User Manual UMAX030141

3 Encoder, 7 Signal Inputs Controller, CANopen®, Isolated

P/N: AX030141

In Europe:
Axiomatic Technologies Oy
Höytämöntie 6
33880 LEMPÄÄLÄ - Finland
Tel. +358 103 375 750
Fax. +358 3 3595 660
www.axiomatic.fi

In North America:
Axiomatic Technologies Corporation
5915 Wallace Street
Mississauga, ON Canada L4Z 1Z8
Tel. 1 905 602 9270
Fax. 1 905 602 9279
www.axiomatic.com

VERSION HISTORY

Version	Date	Author	Modification
1.00	March 24, 2020	Erik Sasse	Initial Draft
-	March 24, 2020	Amanda Wilkins	Marketing Review Added dimensional drawing
1.01	April 7, 2020	Erik Sasse	Updated drawing in chapter 5.4 Dimensions and Typical Connections

ACRONYMS

CAN	Controller Area Network
DM	Diagnostic message. Defined in J1939/73 standard
EA	Electronic Assistant. PC application software from Axiomatic, primarily designed to view and program Axiomatic control configuration parameters (setpoints) through CAN bus using J1939 Memory Access Protocol
ECU	Electronic control unit
EMI	Electromagnetic Interference
LIN	Local Interconnect Network. Automotive network maintained by the LIN Consortium
LSB	Less Significant Byte
PC	Personal Computer
PGN	Parameter Group Number. Defined in J1939/73 standard
P/N	Part Number
RS-232	PC serial port interface
CANopen	CAN-based higher level protocol designed and supported by CAN in Automation (CiA)
USB	Universal Serial Bus
UTP	Un-shielded twisted pair
BCD	Binary Code Decimal

Contents

1 INTRODUCTION

The following user manual describes: architecture, functionality, configuration parameters and flashing instructions for 3 Encoder, 7 Signal Inputs Controller, CANopen®, Isolated. It also contains technical specifications and installation instructions to help users build a custom solution on the base of this controller.

The user should check whether the application firmware installed in the controller is covered by this user manual. The user manual is valid for application firmware with the same major version number as the user manual. For example, this user manual is valid for any controller application firmware V1.xx. Updates specific to the user manual are done by adding letters: A, B, …, Z to the user manual version number.

The controller supports three Encoder Inputs, one universal input and six digital inputs. Its flexible hardware design allows the controller to have a wide range of input-types for each of the inputs. The sophisticated control algorithms/logical function blocks allow the user to configure the controller for a wide range of applications without the need for custom firmware.

This unit uses the CANopen standard protocol to transmit and receive data, which operates at a baud rate of 125 kbit/s by default.

The 3 Encoder, 7 Signal Inputs Controller, CANopen®, Isolated accepts power supply voltages from 9 to 60 VDC.

2 CONTROLLER DESCRIPTION

The converter is internally organized as a set of function blocks, which can be individually configured and arbitrarily connected to achieve the required system functionality.

The application parameters for this controller are used for the Digital Inputs, Universal Inputs and Encoder Inputs. In order to use the right input type, Table 1 shows what subindex is used for each input.

Sub-Index	Input Type
1	Encoder 1A
2	Encoder 1B
3	Encoder 2A
4	Encoder 2B
5	Encoder 3A
6	Encoder 3B
7	Universal Input
8	Digital Input 1
9	Digital Input 2
10	Digital Input 3
11	Digital Input 4
12	Digital Input 5
13	Digital Input 6

Table 1: Input Subindices

Furthermore, object AI Input range 2010h and AI Decimal Digits 2011h also having 13 subindices since they are used in combination with the application parameters. All other objects related to a certain input type having only the number of subindices as the amount existing of this input type. Input type related objects having therefore a specific numbering which is shown in Table 2.

Input Type	Index
Digital Input	21XXh
Universal Input	22XXh
Encoder Input	23XXh

Table 2: Input Indices Numbering

For example, all objects beginning with 22XXh are related to the universal input. The same principle applies for the other input types shown in the table above.

2.1 Encoder Input Function Block

The controller provides three Encoder inputs, each of them has two physical inputs for connection A and B. These inputs can be configured to measure digital signals generated by up to three incremental encoders.

Moreover, it is possible to use all 6 inputs of the encoder function block as Digital Inputs. For example, if only one encoder Input (two physical inputs) is used and more than six digital inputs are required, the other four inputs of the encoder function block can be used as additional digital inputs. To be able to set every input to a certain type, the most important object associated with the AI function block, object

6110h **AI Sensor Type**, needs to be set. By changing this value, other objects will be automatically updated by the controller. The options for object 6110h are shown in Table 3.

Value	Meaning
0	Input OFF
10002	Encoder Input
10003	Digital ON/OFF

Table 3: AI Sensor Type Options for Encoder Inputs

The controller has two inputs for detecting the pulses generated by an incremental encoder. The two measurements, step count and direction, are determined using these two input signals.

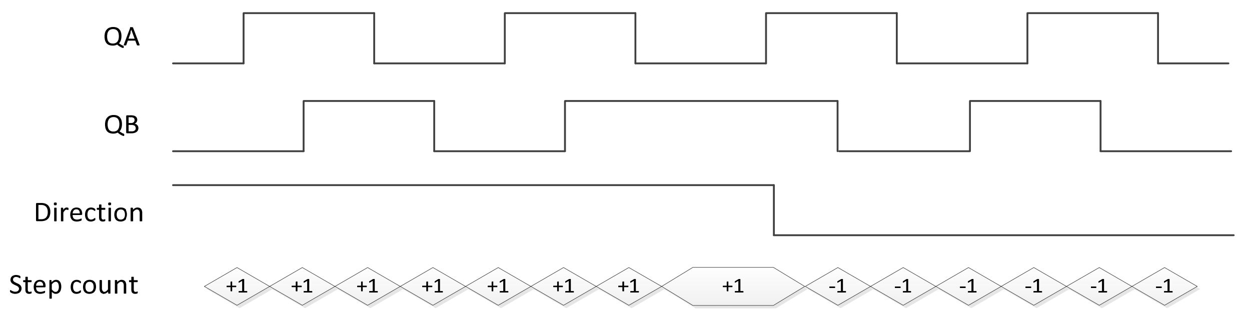


Figure 1: Incremental Decoder Signals (QA&QB) and resulting Direction and Step count

A positive direction is indicated when the edge of signal A rises before the edge of signal B is rising. If the encoder rotates in a positive direction, the **Encoder Counter value** increases, and this value is then be written to object 7100h **AI Input Field Value**. In case the signal changes and input B becomes the leading signal, the counter value decreases, and the direction is indicated as negative.

The encoder inputs have additionally a pullup and pulldown resistor which can be set with object 2313h **ENC A Pull UpDown Value** and object 2323h **ENC B Pull UpDown Value**.

0	*Pullup/Pulldown Off*
1	*10kΩ Pullup*
2	*10kΩ Pulldown*

Table 4: Pull Up/Down Options

2.1.1 Encoder Operating Mode

Object 6112h **AI Operating Mode** is required to select if the encoder should measure either direction or counter value of the encoder input. The input signal to be measured can be chosen by setting object 6112h to one of the values shown in Table 5.

Value	16-Bit Counter
0	Disabled
1	**Analog Normal**
10	Encoder Counter

Table 5: Encoder AI Operating Mode Options

If index 6112h is set to Analog Normal, the direction will be measured and then written to object 7100h **AI Input Field Value**. Table 6 shows what values are representing what direction.

Value	Direction
0	Stationary
1	Negative
2	Positive

Table 6: Encoder Directions

2.1.2 Encoder Stationary Delay

The stationary object **Enc Stationary Delay** 2303h decides after how many milliseconds the direction is set to stationary (0) if no rising edge or falling edge is detected on signal A or B on the encoder input. This way, the performance of the encoder input can be adjusted for slower and faster rotating encoders. The default value for this parameter is set to 100[ms] by default.

2.1.3 Input Field Values and Scaling

As already mentioned above, the measured values of the encoder input are written to read-only and mappable object 7100h **AI Input Field Value**. Before the value is written to this object, the value is shifted according to read-only object 2011h **AI Decimal Digits FV.**
The value of 2011h will depend on the AI Sensor Type and Input Range selected and will be automatically updated per Table 7 when either 6110h or 2010h are changed. All other objects associated with the input field value also apply this object. These objects are 7120h **AI Scaling 1 FV**, 7122h **AI Scaling 2 FV**, 7148h AI **Span Start**, 7149h **AI Span End**, and 2305h **Enc Error Clear Hysteresis**. These objects are also automatically updated when the Type or Range is changed.

Sensor Type and Range	Decimal Digits
Digital Input	0 [On/Off]
Encoder Input	1

Table 7: AI Decimal Digits FV Depending on Sensor Type

2.1.4 Input Process Values and Scaling

Read-only object 7130h **AI Input Process Value** is also mappable. However, the default values for objects 7121h **AI Scaling 1 PV** and 7123h **AI Scaling 2 PV** are set equal to 7120h and 7122h, respectively, while object 6132h **AI Decimal Digits PV** is automatically initialized to equal 2011h. This means that the default relationship between the FV and PV is one-to-one, so object 7130h is not mapped to a TPDO by default.

Should a different linear relationship between what is measured versus what is sent to the CANopen bus be desired, objects 6132h, 7121h and 7123h can be changed. The linear relationship profile is shown in Figure 2Figure 7. Should a non-linear response be desired, the lookup table function block can be used instead, as described in 2.5

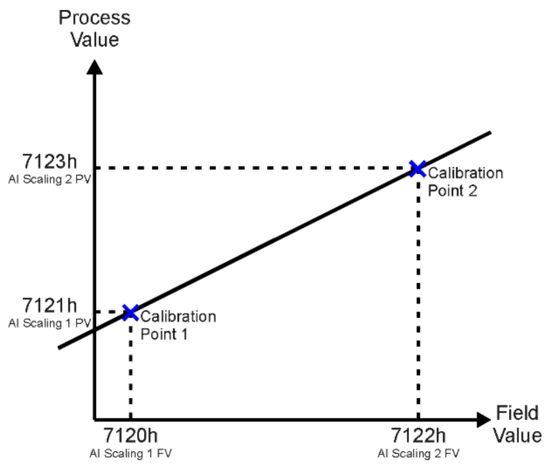


Figure 2: Analog Input Linear Scaling FV to PV

As stated earlier, the FV scaling objects are automatically updated with the Sensor Type or Range changes. This is because objects 7120h and 7122h are not only used in a linear conversion from FV to PV as described above, but also as the minimum and maximum limits when the input is used to control another logic block. Therefore, the values in these objects are important, even when the AI Input PV object is not being used.

The **AI Span Start** and **AI Span End** objects are used for fault detection, so they too are automatically updated for sensible values as the Type/Range changes. The Error Clear Hysteresis object is also updated, as it too is measured in the same units as the AI Input FV object.

Table 10 lists Table 8 the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2305h for each Sensor Type and Input Range combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 7.

Sensor Type/ Input Range	**7148h AI Span Start (i.e. Error Min)**	**7120h AI Scaling 1 FV (i.e. Input Min)**	**7122h AI Scaling 2 FV (i.e. Input Max)**	**7149h AI Span End (i.e. Error Max)**	**2305h Error Clear Hysteresis**
Encoder Direction	0	0	0	0	0
Encoder Counter	0	0	0	0	0
Digital Input	OFF	OFF	ON	ON	0

Table 8: AI Object Defaults Based on Sensor Type and Input Range

2.1.5 Automatic Update of Input Type Ranges

It might not be desired in an application for the automatic updating of objects when a key object is changed, i.e. AI Sensor Type. In this case, object 5550h **Enable Automatic Updates** can be set to FALSE (true by default) in which case changing an object will have no impact on any other objects. In this mode, the user must manually change all the objects for sensible values, or the controller will not work as expected.

The following section 2.2 explains how to use the encoder inputs as Digital ON/OFF inputs. The table below shows the digital input setpoints of the Encoder and next to it, the indices of the Digital Function Block, which are explained in more detail in section 2.2. Please refer to each parameter in the following section for more details.

Parameter	Encoder Input (Enc A) Index	Digital Input (DI) Index	Refer to
Logic Type	2314h, 2324h	2101h	Table 13
Active State	2310h, 2320h	2102h	Table 14
Debounce Time	2311h, 2321h	2103h	2.2.3
Debounce Filter	2312h, 2322h	2104h	Table 12
Pull Up/Down Value	2313h, 2323h	2105h	2.2.7

Table 9: Encoder Input parameter reference

2.2 Digital Input Function Block

2.2.1 Input Sensor Type

The unit provides six Digital Inputs which can be configured to different input types. To be able to set every input to a certain type, the most important object associated with the AI function block, object 6110h **AI Sensor Type**, needs to be set. By changing this value other objects will be automatically updated by the controller. The options for object 6110h are shown in Table 10, and no values other than what are shown here will be accepted.

Value	Meaning
60	Frequency Input (or RPM)
10000	PWM Input
10001	16-Bit Counter
10003	Digital ON/OFF

Table 10: AI Sensor Type Options for Digital Input 1 to 6

While four of these inputs have a wider range to choose from, the other two inputs can work as a digital on/off input only. Table 11 shows what input types are available for each Digital Input.

Input	AI Sensor Type Sub index	Configurable Inputs Types
Encoder 1 A	1	Encoder, Digital ON/OFF
Encoder 1 B	2	Encoder, Digital ON/OFF
Encoder 2 A	3	Encoder, Digital ON/OFF
Encoder 2 B	4	Encoder, Digital ON/OFF
Encoder 3 A	5	Encoder, Digital ON/OFF
Encoder 3 B	6	Encoder, Digital ON/OFF
Universal Input	7	PWM, Frequency, Counter, Digital ON/OFF, Voltage, Current
Digital Input 1	8	PWM, Frequency, Counter, Digital ON/OFF
Digital Input 2	9	PWM, Frequency, Counter, Digital ON/OFF
Digital Input 3	10	Digital ON/OFF
Digital Input 4	11	PWM, Frequency, Counter, Digital ON/OFF
Digital Input 5	12	PWM, Frequency, Counter, Digital ON/OFF
Digital Input 6	13	Digital ON/OFF

Table 11: Available Input Types for Digital Input 1 to 6

The input type of input 1, 2, 4, and 5 are set to PWM. Input 3 and 6 are configured to Digital ON/OFF as input type by default.

Figure 3 shows the hysteresis on the input when switching a discrete signal. A digital input can be switched up to +Vcc (16Vmax.).

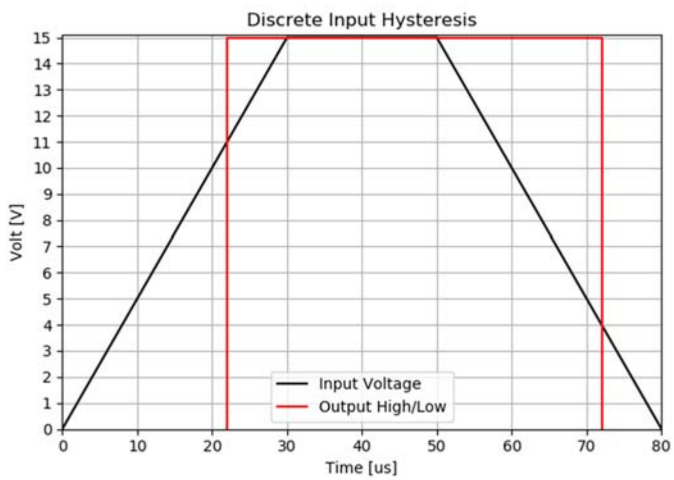


Figure 3: Discrete Input Hysteresis

The input state will be ON(1) in case the input voltage reaches a high of 11V. If then the voltage drops below 4V, the state will be OFF(0).

The digital input (DI) function block only becomes applicable on the input when object 6110h, **AI Sensor Type**, is set to a digital input response. Object 6000h will then provide the actual state of the corresponding input.

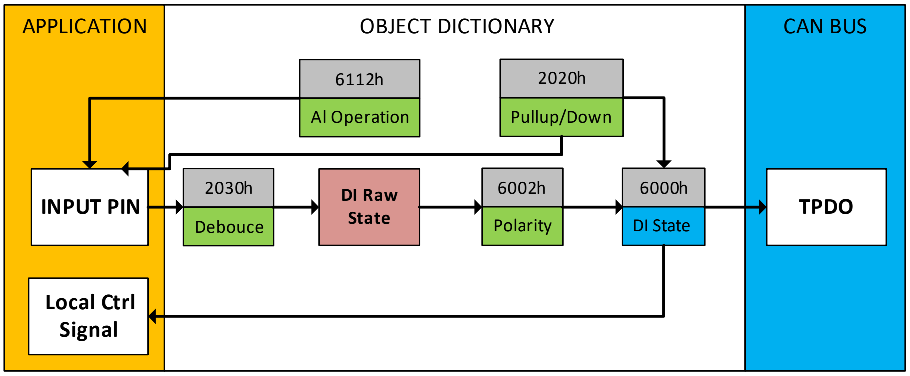


Figure 4: Digital Input Objects

2.2.2 Debounce Filter

Object 2104h **DI Debounce Filter** is applied to the input before the state is read by the processor. The options for object 2104h are shown in Table 2, with the default bolded.

Value	Meaning
0	Filter Disabled

1	Filter 111ns
2	**Filter 1.78 us**
3	Filter 14.22 us

Table 12: DI Debounce Filter Options

2.2.3 Debounce Time

In addition to the debounce filter, the object 2103h **DI Debounce Time** can be a useful parameter in applications where the digital input signal received by the controller is very noisy. shows how the Debounce Time helps detecting a correct input signal.

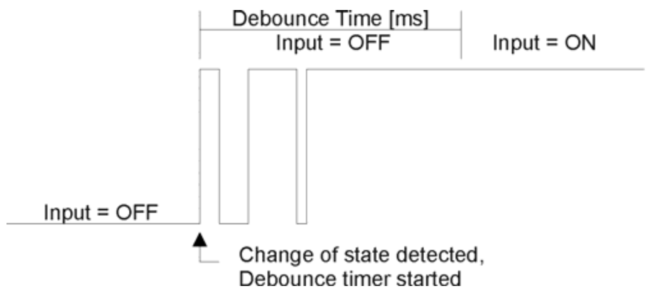


Figure 5: Digital Input Debounce Time

2.2.4 Input Logic Type

If the input is configured as a digital ON/OFF input type, object 2101h **DI Logic Type** can be used to set the logical input type of the corresponding input. The Digital Input Type parameter allows flexibility in the response of the input. Table 13 shows the options available for this parameter.

Value	Meaning
0	**Normal Logic**
1	Inverse Logic
2	Latched Logic

Table 13: Digital Input logic Types

Normal Logic Type is used for the inputs by default.

In *Normal Logic* mode, the input state is 1 in case the input signal is interpreted as an ON-signal. The input state turns 0 if the input signal is interpreted as an OFF-signal.

For the *Inverse Logic* type, the opposite behavior applies. If the input signal is ON, the state turns 0 and if the input signal is OFF, the state turns 1.

Setting the Input to *Latched Logic*, the input state is toggled between 1 and 0 every time the input signal of the respective digital input changes from OFF to ON.

2.2.5 Active High/Low

The Active High/Low object 2102h **DI Active State** allows the user to select how the controller responds to the behaviour of the digital input. Table 14 shows the different Active High/Low options with the default being highlighted.

Value	Meaning
0	***Active High***
1	*Active Low*

Table 14: Active High/Low

The inputs of the Digital Inputs have a fixed 10kOhm pull-up resistor. Given that by default, the inputs are configured to *Active High*, an ON response by the input is achieved when the input is grounded.

2.2.6 Input States

Once the raw state has been evaluated, the logical state of the input is determined by **object 6002h DI_Polarity_8_Input_Lines.** The options for object 6002h are shown in Table 15. The state of the DI will be written to read-only object 6000h **DI_Read_state_8_Input_Lines**. By default, normal on/off logic is used.

Value	Meaning
0	**Normal On/Off**
1	Inverse On/Off

Table 15: Object 6002h DI_Polarity_8_Input_Lines options

The format to write to object 6002h is as follows:

Sub-index 1 will determine the following inputs' polarities:

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AI8	A7	AI6	AI5	AI4	AI3	AI2	AI1

Sub-index 2 will determine the following inputs' polarities:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	**Bit 1**	**Bit 0**
-	-	-	AI13	AI12	AI11	AI10	AI9

As per the format of object 6002h, the bits in object 6000h **DI_Read_state_8_Input_Lines** will be written to represent the same inputs' states.

2.2.7 Frequency/RPM Input

There is another type of 'digital' input that can be selected on Digital Input 1,2,4, and 5, they are connected to the microcontroller timers, when 6110h is not set to 1003 Digital On/Off. In this case, the input is configured as an analog input and, therefore, the objects from the Analog Input (AI) block are applied instead of those discussed above.

A frequency input can be automatically turned into an RPM measurement by setting object 2107h **DI Number of Pulses Per Revolution** to a non-zero value. This means when **Pulses per Revolution** setpoint is set to 0, the measurements taken will be in units of [Hz]. If Pulses per Revolution setpoint is set to higher than 0, the measurements taken will be in units of [RPM]. All other input types ignore this object.

2.2.8 Counter Input Type

The 16-Bit counter has three different types of a Counter input modes: Counter (Pulse Count Reset), Counter (Time Window), Counter (Pulse Window). The type can be chosen by setting object 2010h **AI Range** to one of the values shown in Table 16: Counter Ranges.

Value	**16-Bit Counter**
0	Pulse Count Reset
1	Time Window
2	Pulse Window

Table 16: Counter Ranges

Counter (Pulse Count Reset) is a Counter input type in which the output data of the input function block is the number of pulses measured. The pulses will continue to increment in count as they are measured until the **DI Pulse Count to Reset Counter** value of object 2108h is reached at which the counter will reset back to 0.

Counter (Time Window) is a Counter input type in which object **DI Pulse Count in Time Window** 210Ah is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Counter (Pulse Window) is a Counter input type in which a 'Pulse Window', is selected in number of pulses (**DI pulse window in number of pulses** 2109h). The output of the input block will be the amount of time elapsed to reach the number of pulses set in the Elapsed Time to Each Pulse Count setpoint.

The value of each counter range will be updated to the Field Value object (7100h). Error detection objects are not used, since error detection is not possible in this mode.

2.2.9 Input Filter

Regardless of type selected, all inputs can be further filtered once the raw data has been measured. Object 61A0h **AI Filter Type** determines what kind of filter is used per Table 17. By default, additional software filtering is disabled.

Value	**Meaning**
0	**No Filter**
1	Moving Average
2	Repeating Average

Table 17: AI Filter Type Options

Object 61A1h **AI Filter Constant** is used with all three types of filters as per the formulas below:

Calculation with no filter:
Value = Input
The data is simply a 'snapshot' of the latest value measured by the ADC or timer.

