)
5 = Low Side Current (mA)
6 = Low Side PWM (Duty Cycle)
7 = Bi-Directional (Half-Bridge) *All 505030 outputs are half-bridge by default
8 = Voltage

[2] This parameter only applies if the corresponding output is configured as Current or PWM, per Physical 100-107.

When finished adding instructions in the Configurator, click “Save XML” in the bottom-right corner of the window (as seen in Figure 8). to save the data to an .XML file. To load this data on to the 505030 module, follow the instructions in section 3.4 Loading the User Program.

3.3.3.3 Hardware I/O Configuration Example

Configuration Example for Hardware I/O Mode

Configurator Instruction				
Operator	**A Type**	**A Index**	**Value**	**Result**
SET VARIABLE	Physical	117	2	Module Mode = Hardware I/O Mode
SET VARIABLE	Physical	100	1	Output 1 is an active high digital output
SET VARIABLE	Physical	101	2	Output 2 is an active low digital output
SET VARIABLE	Physical	102	3	Output 3 is an active high PWM closed loop output
SET VARIABLE	Physical	103	6	Output 4 is an active low PWM open loop output
SET VARIABLE	Physical	99	100	Output PWM Frequency = 100Hz
SET VARIABLE	Physical	56	250	Output 3 current setpoint is 250mA
SET VARIABLE	Physical	57	750	Output 4 Duty cycle is 75.0%
SET VARIABLE	Physical	83	100	Output 1 On-Delay is 1.0 seconds.
SET VARIABLE	Physical	90	150	Output 1 Off-Delay is 1.5 seconds.
SET VARIABLE	Physical	84	200	Output 2 On-Delay is 2.0 seconds.

 ©2023 Marlin Technologies, Inc

Configurator Instruction				Result
Operator	A Type	A Index	Value	
SET VARIABLE	Physical	91	250	Output 2 Off-Delay is 2.5 seconds.
SET VARIABLE	Physical	85	300	Output 3 On-Delay is 3.0 seconds.
SET VARIABLE	Physical	92	350	Output 3 Off-Delay is 3.5 seconds.
SET VARIABLE	Physical	86	400	Output 4 On-Delay is 4.0 seconds.
SET VARIABLE	Physical	93	450	Output 4 Off-Delay is 4.5 seconds.
SET VARIABLE	Physical	78	2000	Input digital Thresholds = Vbatt-2V & GND+2V
SET VARIABLE	Physical	79	1000	Input digital hysteresis = 1V
SET VARIABLE	Physical	80	2500	P-Term = 2500
SET VARIABLE	Physical	81	750	I-Term = 750
SET VARIABLE	Physical	82	0	D-Term = 0

3.3.4 Keypad Mode

3.3.4.1 Keypad Mode Description

In Keypad mode, the 505030 Module's outputs are driven by button presses on a Marlin 5052xx keypad. LEDs on each button are turned on or off, giving the operator visual feedback when a button is pressed.

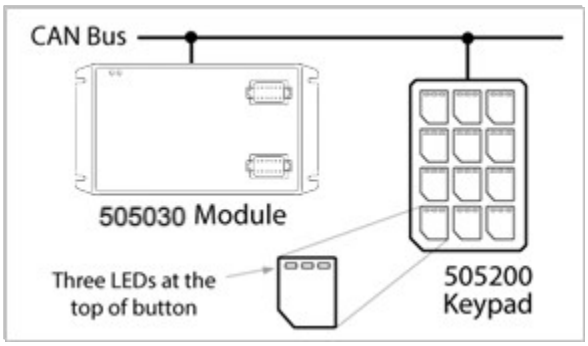


Figure 20

Individual keypad buttons can be assigned to drive individual outputs on the 505030 module. The type of output (digital, current, PWM) can also be assigned. Furthermore, the button's functionality, can be configured to be one of three types:

- **Momentary:** The output is set to its On state for as long as the button is held down. When the button is released, the output resumes its Off state. All the button's LEDs are On when the output is On, and Off when the output is Off.
    - Note: for outputs configured as PWM or Current, the On state is defined by the output's respective *Output Duty Cycle/Current* parameter. See Physical parameters 54-61 in the *List of Keypad Mode Parameters* below.
- **On/Off Toggle:** The output is toggled between its On and Off state each time the button is pushed downwards. For example, with the output initially off, the operator pushes the

 ©2023 Marlin Technologies, Inc

button down and thus causes the output to turn On. The output remains On as the operator releases the button. The output is turned Off when the operator again presses the button down, and remains Off as the button is released, and so on. All the button's LEDs are On when the output is On, and Off when the output is Off.

