User Manual UMAX100311

Universal 25A DC Motor Controller, CANopen®

P/N: AX100311

Axiomatic Technologies Oy
Höytämöntie 6
33880 LEMPÄÄLÄ, Finland
Tel. +358 103 375 750
salesfinland@axiomatic.com
www.axiomatic.fi

Axiomatic Technologies Corporation
1445 Courtneypark Dr. E.
Mississauga, ON Canada L5T 2E3
Tel. 1 905 602 9270
sales@axiomatic.com
www.axiomatic.com

VERSION HISTORY

Version	Date	Author	Modification
1.0.0	July 12, 2020	Erik Sasse	Initial Draft
1.0.1	October 28, 2020	Erik Sasse	Ramp Up and Ramp Down feature in section 2.1.1
-	November 22, 2021	Amanda Wilkins	Marketing Review. Added AX100311 drawing.
1.0.2	September 6, 2023	Kiril Mojsov	Performed Legacy Updates

ACRONYMS

CAN	Controller Area Network
DM	Diagnostic message. Defined in J1939/73 standard
EA	The Axiomatic Electronic Assistant. The EA is a 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®	A 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 ... 1
2 CONTROLLER DESCRIPTION ... 2
2.1 Motor Drive Function Block ... 2
2.1.1 Motor Type ... 2
2.1.2 Motor Drive Control/Enable/Override/ Sources ... 3
2.1.3 Motor Drive Enable ... 4
2.1.4 Motor Drive Override ... 5
2.1.5 Output Field Values and Scaling ... 5
2.1.6 Output Process Values and Scaling ... 5
2.1.7 Automatic Update of Output Type Ranges ... 6
2.1.8 Fault Detection ... 6
2.2 Encoder Input Function Block ... 7
2.2.1 Encoder Operating Mode ... 8
2.2.2 Encoder Stationary Delay ... 9
2.2.3 Input Logic Type ... 9
2.2.4 Active High/Low ... 10
2.2.5 Debounce Time ... 10
2.2.6 Input Software Filter Types ... 11
2.2.7 Input States ... 11
2.2.8 Frequency/RPM Input ... 12
2.2.9 Counter Input Type ... 12
2.2.10 Debounce Filter ... 12
2.2.11 Minimum and Maximum Ranges/Errors ... 13
2.2.12 Input Field Values and Scaling ... 13
2.2.13 Input Process Values and Scaling ... 14
2.2.14 Automatic Update of Input Type Ranges ... 15
2.3 Universal Input Function Block ... 15
2.3.1 Pull Up/Down Options ... 16
2.4 Internal Function Block Control Sources ... 17
2.5 PID Control Function Block ... 20
2.6 Lookup Table Function Block ... 21
2.6.1 X-Axis, Input Data Response ... 21
2.6.2 Y-Axis, Lookup Table Output ... 22
2.6.3 Point To Point Response ... 22
2.6.4 X-Axis, Time Response ... 24
2.7 Programmable Logic Function Block ... 25
2.7.1 Conditions Evaluation ... 27
2.7.2 Table Selection ... 28
2.7.3 Logic Block Output ... 29
2.8 Math Function Block ... 30
2.9 Simple Conditional Function Block ... 32
2.10 LED Control Function Block ... 33
2.10.1 LED Control Sources ... 33
2.10.2 LED Output/Response Type ... 34
3 Installation Instructions ... 35
3.1 Node ID and Baud rate ... 35
3.1.1 LSS Protocol to Update ... 35

3.1.2 Setting Node-ID 35
3.1.3 Setting Baud Rate 36
3.2 Communication Objects (DS-301) 39
3.2.1 PDO Parameters 40
3.2.2 Object 1000h: Device Type 42
3.2.3 Object 1001h: Error Register 42
3.2.4 Object 1003h: Pre-Defined Error Field 42
3.2.5 Object 1010h: Store Parameters 43
3.2.6 Object 1011h: Restore Default Parameters 45
3.2.7 Object 1016h: Consumer Heartbeat Time 46
3.2.8 Object 1017h: Producer Heartbeat Time 47
3.2.9 Object 1018h: Identity Object 47
3.2.10 Object 1029h: Error Behaviour 48
3.3 Manufacturer Objects 50
3.3.1 Object 2010h: AI Input Range 52
3.3.2 Object 2011h: AI Decimal Digits FV 53
3.3.3 Object 2100h: MO Ramp Up (Min to Max) 54
3.3.4 Object 2100h: MO Ramp Up (Min to Max) 54
3.3.5 Object 2102h: MO Error Minimum 54
3.3.6 Object 2103h: MO Error Maximum 55
3.3.7 Object 2104h: MO Control Source 55
3.3.8 Object 2105h: MO Control Number 56
3.3.9 Object 2106h: MO Enable Source 56
3.3.10 Object 2107h: MO Enable Number 56
3.3.11 Object 2108h: MO Enable Response 57
3.3.12 Object 2109h: MO Override Source 57
3.3.13 Object 210Ah: MO Override Number 57
3.3.14 Object 210Bh: MO Override Control Response 58
3.3.15 Object 210Ch: MO Override Response Value 58
3.3.16 Object 210Dh: MO Output Frequency 58
3.3.17 Object 210Eh: MO Gain 59
3.3.18 Object 210Fh: MO Integral 59
3.3.19 Object 2110h: MO Derivative 60
3.3.20 Object 2111h: MO Over Current Reset Timer 60
3.3.21 Object 2113h: MO Dead Time 61
3.3.22 Object 2114h: MO Error Detection Enable 61
3.3.23 Object 2115h: MO Error Clear Hysteresis 62
3.3.24 Object 2116h: MO Error Reaction Delay 62
3.3.25 Object 2117h: MO Auto Update 63
3.3.26 Object 2201h: ENC Stationary Delay 63
3.3.27 Object 2202h: ENC Debounce Filter 63
3.3.28 Object 2203h: ENC Error Detection Enable 64
3.3.29 Object 2204h: ENC Error Clear Hysteresis 64
3.3.30 Object 2205h: ENC Error Reaction Delay 65
3.3.31 Object 2206h: ENC Auto Update 65
3.3.32 Object 22X0h: ENC X Input Logic Type 66
3.3.33 Object 22X1h: ENC Active State 66
3.3.34 Object 22X2h: ENC X Debounce Time 67
3.3.35 Object 2233h: ENC Z Debounce Filter 67
3.3.36 Object 2234h: ENC Number of Pulses Per Revolution 67

3.3.37 Object 2235h: ENC Number to Reset Pulse Counter ... 68
3.3.38 Object 2236h: ENC Pulse Count Pulse Window ... 68
3.3.39 Object 2237h: ENC Pulse Count Time Window ... 69
3.3.40 Object 2238h: ENC Overflow Value ... 69
3.3.41 Object 2300h: UNI Input Logic Type ... 70
3.3.42 Object 2301h: UNI Active State ... 70
3.3.43 Object 2302h: UNI Debounce Time ... 71
3.3.44 Object 2303h: UNI Debounce Filter ... 71
3.3.45 Object 2304h: UNI Overflow Value ... 72
3.3.46 Object 2305h: UNI Number of Pulses Per Revolution ... 72
3.3.47 Object 2306h: UNI Number to Reset Pulse Counter ... 73
3.3.48 Object 2307h: UNI Pulse Count Pulse Window ... 74
3.3.49 Object 2308h: UNI Pulse Count Time Window ... 74
3.3.50 Object 2309h: UNI Pullup/Down Mode ... 75
3.3.51 Object 220Ah: UNI Auto Update ... 75
3.3.52 Object 220Bh: UNI Error Detection Enable ... 76
3.3.53 Object 220Ch: UNI Error Clear Hysteresis ... 77
3.3.54 Object 220Dh: UNI Error Reaction Delay ... 77
3.3.55 Object 2500h: EC Extra Received Process Value ... 78
3.3.56 Object 2502h: EC Decimal Digits PV ... 79
3.3.57 Object 2520h: EC Scaling 1 PV ... 79
3.3.58 Object 2522h: EC Scaling 2 PV ... 80
3.3.59 Object 260X: LED Control Source ... 81
3.3.60 Object 261X: LED Control Number ... 81
3.3.61 Object 262X: LED Response ... 82
3.3.62 Object 263X: LED Blink Rate ... 82
3.3.63 Object 2700: EX Under Voltage ... 83
3.3.64 Object 2701: EX Over Voltage ... 83
3.3.65 Object 2702: EX Shut Down Temperature ... 84
3.3.66 Object 2703: EX Disable Output ... 84
3.3.67 Object 2704: EX Error React Enable ... 85
3.3.68 Object 2705: EX Error React Enable ... 85
3.3.69 Object 3yz0h: LTyz Input X-Axis Source ... 86
3.3.70 Object 3yz1h: LTyz Input X-Axis Number ... 87
3.3.71 Object 3yz2h: LTyz Auto Repeat ... 87
3.3.72 Object 3yz3h: LTyz X-Axis Decimal Digits PV ... 88
3.3.73 Object 3yz4h: LTyz Y-Axis Decimal Digits PV ... 88
3.3.74 Object 3yz5h: LTyz Point Response ... 88
3.3.75 Object 3yz6h: LTyz Point X-Axis PV ... 89
3.3.76 Object 3yz7h: LTyz Point Y-Axis PV ... 90
3.3.77 Object 3yz8h: LTyz Output Y-Axis PV ... 91
3.3.78 Object 3300h: Logic Block Enable ... 91
3.3.79 Object 3310h: Logic Block Selected Table ... 92
3.3.80 Object 3320h: Logic Block Output PV ... 93
3.3.81 Object 3x01h: LB(3-x) Lookup Table Numbers ... 93
3.3.82 Object 3x02h: LB(3-x) Function Logical Operator ... 94
3.3.83 Object 3x11h: LB(3-x) Function A Condition 1 ... 95
3.3.84 Object 3x12h: LB(3-x) Function A Condition 2 ... 95
3.3.85 Object 3x13h: LB(3-x) Function A Condition 3 ... 95
3.3.86 Object 3x21h: LB(3-x) Function B Condition 1 ... 95