Calculation with the moving average filter:

$$Value_N = Value_{N-1} + \frac{(Input - Value_{N-1})}{FilterConstant}$$

This filter is called every 1ms. The value FilterConstant stored in object 61A1h is 10 by default.

Calculation with the repeating average filter:

$$Value = \frac{\sum Input_N}{N}$$

At every reading of the input value, it is added to the sum. At every N^{th} read, the sum is divided by N, and the result is the new input value. The value and counter will be set to zero for the next read. The value of N is stored in object 61A1h and is set to 10 by default. This filter is called every 1ms.

2.2.10 Input Field Values and Scaling

The value from the filter is shifted according to read-only object 2011h **AI Decimal Digits FV** and then written to read-only and mappable object 7100h **AI Input Field Value.**
The value of 2011h will depend on the AI Sensor Type and Input Range selected and will be automatically updated per Table 18: AI Decimal Digits FV Depending on Sensor Type when either 6110h or 2010h are changed. All other objects associated with the input field value also apply this object. These objects are 7120h **AI Scaling 1 FV**, 7122h **AI Scaling 2 FV**, 7148h AI **Span Start**, 7149h **AI Span End**, and 2110h **DI Error Clear Hysteresis**. These objects are also automatically updated when the Type or Range is changed.

Sensor Type and Range	**Decimal Digits**
Frequency: 0.5Hz to 50Hz	2 [0.01 Hz]
Frequency: 10Hz to 1kHz	1 [0.1 Hz]
Frequency: 100Hz to 10kHz	0 [Hz]
Frequency: RPM Mode	1 [0.1 RPM]
PWM: All Ranges	1 [0.1 %]
16-Bit Counter	0 [ms]
Digital Input	0 [On/Off]

Table 18: AI Decimal Digits FV Depending on Sensor Type

2.2.11 Input Process Values and Scaling

Read-only object 7130h **AI Input Process Value** is also mappable. However, the default values for objects 7121h **AI Scaling 1 PV** and 7123h **AI Scaling 2 PV** are set to equal 7120h and 7122h respectively, while object 6132h **AI Decimal Digits PV** is automatically initialized to equal 2011h. This means that the default relationship between the FV and PV is one-to-one, so object 7130h is not mapped to a TPDO by default.

Should a different linear relationship between what is measured versus what is sent to the CANopen bus be desired, objects 6132h, 7121h and 7123h can be changed. The linear relationship profile is shown in Figure 6 Figure 7 below. Should a non-linear response be desired, the lookup table function block can be used instead, as described in section 2.5.

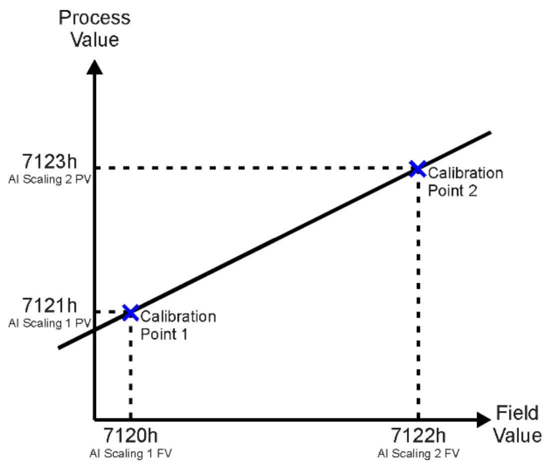


Figure 6: Analog Input Linear Scaling FV to PV

As stated earlier, the FV scaling objects are automatically updated with the Sensor Type or Range changes. This is because objects 7120h and 7122h are not only used in a linear conversion from FV to PV as described above, but also as the minimum and maximum limits when the input is used to control another logic block. Therefore, the values in these objects are important, even when the AI Input PV object is not being used.

The **AI Span Start** and **AI Span End** objects are used for fault detection, so they too are automatically updated for sensible values as the Type/Range changes. The Error Clear Hysteresis object is also updated, as it too is measured in the same unit as the AI Input FV object.

Table 19 lists the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2111h for each Sensor Type and Input Range combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 9.

Sensor Type/ Input Range	**7148h AI Span Start (i.e. Error Min)**	**7120h AI Scaling 1 FV (i.e. Input Min)**	**7122h AI Scaling 2 FV (i.e. Input Max)**	**7149h AI Span End (i.e. Error Max)**	**2111h Error Clear Hysteresis**
Frequency	100 [0.01Hz]	500 [0.01Hz]	100000 [0.01Hz]	105000 [0.01Hz]	100 [0.01Hz]
Freq: RPM Mode	500 [0.1RPM]	1000 [0.1RPM]	30000 [0.1RPM]	33000 [0.1RPM]	100 [0.1RPM]
PWM: 0 to 100%	10 [0.1%]	50 [0.1%]	950 [0.1%]	990 [0.1%]	10 [0.1%]
16-Bit Counter	OFF	OFF	ON	ON	0
Digital Input	OFF	OFF	ON	ON	0

Table 19: AI Object Defaults Based on Sensor Type and Input Range

2.2.12 Automatic Update of Input Type Ranges

It might not be desired in an application for the automatic updating of objects when a key object is changed, i.e. AI Sensor Type. In this case, object 5550h **Enable Automatic Updates** can be set to FALSE (true by default) in which case changing an object will have no impact on any other objects. In this mode, the user must manually change all the objects for sensible values, or the controller will not work as expected.

When changing these objects, Table 20 outlines the range constraints placed on each based on its Sensor Type and Input Range combination selected. In all cases, the MAX-value is the upper end of the range (i.e. 5V or) Object 7122h cannot be set higher than MAX, whereas 7149h can be set up to 110% of MAX. Object 2111h on the other hand can only be set up to maximum value of 10% of MAX. Table 20 uses the base unit of the input, but recall the limits will also have object 2102h apply to them as per Table 18.

<table>
<tr><th>Sensor Type/
Input Range</th><th>7148h</th><th>7120h</th><th>7122h</th><th>7149h</th><th>2110h</th></tr>
<tr><td>RPM: 0 to 6000RPM
PWM: 0 to 100%</td><td>0 to 7120h</td><td>7148h to 7122h</td><td rowspan="2">7120h to 7149h
If(7149h>MAX)
7120h to MAX</td><td rowspan="2">7122h
to
110%
of MAX</td><td rowspan="2">10% of
MAX</td></tr>
<tr><td>Frequency</td><td>0.1Hz to 7120h</td><td>7148h to 7122h
If(7148h<0.5Hz) 0.5Hz to 7122h</td></tr>
</table>

Table 20: AI Object Ranges Based on Sensor Type and Input Range

2.2.13 Input Error Behavior

The last objects associated with the digital input block left to discuss are those associated with fault detection. Should the calculated input (after measuring and filtering) fall outside of the allowable range, as defined by the AI Span Start and AI Span End objects, an error flag will be set in the application if object 210Fh **AI Error Detect Enabled** is set to TRUE (1).

When (7100h AI Input FV < 7148h AI Span Start), an “Out of Range Low” flag is set. If the flag stays active for the 2111h **AI Error Reaction Delay** time, an Input Overload Emergency (EMCY) message will be added to object 1003h **Pre-Defined Error Field**. Similarly, when (7100h AI Input FV > 7149h AI Span End), an “Out of Range High” flag is set and will create an EMCY message should it stay active throughout the delay period. In either case, the application will react to the EMCY message as defined by object 1029h **Error Behavior** at the sub-index corresponding to an Input Fault. Refer to section 3.4 and 3.5 for more information about objects 1003h and 1029h.

Once the fault has been detected, the associate flag will be cleared only once the input comes back into range. Object 2111h **AI Error Clear Hysteresis** is used here so that the error flag will not be set/cleared continuously while the AI Input FV hovers around the AI Span Start/End value.

To clear an “Out of Range Low” flag, AI Input FV >= (AI Span Start + AI Error Clear Hysteresis)
To clear an “Out of Range High” flag, AI Input FV <= (AI Span End - AI Error Clear Hysteresis)
Both flags cannot be active at once. Setting either one of these flags automatically clears the other.

2.3 Universal Input Function Block

The universal Input on this controller can be configured to a wide range of input types. To be able to set every input to a certain type, the most important object associated with the AI function block, object 6110h **AI Sensor Type**, needs to be set as discussed in the previous chapter. By changing this value other objects will be automatically updated by the controller. The options for object 6110h sub-index 7, which represents the universal input, are shown in Table 10.

Value	Meaning
40	**Voltage Input**
50	Current Input
60	Frequency Input (or RPM)
10000	PWM Input
10001	16-Bit Counter
10003	Digital ON/OFF

Table 21: AI Sensor Type Options for Universal Input

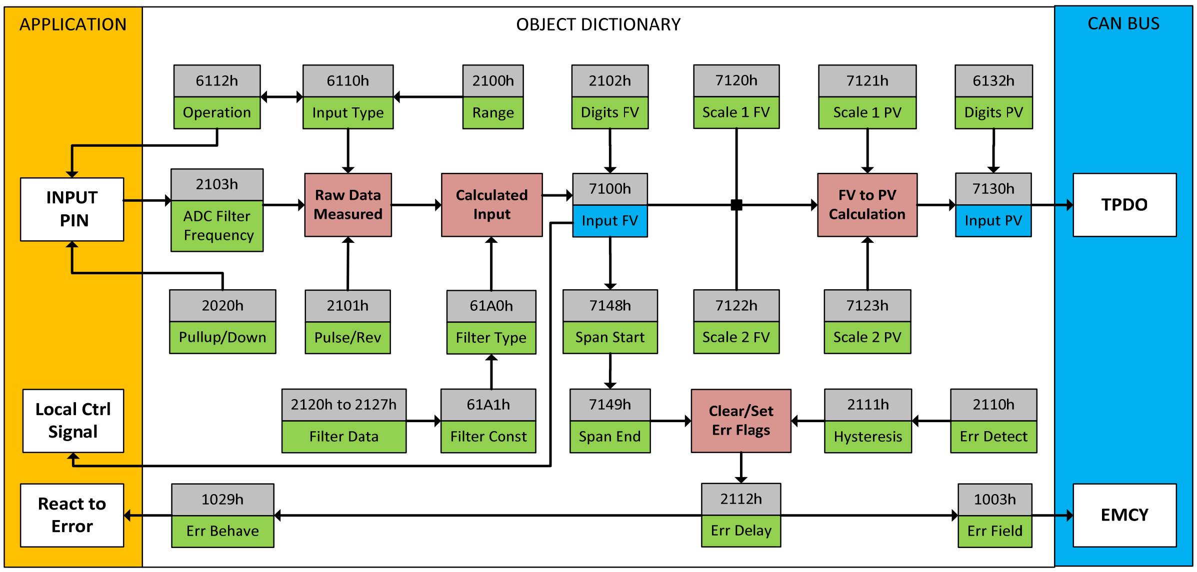


Figure 7 Analog Input Objects

Since the universal input can be used as a type of Digital ON/OFF, Frequency, PWM, and Counter input, the parameters explained in the previous chapter, Digital Input Function, are also available for the Universal Input except that the indices are different for this input. Table 22 shows what indices are used for the Universal Input.

Parameter	Universal Input (UNI) Index	Digital Input (DI) Index	Refer to
Logic Type	220Bh	2101h	Table 13
Active State	2200h	2102h	Table 14
Debounce Time	2201h	2103h	2.2.3
Debounce Filter	2202h	2104h	Table 12
Overflow Value	2203h	2105h	2.2.7
Filter Type	220Ah	2106h	**Error! Reference source not found.**
Number pulses per revolution	2204h	2107h	2.2.7
Num to reset pulse counter	2205h	2108h	2.2.8
Pulse count pulse window	2206h	2109h	2.2.8
Pulse count time window	2207h	210Ah	2.2.8

Error detect enable	220Ch	210Fh	2.2.13
Error clear hysteresis	220Dh	2110h	2.2.13
Error react delay	220Eh	2111h	2.2.13

Table 22: Universal Input parameter reference

Since there are more input types for the Universal Input available, the allowable ranges will differ from the Digital Input. Table 23 shows the relationship between the available sensor types and the associated range options. The default value for each range is bolded, and object 2010h **AI Range** will automatically be updated with this value when 6110h is changed. The grayed out cells mean that the associate value is not allowed for the range object when that sensor type has been selected.

Value	Voltage	Current	16-Bit Counter
0	**0 to 5V**	0 to 20mA	Pulse Count Reset
1	0 to 10V	**4 to 20mA**	Time Window
2			Pulse Window

Table 23: Universal Input Range Options Depending on Sensor Type

2.3.1 ADC Filter Frequencies

Not all objects apply to all input types. For example, object 2209h **AI Filter Frequency for ADC** is only applicable if a voltage or current input is being measured. In these cases, the ADC will automatically filter as per Table 24, and is set for 50Hz noise rejection by default.

Value	Meaning
0	Input Filter Off
1	**Filter 50Hz**
2	Filter 60Hz
3	Filter 50Hz and 60Hz

Table 24: ADC Filter Frequency Options

2.3.2 Pull Up/Down Options

Conversely, Frequency, PWM and Counter inputs use object 2208h **UNI Pullup/Down Mode** (see Table 1) while **voltage and current inputs set this object to zero**. The options for this parameter are listed in Table 25.

0	*Pullup/Pulldown Off*
1	*10kΩ Pullup*
2	*10kΩ Pulldown*

Table 25: Pull Up/Down Options

As discussed in sub-chapter 2.2.10, the value from the filter is shifted according to read-only object 2011h **AI Decimal Digits FV** and then written to read-only object 7100h **AI Input Field Value.**
The value of 2011h will depend on the AI Sensor Type and Input Range selected and will be automatically updated per Table 26 when 6110h is changed. All other objects associated with the input field value also apply this object. These objects are 7120h **AI Scaling 1 FV**, 7122h **AI Scaling 2 FV**, 7148h AI **Span Start**, 7149h **AI Span End**, and 220Dh **UNI Error Clear Hysteresis**. These objects are also automatically updated when the Type or Range is changed.

Sensor Type and Range	Decimal Digits
Voltage: All Ranges	3 [mV]
Current: All Ranges	3 [uA]
Frequency	1 [0.1 Hz]
Frequency: RPM Mode	1 [0.1 RPM]
PWM: All Ranges	1 [0.1 %]
16-Bit Counter	0 [ms]
Digital Input	0 [On/Off]

Table 26: AI Decimal Digits FV Depending on Universal Input Sensor Type

Table 27 lists the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2111h for each Sensor Type and Input Range combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 26.

Sensor Type/ Input Range	7148h AI Span Start (i.e. Error Min)	7120h AI Scaling 1 FV (i.e. Input Min)	7122h AI Scaling 2 FV (i.e. Input Max)	7149h AI Span End (i.e. Error Max)	2111h Error Clear Hysteresis
Voltage: 0 to 5V	200 [mV]	500 [mV]	4500 [mV]	4800 [mV]	100 [mV]
Voltage: 0 to 10V	200 [mV]	500 [mV]	9500 [mV]	9800 [mV]	200 [mV]
Current: 0 to 20mA	0 [uA]	0 [uA]	20000 [uA]	20000 [uA]	250 [uA]
Current: 4 to 20mA	1000 [uA]	4000 [uA]	20000 [uA]	21000 [uA]	250 [uA]
Freqemcy	1 [Hz]	10 [Hz]	10000 [Hz]	10500 [Hz]	10 [Hz]
Freq: RPM Mode	500 [0.1RPM]	1000 [0.1RPM]	30000 [0.1RPM]	33000 [0.1RPM]	100 [0.1RPM]
PWM: 0 to 100%	10 [0.1%]	50 [0.1%]	950 [0.1%]	990 [0.1%]	10 [0.1%]
16-Bit Counter	OFF	OFF	ON	ON	0
Digital Input	OFF	OFF	ON	ON	0

Table 27: AI Object Defaults Based on Universal Input Sensor Type and Input Range

It might not be desired in an application for the automatic updating of objects when a key object is changed, i.e. AI Sensor Type. In this case, object 5540h **Enable Automatic Updates** can be set to FALSE (true by default) in which case changing an object will have no impact on any other objects. In this mode, the user must manually change all the objects for sensible values, or the controller will not work as expected.

When changing these objects, Table 11 outlines the range constraints placed on each based on its Sensor Type and Input Range combination selected. In all cases, the MAX value is the upper end of the range (i.e. 5V or) Object 7122h cannot be set higher than MAX, whereas 7149h can be set up to 110% of MAX. Object 2111h on the other hand can only be set up to maximum value of 10% of MAX. Table 11 uses the base unit of the input, but recall the limits will also have object 2102h apply to them as per Table 9.

Sensor Type/ Input Range	7148h	7120h	7122h	7149h	2111h
Voltage: 0 to 5V and 0 to 10V Current: 0 to 20mA and 4 to 20mA RPM: 0 to 6000RPM	0 to 7120h	7148h to 7122h	7120h to 7149h If(7149h>MAX) 7120h to MAX	7122h to 110% of MAX	10% of MAX

PWM: 0 to 100%					
Current: 4 to 20mA	0 to 7120h	7148h to 7122h If(7148h<4mA) 4mA to 7122h			
Freq: 0.5Hz to 50Hz	0.1Hz to 7120h	7148h to 7122h If(7148h<0.5Hz) 0.5Hz to 7122h			
Freq: 10Hz to 1kHz	5Hz to 7120h	7148h to 7122h If(7148h<10Hz) 10Hz to 7122h			
Freq: 100Hz to 10kHz	50Hz to 7120h	7148h to 7122h If(7148h<100Hz) 100Hz to 7122h			

Table 28: AI Object Ranges Based on Universal Input Sensor Type and Input Range

2.4 Internal Function Block Control Sources

The controller allows for internal function block sources to be selected from the list of the logical function blocks supported by the controller. As a result, any output from one function block can be selected as the control source for another. Keep in mind that not all options make sense in all cases, but the complete list of control sources is shown in Table 29.

Value	**Meaning**
0	Control Source Not Used (Ignored)
1	Digital Input Data
2	Universal Input Data
3	Digital Frequency Input
4	Universal Frequency Input
5	Encoder Data A
6	Encoder Data B
7	Encoder Direction
8	Encoder Counter
9	VPS Fault State
10	Temperature Fault State
11	CANopen Message (RPDO)
12	Simple Conditional Block
13	Power Supply Measured
14	Processor Temperature Measured
15	Look Up Table
16	Mathematical Function Block
17	Programmable Logic Function Block
18	Constant Function Block

Table 29: Control Source Options

In addition to a source, each control also has a number which corresponds to the sub-index of the function block in question. Table 13 outlines the ranges supported for the number objects, depending on the source that had been selected.

Control Source	Range	Object (Meaning)
Control Source Not Used	0	Ignored
CANopen Message (RPDO)	1	2500h sub-index 1 (Extra Received PV 1)
	2	2500h sub-index 1 (Extra Received PV 2)
	3	2500h sub-index 2 (Extra Received PV 3)
	4	2500h sub-index 3 (Extra Received PV 4)
	5	2500h sub-index 4 (Extra Received PV 5)
Encoder Input	1	7100h sub-index 1 or 6000h sub-index 1 bit 0
	2	7100h sub-index 2 or 6000h sub-index 1 bit 1
	3	7100h sub-index 3 or 6000h sub-index 1 bit 2
	4	7100h sub-index 4 or 6000h sub-index 1 bit 3
	5	7100h sub-index 5 or 6000h sub-index 1 bit 4
	6	7100h sub-index 6 or 6000h sub-index 1 bit 5
Universal Input	7	7100h sub-index 7 or 6000h sub-index 1 bit 6
Digital Input	8	7100h sub-index 8 or 6000h sub-index 1 bit 7
	9	7100h sub-index 9 or 6000h sub-index 2 bit 0
	10	7100h sub-index 10 or 6000h sub-index 2 bit 1
	11	7100h sub-index 10 or 6000h sub-index 2 bit 2
	12	7100h sub-index 10 or 6000h sub-index 2 bit 3
	13	7100h sub-index 10 or 6000h sub-index 2 bit 4
Constant Function Block	1	5010h sub-index 1 (always FALSE)
	2	5010h sub-index 2 (always TRUE)
	3	5010h sub-index 3 (Constant FV 3)
	4	5010h sub-index 4 (Constant FV 4)
	5	5010h sub-index 5 (Constant FV 5)
	6	5010h sub-index 6 (Constant FV 6)
	7	5010h sub-index 7 (Constant FV 7)
	8	5010h sub-index 8 (Constant FV 8)
	9	5010h sub-index 9 (Constant FV 9)
	10	5010h sub-index 10 (Constant FV 10)
Lookup Table Function Block	1	3017h (Lookup Table 1 Output Y-Axis PV)
	2	3027h (Lookup Table 2 Output Y-Axis PV)
	3	3037h (Lookup Table 3 Output Y-Axis PV)
	4	3047h (Lookup Table 4 Output Y-Axis PV)
	5	3057h (Lookup Table 5 Output Y-Axis PV)
	6	3067h (Lookup Table 6 Output Y-Axis PV)
	7	3077h (Lookup Table 7 Output Y-Axis PV)
	8	3087h (Lookup Table 8 Output Y-Axis PV)
	9	3097h (Lookup Table 9 Output Y-Axis PV)
	10	3107h (Lookup Table 10 Output Y-Axis PV)
	11	3117h (Lookup Table 11 Output Y-Axis PV)
	12	3127h (Lookup Table 12 Output Y-Axis PV)
Mathematical Function Block	1	4350h sub-index 1 (Math Output PV 1)
	2	4350h sub-index 2 (Math Output PV 2)
	3	4350h sub-index 3 (Math Output PV 3)

	4	4350h sub-index 4 (Math Output PV 4)
	5	4350h sub-index 5 (Math Output PV 5)
	6	4350h sub-index 6 (Math Output PV 6)
	7	4350h sub-index 7 (Math Output PV 5)
	8	4350h sub-index 8 (Math Output PV 6)
Programmable Logic Function Block	1	3xy7h (Lookup Table Selected by Logic 1)
	2	3xy7h (Lookup Table Selected by Logic 2)
	3	3xy7h (Lookup Table Selected by Logic 3)
	4	3xy7h (Lookup Table Selected by Logic 4)
Simple Conditional Logic	1	5102h sub-index 1 (Cnd. Logic Output PV 1)
	2	5102h sub-index 2 (Cnd. Logic Output PV 2)
	3	5102h sub-index 3 (Cnd. Logic Output PV 3)
	4	5102h sub-index 4 (Cnd. Logic Output PV 4)
	5	5102h sub-index 5 (Cnd. Logic Output PV 5)
	6	5102h sub-index 6 (Cnd. Logic Output PV 6)
	7	5102h sub-index 7 (Cnd. Logic Output PV 7)
	8	5102h sub-index 8 (Cnd. Logic Output PV 8)
	9	5102h sub-index 9 (Cnd. Logic Output PV 9)
	10	5102h sub-index 10 (Cnd. Logic Output PV 10)
Processor Temperature Measured	N/A	5040h (Power Supply FV) sub-index 1
Power Supply Measured	N/A	5040h (Temperature FV) sub-index 2

Table 30: Control Number Options Depending on Source Selected

When using any control source as the X-Axis input to a function block, the corresponding scaling limits are defined as per Table 14. It is the responsibility of the user to make sure that the scaling objects for any function block are setup appropriately depending on the source selected for the X-Axis input.