  - Note: for outputs configured as PWM or Current, the On state is defined by the output's respective *Output Duty Cycle/Current* parameter. See Physical parameters 54-61 in the *List of Keypad Mode Parameters* below.

- **Off/Low/Medium/High (O/L/M/H**): This functionality applies only to outputs configured as PWM or Current (i.e. not Digital). As with On/Off Toggle functionality, the output state changes only when the button is pressed down (and not when the button is released). But instead of the output simply turning on or off, a button press causes the output signal to cycle between four different states: Off (0% of the fully-on state), Low (33% of the fully-on state), Medium (66% of the fully-on state), and High (100% of the fully-on state). The "fully-on state" is defined by the output's respective *Output Duty Cycle/Current* parameter. See Physical parameters 54-61 in the *List of Keypad Mode Parameters* below.

The three types of button functionality are represented visually in Figure 21.

 ©2023 Marlin Technologies, Inc

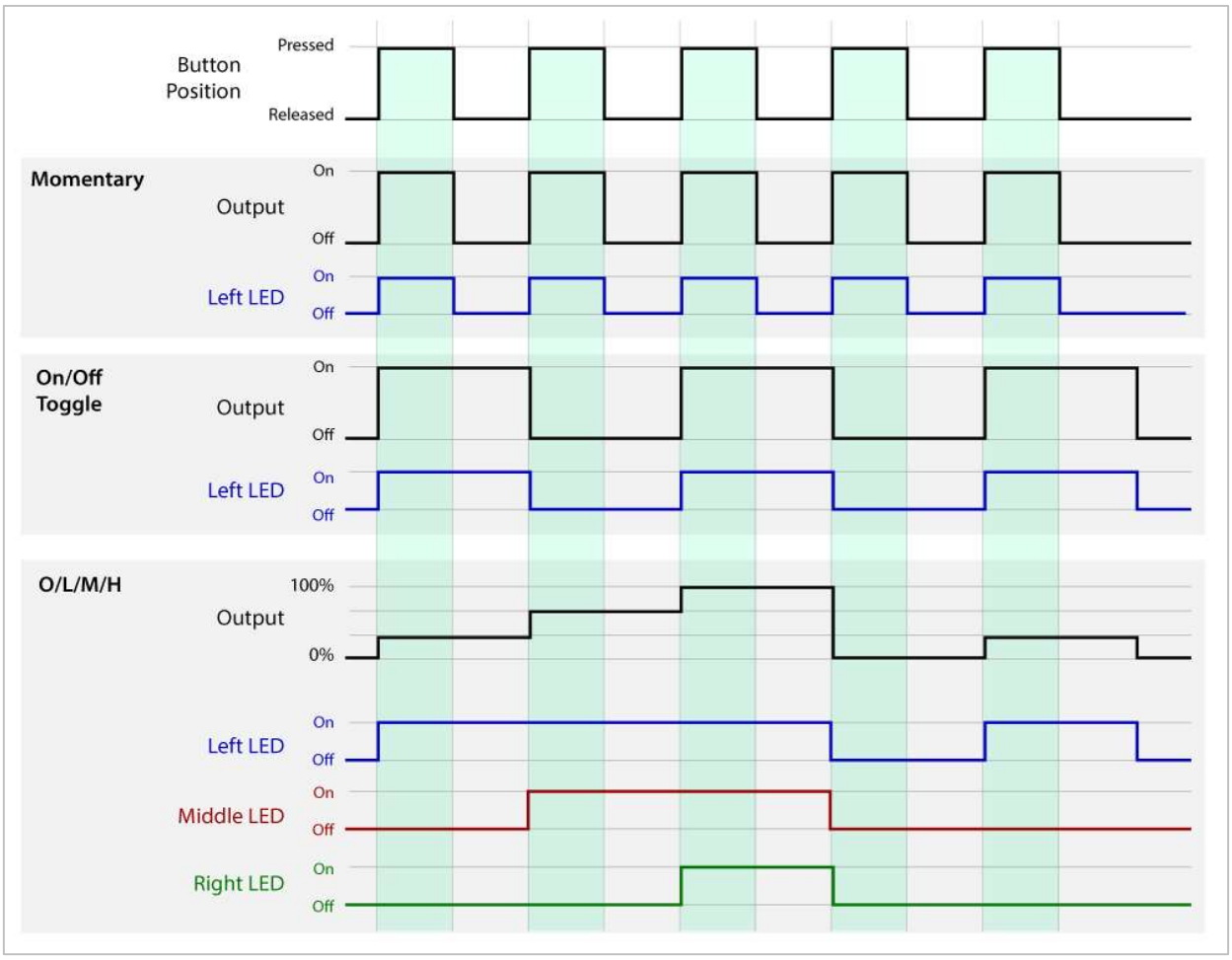


Figure 21

The 505030 module receives CAN messages from the keypad indicating when buttons are pressed. The CAN messages involved are included below for reference. While the 505030 module automatically handles these CAN messages, some CAN functionality is configurable.