3.3.87 Object 3x22h: LB(3-x) Function B Condition 2 95
3.3.88 Object 3x23h: LB(3-x) Function B Condition 3 95
3.3.89 Object 3x31h: LB(3-x) Function C Condition 1 95
3.3.90 Object 3x32h: LB(3-x) Function C Condition 2 95
3.3.91 Object 3x33h: LB(3-x) Function C Condition 3 95
3.3.92 Object 4000h: Math Function Enable 96
3.3.93 Object 4021h: Math Output Scaling 1 PV 97
3.3.94 Object 4023h: Math Output Scaling 2 PV 98
3.3.95 Object 4030h: Math Output Process Value 98
3.3.96 Object 4032h: Math Output Decimal Digits PV 99
3.3.97 Object 4y00h: Math Y Input Source 99
3.3.98 Object 4y01h: Math Y Input Number 100
3.3.99 Object 4y02h: Math Y Input Function Number 101
3.3.100 Object 4y03h: Math Y Input Decimal Digits FV 101
3.3.101 Object 4y20h: Math Y Input Scaling 1 FV 102
3.3.102 Object 4y22h: Math Y Input Scaling 2 FV 103
3.3.103 Object 4y40h: Math Y Input Gain 103
3.3.104 Object 4y50h: Math Y Operator 104
3.3.105 Object 5010h: Constant Field Value 105
3.3.106 Object 5540h: Bootup Message Enabled 107
3.3.107 Object 5555h: Start in Operation Mode 107
3.3.108 Object 55AAh: Start Bootloader 108
3.3.109 Object 5B50h: Change Baud Rate 108
3.3.110 Object 5B51h: Change Node ID 109
3.4 Application Objects 109
3.4.1 Object 6000h: DI Read State 8 Input Lines 110
3.4.2 Object 6002h: DI Polarity 8 Input Lines 110
3.4.3 Object 7100h: AI Input Field Value 111
3.4.4 Object 6110h: AI Sensor Type 112
3.4.5 Object 6112h: AI Operating Mode 112
3.4.6 Object 7120h: AI Input Scaling 1 FV 113
3.4.7 Object 7121h: AI Input Scaling 1 PV 113
3.4.8 Object 7122h: AI Input Scaling 2 FV 114
3.4.9 Object 7123h: AI Input Scaling 2 PV 115
3.4.10 Object 7130h: AI Input Process Value 115
3.4.11 Object 6132h: AI Decimal Digits PV 116
3.4.12 Object 7148h: AI Span Start 117
3.4.13 Object 7149h: AI Span End 117
3.4.14 Object 7124: AI Input Offset 118
3.4.15 Object 6125: AI Autozero 118
3.4.16 Object 61A0h: AI Filter Type 119
3.4.17 Object 61A1h: AI Filter Constant 120
3.4.18 Object 7450h: PID Proportional Band 120
3.4.19 Object 7452h: PID Integral Time 122
3.4.20 Object 7454h: PID Derivative Action Time 122
3.4.21 Object 7456h: PID Cycle Time 123
3.4.22 Object 6458h: PID Physical Unit Timing 123
3.4.23 Object 6459h: PID Decimal Digits Timing 124
3.4.24 Object 7300h: AO Output Process Value 124
3.4.25 Object 6302h: AO Decimal Digits PV 124

3.4.26 Object 6310h: AO Output Type 125
3.4.27 Object 7320h: AO Output Scaling 1 PV 125
3.4.28 Object 7321h: AO Output Scaling 1 FV 126
3.4.29 Object 7322h: AO Output Scaling 2 PV 126
3.4.30 Object 7323h: AO Output Scaling 2 FV 127
3.4.31 Object 7330h: AO Output Field Value 127
3.4.32 Object 6340h: AO Fault Mode 127
3.4.33 Object 7341h: AO Fault FV 128
4 REFLASHING OVER CAN WITH EA ® BOOTLOADER 129
5 TECHNICAL SPECIFICATIONS 134
5.1 Power Input 134
5.2 Input Specifications 134
5.3 Output Specifications 135
5.4 General Specifications 135
5.5 Dimensions and Typical Connections 137

1 INTRODUCTION

The following user manual describes architecture, functionality, configuration parameters and flashing instructions for the Universal 25A DC Motor Controller, CANopen®. 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 motor controller supports one motor output, one Encoder Input with three input signals for A, B, and Z, as well as two universal 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. shows the hardware functional block diagram of the Universal 25A DC Motor Controller.

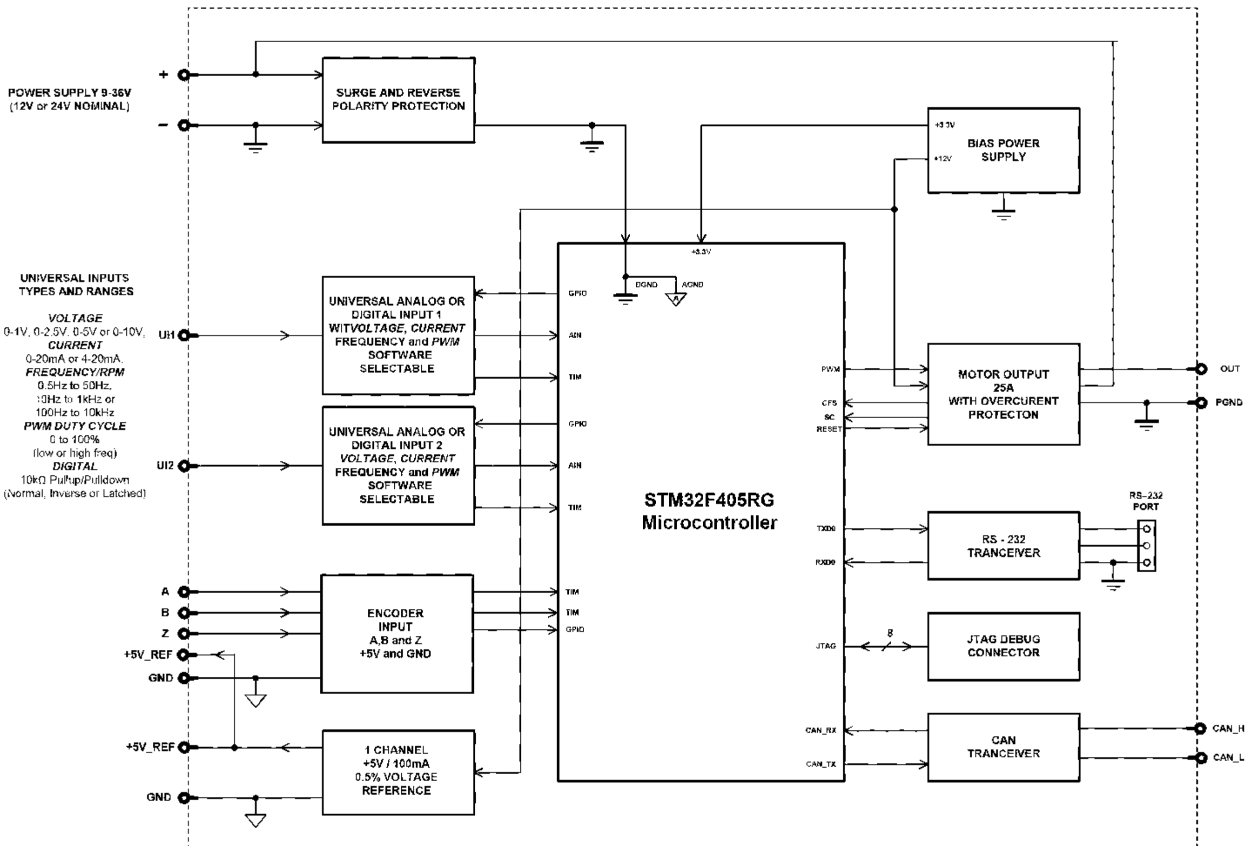


Table 1: Hardware Functional Block Diagram

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

Note: This controller accepts power supply voltages from 9 to 36 VDC.

2 CONTROLLER DESCRIPTION

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

The Universal Motor Controller can drive one motor and has additionally two Universal Inputs as well as three inputs to connect an incremental encoder. Furthermore, the motor controller has several function blocks to use the motor controller in a wider spectrum of applications. All these function blocks are explained in more detail in the upcoming subchapters.