Control Source	Scaling 1	Scaling 2	Dec Digits
CANopen Message – Num 1 to 12	2520h	2522h	2502h
Input Function Block 1-13	7120h	7122h	6132h
Constant Function Block	N/A	N/A	N/A (float)
Lookup Table yz Function Block (where yz = 01 to 12)	0 or lowest from 3yz6h(*)	100 or highest from 3yz6h(**)	3yz3h
Mathematical Function	4021h	4023h	4032h
Programmable Logic Function	0%	100%	1 (fixed)
Power Supply Measured	N/A	N/A	1 (fixed)
Processor Temperature Measured	N/A	N/A	1 (fixed)

Table 31: Scaling Limits per Control Source - ***(*)** - Whichever value is smaller**; (**)** - Whichever value is larger*

2.5 Lookup Table Function Block

The lookup table (LTz) function blocks are not used by default.

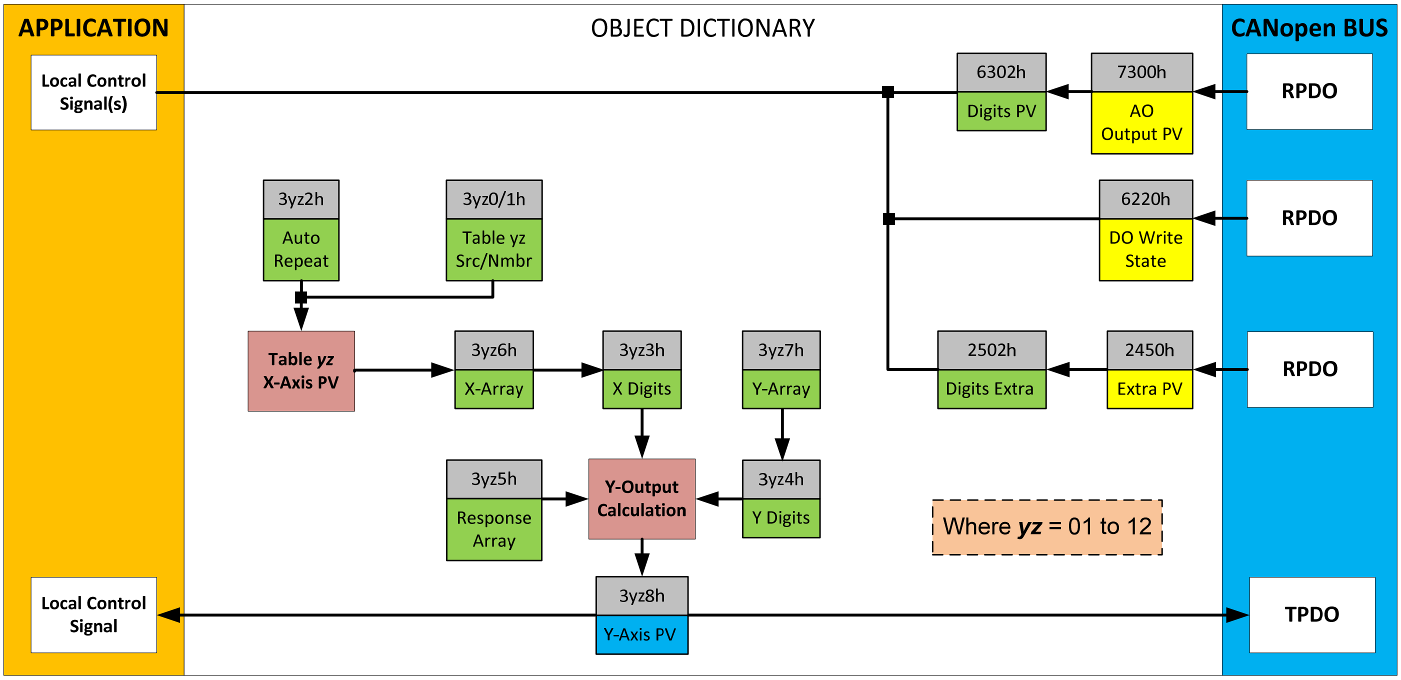


Figure 8: Lookup Table Objects

Lookup tables are used to give an output response of up to 10 slopes per input. The array size of the objects 3yz5h **LTyz Point Response**, 3yz6h **LTyz Point X-Axis PV** and 3yz7h **Point Y-Axis PV** shown in the block diagram above is therefore 11.

Note: If more than 10 slopes are required, a Programmable Logic Block can be used to combine up to three tables to get 30 slopes, as is described in 2.6.

A parameter that will affect the function block is object **3yz5h sub-index 1** which defines the "**X-Axis Type**". By default, the tables have a *'Data Response'* output (0). Alternatively, it can be selected as a *'Time Response'* (1).

There are two (or three) other key parameters that will affect how this function block will behave depending on the "**X-Axis Type**" chosen. If chosen *'Data Response'*, then the objects 3yz0h **Lookup Table yz Input X-Axis Source** and 3yz1h **Lookup Table yz Input X-Axis Number** together define the control source for the function block. When it is changed, the table values in object 3yz6h need to be updated with new defaults based on the X-Axis source selected. If, however, the "**X-Axis Type**" is chosen to be *'Time Response'*, an additional parameter is taken into consideration - object 3yz2h, **Lookup Table yz Auto Repeat**. These will be described in more detail in 2.5.4.

2.5.1 X-Axis, Input Data Response

In the case where the "**X-Axis Type**" = *'Data Response',* the points on the X-Axis represents the data of the control source.

However, should the minimum input be less than zero, for example a resistive input that is reflecting temperature in the range of -40ºC to 210ºC, then the "**LTz Point X-Axis PV sub-index 1**" will be set to the minimum instead, in this case -40ºC.

The constraint on the X-Axis data is that the next index value is greater than or equal to the one below it, as shown in the equation below. Therefore, when adjusting the X-Axis data, it is recommended that X_{11} is changed first, then lower indexes in descending order.

$$MinInputRange <= X_1 <= X_2 <= X_3 <= X_4 <= X_5 <= X_6 <= X_7 <= X_8 <= X_9 <= X_{10} <= X_{11} <= MaxInputRange$$

As stated earlier, MinInputRange and MaxInputRange will be determined by the scaling objects associated with X-Axis Source that has been selected, as outlined in Table 29.

2.5.2 Y-Axis, Lookup Table Output

By default, it is assumed that the output from the lookup table function block will be a percentage value in the range of 0 to 100.

In fact, as long as all the data in the Y-Axis is 0<=Y[i]<=100 (where i = 1 to 11) then other function blocks using the lookup table as a control source will have 0 and 100 as the Scaling 1 and Scaling 2 values used in linear calculations shown in Table 31.

However, the Y-Axis has no constraints on the data that it represents. This means that inverse, increasing/decreasing, or other responses can be easily established. **The Y-Axis does not have to be a percentage output but it could represent full scale process values instead.**
In all cases, the controller looks at the **entire range** of the data in the Y-Axis sub-indices and selects the lowest value of the MinOutRange as well as the highest value of the MaxOutRange. As long as they are not both within the 0 to 100 range, they are passed directly to other function blocks as the limits on the lookup table output. (i.e. Scaling 1 and Scaling 2 values in linear calculations.)

Even if some of the data points are *'Ignored'* as described in 2.5.3, they are still used in the Y-Axis range determination. If not all the data points are going to be used, it is recommended that Y10 be set to the minimum end of the range, and Y11 to the maximum first. This way, the user can get predictable results when using the table to drive another function block, such as an analog output.

2.5.3 Point To Point Response

By default, all six lookup tables have a simple linear response from 0 to 100 in steps of 10 for both the X and Y axes. For a smooth linear response, each point in the 30z5h **LTz Point Response** array is setup for a *'Ramp To'* output.

Alternatively, the user could select a *'Step To'* response for 30z4h, where N = 2 to 11. In this case, any input value between X_{N-1} to X_N will result in an output from the lookup table function block of Y_N. *(Recall: LTz Point Response sub-index 1 defines the X-Axis type)*

Figure 10 shows the difference between these two response profiles with the default settings.

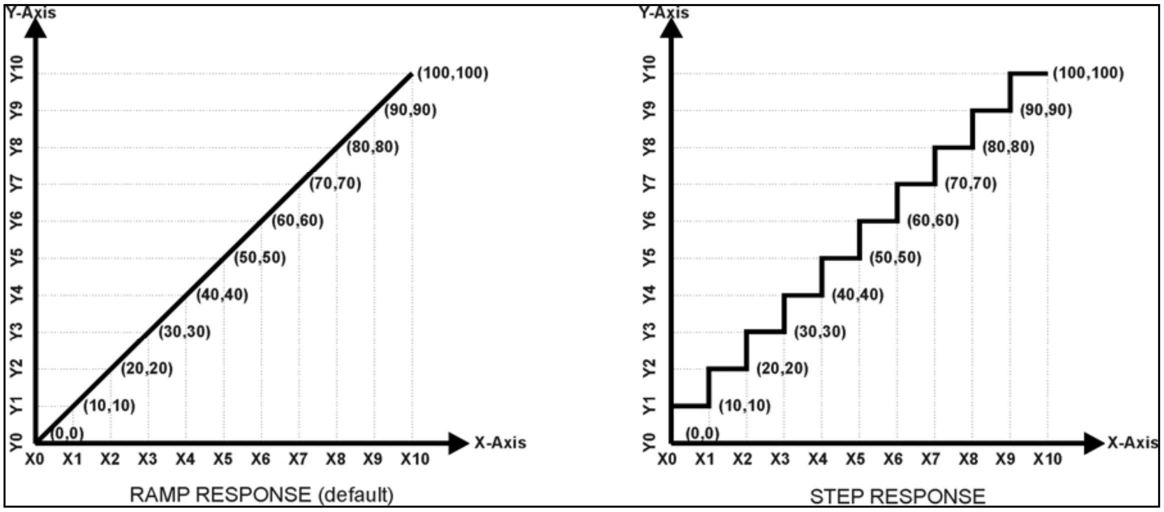


Figure 9: Lookup Table Defaults with Ramp and Step Responses

Lastly, any point except (1,1) can be selected for an *'Ignore'* response. If **LTz Point Response sub-index N** is set to ignore, then all points from (X_N, Y_N) to (X_{11}, Y_{11}) will also be ignored. For all data greater than X_{N-1}, the output from the lookup table function block will be Y_{N-1}.

A combination of *'Ramp To'*, *'Jump To'* and *'Ignore'* responses can be used to create an application specific output profile. An example of where the same input is used as the X-Axis for two tables, but where the output profiles 'mirror' each other for a deadband joystick response is shown in Figure 10. The example shows a dual slope percentage output response for each side of the deadband, but additional slopes can be easily added as needed. *(Note: In this case, since the analog outputs are responding directly to the profile from the lookup tables, both would have object 2342h AO Control Response set to a 'Single Output Profile.')*

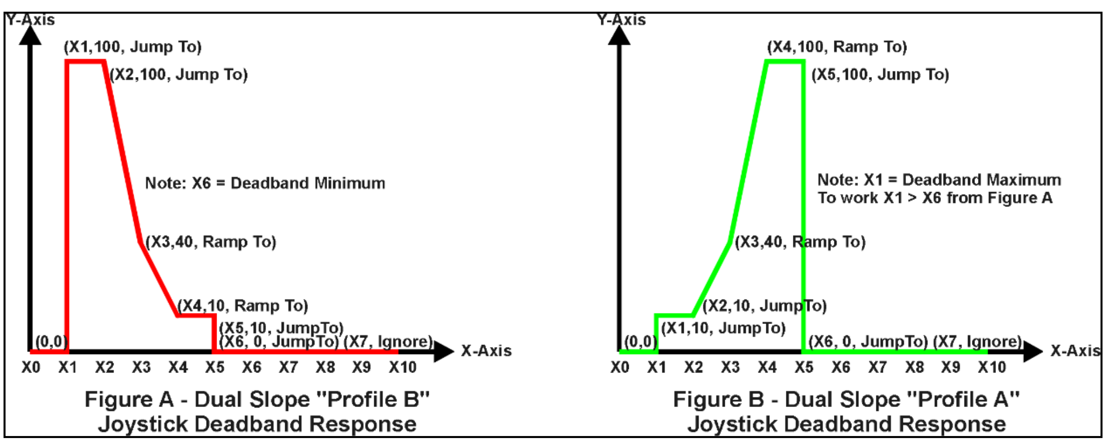


Figure 10: Lookup Table Examples to Setup for Dual-Slope Joystick Deadband Response

To summarize, Table 32 outlines the different responses that can be selected for object 30z4h, both for the X-Axis type and for each point in the table.

Sub-Index	Value	Meaning
1	0	Data Response (X-Axis Type)
2 to 11		Ignore (this point and all following it)
1	1	Time Response (X-Axis Type)
2 to 11		Ramp To (this point)
1	2	N/A (not an allowed option)
2 to 11		Jump To (this point)

Table 32: LTyz Point Response Options

2.5.4 X-Axis, Time Response

A lookup table can also be used to get a custom output response where the "**X-Axis Type**" is a *'Time Response'.* When this is selected, the X-Axis now represents time, in units of milliseconds, while the Y-Axis still represents the output of the function block.

With this response, the sequence will start depending on two parameters:

- **Lookup Table yz Input X-Axis Source** Object 3yz0h and;
- **Lookup Table yz Auto Repeat** Object 3yz2h

By default, the "Auto Repeat" object is set to FALSE (0). In this case, the lookup table will react in the following way:

> The X-Axis control source is treated as a digital input. When the control input is ON, the output will be changed over a period of time based on the profile in the lookup table. Once the profile has finished (i.e. reached index 11, or an 'Ignored' response), the output will remain at the last output at the end of the profile until the control input turns OFF.

However, when the "Auto Repeat" object is set to TRUE (1), the lookup table will react in the following way:

> When the control input is ON, the output will be changed over a period of time based on the profile in the lookup table. Once the profile has finished (i.e. reached index 11, or an 'Ignored' response), the lookup table will revert back to the first point in the table and Auto Repeat the sequence. This will continue for as long as the input remains ON. Once the input turns OFF, the lookup table sequence will stop and the output of the lookup table is zero.

Note: When the control input is OFF, the output is always at zero. When the input comes ON, the profile will **ALWAYS** start at position (X_1, Y_1) which is 0 output for 0ms.

When using the lookup table to drive an output based on **time**, it is mandatory that objects 2330h **Ramp Up** and 2331h **Ramp Down** in the analog output function block be set to **zero**. Otherwise, the output result will not match the profile as expected. Recall, also, that the AO scaling should be set to match the Y-Axis scaling of the table in order to get a 1:1 response of AO Output FV versus LTyz Output Y-Axis PV.
In a time response, the data in object 30z6h **LTyz Point X-Axis PV** is measured in milliseconds, and object 3yz3h **LTyz X-Axis Decimal Digits PV** is automatically set to 0. A minimum value of 1ms must be selected

for all points other than sub-index 1 which is automatically set to [0,0]. The interval time between each point on the X-axis can be set anywhere from 1ms to 24 hours. [86,400,000 ms].

2.6 Programmable Logic Function Block

The programmable logic blocks (LB(3-x)) functions are not used by default.

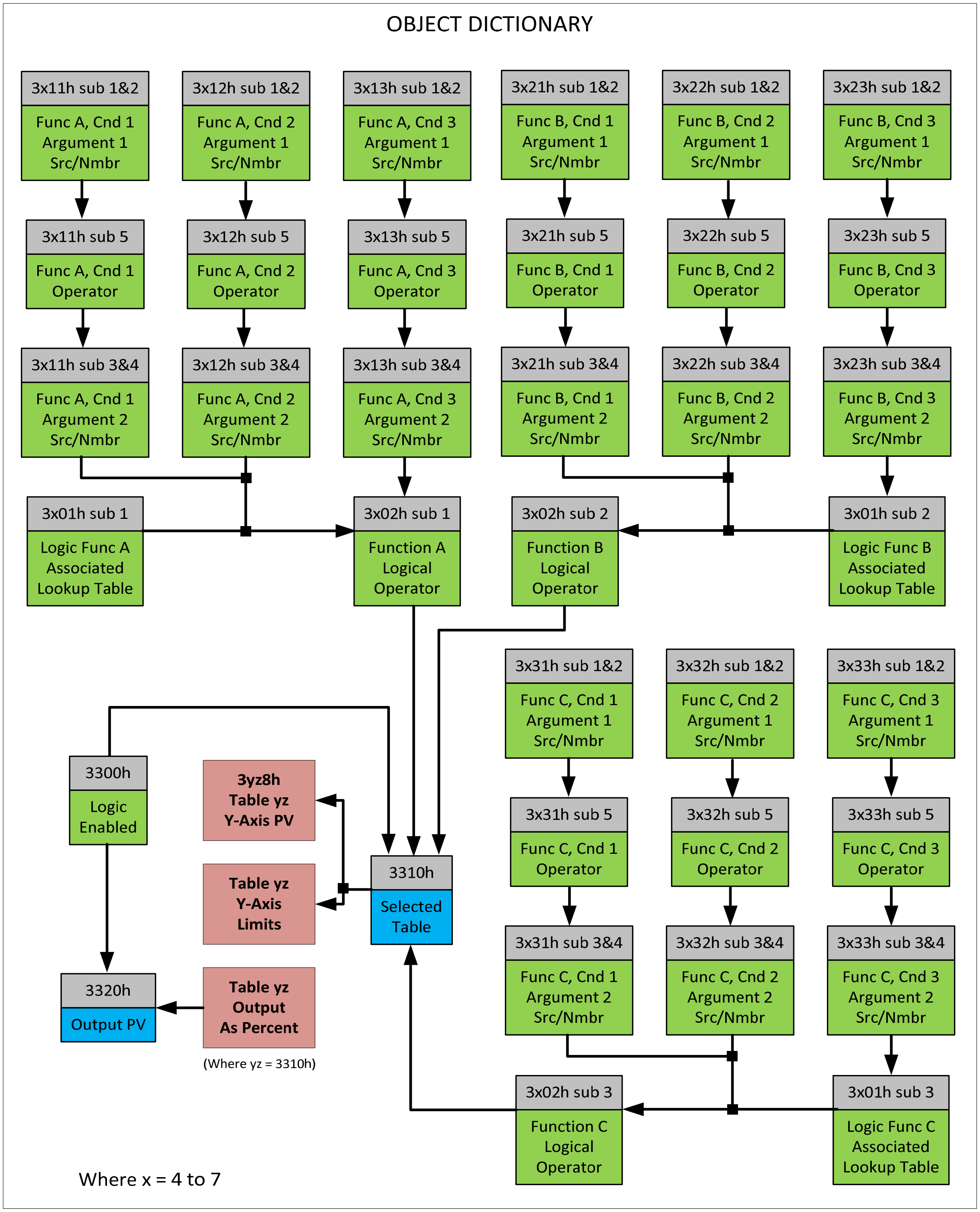


Figure 11: Logic Block Objects

This function block is obviously the most complicated of all function blocks provided, but very powerful. Any LBx (where X= 4 to 7) can be linked with up to three lookup tables, any one of which would be selected only under given conditions. Any three tables (of the available 12) can be associated with the logic, and which ones are used is fully configurable on object 3x01 **LB(3-x) Lookup Table Number**.

Should the conditions be such that a particular table (A, B or C) has been selected as described in 2.5.2, then the output from the selected table, at any given time, will be passed directly to LB(3-x)'s corresponding sub-index X in read-only mappable object 3320h **Logic Block Output PV**. The active table number can read from read-only object 3310h **Logic Block Selected Table**.

Note: In this document, the term *LB(3-x)* refers to Logic Blocks 1 to 4. Due to the CANopen Object indices, Logic Block 1 begins at 3401h where x, in this case, is 4.

Therefore, an LBx allows up to three different responses to the same input, or three different responses to different inputs, to become the control for another function block, such as an analog output. Here, the "**Control Source**" for the reactive block would be selected to be the *'Programmable Logic Function Block,'*.

In order to enable any one of logic blocks, the corresponding sub-index in object 3300h **Logic Block Enable** must be set to TRUE. They are all disabled by default.

Logic is evaluated in the order shown in Figure 12. Only if a lower indexed table (A, B, C) has not been selected will the conditions for the next table be looked at. **The default table is always selected as soon as it is evaluated. It is therefore required that the default table always be the highest index in any configuration.**

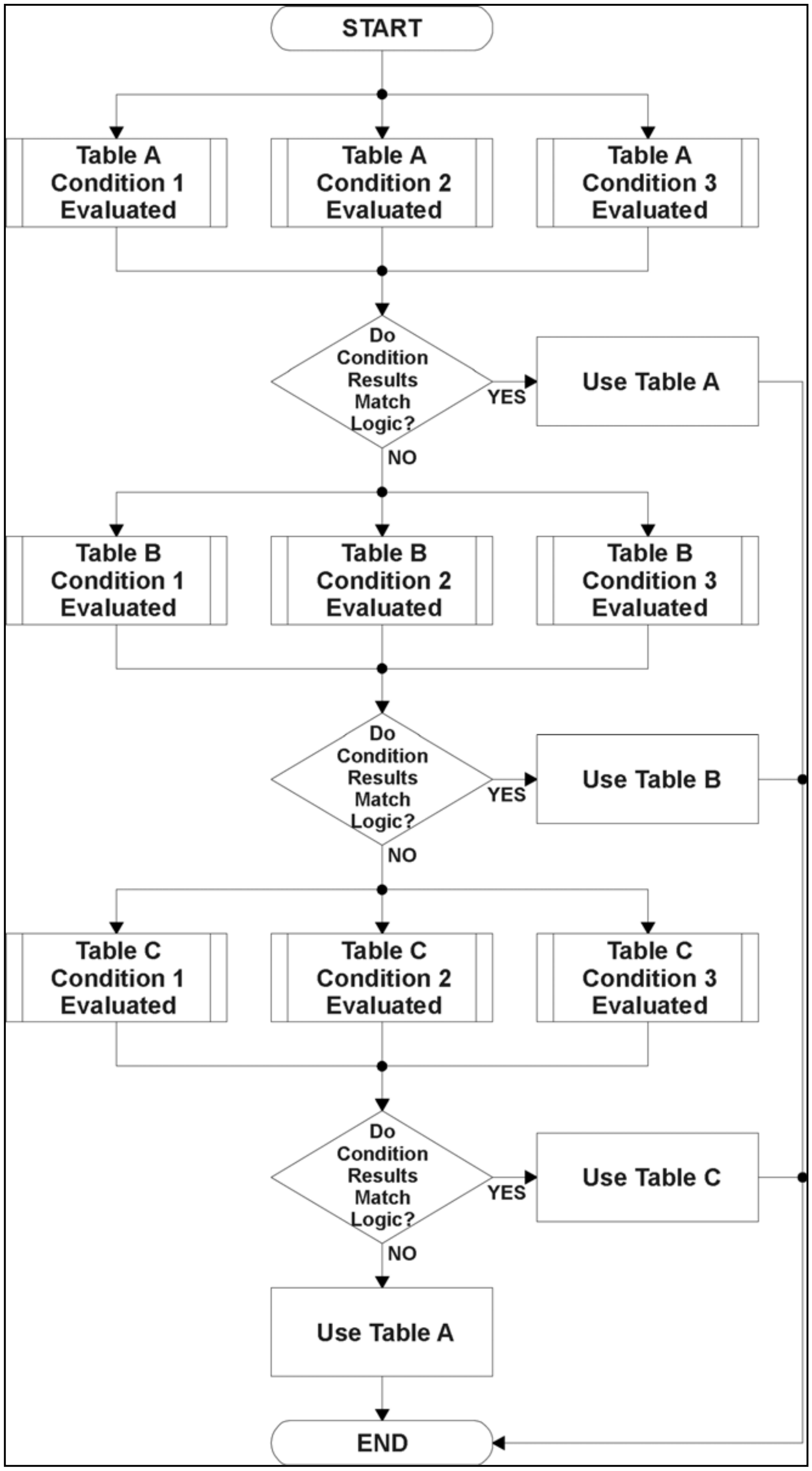


Figure 12: Logic Block Flowchart:

2.6.1 Conditions Evaluation

The first step in determining which table will be selected as the active table is to first evaluate the conditions associated with a given table. Each table has associated with it up to three conditions that can be evaluated. Conditional objects are custom DEFSTRUCT objects defined as shown in Table 33.