CAN messages Transmitted by 505030:

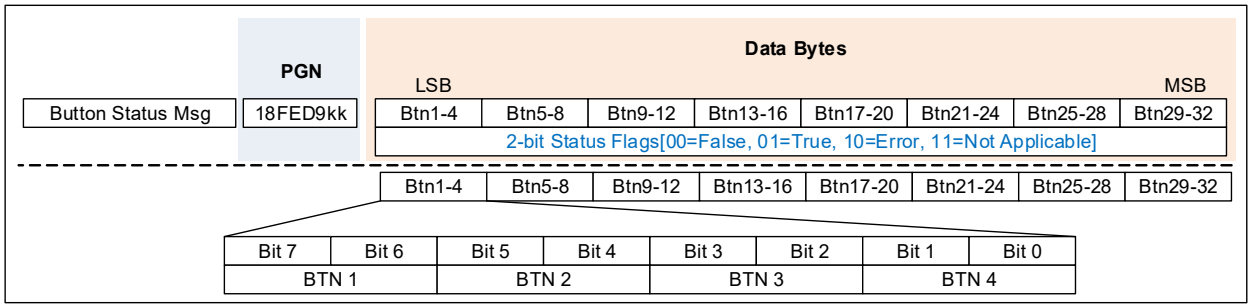
	PGN	Data Bytes (LSB → MSB)							
Indicator LED Msg	18A7kksa	LED1-4	LED5-8	LED9-12	LED13-16	LED17-20	LED21-24	LED25-28	LED29-32
		2-bit flags [00=Off, 01=On, 10=Blink 2Hz, 11=No Change]							
Indicator LED Msg	18A6kksa	LED33-36	LED37-40	LED41-44	LED45-48	LED49-52	LED53-56	LED57-60	LED61-64
		2-bit flags [00=Off, 01=On, 10=Blink 2Hz, 11=No Change]							

LED1-4	LED5-8	LED9-12	LED13-16	LED17-20	LED21-24	LED25-28	LED29-32

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LED 1		LED 2		LED 3		LED 4	

 ©2023 Marlin Technologies, Inc

CAN messages Received by 505030:



Notes:

- Indicator LED Status Messages are sent every 100mS.
- sa = Source Address of the 505030 Module
- kk = Source Address of the Keypad Module
- See the 11713S_ specification for Marlin Keypad Modules

3.3.4.2 Configuring for Keypad Mode

To configure the various module parameters involved in Keypad Mode, open a Configurator window as described in section *3.3 Configuring the Module*. Use the ADD button add *SET VARIABLE* instructions for each parameter listed in the *List of Keypad Mode Parameters* below. Refer to Figure 22 as a guide for filling out the *SET VARIABLE* instructions. If necessary, see section *3.3 Configuring the Module* for more details about adding *SET VARIABLE* instructions.