Measured input data can be sent to a CANopen® Network as is or used in the motor controller function blocks for controlling how the motor is driven. Throughout this document, setpoint names are referred with bolded text and the setpoint option is referred with italicized. For example, **Input Sensor Type** setpoint set to option *Frequency/RPM*.

2.1 Motor Drive Function Block

The motor control output can drive a DC motor or other load with continuous 25A current. The maximum instantaneous current can go beyond this limit up to 35A. The shunt monitor on the board will detect any current value above 35A and shut off the output. This is a fixed value and is not adjustable through software.

The Motor Drive is equipped with several setpoint parameter to adjust its performance. Each parameter is explained in more detail below.

2.1.1 Motor Type

The Motor Output Type determines the output signal related to its input signal. Changing this setpoint causes other setpoints in the group to update and match the selected type, thus the **Output Type** should be selected before configuring other setpoints within the setpoint group. The Output Type can be selected by object 6310h **AO output type** with its options listed in Table 2.

Value	Output Type
0	Output Disabled
10	Proportional Voltage
20	Proportional Current
40	PWM Duty Cycle

Table 2: Motor Output Types

The *Proportional Current* sets the output signal corresponding to its input signal and the current feedback value measured at the output. An internal PID controller can be used for better output control which can be set with the objects 210Eh **MO Gain**, 210Fh MO **Integral**, and 2110h MO **Derivative Value**. The PID-controller is only available for the *Proportional Current* since there is no voltage feedback. The input and output values of the *Proportional Current* are set and measured in milli Ampere. Therefore, the Minimum and Maximum value of the output must be set in Milli Ampere as well. The Output frequency can be adjusted in a range between 4 kHz to 25 kHz using object 210D **MO Output Frequency**.

The *PWM Duty Cycle* allows the user to run the output at a fixed frequency configured by the setpoint **MO Output Frequency**, while the duty cycle changes depending on the control signal.

The *Proportional Voltage* uses the measured value of the power supply and adjusts the duty cycle of the output such that the average value will match the target output voltage.

Object 2100h **Ramp Up** and object 2101h **Ramp Down** setpoints define the ramp duration in milli seconds for the motor control output current/voltage to rise to its target or fall back to zero. Please note if the enable function is used to stop the motor, the ramp function is not applied because the enable functionality utilizes hardware feature for disabling the motor.

For Proportional outputs signal minimum and maximum values are configured with object 7321h **AO Scaling 1 FV** and object 7323h **AO Scaling 2 FV** setpoints. Their value range is limited its by selected 6310h **MO Output Type**.

Regardless of what type of control input is selected, the output will always respond in a linear fashion to changes in the input per Equation

$$y = mx + a$$

$$m = \frac{Y_{max} - Y_{min}}{X_{max} - X_{min}}$$

$$a = Y_{min} - m * X_{min}$$

Equation 1: Linear Slope Calculations

X is defined by its control input minimum and maximum ranges. Therefore, X_{min}= Control Input Minimum and X_{max} = Control Input Maximum. The Y in Equation 1 is defined by Y_{min} = 7321h **AO Scaling 1 FV,** and Y_{max} = 7323h **AO Scaling 2 FV**. While X-axis has the constraint that X_{min}= < X_{max}, the Y-axis does not have such limitation. Thus, configuring **AO Scaling 1 FV** to be greater than **AO Scaling 2 FV** allows the output to follow the control signal inversely.

2.1.2 Motor Drive Control/Enable/Override/ Sources

The Motor Drive output can be configured to be commanded and/or enabled/overridden by the control sources listed in Table 3. This table also displays the number associated to the control sources which can be selected. The default control source is highlighted in Table 3 while the default Enable Source and Override Source is configured to *Source Not Used.*

Value	Meaning	Source Range
0	*Source Not Used*	*[0]*
1	***Universal Input Data***	***[1…2]***
2	*Universal Input Frequency*	*[1…2]*
3	*Encoder Data A*	*[1]*
4	*Encoder Data B*	*[1]*
5	*Encoder Direction*	*[1]*

6	*Encoder Counter*	*[1]*
7	*Encoder Z Data*	*[1]*
8	*Power Supply Voltage Fault State*	*[1]*
9	*Temperature Fault State*	*[1]*
10	*CANopen® Receive*	*[1…10]*
11	*Conditional Block*	*[1…10]*
12	*Global Power Supply Voltage*	*[1]*
13	*Global Temperature*	*[1]*
14	*Look Up Table*	*[1…6]*
15	*Math Function*	*[1…4]*
16	*Programmable Logic*	*[1…2]*
17	*Lookup Table*	*[1…6]*
18	*Constant Data*	*[1…10]*
19	*Current Feedback*	*[1]*
20	*PID Control*	*[1]*

Table 3: Motor Drive Output Control/Enable/Override/Unlatch Source

The selected control source object 2104 **MO Control Source** in combination with object 2105 **MO Control Number** is the main commanding source of the motor output based on the motor type selected in object 6310h **MO Output Type**.

For example, setting **Control Source** to *Universal Input* and **Control Number** to '1', connects signal measured from Universal Input 1 to the output in question. Is the input signal an analog input with a voltage between 0V and 5V, for instance, and 1V is measured at this input, the output signal would be 20% in case the **Motor Type** is set to *PWM Output*. That means that the output signal is always responding linear to changes in the control signal.

2.1.3 Motor Drive Enable

In addition to the Control input, the Motor Drive also supports an Enable and Override input. The **MO Enable Source**, object 2106h, together with its **MO Enable Number**, object 2107h, determines the enable signal for the output in question. Object 2108h **MO Enable Response** is used to select how the output will respond to the selected Enable signal. The options for the enable response are listed in Table 4.

Value	**Meaning**
0	Enable When On, Else Shutoff
1	Enable When On, Else Rampoff
2	Enable When Off, Else Shutoff
3	Enable When Off, Else Rampoff
4	Enable When On, Else Ramp To Minimum
5	Enable When On, Else Ramp To Maximum

Table 4: Enable Response Options

2.1.4 Motor Drive Override

The override input allows the motor drive to be configured to go to a default value in case of the override input being engaged/disengaged, depending on the logic selected in object 210Bh **Override Control Response**, presented in Table 5. When active, the output will be driven to the value in object 210Ch **Output at Override Command** regardless of the value of the Control input which is 0 for all output types. Object 2109h **MO Override Source** and object 210Ah **Override Number** determine the Override input signal as described in an example in subchapter 2.1.2.

Value	Meaning
0	Override When OFF
1	**Override When ON**

Table 5: Override Control Response Options object 210Bh

2.1.5 Output Field Values and Scaling

The output value is shifted according to object 6332h **AO Decimal Digits FV** and then written to read-only and mappable object 7330h **AO Output Field Value.** The value of 6332h will depend on the AO Sensor Type and Output Range selected and will be automatically updated per Table 6: AO Decimal Digits FV Depending on Sensor Type when either 7330h or 6332h are changed. All other objects associated with the input field value also apply to this object. These objects are 7120h **AO Output Scaling 1 FV**, 7122h **AO Output Scaling 2 FV**, 2102h **MO Error Min**, 2103h **MO Error Max**, and 2115h **MO Error Clear Hysteresis**. These objects are also automatically updated when the Type is changed.

Sensor Type and Range	Decimal Digits
Proportional Current	0 [mA]
PWM Duty Cycle	1 [0.1 %]
Proportional Voltage	1 [0.1 V]

Table 6: AO Decimal Digits FV Depending on Sensor Type

2.1.6 Output Process Values and Scaling

Read-only object 7300h **AO Output Process Value** is also mappable. However, the default values for objects 7320h **AO Scaling 1 PV** and 7322h **AO Scaling 2 PV** are set to equal 7321h and 7323h respectively, while object 6302h **AO Decimal Digits PV** is automatically initialized to equal 6332h. This means that the default relationship between the FV and PV is one-to-one.

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, 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 AO Output PV object is not being used.

The **MO Error Min** and **MO Error Max** 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 AO Output FV object.

Error! Reference source not found. lists the default values that are loaded into objects mentioned above f or each Sensor Type Recall that these objects all have the decimal digits applied to them as outlined in Table 7.

Sensor Type	Proportional Current	PWM Duty Cycle	Proportional Voltage
2102h MO Error Min (i.e. Error Min)	0[mA]	0[0.1%]	0[0.1V]
7321h AO Scaling 1 FV (i.e. Output At Min)	0 [mA]	0[0.1%]	0[0.1V]
7323h AO Scaling 2 FV (i.e. Output At Max)	1000[mA]	1000[0.1%]	160[0.1V]
2103h MO Error Max (i.e. Error Max)	1050[mA]	1002[0.1%]	165[0.1V]
2111h Error Clear Hysteresis	100[mA]	5[0.1%]	2[0.1V]

Table 7: AO Object Defaults Based on Sensor Type

2.1.7 Automatic Update of Output 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 2117h MO **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.

2.1.8 Fault Detection

Fault detection is available for all output types. A current feedback signal, for example, is measured and compared to the desired output current value.

The last objects associated with the motor output block left to discuss are those associated with fault detection. Should the calculated output fall outside of the allowable range, as defined by object 2102h **MO Error Min** and object 2103h **MO Error Max**, an error flag will be set in the application if object 2114h **MO Error Detect Enabled** is set to TRUE (1).