Index	Sub-Index	Name	Data Type
3xyz*	0	Highest sub-index supported	UNSIGNED8
	1	Argument 1 Source	UNSIGNED8
	2	Argument 1 Number	UNSIGNED8
	3	Argument 2 Source	UNSIGNED8
	4	Argument 2 Number	UNSIGNED8
	5	Operator	UNSIGNED8

** Logic Block X Function Y Condition Z, where X = 4 to 7, Y = A, B or C, and Z = 1 to 3*

Table 33: LB(3-x) Condition Structure Definition

Objects 3x11h, 3x12h and 3x13h are the conditions evaluated for selecting Table A.
Objects 3x21h, 3x22h and 3x23h are the conditions evaluated for selecting Table B.
Objects 3x31h, 3x32h and 3x33h are the conditions evaluated for selecting Table C.

Argument 1 is always a logical output from another function block. As always, the input is a combination of the functional block objects 3xyzh sub-index 1 **"Argument 1 Source"** and **"Argument 1 Number."**

Argument 2 on the other hand, could either be another logical output such as with Argument 1, OR a constant value set by the user. To use a constant as the second argument in the operation, set **"Argument 2 Source"** to *'Constant Function Block',* and **"Argument 2 Number"** to the desired sub-index. When defining the constant, make sure it uses the same resolution (decimal digits) as the Argument 1 input.

Argument 1 is evaluated against Argument 2 based on the **"Operator"** selected in sub-index 5 of the condition object. The options for the operator are listed in Table 34, and the default value is always *'Equal'* for all condition objects.

Value	Meaning
0	**=, Equal**
1	!=, Not Equal
2	>, Greater Than
3	>=, Greater Than or Equal
4	<, Less Than
5	<=, Less Than or Equal

Table 34: LB(3-x) Condition Operator Options

By default, both arguments are set to *'Control Source Not Used'* which disables the condition, and automatically results in a value of N/A. Although it is generally considered that each condition will be evaluated as either TRUE or FALSE, the reality is that there could be four possible results, as described in Table 35.

Value	Meaning	Reason

0	False	(Argument 1) Operator (Argument 2) = False
1	True	(Argument 1) Operator (Argument 2) = True
2	Error	Argument 1 or 2 output was reported as being in an error state
3	Not Applicable	Argument 1 or 2 is not available (i.e. set to *'Control Source Not Used'*)

Table 35: LB(3-x) Condition Evaluation Results

2.6.2 Table Selection

In order to determine if a particular table will be selected, logical operations are performed on the results of the conditions as determined by the logic in 2.5.1. There are several logical combinations that can be selected, as listed in Table 36. The default value for object 3x02h **LB(3-x) Function Logical Operator** is dependent on the sub-index. For sub-index 1 (Table A) and 2 (Table B), the *'Cnd1 And Cnd2 And Cnd3'* operator is used, whereas sub-index 3 (Table C) is setup as the *'Default Table''* response.

Value	Meaning
0	Default Table
1	Cnd1 And Cnd2 And Cnd3
2	Cnd1 Or Cnd2 Or Cnd3
3	(Cnd1 And Cnd2) Or Cnd3
4	(Cnd1 Or Cnd2) And Cnd3

Table 36: LB(3-x) Function Logical Operator Options

Not every evaluation is going to need all three conditions. The case given in the earlier section, for example, only has one condition listed, i.e. that the Engine RPM be below a certain value. Therefore, it is important to understand how the logical operators would evaluate an Error or N/A result for a condition, as outlined in Table 37.

Logical Operator	Select Conditions Criteria
Default Table	Associated table is automatically selected as soon as it is evaluated.
Cnd1 And Cnd2 And Cnd3	**Should be used when two or three conditions are relevant, and all must be True to select the table.** If any condition equals False or Error, the table is not selected. An N/A is treated like a True. If all three conditions are True (or N/A), the table is selected. If((Cnd1==True) &&(Cnd2==True)&&(Cnd3==True)) Then Use Table
Cnd1 Or Cnd2 Or Cnd3	**Should be used when only one condition is relevant. Can also be used with two or three relevant conditions.** If any condition is evaluated as True, the table is selected. Error or N/A results are treated as False If((Cnd1==True) || (Cnd2==True) || (Cnd3==True)) Then Use Table
(Cnd1 And Cnd2) Or Cnd3	**To be used only when all three conditions are relevant.**

	If both Condition 1 and Condition 2 are True, OR Condition 3 is True, the table is selected. Error or N/A results are treated as False If(((Cnd1==True)&&(Cnd2==True)) || (Cnd3==True)) Then Use Table
(Cnd1 Or Cnd2) And Cnd3	**To be used only when all three conditions are relevant.** If Condition 1 And Condition 3 are True, OR Condition 2 And Condition 3 are True, the table is selected. Error or N/A results are treated as False If(((Cnd1==True)||(Cnd2==True)) && (Cnd3==True)) Then Use Table

Table 37: LB(3-x) Conditions Evaluation Based on Selected Logical Operator

If the result of the function logic is TRUE, then the associated lookup table (see object 4x01h) is immediately selected as the source for the logic output. No further conditions for other tables are evaluated. For this reason, the *'Default Table'* should always be setup as the highest letter table being used (A, B or C) If no default response has been setup, the Table A automatically becomes the default when no conditions are true for any table to be selected. This scenario should be avoided whenever possible to not result in unpredictable output responses.

The table number that has been selected as the output source is written to sub-index X of read-only object 4010h **Logic Block Selected Table.** This will change as different conditions result in different tables being used.

2.6.3 Logic Block Output

Recall that Table Y, where Y = A, B or C in the LB(3-x) function block does NOT mean lookup table 1 to 3. Each table has object 3x01h LB(3-x) **Lookup Table Number** which allows the user to select which lookup tables they want associated with a particular logic block. The default tables associated with each logic block are listed in Table 38.

Programmable Logic Block Number	Table A – Lookup Table Block Number	Table B – Lookup Table Block Number	Table C – Lookup Table Block Number
1	1	2	3
2	4	5	6
3	7	8	9
4	10	11	12

Table 38: LB(3-x) Default Lookup Tables

If the associated Lookup Table YZ (where YZ equals 3310h sub-index X) does not have an **"X-Axis Source"** selected, then the output of LB(3-x) will always be "Not Available" so long as that table is selected. However, should LTyz be configured for a valid response to an input, be it Data or Time, the output of the LTyz function block (i.e. the Y-Axis data that has been selected based on the X-Axis value) will become the output of the LB(3-x) function block so long as that table is selected.

The LB(3-x) output is always setup as a percentage, based on the range of the Y-Axis for the associated table (see Section 2.5.2) It is written to sub-index X of read-only object 3320h **Logic Block Output PV** with a resolution of 1 decimal place.

2.7 Math Function Block

There are six mathematic function blocks that allow the user to define basic algorithms. Math function block Z = 1 to 6 will be enabled based on sub-index Z in object 4000h **Math Enable**.

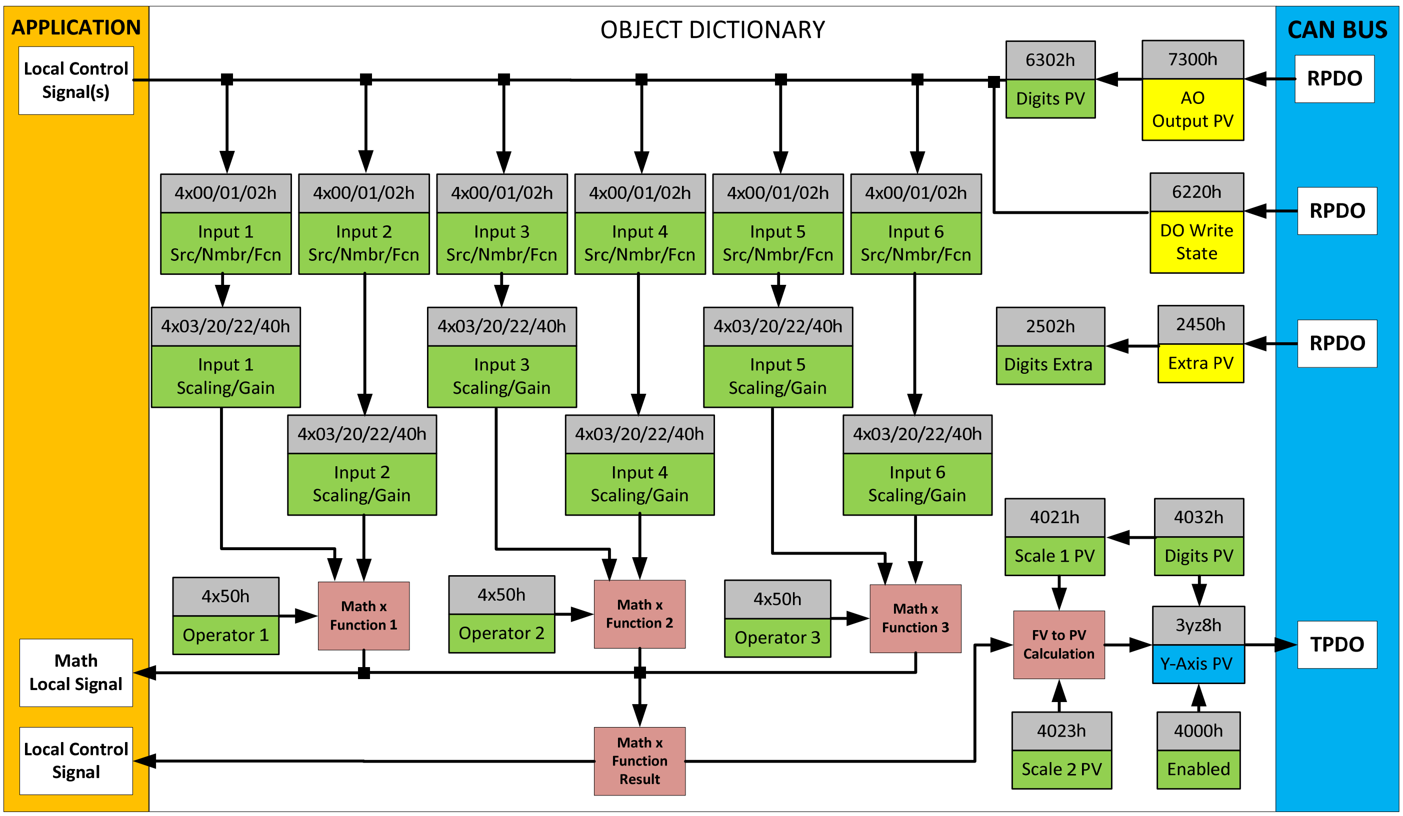


Figure 13: Math Function Block Objects

A math function block can take up to six input signals. Each input is then scaled according the associated scaling and gain objects. A "Math Input X" is determined by the corresponding sub-index X = 1 to 6 of the objects 4y00h **Math Y Input Source** and 4y01h **Math Y Input Number.** Here, y = 1 to 6; corresponding the Math 1- Math 6.

Inputs are converted into a percentage value based on objects 4y20h **Math Y Scaling 1 FV** and 4y22h **Math Y Scaling 2 FV**. Before being used in the calculation, these objects apply the resolution shift defined by object 4y02h **Math Y Decimal Digits FV**. As with any other function block using a control source for the X-Axis in a conversion, the scaling objects should be selected to match the values in the control's corresponding objects as per Table 29.

For additional flexibility, the user can also adjust object 4y40h **Math Y Input Gain.** This object has a fixed decimal digit resolution of 2, and a range of -100 to 100. By default, each input has a gain of 1.0.

For example, in the case where the user may want to combine two inputs such that a joystick (Input 1) is the primary control of an output, but the speed can be incremented or decremented based on a potentiometer (Input 2), it may be desired that 75% of the scale is controlled by the joystick position, while the potentiometer can increase or decrease the min/max output by up to 25%. In this case, Input 1 would have a gain of 0.75, while Input 2 uses 0.25. The resulting addition will give a command from 0 to 100% based on the combined positions of both inputs.

For each input pair, the appropriate arithmetic or logical operation is performed on the two inputs, InA and InB, according the associated function in sub-index of InB in object 4y50h **Math Y Operator**. The list of selectable function operations is defined in Table 39.

0	=	True when InA Equals InB
1	!=	True when InA Not Equal InB
2	>	True when InA Greater Than InB
3	>=	True when InA Greater Than or Equal InB
4	<	True when InA Less Than InB
5	<=	True when InA Less Than or Equal InB
6	OR	True when InA or InB is True
7	NOR	True when InA and InB are False
8	AND	True when InA and InB are True
9	NAND	True when InA and InB are not both True
10	XOR	True when InA/InB is True, but not both
11	XNOR	True when InA and InB are both True or False
12	+	Result = InA plus InB
13	-	Result = InA minus InB
14	x	Result = InA times InB
15	/	Result = InA divided by InB
16	MIN	Result = Smallest of InA and InB
17	MAX	Result = Largest of InA and InB

Table 39: Object 4y50h Math Function Operators

For Function 1, InA and InB are Math Inputs 1 and 2, respectively.
For Function 2, InA and InB are Math Inputs 3 and 4, respectively.
For Function 3, InA and InB are Math Inputs 5 and 6, respectively.

Exclusively **within** a Math Block, there is a third control parameter: Object 4y02h, **Math Y Function Number.** This parameter allows for the result of any Function (1, 2 or 3) to be the input to any **Math Input Y** within the same Math Block. Therefore, **Math Y Input Source** must be a Math Block and **Math Y Input Number** must be the same number as being configured. When these four parameters match, if **Math Y Function Number** is set to 1, 2, or 3, the respective input will be the result of the Function selected. By default, **Math Y Function Number** is set to 0 – in which case this parameter is ignored and uses the Math Block output result. These functions can only be used **within** the Math Block. They can not be used for other Math Blocks or logic blocks.

This allows for more versatility within the Math Block. For a valid result in each Function, both inputs must be non-zero value (other than *'Control Source Not Used'*). Otherwise, the corresponding Function is ignored. Furthermore, for a valid/expected output result in each Math Block, it is necessary to keep in mind how the Functions link to one another within the Math Block. As an example, consider all 6 inputs to be CANopen Messages 1 to 6 (thus using all 3 Functions). Since all 3 functions are used but Function 3 has no relation to Function 1 or 2, the result of the Math Block will be the result of Function 3, thus, ignoring Functions 1 and 2.

For logical operators (6 to 11), any SCALED input greater than or equal to 0.5 is treated as a TRUE input. For logic output operators (0 to 11), the result of the calculation for the function will always be 0 (FALSE) or 1 (TRUE).

Error data (i.e. input measured out of range) is always treated as a 0.0 input into the function.

For the arithmetic functions (12 to 17), it is recommended to scale the data such that the resulting operation will not exceed full scale (0 to 100%) and saturate the output result.

When dividing, a zero InB value will always result is a zero output value for the associated function. When subtracting, a negative result will always be treated as a zero, unless the function is multiplied by a negative one, or the inputs are scaled with a negative coefficient first.

The resulting final mathematical output calculation is in the appropriate physical units using object 4021h Math Output Scaling 1 PV and 4023h Math Output Scaling 2 PV. These objects are also considered the Min and Max values of the Math Block output and apply the resolution shift defined by object 4032h Math Output Decimal Digits PV. The result is written to read-only object 4030h Math Output PV. These scaling objects should also be taken into account when the Math Function is selected as the input source for another function block.

2.8 Simple Conditional Function Block

This function block is a simple version of the Programmable Logic Function Block and compares up to four different input sources with different logical or relational operators. The result of each block can therefore only be true (1) or false (0). Figure 14demonstrates the connections between all parameters.

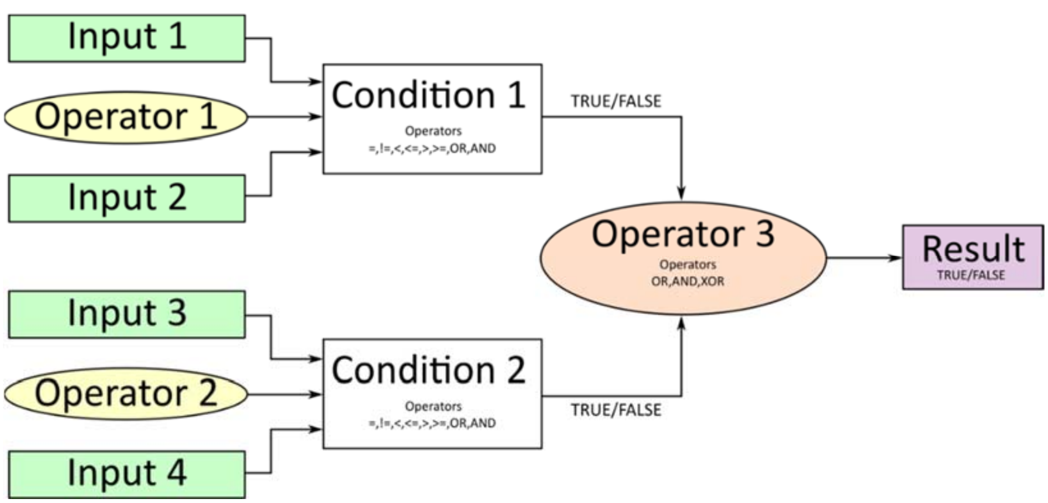


Figure 14: Conditional Block Diagram

Each Conditional Block offers two conditions. Both compare two inputs, which can either be a logical value or an integer value. The result of the conditions can, therefore, only be true or false and will be compared by Operator 3 with a logical operator. This comparison is the result of the Conditional Block and can control any output source.

The value of Input 1 to Input 4 depends on the configured control source and control number. All Inputs are set to No Source as an input source by default.

The value of each source will then be compared to each other with an operator of Table 40. If no source is selected, the output value of an Input will be zero.

Value	Meaning
0	==, True when Argument 1 is equal to Argument 2
1	!=, True when Argument 1 is not equal to Argument 2
2	>, True when Argument 1 is greater than Argument 2
3	>=, True when Argument 1 is greater than Argument 2
4	<, True when Argument 1 is less than Argument 2
5	<=, True when Argument 1 is less than or equal Argument 2
6	OR, True when Argument 1 or Argument 2 is True
7	**AND, True when Argument 1 and Argument 2 are True**

Table 40: Input Operator Options

Operator 1 and Operator 2 are configured to AND by default. The table above cannot be used for comparing the conditions because they can only be compared with logical operators, which are listed in Table 41.

Value	Meaning
0	OR, True when Argument 1 or Argument 2 is True
1	**AND, True when Argument 1 and Argument 2 are True**
2	XOR, True when Argument 1 is not equal to Argument 2

Table 41: Condition Operator Options

If only one condition is used, it is to make sure that Operator 3 is set to **OR** so that the result is based solely on the condition which has been chosen.

2.9 Fault Detection

In addition to the 13 inputs, the unit also has the ability to detect and report other faults. The types of faults it can detect are:

- Module Over-Temperature
- Power Supply Over Voltage
- Power Supply Under Voltage

The objects associated with Fault Detection are all manufacturer defined objects and are described in detail in the Object Dictionary.

3 Flashing new Firmware

This chapter describes a step by step procedure to flash a new application firmware in the field using the unit's embedded bootloader. First, the prerequisites to set up the process are listed below:

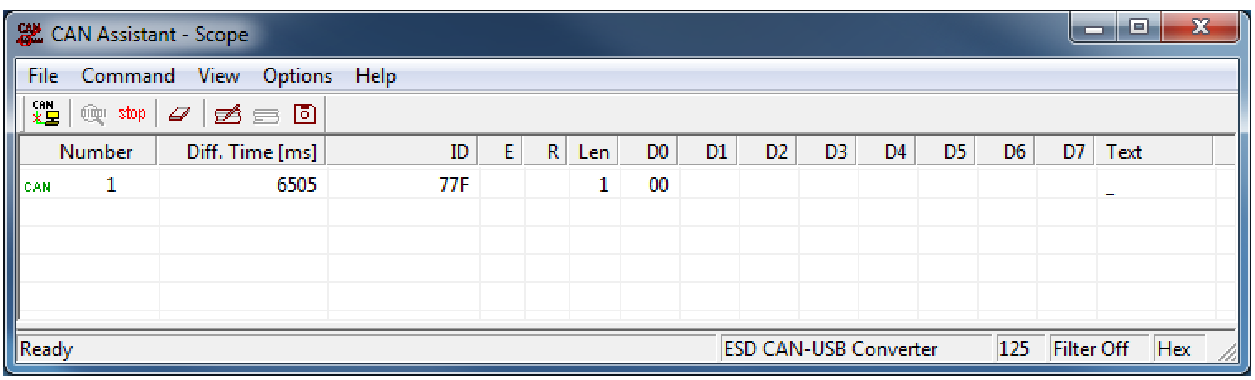
- A personal computer with a USB port running Windows operating system.
- A flash file for AX140603, which should have the following name convention: AF-140603-xx.xx-sss.bin, where *xx.xx* describes the firmware version number, and *sss* are file comments for information purposes.
- Axiomatic Electronic Assistant (EA) software, p/n AX070500.
- Axiomatic CAN Assistant – Scope software, p/n AX070501SCO.
- Axiomatic USB-CAN Converter, p/n AX070501. It should be connected to the USB port of the personal computer.
- Power supply to power the converter.
- Wire harness to connect the converter to the power supply and to the CAN port of the Axiomatic USB-CAN converter with the proper termination resistance.