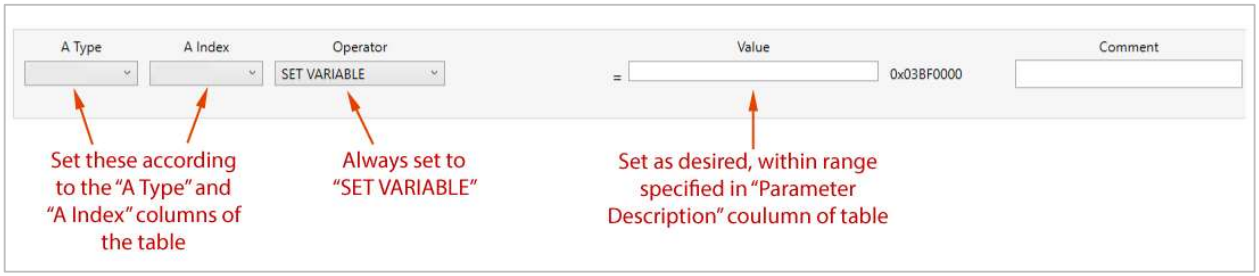


Figure 22

List of Keypad Mode Parameters

A Type	A Index	Parameter Description
Physical	117	Module Mode. Must be 3 for Keypad Mode.
Physical	100	Output 1 Type [1]
Physical	101	Output 2 Type [1]
Physical	102	Output 3 Type [1]
Physical	103	Output 4 Type [1]
Physical	104	Output 5 Type [1]
Physical	105	Output 6 Type [1]
Physical	106	Output 7 Type [1]
Physical	107	Output 8 Type [1]
Physical	70	Output 1 Current Limit [1 mA/bit, 0-3,000mA]
Physical	71	Output 2 Current Limit [1 mA/bit, 0-3,000mA]

 ©2023 Marlin Technologies, Inc

A Type	A Index	Parameter Description
Physical	72	Output 3 Current Limit [1 mA/bit, 0-3,000mA]
Physical	73	Output 4 Current Limit [1 mA/bit, 0-3,000mA]
Physical	74	Output 5 Current Limit [1 mA/bit, 0-3,000mA]
Physical	75	Output 6 Current Limit [1 mA/bit, 0-3,000mA]
Physical	76	Output 7 Current Limit [1 mA/bit, 0-3,000mA]
Physical	77	Output 8 Current Limit [1 mA/bit, 0-3,000mA]
Physical	54	Output 1 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	55	Output 2 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	56	Output 3 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	57	Output 4 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	58	Output 5 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	59	Output 6 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	60	Output 7 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	61	Output 8 DutyCycle/Current setpoint Command [1mA or 0.1% /bit] [2]
Physical	80	P-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	81	I-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	82	D-term for Closed-Loop of all outputs [0-655.35%, 0.01%/bit]
Physical	108	Keypad Button Index and Button Type for Output 1[3]
Physical	109	Keypad Button Index and Button Type for Output 2[3]
Physical	110	Keypad Button Index and Button Type for Output 3[3]
Physical	111	Keypad Button Index and Button Type for Output 4[3]
Physical	112	Keypad Button Index and Button Type for Output 5[3]
Physical	113	Keypad Button Index and Button Type for Output 6[3]
Physical	114	Keypad Button Index and Button Type for Output 7[3]
Physical	115	Keypad Button Index and Button Type for Output 8[3]
Physical	116	Keypad Source Address
CAN	79	Cmd Message Timeouts [10mS/bit, 0=NoTimeout]

1 0 = Off [Default]
1 = High Side Digital (Active High)
2 = Low Side Digital (Active Low)
3 = High Side Current (mA)
4 = High Side PWM (Duty Cycle)
5 = Low Side Current (mA)
6 = Low Side PWM (Duty Cycle)
7 = Bi-Directional (Half-Bridge)
8 = Voltage

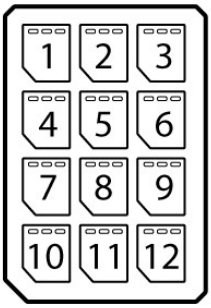
2 This parameter only applies if the corresponding output is configured as Current or PWM, per Physical 100-107. It determines the output level when a keypad button press commands the output to its On state.

 ©2023 Marlin Technologies, Inc

3 This parameter a) identifies which button on the keypad drives the given output (i.e. its Index Value), and b) specifies the button's functionality (i.e. its Type). The parameter's value is in the following 16-bit format:

Bits 15:10	Bits 9:8	Bits 7:0
Not Used	Button Type	Button Index Number

Button Index Number: Buttons on a keypad are typically numbered in order from left to right, top row to bottom as follows, as shown in the example to the right. Thus, the Index Number of the lower-left button on a the 3x4 keypad in the example would be 12.