When 7330h **AO Output FV** < 2102h **MO Error Min**, an “Out of Range Low” flag is set. If the flag stays active for the 2116h **MO Error Reaction Delay** time, which is 0 by default for the motor output, an Output Overload Emergency (EMCY) message will be added to object 1003h **Pre-Defined Error Field**. Similarly, when 7330h **AO Output FV** < 2103h **MO Error Max**, 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 Output Fault. By default, the Error behavior is set to *Pre-operational on Error* for an output faut. This causes the unit being set back to pre-operational which also turns off the motor output. 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 output comes back into range. Object 2115h **MO Error Clear Hysteresis** is used here so that the error flag will not be set/cleared continuously while the AO Output FV hovers around the MO Error Min/Max value.

To clear an “Out of Range Low” flag, **AO Output FV** >= (**MO Error Min** + **MO Error Clear Hysteresis**)
To clear an “Out of Range High” flag, **AO Output FV** <= (**MO Error Max** - **MO Error Clear Hysteresis**)
Both flags cannot be active at once. Setting either one of these flags automatically clears the other.

The outputs are inherently protected against a short to GND or +Vps by circuitry. In case of a dead short, the hardware will automatically disable the output drive, regardless of what the processor is commanding for the output. When this happens, the processor detects output hardware shutdown and commands off the output. An Output Current Overload Emergency (EMCY) message will be added to object 1003h **Pre-Defined Error Field**. It will periodically try to re-engage the short load depending on the time set by the object 2111h **Over Current Reset Time**. If the fault has gone away since the last time the output was engaged while shorted, the controller will automatically resume normal operation.

In case of an open circuit, there will be no interruption of the control for any of the outputs. If the current feedback value is lower than 500mA compared to the set value, an EMCY message will be triggered. If the current threshold of current feedback and set value reaches 400mA, the Open Circuit EMCY message will be deleted from object 1003h **Pre-Defined Error Field.** The processor will continue to attempt to drive the open load. The measured current through the load is available to be broadcasted on a CAN message if desired.

2.2 Encoder Input Function Block

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 input application parameters for this controller are used for the Universal Inputs and Encoder Inputs. To use the right input type, Table 8 shows what subindex is used for each input.

Sub-Index	**Input Type**
1	Encoder A
2	Encoder B
3	Encoder Z
4	Universal Input 1
5	Universal Input 2

Table 8: Input Subindices

Furthermore, object AI Input range 2010h and AI Decimal Digits 2011h also having 5 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 9.

Input Type	**Index**
Encoder Input	22XXh
Universal Input	23XXh

Table 9: Input Indices Numbering

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

The controller provides an Encoder input, which has three physical inputs to support the incremental encoder's A, B, and Z signal. In case no encoder is connected to the unit, these inputs can be used for other purposes. Signal A and B can be used as digital on/off signals and signal Z can measure frequency, PWM and digital on/off signals. The available Input Types are listed in Table 10.

Setpoint Value	Input Type	Supported by Encoder A, B	Supported by Encoder Z
0	Not Implemented	Yes	Yes
1	Digital Input	Yes	Yes
2	Encoder Input	Yes	No
2	Frequency input	No	Yes
3	PWM: 0 to 100%	No	Yes
4	16-Bit Counter	No	Yes
5	Time Window	No	Yes
6	Pulse Window	No	Yes

Table 10: Encoder Input Types

The encoder signals A and B on the motor controller unit are detecting the pulses generated by the incremental encoder. The two measurements, step count and direction, are determined using these two input signals. The Z signal of the encoder works like a common frequency input and measures the input signal every time a rising edge, or falling edge dependent on the configurations, occurs.

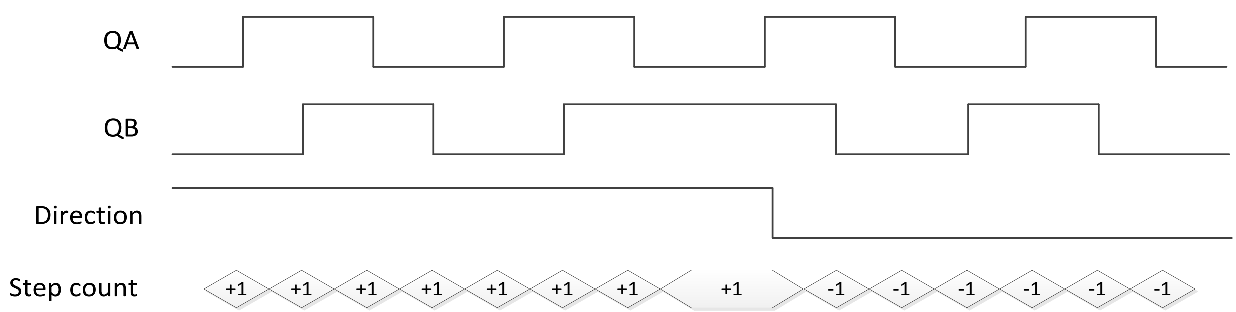


Figure 1: Incremental Decoder Signals (QA&QB) and resulting Direction and Step count

A positive direction is indicated when the edge of signal A is rising before the edge of signal B is rising. If the encoder rotates in a positive direction, the **Encoder Counter value** increases. In case the signal changes and input B becomes the leading signal, the counter value decreases, and the direction is indicated as negative.

2.2.1 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 11.

Value	Option
0	Disabled
1	**Analog Normal**
10	Encoder Counter

Table 11: 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**. **Error! Reference source not found.** shows what values are representing what d irection.

Value	Direction
0	Stationary
1	Negative
2	Positive

The Signal Z Input, on the other hand, has more available input types listed in Table 12.

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

Table 12: Encoder Signal Z Input Types

The Input Type can be selected by object 6310h **AO output type**

2.2.2 Encoder Stationary Delay

Object **Enc Stationary Delay** 2201h 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.

2.2.3 Input Logic Type

When Encoder Signal A/B/Z is set to Digital ON/OFF, further setpoints are available. The Encoder Signal parameters are arranged as shown in Table 13.

Encoder Signal	Index
A	XX1X
B	XX2X
Z	XX3X

Table 13: Encoder Signal Indeces

The object 22X0h **ENC A/B/Z Input logic Type** is 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 14 shows the options available for this parameter.

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

Table 14: 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.

2.2.4 Active High/Low

This object 22X1h **ENC A/B/Z Input Active State** allows the user to select how the controller responds to the behaviour of the digital input. Table 15 shows the different Active High/Low options with the default being highlighted.

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

Table 15: Active High/Low

The inputs of the Digital Inputs have a fixed 10kOhm pull-down resistor. Given that by default, the inputs are configured to *Active High*, an OFF response by the input is achieved when the input is grounded. The opposite applies if the parameter is set to Active Low.

2.2.5 Debounce Time

The Digital Input Debounce Time can be set with object 22X2h **ENC A/B/Z Input Debounce Time** and is a useful parameter in case the digital input signal is noisy. Figure 2 shows how the Debounce Time helps detect a correct input signal.

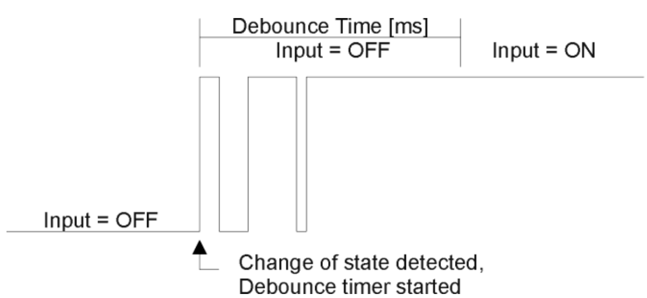


Figure 2: Digital Input Debounce Time

2.2.6 Input Software Filter Types

This parameter is only relevant for the Encoder Z Input. Its data can be filtered using object **61A0h AI Filter Type** and object 61A1h **AI Filter Constant**. There are three (3) filter types available as listed in Table 16.

0	*No Filtering*
1	*Moving Average*
2	*Repeating Average*

Table 16: Input Filtering Types

The first filter option, *No Filtering,* provides no filtering to the measured data. Thus, the measured data will be directly used to the any function block which uses this data.

The second option, *Moving Average*, applies the 'Equation 1' below to measured input data, where Value_N represents the current input measured data, while Value_{N-1} represents the previous filtered data. The Filter Constant is the **Filter Constant** setpoint.

Equation 2 - Moving Average Filter Function:

$$\text{Value}_{\text{N}} = \text{Value}_{\text{N}-1} + \frac{(\text{Input} - \text{Value}_{\text{N}-1})}{\text{Filter Constant}}$$

The third option, Repeating Average, applies the 'Equation 2' below to measured input data, where N is the value of **Filter Constant** setpoint. The filtered input, Value, is the average of all input measurements taken in N (**Filter Constant)** number of reads. When the average is taken, the filtered input will remain until the next average is ready.

Equation 3 - Repeating Average Transfer Function:

$$\text{Value} = \frac{\sum_0^N \text{Input}_N}{N}$$

2.2.7 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 17. 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 17: 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.

2.2.8 Frequency/RPM Input

Note: This parameter is only available for Signal Z of the Encoder.

There is another type of 'digital' input that can be selected which are connected to the microcontroller timers when the Encoder Z Input Type is set to *Frequency Input*. This input type can be automatically turned into an RPM measurement by setting object 2234h **ENC Z Number of Pulses per Revolution** to a non-zero value. This means when this object 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.