Next, the procedure itself will be shown step by step by enumeration.

1. Connect the unit, AX140603, to the power supply and Axiomatic USB-CAN converter.

2. Open CAN port and start monitoring the CAN bus in CAN Assistant – Scope. Make sure that the baud rate is set to **125 kBit/s**.

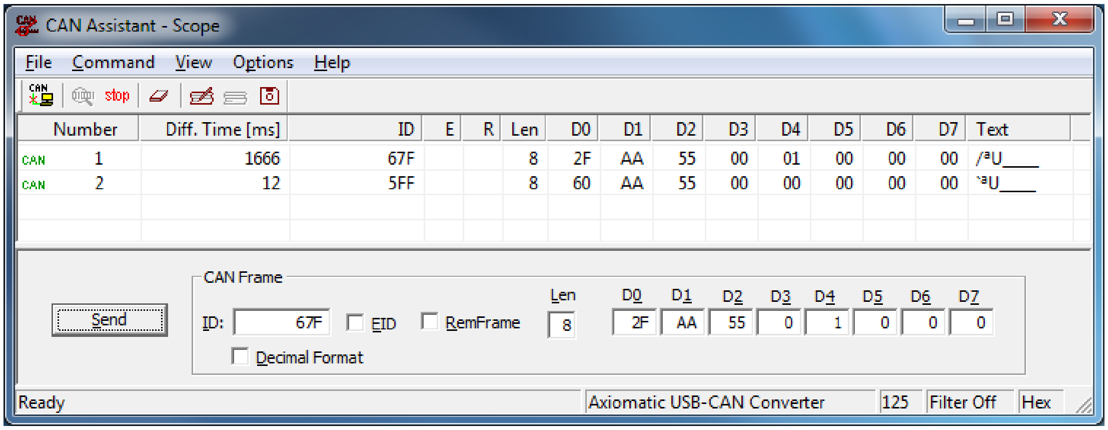
3. Power-up the converter. The user should see a single boot-up message from the converter:



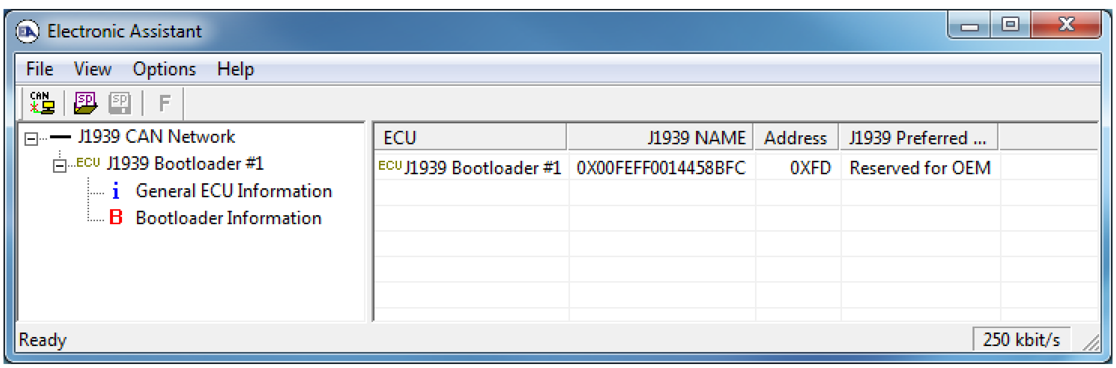
4. Set the CANopen object 0x55AA to 1 using SDO protocol. To do so, send the following message using CAN Assistant – Scope:

ID	Len	D0	D1	D2	D3	D4	D5	D6	D7
67F	**8**	**2F**	**AA**	**55**	**00**	**01**	**00**	**00**	**00**

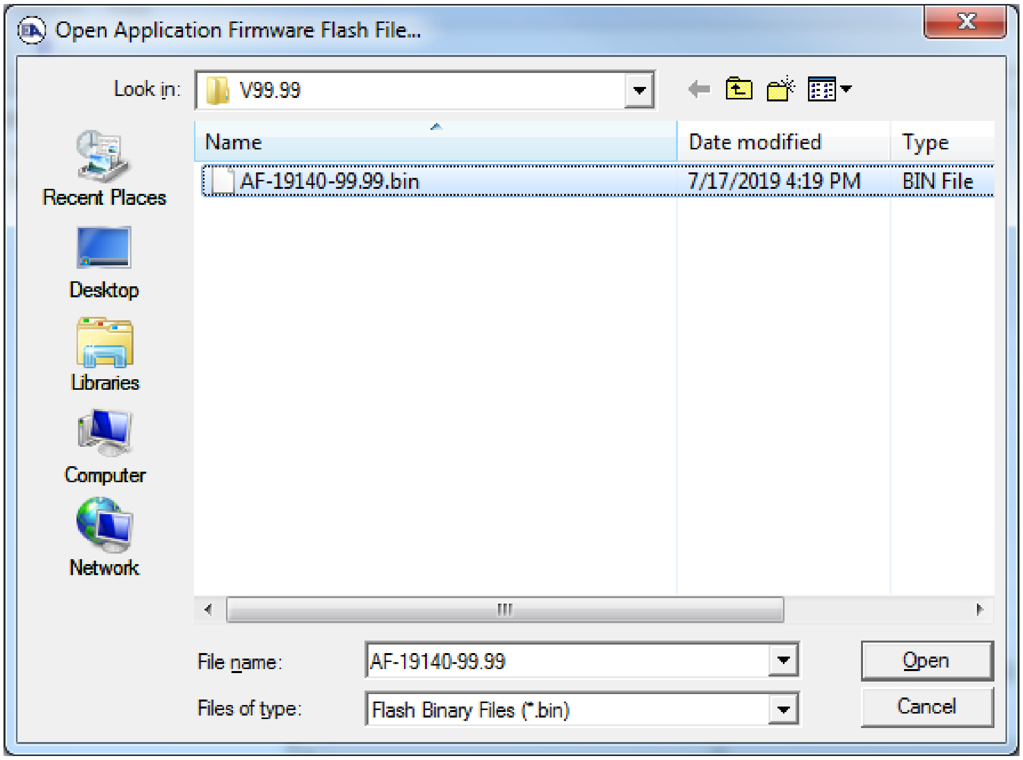
If everything is correct, the converter acknowledges the writing operation and activates the bootloader after 5 seconds after the message is sent.



5. Run the Electronic Assistant (EA) software and connect to the CAN network. Make sure that under Options -> CAN Interface Setup, the Baud Rate is set to 250 kbit/s. The user should see the following screen with these settings:



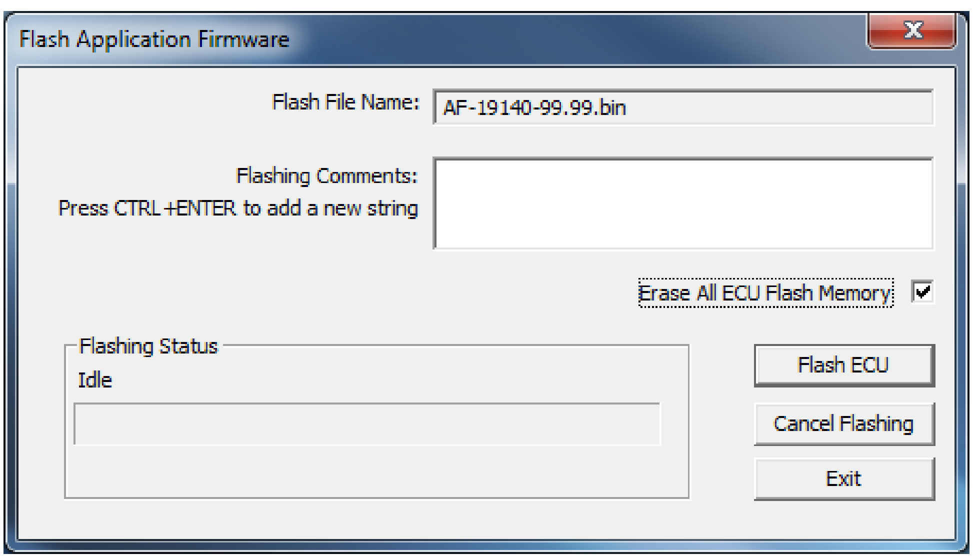
6. Click on the *Bootloader Information* group in the left pane and then on the ***F*** button in the EA toolbar. Select the flash file and click the 'open' button:



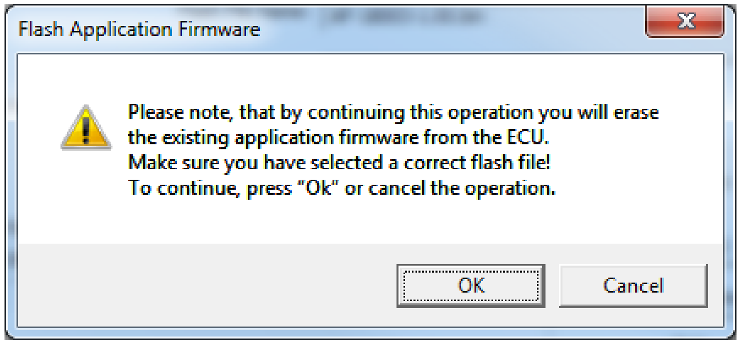
7. Open the flash file and start flashing operation by pressing the *Flash ECU* button.

Note: The erase all ECU Flash Memory option will erase the converted configuration and will set all parameters to default of the new firmware file. If unaware, please contact Axiomatic Technologies.

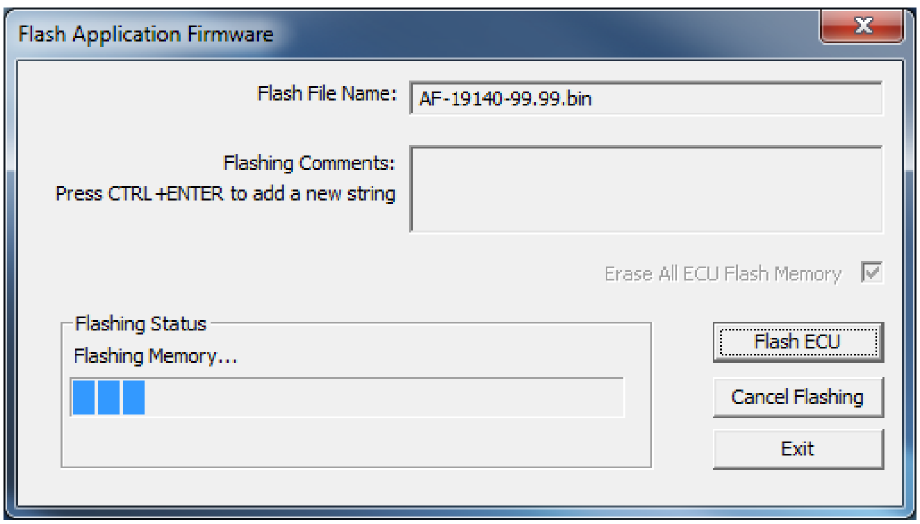
Optionally, the user can write their comments in the *Flashing Comments* field.



8. Confirm the warning message from EA.



After confirming flashing, the user will see the flashing operation in dynamics on the EA screen



9. When flashing is done, reset the ECU.

After ECU is reset, the J1939 bootloader is no longer active, and the new CANopen application firmware should be running.

10. Check the new application firmware. Cycle the power to the controller. In the CAN Assistant – Scope the user should see a single message from the new application firmware reporting that the unit is in the operational mode.

11. End of procedure.

4 Installation Instructions

The CANopen ® object dictionary of the AX140603 Controller is based on CiA device profile DS-402 V4.1.0. The object dictionary includes Communication Objects beyond the minimum requirements in the profile, as well as several manufacturer-specific objects for extended functionality.

4.1 Node ID and Baud rate

By default, the AX140603 controller ships factory programmed with a

Node ID = 127 (0x7F)

and with

Baud rate = 125 kbps.

4.1.1 LSS Protocol to Update

The only means by which the Node-ID and Baudrate can be changed is to use Layer Settling Services (LSS) and protocols as defined by CANopen ® standard DS-305.

Follow the steps below to configure either variable using LSS protocol. If required, please refer to the standard for more detailed information about how to use the protocol

4.1.2 Setting Node-ID

- Set the module state to LSS-configuration by **sending** the following message:

Item	***Value***
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)
Data 1	0x01 (switches to configuration state)

- Set the Node-ID by **sending** the following message:

Item	***Value***
COB-ID	0x7E5
Length	2

Data 0	0x11	(cs=17 for configure node-id)
Data 1	Node-ID	(set new Node-ID as a hexadecimal number)

- The module will send the following response (any other response is a failure).

Item	***Value***	
COB-ID	0x7E4	
Length	3	
Data 0	0x11	(cs=17 for configure node-id)
Data 1	0x00	
Data 2	0x00	

- Save the configuration by **sending** the following message:

Item	***Value***	
COB-ID	0x7E5	
Length	1	
Data 0	0x17	(cs=23 for store configuration)

- The module will send the following response (any other response is a failure)

Item	***Value***	
COB-ID	0x7E4	
Length	3	
Data 0	0x17	(cs=23 for store configuration)
Data 1	0x00	
Data 2	0x00	

- Set the module state to LSS-operation by **sending** the following message: (Note, the module will reset itself back to the pre-operational state)

Item	***Value***	
COB-ID	0x7E5	
Length	2	
Data 0	0x04	(cs=4 for switch state global)
Data 1	0x00	(switches to waiting state)

4.1.3 Setting Baudrate

- Set the module state to LSS-configuration by sending the following message:

Item	***Value***
COB-ID	0x7E5
Length	2

Data 0	0x04	(cs=4 for switch state global)
Data 1	0x01	(switches to configuration state)

- Set the baudrate by sending the following message:

Item	***Value***	
COB-ID	0x7E5	
Length	3	
Data 0	0x13	(cs=19 for configure bit timing parameters)
Data 1	0x00	(switches to waiting state)
Data 2	Index	(select baudrate index per Table 32)

Table 42 – LSS Baudrate Indices

Index	***Bit Rate***
0	1 Mbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s (default)
5	reserved (100 kbit/s)
6	50 kbit/s
7	20 kbit/s
8	10 kbit/s

- The module will send the following response (any other response is a failure):

Item	***Value***	
COB-ID	0x7E4	
Length	3	
Data 0	0x13	(cs=19 for configure bit timing parameters)
Data 1	0x00	
Data 2	0x00	

- Activate bit timing parameters by sending the following message:

Item	***Value***	
COB-ID	0x7E5	
Length	3	
Data 0	0x15	(cs=19 for activate bit timing parameters)
Data 1	<delay_lsb>	
Data 2	<delay_msb>	

The delay individually defines the duration of the two periods of time to wait until the bit timing parameters switch is done (first period) and before transmitting any CAN message with the new bit timing parameters after performing the switch (second period). The time unit of switch delay is 1 ms.

- Save the configuration by sending the following message (on the NEW baudrate):

Item	***Value***
COB-ID	0x7E5
Length	1
Data 0	0x17 (cs=23 for store configuration)

- The module will send the following response (any other response is a failure):

Item	***Value***
COB-ID	0x7E4
Length	3
Data 0	0x17 (cs=23 for store configuration)
Data 1	0x00
Data 2	0x00

- Set the module state to LSS-operation by sending the following message: (Note, the module will reset itself back to the pre-operational state)

Item	***Value***
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)
Data 1	0x00 (switches to waiting state)

4.2 Communication Objects (DS-301)

The communication objects supported by the AX030140 Controller are listed in the following table. A more detailed description of some of the objects is given in the following subchapters. Only those objects that have device-profile specific information are described. For more information on the other objects, refer to the generic CANopen protocol specification DS-301.

Index (hex)	*Object*	*Object Type*	*Data Type*	*Access*	*PDO Mapping*
1000	Device Type	VAR	UNSIGNED32	RO	No
1001	Error Register	VAR	UNSIGNED8	RO	No
1002	Manufacturer Status Register	VAR	UNSIGNED32	RO	No
1003	Pre-Defined Error Field	ARRAY	UNSIGNED32	RO	No
1009	Manufacturer Hardware Version	ARRAY	UNSIGNED8	RO	No
100A	Manufacturer Software Version	ARRAY	UNSIGNED8	RO	No
100C	Guard Time	VAR	UNSIGNED16	RW	No
100D	Life Time Factor	VAR	UNSIGNED8	RW	No
1010	Store Parameters	ARRAY	UNSIGNED32	RW	No
1011	Restore Default Parameters	ARRAY	UNSIGNED32	RW	No
1016	Consumer Heartbeat Time	ARRAY	UNSIGNED32	RW	No
1017	Producer Heartbeat Time	VAR	UNSIGNED16	RW	No
1018	Identity Object	RECORD		RO	No
1020	Verify Configuration	ARRAY	UNSIGNED32	RO	No
1029	Error Behavior	ARRAY	UNSIGNED8	RW	No
1400	RPDO1 Communication Parameter	RECORD		RW	No
1401	RPDO2 Communication Parameter	RECORD		RW	No
1402	RPDO3 Communication Parameter	RECORD		RW	No
1403	RPDO4 Communication Parameter	RECORD		RW	No
1600	RPDO1 Mapping Parameter	RECORD		RO	No
1601	RPDO2 Mapping Parameter	RECORD		RO	No
1602	RPDO3 Mapping Parameter	RECORD		RO	No
1603	RPDO4 Mapping Parameter	RECORD		RO	No
1800	TPDO1 Communication Parameter	RECORD		RW	No
1801	TPDO2 Communication Parameter	RECORD		RW	No
1802	TPDO3 Communication Parameter	RECORD		RW	No
1803	TPDO4 Communication Parameter	RECORD		RW	No
1A00	TPDO1 Mapping Parameter	RECORD		RW	No
1A01	TPDO2 Mapping Parameter	RECORD		RW	No
1A02	TPDO3 Mapping Parameter	RECORD		RW	No
1A03	TPDO4 Mapping Parameter	RECORD		RW	No

4.2.1 PDO Parameters

All RPDOs and TPDOs in the 3 Encoder, 7 Signal Inputs Controller use the same default communication parameters, respectively. The PDO IDs are set according to the pre-defined connection set described in [DS-301]. All receive PDOs are set to transmission type 255, and all transmit PDOs to transmission type 254, with the event timer (sub-index 5) set to 100 (100ms).

All PDOs are dynamically mappable, and the user can therefore change the mapping of the PDOs. The granularity is 8-bits, so the objects can be mapped with byte offsets. The mapping parameter records include 4 subindexes for the PDO mapping. The default PDO mappings are listed in the following tables.

RPDO1: default ID 0x200 + node ID

Subindex	***Value***	*Object*
0	4	Number of mapped application objects in PDO
1	0x25000110	Extra Received PV 1
2	0x25000210	Extra Received PV 2
3	0x25000310	Extra Received PV 3
4	0x25000410	Extra Received PV 4

RPDO2: default ID 0x300 + node ID

Subindex	***Value***	***Object***
0	8	Number of mapped application objects in PDO
1	0x25000108	Extra Received PV 1
2	0x25000208	Extra Received PV 2
3	0x25000308	Extra Received PV 3
4	0x25000408	Extra Received PV 4
5	0x25000508	Extra Received PV 5
6	0x0	
7	0x0	
8	0x0	

RPDO3: default ID 0x400 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0x0	
2	0x0	
3	0x0	
4	0x0	

RPDO4: default ID 0x500 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

TPDO1: default ID 0x180 + node ID

Subindex	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x71000108	AI Input Field Value for AI 1
2	0x71000308	AI Input Field Value for AI 3
3	0x71000508	AI Input Field Value for AI 5
4	0x71000708	AI Input Field Value for AI 7
5	0x71000808	AI Input Field Value for AI 8
6	0x71000908	AI Input Field Value for AI 9
7	0x71000A08	AI Input Field Value for AI 10
8	0x71000B08	AI Input Field Value for AI 11

TPDO2: default ID 0x280 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0x0	
2	0x0	
3	0x0	
4	0x0	

TPDO3: default ID 0x380 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

TPDO4: default ID 0x480 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

4.2.2 Object 1000h: Device Type

This object contains information about the device type as per device profile DS-404. The value stored in this object is 0x00030140, indicating that the controller includes the following function blocks defined in the device profile.

- Digital Input (DI)
- Analog Input (AI)

Object Description

Index	1000h
Name	Device Type
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Access	RO
PDO Mapping	No
Value Range	0x00030140
Default Value	0x00030140

4.2.3 Object 1001h: Error Register

This object is an error register for the device. Any time there is an error detected by the controller, the Generic Error Bit (bit 0) is set. Only if there are no errors in the module will this bit be cleared. No other bits in this register are used by this unit.

Object Description

Index	1001h
Name	Error Register
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Access	RO
PDO Mapping	No
Value Range	00h or 01h
Default Value	0

4.2.4 Object 1003h: Pre-Defined Error Field

The object 1003h provides an error history by listing the errors in the order that they have occurred. An error is added to the top of the list when it occurs and is immediately removed when the error condition has been cleared. The latest error is always at subindex 1, with subindex 0 containing the number of errors currently in the list. When the device is in an error-free state, the value of subindex 0 is zero.

The error list may be cleared by writing a zero to subindex 0, which will clear all errors from the list, regardless of whether or not they are still present. Clearing the list does NOT mean that the module will return to the error-free behaviour state if at least one error is still active.
The controller has a limitation of a maximum of 20 errors in the list. If the device registers more errors, the list will be truncated, and the oldest entries will be lost.

The error codes stored in the list are 32-bit unsigned numbers, consisting of two 16-bit fields. The lower 16-bit field is the EMCY error code, and the higher 16-bit field is a manufacturer-specific code. The manufacturer-specific code is divided into two 8-bit fields, with the higher byte indicating the error description, and the lower byte indicating the channel number where the error occurred.

MSB			**LSB**
Error Description	Channel	EMCY Error Code	

See Section 4 for a complete list of the error code fields.

Object Description

Index	1003h
Name	Pre-Defined Error Field
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Subindex	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	0 to 16
Default Value	0

Subindex	1h to 10h
Description	Standard error field
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

4.2.5 Object 1010h: Store Parameters

This object supports the saving of parameters in non-volatile memory. In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate subindex. The signature is “save”.