Button Type: This following values select the types of button functionality as follows:

- 0 = Momentary
- 1 = On/Off Toggle
- 2 = Off/Low/Medium/High (for PWM and Current outputs only)

See above (including Figure 21) for details about the types of button functionality.

When finished adding instructions in the Configurator, click "Save XML" in the bottom-right corner of the window to save the data to an .XML file. To load this data on to the 505030 module, follow the instructions in section *3.4 Loading the User* Program.

3.3.4.3 Keypad Mode Configuration Example

Configuration Example for Keypad Mode

Configurator Instruction				Result
Operator	**A Type**	**A Index**	**Value**	
SET VARIABLE	Physical	117	3	Module Mode = Keypad Mode
SET VARIABLE	Physical	100	1	Output 1 is an active high digital output
SET VARIABLE	Physical	101	2	Output 2 is an active low digital output
SET VARIABLE	Physical	102	3	Output 3 is an active high PWM closed loop output
SET VARIABLE	Physical	103	6	Output 4 is an active low PWM open loop output
SET VARIABLE	Physical	56	250	Output 3 current setpoint is 250mA
SET VARIABLE	Physical	57	750	Output 4 Duty cycle is 75.0%
SET VARIABLE	Physical	116	64	Keypad source address is 64 (or 0x40)
SET VARIABLE	Physical	108	3	Keypad Button 3 drives Output 1; Type = Momentary
SET VARIABLE	Physical	109	1	Keypad Button 1 drives Output 2; Type = Momentary
SET VARIABLE	Physical	110	2	Keypad Button 2 drives Output 3; Type = Momentary

 ©2023 Marlin Technologies, Inc

Configurator Instruction				Result
Operator	A Type	A Index	Value	
SET VARIABLE	Physical	111	260	Keypad Button 4 drives Output 4; Type = Toggle
SET VARIABLE	Physical	80	2500	P-Term = 2500
SET VARIABLE	Physical	81	750	I-Term = 750
SET VARIABLE	Physical	82	0	D-Term = 0

3.4 Loading the User Program

Once a user program has been created in the Programming Tool's Configurator utility and saved to an .XML file, it can then be loaded to the 505030 module using the Programming Tool's EEPROM Read/Write window. With the Program Tool running and connected to the module (see section *3.1 Connecting to the Module with the Programming Tool* for details), click the menu item "XML", then click the item "Single," both shown circled in Figure 23.