2.2.9 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).

Object 2235h **ENC Z Number to Reset Pulse Count** (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 object's value is reached at which the counter will reset back to 0.
Object 2236h **ENC Z Pulse Count Pulse Window** (Pulse Window Counter) 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 2237h **ENC Z Pulse Count Time Window** (Time Window Counter) is a Counter input type in which the object is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

2.2.10 Debounce Filter

This filter is applied to the input before the state is read by the processor. The options for the debounce filter are shown in **Error! Reference source not found.**, with the default bolded, and can be changed with o bject 2233h **ENC Z Debounce Filter.**

Value	**Meaning**

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

Table 18: Debounce Filter Options

2.2.11 Minimum and Maximum Ranges/Errors

The object 7120h **AI Scaling 1 FV**, and 7122h **AI Scaling 2 FV** setpoints are used to create the overall useful range of the inputs. For example, if AI Scaling 1 FV is set to 100 RPM and AI Scaling 2 FV is set to 1900 RPM, the overall useful range (0-100%) is between 100 to 1900 RPM. Anything below the AI Scaling 1 FV will saturate at AI Scaling 1 FV. Similarly, anything above the AI Scaling 2 FV will saturate at AI Scaling 2 FV.

The other two setpoints, 7148h **AI Span Start** (Error Minimum) and 7149h **AI Span End** (Error Maximum) must be lower than the **AI Scaling 1 FV** and the **AI Scaling 2 FV**, respectively because any value below the **AI Span Start** or above the **AI Span End** will create a fault. If that input is commanding an output, the output will shut off. If requiring, the output to remain active, the fault response in object 1029h **Error Behaviour** can be changed to a different value.

To enable the error detection, object 2203h **Enc Error Detect Enable** must be set to Enabled. Once the fault has been detected, the associate flag will be cleared only once the input value comes back into range of the maximum and minimum error value. Object 2204h **ENC 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 Encoder **AI Span Start and AI Span End.**

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

2.2.12 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 19 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 2204h **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	0
Frequency Input	1
RPM	0
PWM Input	1

Counter Input	0

Table 19: AI Decimal Digits FV Depending on Sensor Type

2.2.13 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 3 **Error! Reference source not found.**. Should a non-linear response be desired, the lookup table function block can be used instead, as described in 2.6

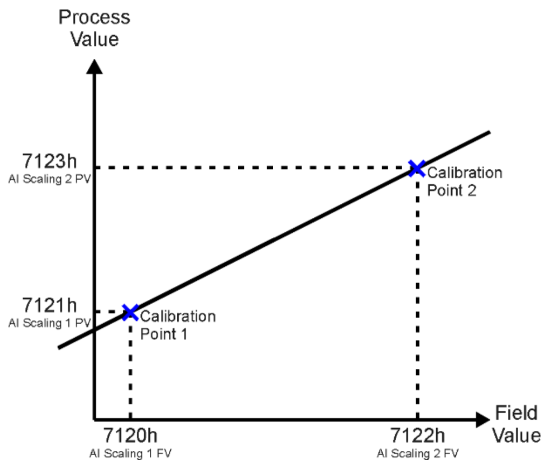


Figure 3: Analog Input Linear Scaling FV to PV

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 20 lists the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2204h for each Sensor Type and Input Range combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 19.

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)	2204h Error Clear Hysteresis
Not Implemented	-	-	-	-	-
Digital Input	OFF	OFF	ON	ON	0
Frequency input	0	5	950	1000	-
PWM: 0 to 100%	1	5	95	99	1

16-Bit Counter	0	0	1	1000000	0
Time Window	0	0	1	1000000	0
Pulse Window	0	0	1	1000000	0

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

2.2.14 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 2206h **ENC 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.

2.3 Universal Input Function Block

The universal Inputs on the motor controller can be configured to an even wider range of input types compared to the Encoder Z Signal. The available input types are listed in Table 21 with its default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2305h for each Sensor Type and Input Range combination.

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)	2204h Error Clear Hysteresis
Not Implemented	-	-	-	-	-
Digital Input	OFF	OFF	ON	ON	0
Frequency input	0	5	950	1000	-
PWM: 0 to 100%	1	5	95	99	1
16-Bit Counter	0	0	1	1000000	0
Time Window	0	0	1	1000000	0
Pulse Window	0	0	1	1000000	0
Voltage: 0 to 5V	100 [mV]	200 [mV]	4800 [mV]	4900 [mV]	100 [mV]
Voltage: 0 to 10V	200 [mV]	500 [mV]	9500 [mV]	9800 [mV]	100 [mV]
Current: 0 to 20mA	0 [uA]	0 [uA]	20000 [uA]	20000 [uA]	500 [uA]
Current: 4 to 20mA	1000 [uA]	4000 [uA]	20000 [uA]	21000 [uA]	500 [uA]
Encoder Input	-	-	-	-	-

Table 21: Input Type Options for Universal Input

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, Encoder Input Function Block, are also available for the Universal Input except that the indices are different for the Universal Input. Table 22 shows what indices are used for the Universal Input compared to the Encoder Inputs.

Parameter	Universal Input (UNI)	Encoder Input (ENC)	Refer to
Logic Type	2300h	2210h, 2220h, 2230h	Table 14
Active State	2301h	2211h, 2221h, 2231h	Table 15
Debounce Time	2302h	2212h, 2222h, 2232h	2.2.5
Debounce Filter	2303h	2202h	**Error! R eference**

			source not found.
Overflow Value	2304h	2238h	2.2.8
Number pulses per revolution	2305h	2234h	2.2.8
Num to reset pulse counter	2306h	2235h	2.2.9
Pulse count pulse window	2307h	2236h	2.2.9
Pulse count time window	2308h	2237h	2.2.9
Auto Update	230Ah	2206h	2.2.14
Error detect enable	230Ch	2203h	2.2.11
Error clear hysteresis	230Dh	2204h	**Error! Reference source not found.**2.2.11
Error react delay	230Eh	2205h	2.2.11**Error! Reference source not found.**

Table 22: Universal Input parameter reference

There are additional input types available for the Universal Input which differ in ranges to the Encoder Inputs. 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 Pull Up/Down Options

Conversely, Digital Input ON/OFF, Frequency, PWM and Counter inputs use object 2309h **UNI Pullup/Down Mode** while **voltage and current inputs set this object to zero**. The options for this parameter are listed in Table 24.

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

Table 24: Pull Up/Down Options

Note: Please make sure to set the Pull Up/Down type parameter after changing the input type from analog to any digital input type since this parameter will not be automatically updated for digital input types.

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 25.

Value	Meaning	Source Range
0	*Source Not Used*	*[0]*
1	***Universal Input Data***	***[1…2]***
2	*Universal Input Frequency*	*[1…2]*
3	*Encoder Data A*	*[1]*
4	*Encoder Data B*	*[1]*
5	*Encoder Direction*	*[1]*
6	*Encoder Counter*	*[1]*
7	*Encoder Z Data*	*[1]*
8	*Power Supply Voltage Fault State*	*[1]*
9	*Temperature Fault State*	*[1]*
10	*CANopen® Receive*	*[1…10]*
11	*Conditional Block*	*[1…10]*
12	*Global Power Supply Voltage*	*[1]*
13	*Global Temperature*	*[1]*
14	*Look Up Table*	*[1…6]*
15	*Math Function*	*[1…4]*
16	*Programmable Logic*	*[1…2]*
17	*Lookup Table*	*[1…6]*
18	*Constant Data*	*[1…10]*
19	*Current Feedback*	*[1]*
20	*PID Control*	*[1]*

Table 25: 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 26: 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 27: Scaling Limits per Control Source - ***(*)*** *- Whichever value is smaller;* ***(**)*** *- Whichever value is larger*

2.5 PID Control Function Block

The PID Control function block is an independent logic block, but it is normally intended to be associated with proportional output control blocks described in the chapters before. When the **Control Source** for an output has been setup as a *PID Function Block*, the command from the selected PID block drives the physical output on the motor controller unit.

The object 3903h **PID Target Command Source** and 390Ah **PID Target Command Number** determine control input and object 3905h **PID Feedback Input Source** and 3906h **PID Feedback Input Number** determine the established the feedback signal to the PID function block. The object 3907h **PID Response Profile** will use the selected inputs as per the options listed in Table 28**Error! Reference source not found.**. When active, the PID algorithm will be called every time set in object 7456h **PID Loop Update Rate** in milliseconds.

0	Single Output
1	Setpoint Control
2	On When Over Target
3	On When Below Target

Table 28: PID Response Options

When a '*Single Output*' response is selected, the Target and Feedback inputs do not have to share the same units. In both cases, the signals are converted to a percentage values based on the minimum and maximum values associated with the source function block.

For example, a CAN command could be used to set the target value, in which case it would be converted to a percentage value using **Receive Data Min** and **Receive Data Max** setpoints in the appropriate *CAN Receive X* function block. The closed-loop feedback signal, i.e. a Current Feedback from the motor controller output, could be selected as the feedback source. In this case the value of the input would be converted to a percentage based on object 2520 **EC scaling 1 PV** and **object 2522 EC scaling 2 PV**. The output of the PID function would depend on the difference between the commanded target and the measured feedback as a percentage of each signals range. In this mode, the output of the block would be a value from -100% to 100%.