The signature is a 32-bit unsigned number, composed of the ASCII codes of the signature characters, according to the following table:

MSB			LSB
e	v	a	s

65h	76h	61h	73h

On reception of the correct signature to an appropriate subindex, the controller will store the parameters in non-volatile memory, and then confirm the SDO transmission.

By read access, the object provides information about the module's saving capabilities.

Object Description

Index	1010h
Name	Store Parameters
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Subindex	0h
Description	Largest subindex supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Subindex	1h
Description	Save all parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h (saves parameters on command)

Subindex	2h
Description	Save communication parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h (saves parameters on command)

Subindex	3h
Description	Save application parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h (saves parameters on command)

Subindex	4h
Description	Save manufacturer parameters

Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h (saves parameters on command)

4.2.6 Object 1011h: Restore Default Parameters

This object supports the restoring of the default values for the object dictionary in non-volatile memory. In order to avoid restoring of parameters by mistake, the device restores the defaults only when a specific signature is written to the appropriate subindex. The signature is “load”.

The signature is a 32-bit unsigned number, composed of the ASCII codes of the signature characters, according to the following table:

MSB			LSB
d	a	o	l
64h	61h	6Fh	6Ch

On reception of the correct signature to an appropriate subindex, the controller will restore the defaults in non-volatile memory, and then confirm the SDO transmission. The default values are set valid after the device is reset or power-cycled.

By read access, the object provides information about the module’s default parameter restoring capabilities.

Object Description

Index	1011h
Name	Restore Default Parameters
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Subindex	0h
Description	Largest subindex supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Subindex	1h
Description	Restore all default parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access) 1h (read access)
Default Value	1h (restores defaults on command)

Subindex	2h

Description	Restore default communication parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access) 1h (read access)
Default Value	1h (restores defaults on command)

Subindex	3h
Description	Restore default application parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access) 1h (read access)
Default Value	1h (restores defaults on command)

Subindex	4h
Description	Restore default manufacturer parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access) 1h (read access)
Default Value	1h (restores defaults on command)

4.2.7 Object 1016h: Consumer Heartbeat Time

The unit can be a consumer of heartbeat objects for up to four modules. This object defines the expected heartbeat cycle time for those modules, and if set to zero, it is not used. When value is non-zero, the time is a multiple of 1ms, and monitoring will start after the reception of the first heartbeat from the module. If the controller fails to receive a heartbeat from a node in the expected timeframe, it will indicate a communication error, and respond as per object 1029h.

Bits	31-24	23-16	15-0
Value	Reserved 00h	Node-ID	Heartbeat time
Encoded as		UNSIGNED8	UNSIGNED16

Object Description

Index	1016h
Name	Consumer heartbeat time
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Subindex	0h
Description	Number of entries
Access	RO
PDO Mapping	No

Value Range	4
Default Value	4

Subindex	1h to 4h
Description	Consumer heartbeat time
Access	RW
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

4.2.8 Object 1017h: Producer Heartbeat Time

The controller could be configured to produce a cyclical heartbeat by writing a non-zero value to this object. The value will be given in multiples of 1ms, and a value of 0 shall disable the heartbeat.

Object Description

Index	1017h
Name	Producer heartbeat time
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	10 to 65535
Default Value	0

4.2.9 Object 1018h: Identity Object

The identity object indicates the data of the controller, including vendor id, device id, software and hardware version numbers, and the serial number.

In the Revision Number entry at subindex 3, the format of the data is as shown below

MSB	Byte 2	Byte 1	Byte 0
0	Major revision number (object dictionary)	Hardware Revision	Software Revision

Object Description

Index	1018h
Name	Identity
Object Type	RECORD
Data Type	Identity Record

Entry Description

Subindex	0h
Description	Number of entries
Access	RO

PDO Mapping	No
Value Range	4
Default Value	4

Subindex	1h
Description	Vendor ID
Access	RO
PDO Mapping	No
Value Range	0x00000055
Default Value	0x00000055 (Axiomatic)

Subindex	2h
Description	Product Code
Access	RO
PDO Mapping	No
Value Range	0xAA020502
Default Value	0xAA020502
Subindex	3h
Description	Revision Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Subindex	4h
Description	Serial Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

4.2.10 Object 1029h: Error Behaviour

This object controls the state that the controller will be set into in case of an error of the type associated with the subindex. The behaviour of the controller in each state is described in more detail in section 3.5.

Object Description

Index	1029h
Name	Error Behaviour
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Subindex	0h
Description	Number of entries
Access	RO

PDO Mapping	No
Value Range	6
Default Value	6

Subindex	1h
Description	Communication Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change 2 = Stopped
Default Value	0 (Pre-Operational)

Subindex	2h
Description	Digital Input Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change 2 = Stopped
Default Value	1 (no state change)

Subindex	3h
Description	Analog Input Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change 2 = Stopped
Default Value	1 (no state change)

Subindex	4h
Description	Digital Output Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change 2 = Stopped
Default Value	1 (no state change)

Subindex	5h
Description	Analog Output Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change

	2 = Stopped
Default Value	1 (no state change)

Subindex	6h
Description	Fault Detection Error
Access	RW
PDO Mapping	No
Value Range	0 = Pre-Operational 1 = No State Change 2 = Stopped
Default Value	1 (no state change)

NOTE: Subindexes 4 and 5 are not used by this module, and have been left for compatibility with other Axiomatic I/O Controllers

4.3 Manufacturer Objects

Index [hex]	Object	Object Type	Data Type	Access	PDO Mapping
2010	AI Input range	ARRAY	UNSIGNED8	RW	No
2011	AI Decimal Digits	ARRAY	UNSIGNED8	RW	No
2101	DI Input Logic Type	ARRAY	UNSIGNED8	RW	No
2102	DI Active State	ARRAY	UNSIGNED8	RW	No
2103	DI Debounce Time	ARRAY	UNSIGNED16	RW	No
2104	DI Debounce Filter	ARRAY	UNSIGNED8	RW	No
2105	DI Overflow Value	ARRAY	FLOAT32	RW	No
2106	DI Filter Type	ARRAY	UNSIGNED8	RW	No
2107	DI Number of Pulses per Revolution	ARRAY	UNSIGNED16	RW	No
2108	DI Number to Reset Pulse Counter	ARRAY	UNSIGNED32	RW	No
2109	DI Pulse Count Pulse Window	ARRAY	FLOAT32	RW	No
210A	DI Pulse Count Time Window	ARRAY	FLOAT32	RW	No
210F	DI Error Detection Enable	ARRAY	UNSIGNED8	RW	No
2110	DI Clear Error Hysteresis	ARRAY	UNSIGNED16	RW	No
2111	DI Error React Delay	ARRAY	UNSIGNED16	RW	No
2200	UNI Active State	ARRAY	UNSIGNED8	RW	No
2201	UNI Debounce Time	ARRAY	UNSIGNED16	RW	No
2202	UNI Debounce Filter	ARRAY	UNSIGNED8	RW	No
2203	UNI Overflow Value	ARRAY	FLOAT32	RW	No
2204	UNI Number of Pulses per Revolution	ARRAY	UNSIGNED16	RW	No
2205	UNI Number to Reset Pulse Counter	ARRAY	UNSIGNED32	RW	No
2206	UNI Pulse Count Pulse Window	ARRAY	FLOAT32	RW	No
2207	UNI Pulse Count Time Window	ARRAY	FLOAT32	RW	No
2208	UNI Pull Up/Down Mode	ARRAY	UNSIGNED8	RW	No
2209	UNI ADC Filter Frequency	ARRAY	UNSIGNED8	RW	No
220A	UNI Filter Type	ARRAY	UNSIGNED8	RW	No
220B	UNI Input Logic Type	ARRAY	UNSIGNED8	RW	No
220C	UNI Error Detection Enable	ARRAY	UNSIGNED8	RW	No
220D	UNI Clear Error Hysteresis	ARRAY	UNSIGNED16	RW	No
220E	UNI Error React Delay	ARRAY	UNSIGNED16	RW	No
2301	ENC Input Resolution	ARRAY	UNSIGNED8	RW	No
2302	ENC Filter Type	ARRAY	UNSIGNED8	RW	No
2303	ENC Stationary Delay	ARRAY	UNSIGNED16	RW	No
2304	ENC Error Detection Enable	ARRAY	UNSIGNED8	RW	No

2305	ENC Clear Error Hysteresis	ARRAY	UNSIGNED16	RW	No
2306	ENC Error React Delay	ARRAY	UNSIGNED16	RW	No
2310	ENC A Input Active State	ARRAY	UNSIGNED8	RW	No
2311	ENC A Input Debounce Time	ARRAY	UNSIGNED16	RW	No
2312	ENC A Debounce Filter	ARRAY	UNSIGNED8	RW	No
2313	ENC A Pull Up/Down Mode	ARRAY	UNSIGNED8	RW	No
2314	ENC A Input Logic Type	ARRAY	UNSIGNED8	RW	No
2320	ENC B Input Active State	ARRAY	UNSIGNED8	RW	No
2321	ENC B Input Debounce Time	ARRAY	UNSIGNED16	RW	No
2322	ENC B Debounce Filter	ARRAY	UNSIGNED8	RW	No
2323	ENC B Pull Up/Down Mode	ARRAY	UNSIGNED8	RW	No
2324	ENC B Input Logic Type	ARRAY	UNSIGNED8	RW	No
2500	EC Extra Received Process Value	ARRAY	INTEGER16	RW	Yes
2502	EC Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
2520	EC Scaling 1 PV	ARRAY	INTEGER16	RW	No
2522	EC Scaling 2 PV	ARRAY	INTEGER16	RW	No
3yz0	LTyz Input X-Axis Source	VAR	UNSIGNED8	RW	No
3yz1	LTyz Input X-Axis Number	VAR	UNSIGNED8	RW	No
3yz2	LTyz Auto Repeat	VAR	UNSIGNED8	RW	No
3yz3	LTyz X-Axis Decimal Digits PV	VAR	UNSIGNED8	RW	No
3yz4	LTyz Y-Axis Decimal Digits PV	VAR	UNSIGNED8	RW	No
3yz5	LTyz Point Response	ARRAY	UNSIGNED8	RW	No
3yz6	LTyz Point X-Axis PV	ARRAY	INTEGER32	RW	No
3yz7	LTyz Point Y-Axis PV	ARRAY	INTEGER16	RW	No
3yz8	LTyz Output Y-Axis PV	VAR	INTEGER16	RO	Yes
3300	Logic Block Enable	ARRAY	BOOLEAN	RW	No
3310	Logic Block Selected Table	ARRAY	UNSIGNED8	RO	Yes
3320	Logic Output Process Value	ARRAY	INTEGER16	RO	Yes
3x01	LB(3-x) Lookup Table Number	ARRAY	UNSIGNED8	RW	No
3x02	LB(3-x) Function Logical Operator	ARRAY	UNSIGNED8	RW	No
3x11	LB(3-x) Function A Condition 1	RECORD	UNSIGNED8	RW	No
3x12	LB(3-x) Function A Condition 2	RECORD	UNSIGNED8	RW	No
3x13	LB(3-x) Function A Condition 3	RECORD	UNSIGNED8	RW	No
3x21	LB(3-x) Function B Condition 1	RECORD	UNSIGNED8	RW	No
3x22	LB(3-x) Function B Condition 2	RECORD	UNSIGNED8	RW	No
3x23	LB(3-x) Function B Condition 3	RECORD	UNSIGNED8	RW	No
3x31	LB(3-x) Function C Condition 1	RECORD	UNSIGNED8	RW	No
3x32	LB(3-x) Function C Condition 2	RECORD	UNSIGNED8	RW	No
3x33	LB(3-x) Function C Condition 3	RECORD	UNSIGNED8	RW	No
4500	Math Block Enable	ARRAY	BOOLEAN	RW	No
4521	Math Output Scaling 1 PV	ARRAY	INTEGER16	RW	No
4523	Math Output Scaling 2 PV	ARRAY	INTEGER16	RW	No
4530	Math Output Process Value	ARRAY	INTEGER16	RO	Yes
4532	Math Output Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
4y00	Math Y Input Source	ARRAY	UNSIGNED8	RW	No
4y01	Math Y Input Number	ARRAY	UNSIGNED8	RW	No
4y02	Math Y Function Number	ARRAY	UNSIGNED8	RW	No
4y03	Math Y Input Decimal Digits FV	ARRAY	UNSIGNED8	RW	No
4y20	Math Y Input Scaling 1 FV	ARRAY	INTEGER16	RW	No
4y22	Math Y Input Scaling 2 FV	ARRAY	INTEGER16	RW	No
4y40	Math Y Input Gain	ARRAY	INTEGER8	RW	No
4y50	Math Y Operator	ARRAY	UNSIGNED8	RW	No
5010	Constant Field Value	ARRAY	FLOAT32	RW	No
5540	Bootup Message Enable	VAR	UNSIGNED8	RW	No
5550	Enable Auto Updates	VAR	UNSIGNED8	RW	No
5555	Start in Operational Mode	VAR	UNSIGNED8	RW	No

55AA	Start Bootloader	VAR	BOOLEAN	RW	No
5B50	Change Baud Rate	VAR	UNSIGNED8	RW	No
5B51	Change Node ID	VAR	UNSIGNED8	RW	No

4.3.1 Object 2010h: AI Input Range

This object, in conjunction with 6110h AI Sensor Type, defines the analog input defaults and allowable ranges for objects 2110h, 220D, 2305h, 7120h, 7122h, 7148h and 7149h. The number and types of ranges will vary according to what type of sensor is connected to the input.

Object Description

Index	2010h
Name	AI Input Range
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	Input X Range
Access	RW
PDO Mapping	No
Value Range	See Table 16 and Table 23
Default Value	0

4.3.2 Object 2011h: AI Decimal Digits FV

This object describes the number of digits following the decimal point (i.e. resolution) of the input data, which is interpreted with data type Integer16 in the field value object.

Example: A field value of 1.230 (Float) will be coded as 1230 in Integer16 format if the number of decimal digits is set to 3.

In addition to the FV object 7100h, objects 2110h, 220D, 2305h, 7120h, 7122h, 7148h and 7149h will also be specified with this resolution. This object is normally read-only, and will be automatically adjusted by the controller as per Table 9 depending on the analog input type and range that has been selected. When object 5550h is set to FALSE (disables automatic updates), this object becomes writeable.

Object Description

Index	2011h
Name	AI Decimal Digits FV

Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	Input X Decimal Digits FV
Access	RW (only when object 5550h is false)
PDO Mapping	No
Value Range	See Table 7, Table 18, Table 26
Default Value Encoder	0
Default Value Universal Input	1
Default Value Digital Input	PWM-Mode: 1 Digital ON/OFF: 0

4.3.3 Object 2101h: DI Input Logic Type

This object allows flexibility in the response of the input. Table 13 demonstrates all possible options for this parameter.

Object Description

Index	2101h
Name	DI Input Logic Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Input X Decimal Digits FV
Access	RW
PDO Mapping	No

Value Range	See Table 13
Default Value	0

4.3.4 Object 2102h: DI Active State

The Active High/Low parameter allows the user to select how the controller responds to the behaviour of the digital input.

Object Description

Index	2102h
Name	DI Active State
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Input X Active State
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	0

4.3.5 Object 2103h: DI Debounce Time

The Digital Input Debounce Time parameter is a useful parameter in cases where the digital input signal coming into the controller is noisy. Please refer to 2.2.3 for more information.

Object Description

Index	2103h
Name	DI Debounce Time
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Input X Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 to 65,535
Default Value	250

4.3.6 Object 2104h: DI Debounce Filter 1 Input Line

This object will debounce the input signal applied on a single digital input.The options for this object are listed in Table 12.

Object Description

Index	2104h
Name	DI Debounce Filter 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Digital Input X Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 12
Default Value	0 [Debounce Filter OFF]

4.3.7 Object 2105h: DI Overflow Value

Object Description

Index	2105h
Name	DI Overflow Value
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO

PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Digital Input X Overflow Value
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	1

4.3.8 Object 2107h: DI Number of Pulses Per Revolution

This object is only used when a "Frequency" input type has been selected by object 6110h. The controller will automatically convert frequency measurement from Hz to RPM when a non-zero value is specified. In this case, objects 2110h, 7120h, 7122h, 7148h and 7149h will be interpreted as RPM data.

Object Description

Index	2107h
Name	DI Number of Pulses Per Revolution
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Digital Input x Pulses per Revolution
Access	RW
PDO Mapping	No
Value Range	0 to 1000
Default Value	0

4.3.9 Object 2108h: DI Number to Reset Pulse Counter

This object is a Counter input type in which the output data of the input function block is the number of pulses measured. The pulses will continue to increment in count as they are measured until the **DI Pulse Count to Reset Counter** value of object 2108h is reached at which the counter will reset back to 0.

Object Description

Index	2108h
Name	DI Number to Reset Pulse Counter
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Digital Input x Number to Reset Pulse Counter
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.10 Object 2109h: DI Pulse Count Pulse Window

This object is a Counter input type in which a 'Pulse Window', is selected in number of pulses. The output of the input block will be the amount of time elapsed to reach the number of pulses set in the Elapsed Time to Each Pulse Count setpoint.

Object Description

Index	2109h
Name	DI Pulse Counter Pulse Window
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)

Description	Digital Input x Pulse Count Pulse Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.11 Object 210Ah: DI Pulse Count Time Window

This object is a Counter input type in which object **DI Pulse Count in Time Window** 210Ah is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Object Description

Index	210Ah
Name	DI Pulse Counter Pulse Window
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Digital Input x Pulse Count Time Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.12 Object 210Fh: DI Error Detection Enable

This object enables error detection and reaction associated with the digital input function block. When disabled, the input will not generate an EMCY code in object 1003h Pre-Defined Error Field, nor will it disable any output controlled by the input should the input go out of range as defined by the objects 7148h AI Span Start and 7149h AI Span End.

Object Description

Index	210Fh
Name	DI Error Detect Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Input x Error Detect Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

4.3.13 Object 2110h: DI Error Clear Hysteresis

This object is used to prevent rapid activation/clearing of an input fault flag and sending of object 1003h to the CANopen® network. Once the input has gone above/below the thresholds that define the valid operating range, it must come back into range minus/plus this value to clear the fault. It is scaled in the physical unit of the FV, i.e. object 2011h applies to this object.

Object Description

Index	2110h
Name	DI Error Clear Hysteresis
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	Input x Error Clear Hysteresis
Access	RW
PDO Mapping	No

Value Range	See Table 20
Default Value	10 [0.1%] (duty cycle)

4.3.14 Object 2111h: DI Error Reaction Delay

This object is used to filter out spurious signals and to prevent saturating the CANopen ® network with broadcasts of object 1003h as the fault is set/cleared. Before the fault is recognized (i.e. the EMCY code is added to the pre-defined error field list), it must remain active throughout the period of time defined in this object. The physical unit for this object is milliseconds.

Object Description

Index	2111h
Name	DI Error Reaction Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	DI x Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

4.3.15 Object 2200h: UNI Active State

The Active High/Low parameter allows the user to select how the controller responds to the behaviour of the digital input.

Object Description

Index	2200h
Name	UNI Active State
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Active State
Access	RW
PDO Mapping	No
Value Range	0(Active High) to 1(Active Low)
Default Value	0(Active High)

4.3.16 Object 2201h: UNI Debounce Time

The Digital Input Debounce Time parameter is a useful parameter in cases where the digital input signal coming into the controller is noisy. Please refer to 2.2.3 for more information.

Object Description

Index	2201h
Name	UNI Debounce Time
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 to 65,535
Default Value	250

4.3.17 Object 2202h: UNI Debounce Filter

This object will debounce the input signal applied on a single digital input. The options for this object are listed in Table 12.

Object Description

Index	2202h

Name	DI Debounce Filter 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 12
Default Value	0 [Debounce Filter OFF]

4.3.18 Object 2203h: UNI Overflow Value

Object Description

Index	2203h
Name	UNI Overflow Value
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Overflow Value
Access	RW
PDO Mapping	No
Value Range	1 to 50
Default Value	1

4.3.19 Object 2204h: UNI Number of Pulses Per Revolution

This object is only used when a "Frequency" input type has been selected by object 6110h. The controller will automatically convert frequency measurement from Hz to RPM when a non-zero value is specified. In this case, objects 220Dh, 7120h, 7122h, 7148h and 7149h will be interpreted as RPM data.

Object Description

Index	220Dh
Name	DI Number of Pulses Per Revolution
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1
Description	Universal Input 1 Pulses per Revolution
Access	RW
PDO Mapping	No
Value Range	0 to 1000
Default Value	0

4.3.20 Object 2205h: UNI Number to Reset Pulse Counter

This object is a Counter input type in which the output data of the input function block is the number of pulses measured. The pulses will continue to increment in count as they are measured until the **UNI Pulse Count to Reset Counter** value of object 2205h is reached at which the counter will reset back to 0.

Object Description

Index	2205h
Name	UNI Number to Reset Pulse Counter
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO

PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1
Description	Universal Input 1 Number to Reset Pulse Counter
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.21 Object 2206h: UNI Pulse Count Pulse Window

This object is a Counter input type in which a ‘Pulse Window’, is selected in number of pulses. The output of the input block will be the amount of time elapsed to reach the number of pulses set in the Elapsed Time to Each Pulse Count setpoint.

Object Description

Index	2109h
Name	DI Pulse Counter Pulse Window
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1
Description	Universal Input 1 Pulse Count Pulse Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.22 Object 2207h: UNI Pulse Count Time Window

This object is a Counter input type in which object **UNI Pulse Count in Time Window** 2207h is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Object Description

Index	2207h
Name	UNI Pulse Counter Pulse Window
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1
Description	Universal Input x Pulse Count Time Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

4.3.23 Object 2208h: UNI Pullup/Down Mode

This object determines how the state read on the input pin corresponds to the logic state, in conjunction with application object 6020h. The options for this object are listed in Table 25, and the controller will adjust the input hardware according to what is specified.

Object Description

Index	2208h
Name	DI Pullup/Down Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Pullup/Down
Access	RW
PDO Mapping	No

Value Range	See Table 25
Default Value	0 (pullup/down disabled)

4.3.24 Object 2209h: UNI Filter Frequency for ADC

This object is used to specify the cutoff filter frequency for the ADC peripheral on the processor. The analog-to-digital converter is used with analog input types: voltage; current; and resistive. It is also used to measure: analog output current feedback; power supply voltage, and processor temperature. The available filters are listed in Table 24.

Object Description

Index	2209h
Name	UNI Filter Frequency for ADC
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	UNI 1 ADC Filter Frequency
Access	RW
PDO Mapping	No
Value Range	See Table 24
Default Value	0 [Input Filter Off]

4.3.25 Object 220Bh: UNI Input Logic Type

This object allows flexibility in the response of the input. Table 13 demonstrates all possible options for this parameter.