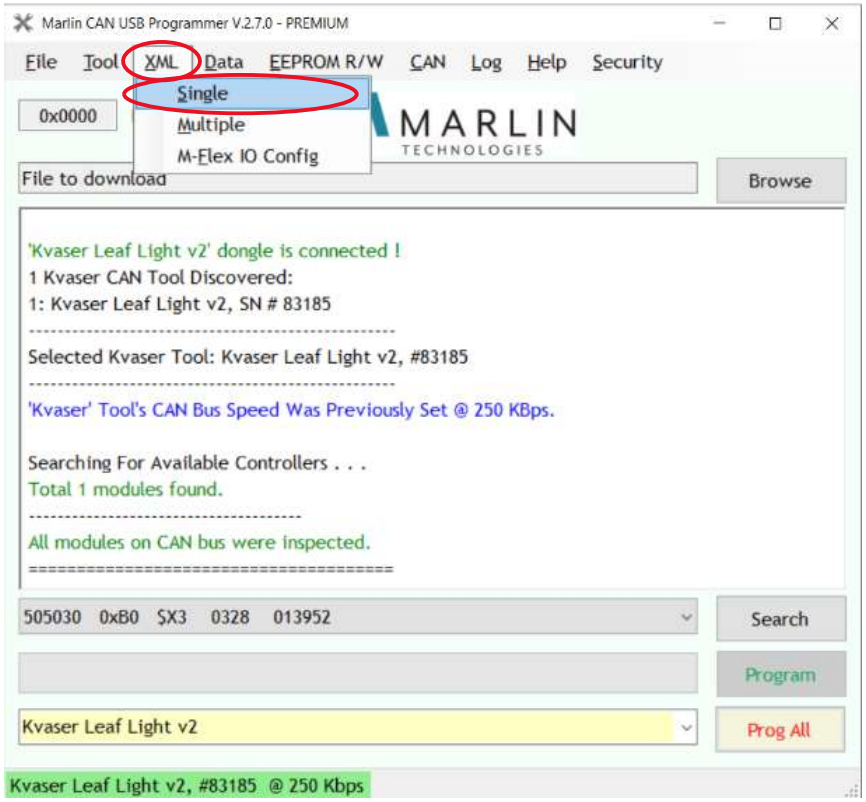


Figure 23

 ©2023 Marlin Technologies, Inc

A window will pop up in which the desired .XML file can be browsed and selected. After the file is selected, the EEPROM Read/Write will pop up, as show in Figure 24. If a different .XML file is desired, click on the yellow “Open XML” button near the top right corner and select a new file. Otherwise, locate source the address field on the right side of the window, just above the green buttons. Fill in the field with current source address of the 505030 module. (If the source address is unknown, use the Search button in the Programming Tool window, as shown in Figure 23, to request the module’s information. The source address in the figure is 0xB0.) Click the “Write to EEPROM button.” A pop-up warning will be displayed indicating the controller’s EEPROM data will be overwritten. Click “OK” to proceed. If the data from the .XML file is programmed successfully, the text “All Attribute were Successfully Written to EEPROM!” will appear (in green) in the text box at the bottom of the window (see Figure 24). If there are errors, consult the Programming Tool user guide mentioned in section *2.* Software Installation to troubleshoot.

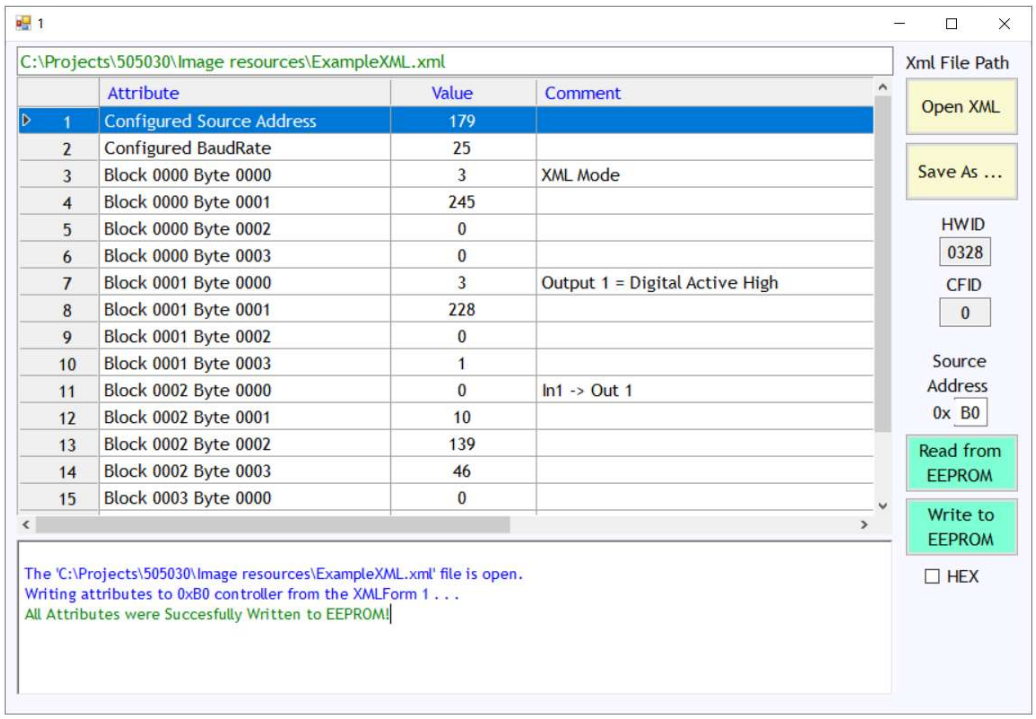


Figure 24

Following the successful write of all XML attributes, a system power cycle is required to apply any changes made.