In Order to allow the output to stabilize, the user can select a non-zero value for object 3900h **PID Delta Tolerance**. If the absolute value of $Error_K$ is less than this value, $Error_K$ in the formula below will be set to zero.

The PID algorithm used is shown below, where G (object 7450h Proportional Band), Ki (object 3901h Integral Gain), Ti (object 7452h Integral Time), Kd (object 3902 Derivative Gain), Td (object 7454h Derivative Time) and Loop_Update_Rate are configurable parameters.

$$PIDOutput_k = P_k + I_k + D_k$$

$$P_k = P\_Gain * Error_k$$
$$I_k = I\_Gain * ErrorSum_k$$
$$D_k = D\_Gain * (Error_k - Error_{k-1})$$

$$Error_k = Target - Feedback$$
$$ErrorSum_k = ErrorSum_{k-1} + Error_k$$

$$P\_Gain = G$$
$$I\_Gain = G * Ki * T/Ti \text{ (Note: If Ti is zero, I\_Gain = 0)}$$
$$D\_Gain = G * Kd * Td/T$$

$$T = Loop\_Update\_Rate * 0.001$$

Note: Each system will have to be turned for the optimum output response. Response times, overshoots and other variables will have to be decided by the customer using an appropriate PID tuning strategy. Axiomatic is not responsible for tuning the control system.

2.6 Lookup Table Function Block

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 2.7.

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 2.6.4.

2.6.1 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 25.

2.6.2 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 27.

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 2.6.3, 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.

2.6.3 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.

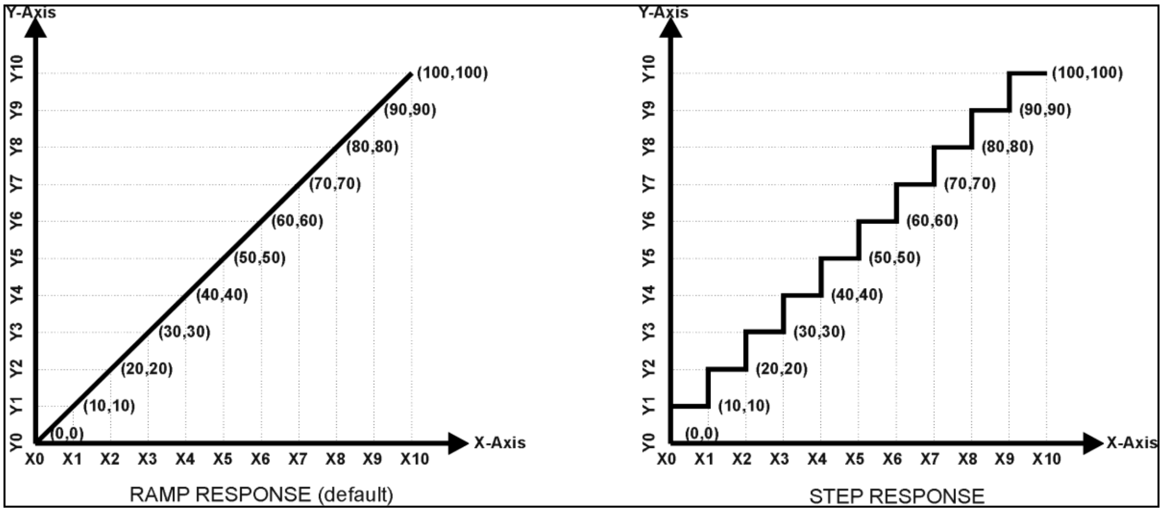


Figure 4: Lookup Table Defaults with Ramp and Step Responses

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 5. 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.')*

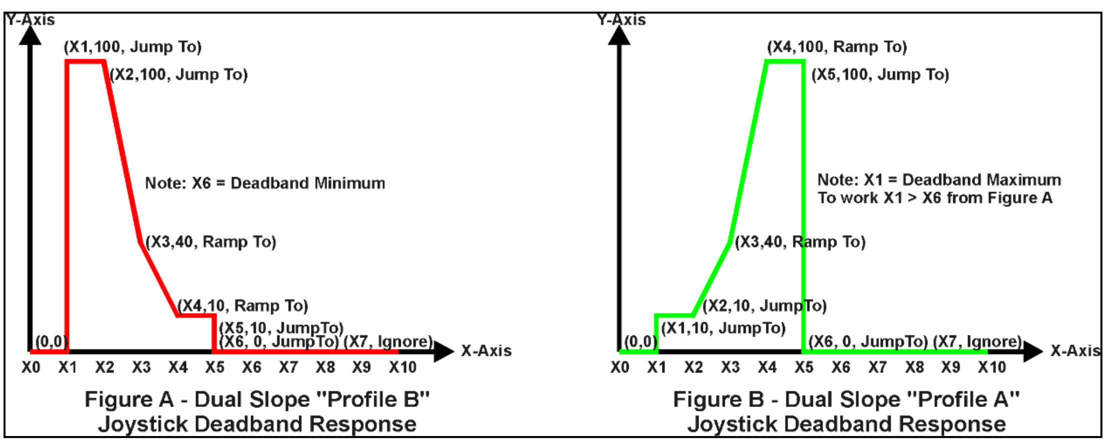


Figure 5: Lookup Table Examples to Setup for Dual-Slope Joystick Deadband Response

To summarize, Table 29 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 29: LTyz Point Response Options

2.6.4 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].

2.7 Programmable Logic Function Block

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 2.6.2, 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 6. 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.**

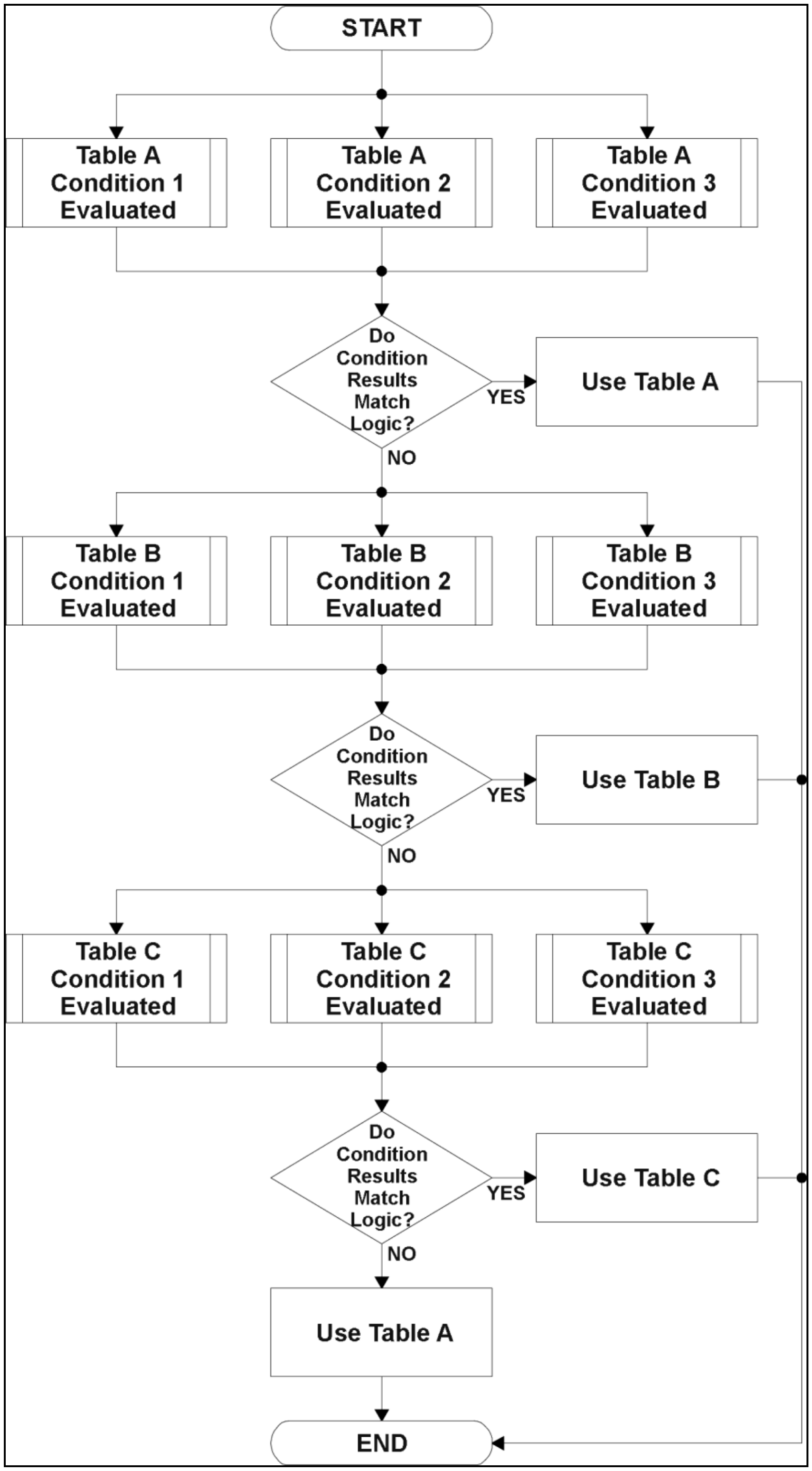


Figure 6: Logic Block Flowchart:

2.7.1 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 30.