Object Description

Index	220Bh
Name	UNI Input Logic Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h

Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Input Logic Type
Access	RW
PDO Mapping	No
Value Range	See Table 13
Default Value	0

4.3.26 Object 220Ch: UNI Error Detection Enable

This object enables error detection and reaction associated with the digital input function block. When disabled, the input will not generate an EMCY code in object 1003h Pre-Defined Error Field, nor will it disable any output controlled by the input should the input go out of range as defined by the objects 7148h AI Span Start and 7149h AI Span End.

Object Description

Index	220CFh
Name	UNI Error Detect Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Error Detect Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

4.3.27 Object 220Dh: UNI Error Clear Hysteresis

This object is used to prevent rapid activation/clearing of an input fault flag and sending of object 1003h to the CANopen® network. Once the input has gone above/below the thresholds that define the valid operating range, it must come back into range minus/plus this value to clear the fault. It is scaled in the physical unit of the FV, i.e. object 2011h applies to this object.

Object Description

Index	220Dh
Name	UNI Error Clear Hysteresis
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Universal Input 1 Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	See Table 20
Default Value	10 [0.1V]

4.3.28 Object 220Eh: UNI Error Reaction Delay

This object is used to filter out spurious signals and to prevent saturating the CANopen ® network with broadcasts of object 1003h as the fault is set/cleared. Before the fault is recognized (i.e. the EMCY code is added to the pre-defined error field list), it must remain active throughout the period of time defined in this object. The physical unit for this object is milliseconds.

Object Description

Index	220Eh
Name	UNI Error Reaction Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h

Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	UNI x Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

4.3.29 Object 2303h: ENC Stationary Delay

This object decides after how many milliseconds the direction is set to stationary (0) if no rising edge or falling edge is detected on signal A or B on the encoder input. This way, the performance of the encoder input can be adjusted for slower and faster rotating encoders.

Object Description

Index	2303h
Name	ENC Stationary Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	ENC x Stationary Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	100 [ms]

4.3.30 Object 2304h: ENC Error Detection Enable

This object enables error detection and reaction associated with the encoder input function block. When disabled, the input will not generate an EMCY code in object 1003h Pre-Defined Error Field, nor will it disable any output controlled by the input should the input go out of range as defined by the objects 7148h AI Span Start and 7149h AI Span End.

Object Description

Index	2304h
Name	ENC Error Detect Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder Input X Error Detect Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

4.3.31 Object 2305h: ENC Error Clear Hysteresis

This object is used to prevent rapid activation/clearing of an input fault flag and sending of object 1003h to the CANopen® network. Once the input has gone above/below the thresholds that define the valid operating range, it must come back into range minus/plus this value to clear the fault. It is scaled in the physical unit of the FV, i.e. object 2011h applies to this object.

Object Description

Index	2305h
Name	ENC Error Clear Hysteresis
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported

Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Universal Input X Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	See Table 20
Default Value	10 [0.1V]

4.3.32 Object 2306h: ENC Error Reaction Delay

This object is used to filter out spurious signals and to prevent saturating the CANopen ® network with broadcasts of object 1003h as the fault is set/cleared. Before the fault is recognized (i.e. the EMCY code is added to the pre-defined error field list), it must remain active throughout the period of time defined in this object. The physical unit for this object is milliseconds.

Object Description

Index	2306h
Name	ENC Error Reaction Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	ENC x Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

4.3.33 Object 2310h: ENC A Active State

The Active High/Low parameter allows the user to select how the controller responds to the behaviour of the digital input.

Object Description

Index	2310h
Name	ENC A Active State
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder Input X Active State
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	0

4.3.34 Object 2311h: ENC A Debounce Time

The Digital Input Debounce Time parameter is a useful parameter in cases where the digital input signal coming into the controller is noisy. Please refer to 2.2.3 for more information.

Object Description

Index	2311h
Name	ENC A Debounce Time
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)

Description	Input X Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 to 65,535
Default Value	250

4.3.35 Object 2312h: ENC A Debounce Filter

This object will debounce the input signal applied on a single digital input. The options for this object are listed in Table 12.

Object Description

Index	2312h
Name	ENC A Debounce Filter
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder A Input X Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 12
Default Value	0 [Debounce Filter OFF]

4.3.36 Object 2313h: ENC A Pullup/Down Mode

This object determines how the state read on the input pin corresponds to the logic state, in conjunction with application object 6020h. The options for this object are listed in Table 25, and the controller will adjust the input hardware according to what is specified.

Object Description

Index	2313h
Name	ENC A Pullup/Down Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported

Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder A Input X Pullup/Down
Access	RW
PDO Mapping	No
Value Range	See Table 25
Default Value	2 (Pulldown Enabled)

4.3.37 Object 2314h: ENC A Input Logic Type

This object allows flexibility in the response of the input. Table 13 demonstrates all possible options for this parameter.

Object Description

Index	2314h
Name	ENC A Input Logic Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Input X Decimal Digits FV
Access	RW
PDO Mapping	No
Value Range	See Table 13
Default Value	0

4.3.38 Object 2320h: ENC B Active State

The Active High/Low parameter allows the user to select how the controller responds to the behaviour of the digital input.

Object Description

Index	2320h
Name	ENC B Active State
Object Type	ARRAY

Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder B Input X Active State
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	0

4.3.39 Object 2321h: ENC B Debounce Time

The Digital Input Debounce Time parameter is a useful parameter in cases where the digital input signal coming into the controller is noisy. Please refer to 2.2.3 for more information.

Object Description

Index	2321h
Name	ENC B Debounce Time
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Input X Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 to 65,535
Default Value	250

4.3.40 Object 2322h: ENC B Debounce Filter

This object will debounce the input signal applied on a single digital input. The options for this object are listed in Table 12.

Object Description

Index	2322h
Name	ENC B Debounce Filter
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Encoder A Input X Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 12
Default Value	0 [Debounce Filter OFF]

4.3.41 Object 2323h: ENC A Pullup/Down Mode

This object determines how the state read on the input pin corresponds to the logic state, in conjunction with application object 6020h. The options for this object are listed in Table 25, and the controller will adjust the input hardware according to what is specified.

Object Description

Index	2323h
Name	ENC A Pullup/Down Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)

Description	Encoder B Input X Pullup/Down
Access	RW
PDO Mapping	No
Value Range	See Table 25
Default Value	2 (Pulldown Enabled)

4.3.42 Object 2324h: ENC B Input Logic Type

This object allows flexibility in the response of the input. Table 13 demonstrates all possible options for this parameter.

Object Description

Index	2324h
Name	DI/UNI/ENC A/ENC B Input Logic Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x = 1 to 3)
Description	Input X Decimal Digits FV
Access	RW
PDO Mapping	No
Value Range	See Table 13
Default Value	0

4.3.43 Object 2500h: EC Extra Received Process Value

This object provides an extra control source in order to allow other function blocks to be controlled by data received from a CANopen ® RPDO. It functions similarly to any other writeable, mappable PV object, such as 7300h AO Output PV.

Object Description

Index	2500h
Name	EC Extra Received Process Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h to 5h (x = 1 to 5)
Description	EC x Received Process Value
Access	RW
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

4.3.44 Object 2502h: EC Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the extra control data, which is interpreted with data type Integer16 in the process value object.

Object Description

Index	2502h
Name	EC Decimal Digits PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h to 5h (x = 1 to 5)
Description	ECx Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	1 (0.1 resolution)

4.3.45 Object 2520h: EC Scaling 1 PV

This object defines the minimum value of the extra control source. It used as the Scaling 1 value by other functions blocks when the EC has been selected as the source for the X-Axis data. There is no physical unit associate with the data, but it uses the same resolution as the received PV as defined in object 2502h, EC Decimal Digits PV. This object must always be smaller than object 2522h EC Scaling 2 PV.

Object Description

Index	2520h
Name	EC Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h to 5h (x = 1 to 5)
Description	ECx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	-32768 to 2522h sub-index X
Default Value	0

4.3.46 Object 2522h: EC Scaling 2 PV

This object defines the maximum value of the extra control source. It used as the Scaling 2 value by other functions blocks when the EC has been selected as the source for the X-Axis dataThere is no physical unit associate with the data, but it uses the same resolution as the received PV as defined in object 2502h, EC Decimal Digits PV. This object must always be larger than object 2520h EC Scaling 1 PV.

Object Description

Index	2522h
Name	EC Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h to 5h (x = 1 to 5)
Description	ECx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	2520h sub-index X to 32767
Default Value	1000 (100.0)

4.3.47 Object 3yz0h: LTyz Input X-Axis Source

This object defines the type of input that will be used to determine the X-Axis input process value for the lookup table function. The available control sources on this controller are listed in Table 29. Not all sources would make sense to use as an X-Axis input, and it is the user's responsibility to select a source that makes sense for the application. A selection of "Control Source Not Used" disables the associated lookup table function block.

Object Description

Index	3yz0h (where yz = 01 to 09)
Name	LTyz Input X-Axis Source
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	See Table 12
Default Value	0 (control not used)

4.3.48 Object 3yz1h: LTyz Input X-Axis Number

This object defines the number of the source that will be used as the X-Axis input PV for the lookup table function. The available control numbers are dependent on the source selected, as shown in Table 29. Once selected, the limits for the points on the X-Axis will be constrained by the scaling objects of the control source/number as defined in Table 31.

Object Description

Index	3yz1h (where yz = 01 to 09)
Name	LTyz Input X-Axis Number
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	See Table 30
Default Value	0 (null control source)

4.3.49 Object 3yz2h: LTyz Auto Repeat

This object determines whether the lookup table sequence will repeat automatically once the last point in the lookup table has been completed. This object is only taken into effect when the response is set to *'Time Response'*. For more details on the functionality of this object and its effect on the lookup table, please refer to section 2.5.4.

Object Description

Index	3yz2h (where yz = 01 to 09)
Name	LTyz X-Axis Decimal Digits PV
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (OFF) to 1 (ON)
Default Value	0 [OFF]

4.3.50 Object 3yz3h: LTyz X-Axis Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the X-Axis input data and the points in the lookup table. It should be set equal to the decimal digits used by the PV from the control source/number as defined in Table 31.

Object Description

Index	3yz3h (where yz = 01 to 09)
Name	LTyz X-Axis Decimal Digits PV
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 4 (see Table 31)
Default Value	0

4.3.51 Object 3yz4h: LTyz Y-Axis Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the Y-Axis points in the lookup table. When the Y-Axis output is going to be the input to another function block (i.e. an analog output), it is recommended that this value be set equal to the decimal digits used by the block that is using the lookup table as the control source/number.

Object Description

Index	3yz4h (where yz = 01 to 09)
Name	LTyz Y-Axis Decimal Digits PV
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	0

4.3.52 Object 3yz5h: LTyz Point Response

This object determines the Y-Axis output response to changes in the X-Axis input. The value set in sub-index 1 determines the X-Axis type (i.e. data or time), while all other sub-indexes determine the response (ramp, step, ignore) between two points on the curve. The options for this object are listed in Table 32. See Figure 9 for an example of the difference between a step and ramp response.

Object Description

Index	3yz5h (where yz = 01 to 12)

Name	LTyz Point Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	11
Default Value	11

Sub-Index	1h
Description	X-Axis Type
Access	RW
PDO Mapping	No
Value Range	See Table 23 (0 or 1)
Default Value	0 (x-axis data response)

Sub-Index	2h to Bh (x = 2 to 11)
Description	LTyz Point X Response
Access	RW
PDO Mapping	No
Value Range	See Table 32 (0, 1 or 2)
Default Value	1 (ramp to response)

4.3.53 Object 3yz6h: LTyz Point X-Axis PV

This object defines the X-Axis data for the 11 calibration points on the lookup table, resulting in 10 different output slopes.

When a data response is selected for the X-Axis type (sub-index 1 of object 3yz5), this object is constrained such that X1 cannot be less than the Scaling 1 value of the selected control source/number, and X11 cannot be more than the Scaling 2 value. The rest of the points are constrained by the formula below. The physical unit associate with the data will be that of the selected input, and it will use the resolution defined in object 3yz3h, LTz X-Axis Decimal Digits PV.

$$MinInt16 <= X_1 <= X_2 <= X_3 <= X_4 <= X_5 <= X_6 <= X_7 <= X_8 <= X_9 <= X_{10} <= X_{11} <= MaxInt16$$

When a time response has been selected, each point on the X-Axis can be set anywhere from 1 to 86,400,000ms.

Object Description

Index	3yz6h (where yz = 01 to 12)
Name	LTyz Point X-Axis PV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	11
Default Value	11

Sub-Index	1h to Ah (x = 1 to 10)	
Description	LTyz Point X-Axis PVx	
Access	RW	
PDO Mapping	No	
Value Range	See above (data)	1 to 86400000 (time)
Default Value	10*(x-1)	No

4.3.54 Object 3yz7h: LTyz Point Y-Axis PV

This object defines the Y-Axis data for the 11 calibration points on the lookup table, resulting in 10 different output slopes. The data is unconstrained and has no physical unit associate with it. It will use the resolution defined in object 3yz4h, LTyz Y-Axis Decimal Digits PV.

Object Description

Index	3yz7h (where yz = 01 to 12)
Name	LTz Point Y-Axis PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	11
Default Value	11

Sub-Index	1h to Ah (x = 1 to 10)

Description	LTyz Point Y-Axis PVx
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	10*(x-1) [i.e. 0, 10, 20, 30, … 100]

4.3.55 Object 3yz8h: LTyz Output Y-Axis PV

This read-only object contains the lookup table function block PV that can be used as the input source for another function block (i.e. analog output.) The physical unit for this object is undefined, and it will use the resolution defined in object 3yz4h, LTz Y-Axis Decimal Digits PV.

Object Description

Index	3yz8h (where yz = 01 to 09)
Name	LTyz Output Y-Axis PV
Object Type	VARIABLE
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

4.3.56 Object 3300h: Logic Block Enable

This object defines whether or not the logic shown in Figure 12 will be evaluated.

Object Description

Index	3300h
Name	Logic Block Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4

Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	LBx Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

4.3.57 Object 3310h: Logic Block Selected Table

This read-only object reflects what table has been selected as the output source for the logic block after the evaluation shown in Figure 12 has been performed.

Object Description

Index	3310h
Name	Logic Block Selected Table
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	LBx Selected Table
Access	RO
PDO Mapping	Yes
Value Range	1 to 12
Default Value	No

4.3.58 Object 3320h: Logic Block Output PV

This read-only object reflects the output from the selected table, interpreted as a percentage. The limits for the percentage conversion are based on the range of the lookup tables Y-Axis Output PV. This value has a fixed decimal digit value of 1 giving a resolution of 0.1%.

Object Description

Index	3320h
Name	Logic Block Output PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	LBx Output PV
Access	RO
PDO Mapping	Yes
Value Range	Dependent on Selected Table
Default Value	No

4.3.59 Object 3x01h: LB(3-x) Lookup Table Numbers

This object determines which of the six lookup tables supports the controller on the associated with a particular function within the given logic block. Up to three tables can be linked to each logic function.

Object Description

Index	3x01h (where x = 4 to 5)
Name	LB(3-x) Lookup Table Numbers
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h to 4h (y = A to C)
Description	LB(3-x) Lookup Table Y Number

Access	RW
PDO Mapping	No
Value Range	1 to 12
Default Value	See Table 18

4.3.60 Object 3x02h: LB(3-x) Function Logical Operator

This object determines how the results of the three conditions for each function are to be compared to one another to determine the overall state of the function output. There are up to three functions that can be evaluated in each logic block. The options for this object are defined in Table 34. See Section 2.6 for more information about how this object is used.

Object Description

Index	3x02h (where x = 4 to 5)
Name	LB(3-x) Function Logical Operator
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h to 4h (y = A to C)
Description	LB(3-x) Function Y Logical Operator
Access	RW
PDO Mapping	No
Value Range	See Table 34
Default Value	Function A = 1 (and all) Function B = 1 (and all) Function C = 0 (default)

4.3.61 Object 3x11h: LB(3-x) Function A Condition 1

4.3.62 Object 3x12h: LB(3-x) Function A Condition 2

4.3.63 Object 3x13h: LB(3-x) Function A Condition 3

4.3.64 Object 3x21h: LB(3-x) Function B Condition 1

4.3.65 Object 3x22h: LB(3-x) Function B Condition 2

4.3.66 Object 3x23h: LB(3-x) Function B Condition 3

4.3.67 Object 3x31h: LB(3-x) Function C Condition 1

4.3.68 Object 3x32h: LB(3-x) Function C Condition 2

4.3.69 Object 3x33h: LB(3-x) Function C Condition 3

These objects, 3xyzh, represent Logic Block z, Function y, Condition z, where x = 4 to 7, y = 1 (A) to 3 (C), and z = 1 to 3. All of these objects are a special type of record, defined in Table 16. Information on how to use these objects is defined in Section 2.6.

Object Description

Index	3xyzh
Name	LB(3-x) Function y Condition z
Object Type	RECORD
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	Argument 1 Source
Access	RW
PDO Mapping	No
Value Range	See Table 29
Default Value	1 (CANopen Message)

Sub-Index	2h
Description	Argument 1 Number
Access	RW
PDO Mapping	No

Value Range	See Table 30
Default Value	11 (EC Received PV 1)

Sub-Index	3h
Description	Argument 2 Source
Access	RW
PDO Mapping	No
Value Range	See Table 29
Default Value	5 (Constant PV)

Sub-Index	4h
Description	Argument 2 Number
Access	RW
PDO Mapping	No
Value Range	See Table 30
Default Value	3 (Constant FV 3)

Sub-Index	5h
Description	Operator
Access	RW
PDO Mapping	No
Value Range	See Table 36
Default Value	0 (Equals)

4.3.70 Object 4000h: Math Function Enable

The corresponding sub-index of object must be set TRUE in order for a math function block to be enabled. Otherwise, the output will always be at 0.

Object Description

Index	4000h
Name	Math Function Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1h to 6h (Y = 1 to 6)
Description	Math Y Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

4.3.71 Object 4021h: Math Output Scaling 1 PV

This object defines the process value that would correspond to 0% output from the math calculation. The object would apply the resolution defined in object 4532h Math Output Decimal Digits PV. The physical unit is undefined.

Object Description

Index	4021h
Name	Math Output Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (Y = 1 to 6)
Description	Math Y Output Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	-32768 to 32767
Default Value	0

4.3.72 Object 4023h: Math Output Scaling 2 PV

This object defines the process value that would correspond to 100% output from the math calculation. The object would apply the resolution defined in object 4532h Math Output Decimal Digits PV. The physical unit is undefined.

Object Description

Index	4023h
Name	Math Output Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (Y = 1 to 6)
Description	Math Y Output Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	-32768 to 32767
Default Value	10000 (100.00)

4.3.73 Object 4030h: Math Output Process Value

This read-only object reflects the output from the math function block after it has been scaled by objects 4021h and 4023h. The object would apply the resolution defined in object 4032h Math Output Decimal Digits PV. The physical unit is undefined.

Object Description

Index	4030h
Name	Math Output Process Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (Y = 1 to 6)
Description	Math Y Output Process Value
Access	RO
PDO Mapping	Yes
Value Range	-32768 to 32767
Default Value	No

4.3.74 Object 4032h: Math Output Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the output
data, which is interpreted with data type Integer16 in the process value object.

Object Description

Index	4032h
Name	Math Output Decimal Digits PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (Y = 1 to 6)
Description	Math Y Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	2 (0.01)

4.3.75 Object 4y00h: Math Y Input Source

This object defines the input sources that will be used in the mathematical calculations. Here, y = 1 to 6 – representing Math Block 1 to Math Block 6. If a control source is not used, the associate mathematical calculation would be ignored. The available control sources on the controller are listed in Table 29.

Object Description

Index	4y00h (y = 1 to 6)
Name	Math Y Input Source

Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Source
Access	RW
PDO Mapping	No
Value Range	See Table 29
Default Value	0 (control source not used)

4.3.76 Object 4y01h: Math Y Input Number

This object defines the number of the input source that will be used in the math calculation. The available control numbers are dependent on the source selected, as shown in Table 30. Once selected, the input value will be used in the corresponding calculation as described in Section 2.7.

Object Description

Index	4y01h (y = 1 to 6)
Name	Math Y Input Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Number
Access	RW
PDO Mapping	No

Value Range	See Table 30
Default Value	0 (null input)

4.3.77 Object 4y02h: Math Y Input Function Number

This object defines the number of the function within the Math Block will be used in the math calculation. This object is applicable when the Input Source together with the Input Number match the Math Block that is being configured. If Input Source and Input Number match the Math Block being configured and the Function Number is 0, this object is ignored. For more details, refer to Section 2.7.

Object Description

Index	4y01h (y = 1 to 6)
Name	Math Y Input Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Function Number
Access	RW
PDO Mapping	No
Value Range	See Table 16
Default Value	0 (null input)

4.3.78 Object 4y03h: Math Y Input Decimal Digits FV

This object describes the number of digits following the decimal point (i.e. resolution) of the input data, which is interpreted with data type Integer16 in the field value object.

Object Description

Index	4y03h (y = 1 to 6)
Name	Math Y Input Decimal Digits FV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	2 (0.01)

4.3.79 Object 4y20h: Math Y Input Scaling 1 FV

This object defines the input field value that would correspond to 0% when scaling the input for use in the math calculation. All inputs are normalized to a percentage before being used by the math function block. The object would apply the resolution defined in object 4y03h Math Y Input Decimal Digits FV. The physical unit would match that of the input source.

Object Description

Index	4y20h (y = 1 to 6)
Name	Math Y Input Scaling 1 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

4.3.80 Object 4y22h: Math Y Input Scaling 2 FV

This object defines the input field value that would correspond to 100% when scaling the input for use in the math calculation. All inputs are normalized to a percentage before being used by the math function block. The object would apply the resolution defined in object 4y03h Math Y Input Decimal Digits FV. The physical unit would match that of the input source.

Object Description

Index	4y22h (y = 1 to 6)
Name	Math Y Input Scaling 2 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	10000 (100.00%)

4.3.81 Object 4y40h: Math Y Input Gain

This object can be used to adjust the ‘weight’ of the input in the math calculation. It is a multiplier of the input after it has been converted into a percentage, before it is used in the math calculation. This object has a fixed resolution of 2 decimal digits.

Object Description

Index	4y40h (y = 1 to 6)
Name	Math Y Input Gain
Object Type	ARRAY
Data Type	INTEGER8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h to 6h (X = 1 to 6)
Description	Math Y Input X Gain
Access	RW
PDO Mapping	No
Value Range	-100 to 100
Default Value	100 (1.0)

4.3.82 Object 4y50h: Math Y Operator

This object defines the actual operators that will be used in each stage of a math calculation, as described in Section 2.7. The options for this object are listed in Table 39.