 ©2023 Marlin Technologies, Inc

4. LED Module

The 505030 module has built-in support for the Marlin 5010xx LED Module, which can be used to monitor the module's I/O for diagnostic purposes. The module's I/O statuses are transmitted to the LED module via a CAN message. No configuration is necessary to enable the LED module to work with the 505030 module.

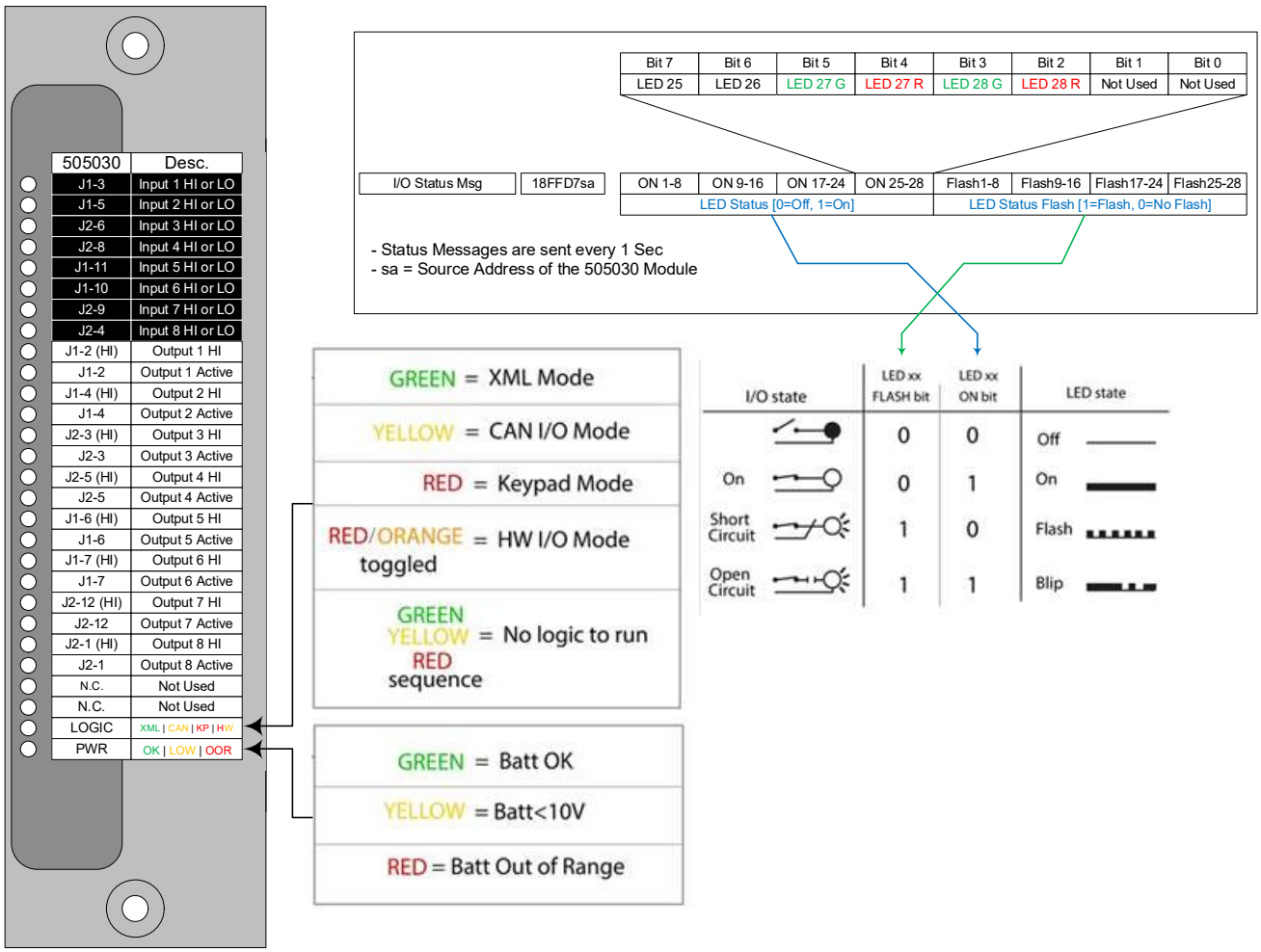


Figure 25

 ©2023 Marlin Technologies, Inc