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 30: 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 31, 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 31: 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 32.

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 32: LB(3-x) Condition Evaluation Results

2.7.2 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 2.6.1. There are several logical combinations that can be selected, as listed in Table 33. 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 33: 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 34.

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 34: 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.

2.7.3 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 35.

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 35: 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.

2.8 Math Function Block

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 25.

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 36.

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 36: 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.

2.9 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 7 demonstrates the connections between all parameters.

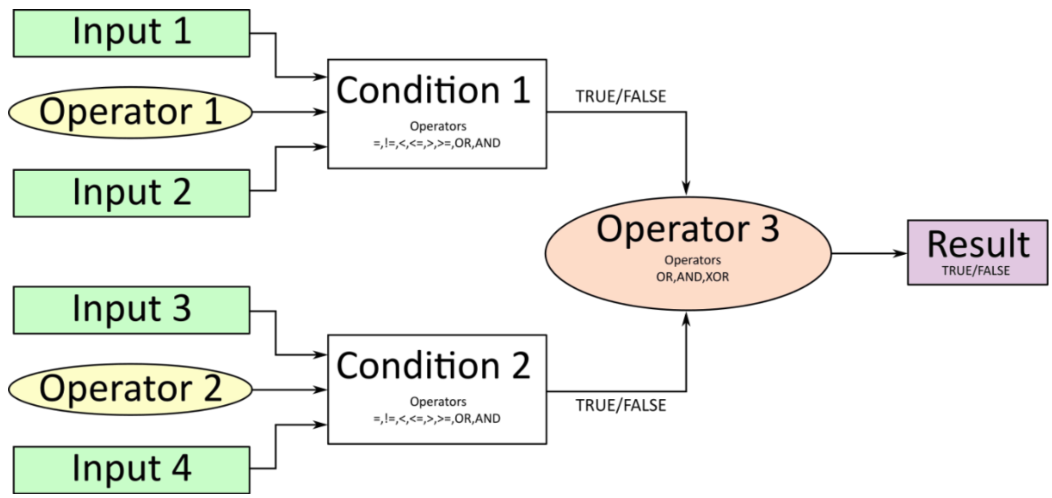


Figure 7: Conditional Block Diagram

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 37*. 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 37: 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 38.

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 38: 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.

2.10 LED Control Function Block

The motor controller has two red LEDs and one green LED mounted on the front panel. Table 39 describes their position on the front panel when the LEDs facing down.

Name	Position	Index
LED 1 (Red)	Right	XXX0h
LED 2 (Red)	Middle	XXX1h
Heartbeat (Green)	Left	XXX2h

Table 39: LED Positions on the front Panel

The LEDs are reflecting internal states in five different stages. Stage 1 has the highest priority and Default Stage has the lowest. While Table 40 shows the settings for stage 1 to 4, the Default stage has no Control Source or Number because its purpose is to indicate the motor controller runs without any issues as well as it informs the user if the unit is powered on.

Stage Settings
Control Source
Output Type
Response Type
Blink Rate

Table 40: LED Stages Settings

The stage with the highest priority, Stage 1, can be reached using sub-index 1 and the default value can be changed using sub-index 5.

2.10.1 LED Control Sources

Each LED can be controlled individually and their LED stages can be configured to be commanded by the LED control sources listed in Table 41. This table displays the control sources which can be selected with object 260X **LED Control Source** and object 261X **LED Control Number**.

Table 41: LED Control Sources

Value	Meaning
0	Control Source Not Used
1	Input Fault
2	Output Fault
3	Power Supply Fault
4	Temperature Fault

5	CAN Receive Timeout
6	Global Fault
7	PID Control
8	Lookup Table
9	Programmable Logic
10	Math Function
11	Constant Data
12	Output Feedback
13	Power Supply Measured
14	Temperature Measured

Setting the stage to **Control Source Not Used** has no affect to the LED. There are several fault-sources to indicate certain errors during runtime which can become 0(false) or 1(true). The Input Fault triggers when any of the Universal Inputs or Encoder Input shows an error. This also applies to the Output Fault in case any error is indicated on the Motor Drive. An overvoltage or undervoltage error can be shown with the source **Power Supply Fault**, while an overheating can be detected with the **Temperature Fault**. In case any of these errors should trigger an LED, the Global Fault source can be used for this purpose.

All other sources can have other values than true or false and therefore need a minimum limit and maximum limit. If the maximum limit is reached or exceeded, the LED will behave as configured for this stage.

2.10.2 LED Output/Response Type

The LED behavior can be configured by object 262Xh **LED Response Type**. **Error! Reference source not f ound.** shows the possible settings for this setpoint.

Table 42: LED Response Types

Value	Response Type
0	Normal OFF
1	Normal ON
2	Blinking Logic

The LED can be simple turned off or on in case the source value becomes true but also set to *Blinking Logic* to make the LED turn on and off in a certain frequency. How fast the toggle frequency is can be set by object 263X **LED Blink Rate**, which is declared in milliseconds.

3 Installation Instructions

The CANopen® object dictionary of the AX100311 Motor 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. For dimensions and connections, refer to Section 5.5.

3.1 Node ID and Baud Rate

By default, the AX100311 Motor controller ships factory programmed with a

Node ID = 127 (0x7F)

and with

Baud rate = 125 kbps.

3.1.1 LSS Protocol to Update

The only means by which the Node-ID and Baud rate 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

3.1.2 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)

3.1.3 Setting Baud rate

- 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 baud rate 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 43 – LSS Baud rate 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 baud rate):

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)

3.2 Communication Objects (DS-301)

The communication objects supported by the AX100311 Motor 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

3.2.1 PDO Parameters

All RPDOs and TPDOs in the Motor 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	0x0	
2	0x0	
3	0x0	
4	0x0	
5	0x0	
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	0x71000208	AI Input Field Value for AI 2
3	0x71000308	AI Input Field Value for AI 3
4	0x71000408	AI Input Field Value for AI 4

TPDO2: default ID 0x280 + node ID

Subindex	***Value***	***Object***
0	0	Number of mapped application objects in PDO
1	0x71000510	AI Input Field Value for AI 5
2	0x73000010	AO Output Field Value for AO 1
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	

3.2.2 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 0x00100311, indicating that the controller includes the following function blocks defined in the device profile.

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

Object Description

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

Entry Description

Access	RO
PDO Mapping	No
Value Range	0x00100311
Default Value	0x00100311

3.2.3 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

3.2.4 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

3.2.5 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)

3.2.6 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)

3.2.7 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

3.2.8 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

3.2.9 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

3.2.10 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	0 (Pre-Operational)

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

3.3 Manufacturer Objects

Index [hex]	Object	Object Type	Data Type	Access	PDO Mapping
2010	AI Input range	VAR	UNSIGNED8	RW	No
2011	AI Decimal Digits	VAR	UNSIGNED8	RW	No
2100	MO Ramp Up(Min to Max)	VAR	UNSIGNED16	RW	No
2101	MO Ramp Down(Max to Min)	VAR	UNSIGNED16	RW	No
2102	MO Error Minimum	VAR	UNSIGNED32	RW	No
2103	MO Error Maximum	VAR	UNSIGNED32	RW	No
2104	MO Control Source	VAR	UNSIGNED8	RW	No
2105	MO Control Number	VAR	UNSIGNED8	RW	No
2106	MO Enable Source	VAR	UNSIGNED8	RW	No
2107	MO Enable Number	VAR	UNSIGNED8	RW	No
2108	MO Enable Response	VAR	UNSIGNED8	RW	No
2109	MO Override Source	VAR	UNSIGNED8	RW	No
210A	MO Override Number	VAR	UNSIGNED8	RW	No
210B	MO Override Control Response	VAR	UNSIGNED8	RW	No
210C	MO Override Response Value	VAR	UNSIGNED32	RW	No
210D	MO Output Frequency	VAR	UNSIGNED16	RW	No
210E	MO Gain	VAR	UNSIGNED32	RW	No
210F	MO Integral	VAR	UNSIGNED32	RW	No
2110	MO Derivative	VAR	UNSIGNED32	RW	No
2111	MO Over Current Reset Timer	VAR	UNSIGNED32	RW	No
2112	MO Over Current Threshold	VAR	UNSIGNED32	RW	No
2113	MO Dead Time	VAR	UNSIGNED8	RW	No
2114	MO Error Detection Enable	VAR	UNSIGNED8	RW	No
2115	MO Clear Error Hysteresis	VAR	UNSIGNED16	RW	No
2116	MO Error React Delay	VAR	UNSIGNED16	RW	No
2117	MO Auto Update	VAR	UNSIGNED8	RW	No
2200	ENC Input Resolution	ARRAY	UNSIGNED8	RW	No
2201	ENC Stationary Delay	ARRAY	UNSIGNED16	RW	No
2202	ENC Debounce Filter	ARRAY	UNSIGNED8	RW	No
2203	ENC Error Detection Enable	ARRAY	UNSIGNED8	RW	No
2204	ENC Clear Error Hysteresis	ARRAY	UNSIGNED16	RW	No
2205	ENC Error React Delay	ARRAY	UNSIGNED16	RW	No
2206	ENC Auto Update	ARRAY	UNSIGNED8	RW	No
22X0	ENC A/B/Z Input Logic Type	ARRAY	UNSIGNED8	RW	No