Object Description

Index	4y50h (y = 1 to 6)
Name	Math Y Operator
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (X = 1 to 3)
Description	Math Y Function X Operator
Access	RW
PDO Mapping	No
Value Range	See Table 30
Default Value	12 (Plus)

4.3.83 Object 5010h: Constant Field Value

This object is provided to allow the user to compare against a fixed value, i.e. for setpoint control in a PID loop, or in a conditional evaluation for a logic block. The first two values in this object are fixed at FALSE (0) and TRUE (1). There are ten other sub-indexes provide for other unconstrained data.

Object Description

Index	5010h
Name	Constant Field Value
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

Sub-Index	1
Description	Constant False
Access	RO
PDO Mapping	No
Value Range	0
Default Value	0 (false)

Sub-Index	2
Description	Constant True
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1 (true)

Sub-Index	3
Description	Constant FV 3
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	10.0

Sub-Index	4
Description	Constant FV 4
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	20.0

Sub-Index	5
Description	Constant FV 5
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	30.0

Sub-Index	6
Description	Constant FV 6
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	40.0

Sub-Index	7
Description	Constant FV 7
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	50.0

Sub-Index	8
Description	Constant FV 8
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	60.0

Sub-Index	9
Description	Constant FV 9
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	70.0

Sub-Index	10
Description	Constant FV 10
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	80.00

4.3.84 Object 5540h: Bootup Message Enabled

If this object is enabled, the controller sends a bootup message via CANopen when the controller is power cycled.

Object Description

Index	5540h
Name	Bootup Message Enabled
Object Type	VARIABLE
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [TRUE]

4.3.85 Object 5555h: Start in Operation Mode

If this object is enabled, the controller will start in operational mode and does not have to be set into this mode manually or by any other unit connected to the CANopen Bus.

Object Description

Index	5555h
Name	Bootup Message Enabled
Object Type	VARIABLE
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)

Default Value	0 [TRUE]

4.3.86 Object 5550h: Enable Automatic Updates

This object allows the controller to update objects to defaults automatically when an output type is changed. Be default this object is TRUE.

Object Description

Index	5550h
Name	Enable Auto Updates
Object Type	VARIABLE
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	1 [TRUE]

4.4 Application Objects

Index [hex]	Object	Object Type	Data Type	Access	PDO Mapping
6000	DI Read State 8 Input Lines	ARRAY	BOOLEAN	RO	Yes
6002	DI Polarity 8 Input Lines	ARRAY	UNSIGNED8	RW	No
7100	AI Input Field Value	ARRAY	INTEGER16	RO	Yes
6110	AI Sensor Type	ARRAY	UNSIGNED16	RW	No
6112	AI Operating Mode	ARRAY	UNSIGNED8	RW	No
7120	AI Input Scaling 1 FV	ARRAY	INTEGER16	RW	No
7121	AI Input Scaling 1 PV	ARRAY	INTEGER16	RW	No
7122	AI Input Scaling 2 FV	ARRAY	INTEGER16	RW	No
7123	AI Input Scaling 2 PV	ARRAY	INTEGER16	RW	No
7130	AI Input Process Value	ARRAY	INTEGER16	RO	Yes
6132	AI Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
7148	AI Input Span Start	ARRAY	INTEGER16	RW	No
7149	AI Input Span End	ARRAY	INTEGER16	RW	No
7124	AI Input Offset	ARRAY	INTEGER16	RW	No
6125	AI Autozero	ARRAY	UNSIGNED32	WO	No
61A0	AI Filter Type	ARRAY	UNSIGNED8	RW	No
61A1	AI Filter Constant	ARRAY	UNSIGNED16	RW	No

4.4.1 Object 6000h: DI Read State 8 Input Lines

This read-only object shall read group of 8 input lines as 8-bit information. Refer to Section 2.2.6 for more information

Object Description

Index	6000h
Name	DI Read State 8 Input Line
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1h
Description	Digital Input State Bitmap D1-D8
Access	RO
PDO Mapping	Yes
Value Range	0 (OFF) or 1 (ON)
Default Value	0

Sub-Index	2h
Description	Digital Input State Bitmap D9-D13
Access	RO
PDO Mapping	Yes
Value Range	0 (OFF) or 1 (ON)
Default Value	0

4.4.2 Object 6002h: DI Polarity 8 Input Lines

This object shall define the polarity of a group of 8 input lines. This object determines how the state read on the input pin corresponds to the logic state.

Object Description

Index	6002h
Name	DI Polarity 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No

Value Range	2
Default Value	2

Sub-Index	1h
Description	Digital Input Polarity Bitmap D1-D8
Access	RW
Section PDO Mapping	No
Value Range	See Table 15
Default Value	0 (Normal On/Off)

Sub-Index	2h
Description	Digital Input Polarity Bitmap D9-D13
Access	RW
PDO Mapping	No
Value Range	See Table 15
Default Value	0 (Normal On/Off)

4.4.3 Object 7100h: AI Input Field Value

This object represents the measured value of an analog input that has been scaled as per manufacturer object 2102h AI Decimal Digits PV. The base unit for each type of input is defined inTable 7, Table 18, and Table 26, as well as the read-only resolution (decimal digits) associated with the FV.

Object Description

Index	7100h
Name	AI Input Field Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	Analog Input X FV
Access	RO
PDO Mapping	Yes
Value Range	Data Type Specific
Default Value	No

4.4.4 Object 6110h: AI Sensor Type

This object defines the type of sensor (input) which is connected to the input pin.

Object Description

Index	6110h
Name	AI Sensor Type
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Ah (x = 1 to 13)
Description	AIx Sensor Type
Access	RW
PDO Mapping	No
Value Range	See Table 3, Table 10, and Table 21
Default Value	Encoder Input 1-3: 1002(Encoder) Universal Input: 40(Voltage) Digital Input 1,2,4,5: 1000(PWM) Digital Input 3 and 6: 1003(Digital ON/OFF)

4.4.5 Object 6112h: AI Operating Mode

This object enables special operating modes for the input.

Object Description

Index	6112h
Name	AI Operating Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Operating Mode
Access	RW
PDO Mapping	No
Value Range	See Table 5
Default Value	1 (normal operation)

4.4.6 Object 7120h: AI Input Scaling 1 FV

This object describes the field value of the first calibration point for the analog input channel. It also defines the "minimum" value of the analog input range when using this input as a control source for another function block. It is scaled in the physical unit of the FV, i.e. object 2011h applies to this object.

Object Description

Index	7120h
Name	AI Input Scaling 1 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	See Table 20 and Table 27
Default Value	Encoder Input 1-3: 0 Universal Input: 500 Digital Input 1,2,4,5: 1 Digital Input 3 and 6: 1

4.4.7 Object 7121h: AI Input Scaling 1 PV

This object defines the process value of the first calibration point for the analog input channel, as shown in Figure 7. It is scaled in the physical unit of the PV, i.e. object 6132h applies to this object.

Object Description

Index	7121h
Name	AI Input Scaling 1 PV
Object Type	ARRAY

Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 or 13)
Description	AIx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	same as 7120h

4.4.8 Object 7122h: AI Input Scaling 2 FV

This object describes the field value of the second calibration point for the analog input channel, as shown in Figure 7. It also defines the “maximum” value of the analog input range when using this input as a control source for another function block, as described in Table 14 in Section 2.4. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

Index	7122h
Name	AI Input Scaling 2 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 or 13)
Description	AIx Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	See Table 31
Default Value	Encoder Input 1-3: 2 Universal Input: 4500 Digital Input 1,2,4,5: 9000 Digital Input 3 and 6: 1

4.4.9 Object 7123h: AI Input Scaling 2 PV

This object defines the process value of the second calibration point for the analog input channel, as shown in Figure 7. It is scaled in the physical unit of the PV, i.e. object 6132h applies to this object.

Object Description

Index	7123h
Name	AI Input Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	same as 7122h

4.4.10 Object 7130h: AI Input Process Value

This object represents the result of the input scaling applied, and gives the measured quantity scaled in the physical unit of the process value (i.e. °C, PSI, RPM, etc) with the resolution defined in object 6132h AI Decimal Digits PV.

Object Description

Index	7130h
Name	AI Input Process Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Process Value
Access	RO
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

4.4.11 Object 6132h: AI Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the input data, which is interpreted with data type Integer16 in the process value object.

Example: A process value of 1.230 (Float) will be coded as 1230 in Integer16 format if the number of decimal digits is set to 3.

Object Description

Index	6123h
Name	AI Decimal Digits PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	Encoder Input 1-3: 0 Universal Input: 1 Digital Input 1,2,4,5: 0 Digital Input 3 and 6: 0

4.4.12 Object 7148h: AI Span Start

This value specifies the lower limit where field values are expected. Field values, which are lower than this limit, are marked as negative overload. It is scaled in the physical unit of the FV.

Object Description

Index	7148h
Name	AI Span Start
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Ah (x = 1 to 10)
Description	AIx Span Start (Error Min)
Access	RW
PDO Mapping	No
Value Range	See Table 20
Default Value	Encoder Input 1-3: 0 Universal Input: 200 Digital Input 1,2,4,5: 10 Digital Input 3 and 6: 0

4.4.13 Object 7149h: AI Span End

This value specifies the upper limit where field values are expected. Field values, which are higher than this limit, are marked as positive overload. It is scaled in the physical unit of the FV.

Object Description

Index	7149h
Name	AI Span End
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Span End (Error Max)
Access	RW
PDO Mapping	No

Value Range	See Table 20
Default Value	Encoder Input 1-3: 2 Universal Input: 4800 Digital Input 1,2,4,5: 10000 Digital Input 3 and 6: 1

4.4.14 Object 7124: AI Input Offset

Thisobject defines an offset for the input channel, which added to the input value after scaling the input field value to process value. The value is scaled in the physical unit of the input process value, object 7130h.

Object Description

Index	7124h
Name	AI Input Offset
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Input Offset
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

4.4.15 Object 6125: AI Autozero

Writing a signature 'zero' to this object causes a modification of object 7124h value, AI Input Offset, such that object 7130h (AI Input PV) becomes zero. This zeroing cycling is performed once, upon reception of the signature to the appropriate subindex, but the zeroing procedure can be performed at any time, as many times as required.

The signature is a 32-bit unsigned number, composed of the ASCII codes of the signature characters, according to the following table:

MSB			LSB
0	r	e	z
6Fh	72h	65h	7Ah

Upon reception of the correct signature to an appropriate subindex, the controller will perform the zeroing operation for that input, and then confirm the SDO transmission.

Object Description

Index	6125h
Name	AI Autozero
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Autozero
Access	WO
PDO Mapping	No
Value Range	0x6F72657A
Default Value	No

4.4.16 Object 61A0h: AI Filter Type

This object defines the type of data filter that will be applied to the raw input data, as read from the ADC or Timer, before it is passed to the field value object. The types of data filters are defined in Table 17, and how they are used is outlined in Section 2.2.

Object Description

Index	61A0h
Name	AI Filter Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Filter Type

Access	RW
PDO Mapping	No
Value Range	See Table 17
Default Value	0 (no filter)

4.4.17 Object 61A1h: AI Filter Constant

This object defines the number of steps used in the various filters, as defined in 2.2.9

Object Description

Index	61A0h
Name	AI Filter Constant
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	13
Default Value	13

Sub-Index	1h to Dh (x = 1 to 13)
Description	AIx Filter Constant
Access	RW
PDO Mapping	No
Value Range	1 to 1000
Default Value	10

5 TECHNICAL SPECIFICATIONS

Typical at nominal input voltage and 25 degrees C unless otherwise specified.

5.1 Power

Power Supply Input - Nominal	12, 24 or 48Vdc nominal operating voltage 8…60 Vdc power supply range for voltage transients
Surge Protection	Provided
Reverse Polarity Protection	Provided
Quiescent Current	55mA @ 12V; 28mA @ 24V Typical

5.2 Signal Input Specifications

<table>
<tr><td>Encoder Inputs</td><td>Three 2-phase, phase A and B incremental encoder inputs
Range: 0.5 to 60 kHz
Amplitude: up to +Vps
1 MOhm impedance or Active High with 10K Pullup or Active Low with 10K Pulldown resistor to GND</td></tr>
<tr><td>Universal Input</td><td>1 user selectable input
Analog 12-bit (0-5V, 0-10V, 0-20 mA, 4-20 mA)
PWM 12-bit (low or high frequency) – auto detect 0.5 to 50 kHz, 0-100%
Frequency/RPM – auto detect 0.5 to 50 kHz, 0-100%
Counter input 16-bit
Digital (active high/active low) [ON when input ≥ 1.5V]
The “Input Sensor Type” setpoint is used to configure input type.

All inputs with the exception of 16-Bit Counter are sampled every 1ms.
Analog Input types have a 12-bit resolution.

With current inputs, short circuit protection is provided.</td></tr>
<tr><td>Digital Inputs 1-6</td><td>4 user selectable inputs
PWM 12-bit (low or high frequency)
Frequency/RPM auto detect 0.5 to 50 kHz, 0-100%
Digital (active high with 10K pullup) [ON when input ≥ 1.5V]
2 digital inputs (inputs 3 and 6)
Digital (active high with 10K pullup) [ON when input ≥ 1.5V]</td></tr>
<tr><td>Minimum and Maximum Ratings</td><td>
<table>
<tr><th colspan="4">Table 1.0. Absolute Maximum and Minimum Ratings</th></tr>
<tr><th>Characteristic</th><th>Min</th><th>Max</th><th>Units</th></tr>
<tr><td>Power Supply</td><td>8</td><td>60</td><td>V dc</td></tr>
<tr><td>Voltage Input</td><td>0</td><td>43</td><td>V dc</td></tr>
<tr><td>Current Input</td><td>0</td><td>21</td><td>mA</td></tr>
<tr><td>Current Input – Voltage Level</td><td>0</td><td>12</td><td>Vdc</td></tr>
<tr><td>Digital Type Input – Voltage Level</td><td>0</td><td>43</td><td>Vdc</td></tr>
<tr><td>PWM Duty Cycle</td><td>0</td><td>100</td><td>%</td></tr>
<tr><td>PWM Frequency</td><td>50</td><td>20 000</td><td>Hz</td></tr>
<tr><td>PWM Voltage pk - pk</td><td>0</td><td>43</td><td>V dc</td></tr>
<tr><td>RPM Frequency</td><td>50</td><td>20 000</td><td>Hz</td></tr>
</table>
</td></tr>
<tr><td>Input Accuracy</td><td>
<table>
<tr><th colspan="3">Table 2.0. Input Accuracy</th></tr>
<tr><th>Input Type</th><th>Accuracy</th><th>Resolution</th></tr>
<tr><td>Voltage</td><td>+/- 1%</td><td>1 [mV]</td></tr>
<tr><td>Current</td><td>+/- 1%</td><td>1 [uA]</td></tr>
<tr><td>PWM</td><td>+/- 1% (<5kHz)
+/- 2% (>5kHz)</td><td>0.1 [%]</td></tr>
<tr><td>Frequency/RPM</td><td>+/- 1%</td><td>0.01 [Hz]</td></tr>
</table>
</td></tr>
<tr><td>Input Impedance</td><td>0-5V: 1 MOhm
0-10V: 170 kOhm
0(4)-20mA: 249 Ohm
Frequency/Digital Input: Pull Up/Pull Down 1 MOhm</td></tr>
<tr><td>Scan Rate</td><td>Each input is scanned in 100uS.
A complete scan of 10 inputs occurs with new measured values every 1mS.</td></tr>
<tr><td>Analog GND</td><td>1 Analog GND connections is provided.</td></tr>
</table>

5.3 General Specifications

Microprocessor	STM32F405RG
Communications	1 CAN port (CANopen®) A SAE J1939 model is available, ordering part number AX030140. Auto baud rate detection allows for high speed SAE J1939 CAN bus connections.
EMC Compliance	CE marking
Vibration	MIL-STD-202G, Method 204D, test condition A – 10 g peak (Sine) MIL-STD-202G, Method 214A, test condition B – 7.68 Grms (Random)
Shock	MIL-STD-202G, Method 213B, test condition A 50 g half sine pulse, 6 ms, 6 pulses per axis
User Interface	User configuration and diagnostics are provided by any CANopen service tool (not supplied).
Network Termination	It is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.
Electrical Connections	TE Deutsch DTM series 24 pin receptacle (DTM13-12PA-12PB-R008) Mating plug: TE Deutsch DTM06-12SA and DTM06-12SB with 2 wedgelocks (WM12S) and 24 contacts (0462-201-20141). 20 AWG wire is recommended for use with contacts 0462-201-20141. Refer to Table 3.0 for pinout.
Enclosure and Dimensions	High Temperature Nylon housing – TE Deutsch PCB Enclosure (EEC-325X4B) 4.63 x 5.25 x 1.41 inches 117.60 x 133.50 x 35.90 mm (W x L x H excluding mating plugs)
Operating Conditions	-40 to 85°C (-40 to 185°F)
Weight	0.55 lb. (0.25kg)
Protection	IP67, Unit is conformal coated in the housing.
Mounting	Mounting holes sized for ¼ inch or M6 bolts. The bolt length will be determined by the end-user's mounting plate thickness. The mounting flange of the controller is 0.63 inches (16 mm) thick. If the module is mounted without an enclosure, it should be mounted vertically with connectors facing left and right to reduce likelihood of moisture entry. The CAN wiring is considered intrinsically safe. The power wires are not considered intrinsically safe and so in hazardous locations, they need to be located in conduit or conduit trays at all times. The module must be mounted in an enclosure in hazardous locations for this purpose. All field wiring should be suitable for the operating temperature range. Install the unit with appropriate space available for servicing and for adequate wire harness access (6 inches or 15 cm) and strain relief (12 inches or 30 cm).

5.4 Dimensions and Typical Connections

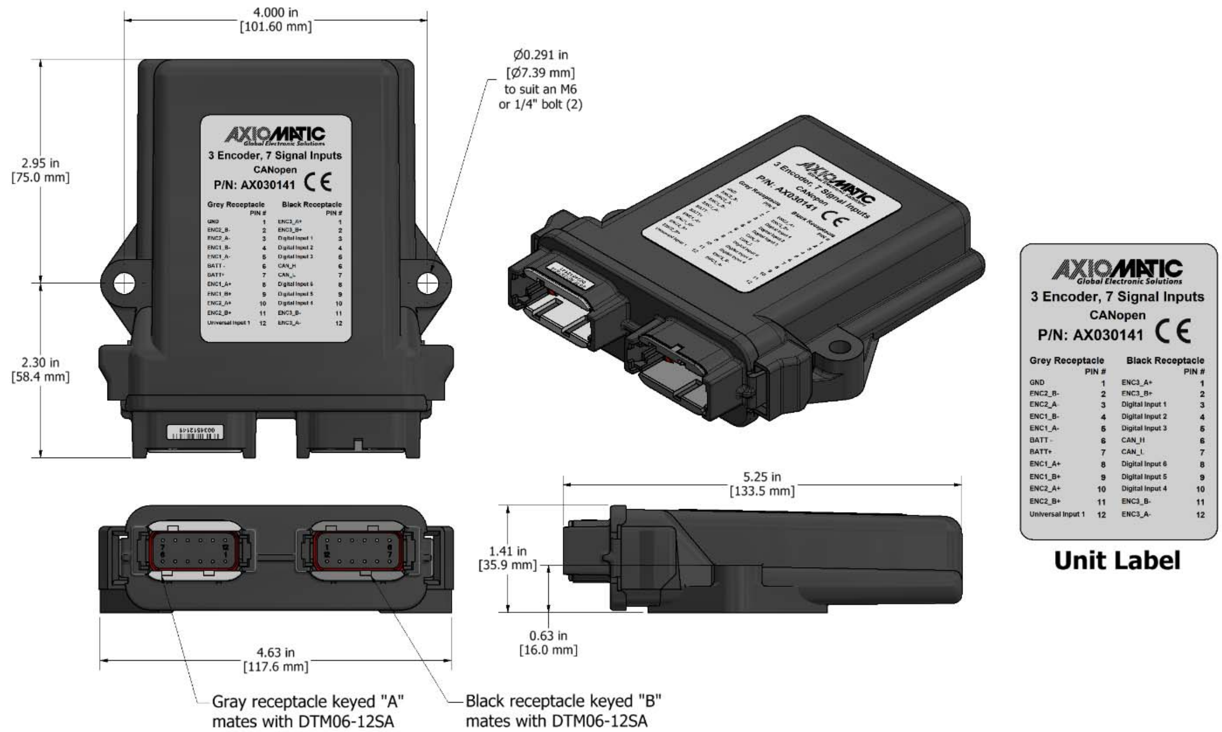


Table 3.0. Electrical Pin Out

Grey Connector		Black Connector	
Pin #	**Function**	**Pin #**	**Function**
1	Analog GND	1	ENC3_A+
2	ENC2_B-	2	ENC3_B+
3	ENC2_A-	3	Digital Input 1
4	ENC1_B-	4	Digital Input 2
5	ENC1_A-	5	Digital Input 3 (Digital only)
6	Batt -	6	CAN_H
7	Batt +	7	CAN_L
8	ENC1_A+	8	Digital Input 6 (Digital only)
9	ENC1_B+	9	Digital Input 5
10	ENC2_A+	10	Digital Input 4
11	ENC2_B+	11	ENC3_B-
12	Universal Input	12	ENC3_A-

Notes:
CANopen® is a registered community trademark of CAN in Automation e.V.

OUR PRODUCTS

Actuator Controls

Automotive Ethernet Converters

Battery Chargers

CAN bus Controls

CAN/Wifi, CAN/Bluetooth

Current/Voltage Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Switches

Fan Drive Controllers

Gateways, CAN/Modbus Protocols

Gyroscope Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Simulators

Machine Controls

Motor Controls

Power Supplies

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

Axiomatic provides electronic machine controls, components, and systems to the off-highway, commercial vehicle, electric vehicle, power generator set, material handling, renewable energy and industrial OEM markets.

We provide efficient, innovative solutions that focus on adding value for our customers.

We innovate with engineered and off-the-shelf machine controls.

We emphasize service and partnership with our customers, suppliers, and employees to build long term relationships and mutual trust.

QUALITY DESIGN AND MANUFACTURING

Axiomatic in Canada operates an ISO 9001:2015 registered design and manufacturing facility.

SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#). Please request an RMA# from rma@axiomatic.com.

Please provide the following information when requesting an RMA number:

- Serial number, part number
- Axiomatic invoice number and date
- Hours of operation, description of problem
- Wiring set up diagram, application
- Other comments as needed

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

Axiomatic Technologies Corporation reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on www.axiomatic.com/service.html.

CONTACTS

Axiomatic Technologies Corporation
5915 Wallace Street
Mississauga, ON
CANADA L4Z 1Z8
TEL: +1 905 602 9270
FAX: +1 905 602 9279
www.axiomatic.com

Axiomatic Technologies Oy
Höytämöntie 6
33880 Lempäälä
FINLAND
TEL: +358 103 375 750
FAX: +358 3 3595 660
www.axiomatic.fi