22X1	ENC A/B/Z Input Active State	ARRAY	UNSIGNED8	RW	No
22X2	ENC A/B/Z Input Debounce Time	ARRAY	UNSIGNED16	RW	No
2234	ENC Number of Pulses per Revolution	ARRAY	UNSIGNED16	RW	No
2235	ENC Number to Reset Pulse Counter	ARRAY	UNSIGNED32	RW	No
2236	ENC Pulse Count Pulse Window	ARRAY	UNSIGNED32	RW	No
2237	ENC Pulse Count Time Window	ARRAY	UNSIGNED32	RW	No
2238	ENC Overflow Value	ARRAY	UNSIGNED32	RW	No
2300	UNI Input Logic Type				
2301	UNI Active State	ARRAY	UNSIGNED8	RW	No
2302	UNI Debounce Time	ARRAY	UNSIGNED16	RW	No
2303	UNI Debounce Filter	ARRAY	UNSIGNED8	RW	No
2304	UNI Overflow Value	ARRAY	FLOAT32	RW	No
2305	UNI Number of Pulses per Revolution	ARRAY	UNSIGNED16	RW	No
2306	UNI Number to Reset Pulse Counter	ARRAY	UNSIGNED32	RW	No
2307	UNI Pulse Count Pulse Window	ARRAY	FLOAT32	RW	No
2308	UNI Pulse Count Time Window	ARRAY	FLOAT32	RW	No
2309	UNI Pull Up/Down Mode	ARRAY	UNSIGNED8	RW	No
230A	UNI Auto Update	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
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
260X	LED Control Source	ARRAY	UNSIGNED8	RW	No
261X	LED Control Number	ARRAY	UNSIGNED8	RW	No
262X	LED Response	ARRAY	UNSIGNED8	RW	No
263X	LED Blink Rate	ARRAY	UNSIGNED16	RW	No
2700	EX Set Under Voltage	VAR	UNSIGNED32	RW	No
2701	EX Set Over Voltage	VAR	UNSIGNED32	RW	No
2702	EX Set Shut Down Temperature	VAR	UNSIGNED32	RW	No
2703	EX VPS Fault Disable Outputs	VAR	UNSIGNED32	RW	No
2704	EX Error React Enable	ARRAY	UNSIGNED8	RW	No
2705	EX Clear Hysteresis	ARRAY	UNSIGNED32	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
3900	PID Output Tolerance	ARRAY	INTEGER16	RW	No
3901	PID Integral Gain	ARRAY	INTEGER16	RW	No
3902	PID Derivative Gain	ARRAY	INTEGER16	RW	No
3903	PID Target Source	ARRAY	UNSIGNED8	RW	No
3904	PID Target Number	ARRAY	UNSIGNED8	RW	No
3905	PID Feedback Source	ARRAY	UNSIGNED8	RW	No
3906	PID Feedback Number	ARRAY	UNSIGNED8	RW	No
3907	PID Control Response	ARRAY	UNSIGNED8	RW	No
3910	PID Output FV	ARRAY	INTEGER16	RW	No
4000	Math Block Enable	ARRAY	BOOLEAN	RW	No
4021	Math Output Scaling 1 PV	ARRAY	INTEGER16	RW	No
4023	Math Output Scaling 2 PV	ARRAY	INTEGER16	RW	No
4030	Math Output Process Value	ARRAY	INTEGER16	RO	Yes
4032	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
5100	CND Logic Enabled	ARRAY	UNSIGNED8	RW	No
5101	CND Logic Logical Operator	ARRAY	UNSIGNED8	RW	No
5102	CND Output PV	ARRAY	UNSIGNED8	RW	No
5103	CND Decimal Digits	ARRAY	UNSIGNED8	RW	No
52X0	Logic X Condition 1 Argument 1	RECORD	UNSIGNED8	RW	No
52X1	Logic X Condition 1 Argument 2	RECORD	UNSIGNED8	RW	No
52X2	Logic X Condition 2 Argument 1	RECORD	UNSIGNED8	RW	No
52X3	Logic X Condition 2 Argument 2	RECORD	UNSIGNED8	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

3.3.1 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 **Error! Reference source not found.** and Table 23
Default Value	0

3.3.2 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 19**, Table 6, **Error! Reference source not found.**
Default Value Encoder	0
Default Value Universal Input	1
Default Value Digital Input	PWM-Mode: 1 Digital ON/OFF: 0

3.3.3 Object 2100h: MO Ramp Up (Min to Max)

Object Description

Index	2100h
Name	MO Ramp Up
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Ramp Up value for motor output
Access	RW
PDO Mapping	No
Value Range	0 to 65,000
Default Value	0

3.3.4 Object 2100h: MO Ramp Up (Min to Max)

Object Description

Index	2101h
Name	MO Ramp Down
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Ramp Down value for motor output
Access	RW
PDO Mapping	No
Value Range	0 to 65,000
Default Value	0

3.3.5 Object 2102h: MO Error Minimum

Object Description

Index	2102h
Name	MO Error Min

Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Error Minimum Value of Motor Output Setpoint
Access	RW
PDO Mapping	No
Value Range	Depending on Motor Output Type
Default Value	0

3.3.6 Object 2103h: MO Error Maximum

Object Description

Index	2103h
Name	MO Error Maximum
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RW
PDO Mapping	No
Value Range	Depending on Motor Output
Default Value	165

3.3.7 Object 2104h: MO Control Source

Object Description

Index	2104h
Name	MO Control Source
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Control Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	1

3.3.8 Object 2105h: MO Control Number

Object Description

Index	2105h
Name	MO Control Number
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Control Number
Access	RW
PDO Mapping	No
Value Range	Depends on Source selected. See Table 3
Default Value	1

3.3.9 Object 2106h: MO Enable Source

Object Description

Index	2106h
Name	MO Enable Source
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Enable Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0

3.3.10 Object 2107h: MO Enable Number

Object Description

Index	2107h
Name	MO Enable Number
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Enable Number
Access	RW
PDO Mapping	No

Value Range	Depends on Source selected. See Table 3
Default Value	0

3.3.11 Object 2108h: MO Enable Response

Object Description

Index	2108h
Name	MO Enable Response
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Enable Response
Access	RW
PDO Mapping	No
Value Range	Table 4
Default Value	2(ENABLE WHEN OFF ELSE SHUTOFF)

3.3.12 Object 2109h: MO Override Source

Object Description

Index	2104h
Name	MO Override Source
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Override Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0

3.3.13 Object 210Ah: MO Override Number

Table 5

Object Description

Index	2105h
Name	MO Override Number
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Control Number
Access	RW
PDO Mapping	No
Value Range	Depends on Source selected. See Table 3
Default Value	1

3.3.14 Object 210Bh: MO Override Control Response

Object Description

Index	210Bh
Name	MO Override Control Response
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Control Response
Access	RW
PDO Mapping	No
Value Range	Table 5
Default Value	0 (Override When On)

3.3.15 Object 210Ch: MO Override Response Value

If the Override Source becomes active, this value will be set on the Motor Output.

Object Description

Index	210Ch
Name	Motor Override Response Value
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Motor Override Response Value
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

3.3.16 Object 210Dh: MO Output Frequency

Object Description

Index	210Dh
Name	Motor Output Frequency

Object Type	VAR
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	PWM Output Frequency
Access	RW
PDO Mapping	No
Value Range	4,000 to 25,000 (Hz)
Default Value	18,000 (Hz)

3.3.17 Object 210Eh: MO Gain

If Motor Output Type is set to 'Proportional Current', This parameter represents the gain of the PID-Control.

Object Description

Index	210Eh
Name	MO Gain
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Gain of PID-Control
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	1

3.3.18 Object 210Fh: MO Integral

If Motor Output Type is set to 'Proportional Current', This parameter represents the Integral Time in ms of the PID-Control.

Object Description

Index	210Fh
Name	MO Gain
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Integral of PID-Control
Access	RW

PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	10 [ms]

3.3.19 Object 2110h: MO Derivative

If Motor Output Type is set to 'Proportional Current', This parameter represents the Integral Time in ms of the PID-Control.

Object Description

Index	2110h
Name	MO Derivative
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Derivative of PID-Control
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	10 [ms]

3.3.20 Object 2111h: MO Over Current Reset Timer

This object is to reset the Current Shunt Monitor after an overcurrent on the output occurred.

Object Description

Index	2111h
Name	MO Over Current Reset Timer
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Motor Output over current reset timer
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	2000 [ms]

3.3.21 Object 2113h: MO Dead Time

The 'Dead Time' decides for how long both MOSFETs of the low side and high side stay off before switching one of them.

Object Description

Index	2113h
Name	MO Dead Time
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output dead time. Please refer to STM32F405 User Manual or contact Axiomatic for help calculation the dead time.
Access	RW
PDO Mapping	No
Value Range	0x1C to 0x3C
Default Value	0x1C

3.3.22 Object 2114h: MO 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	2114h
Name	MO Error Detect Enable
Object Type	VAR
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Description	Motor Output Error Detection Enabled
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	1

3.3.23 Object 2115h: MO 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 Output FV.

Object Description

Index	2115h
Name	MO Error Clear Hysteresis
Object Type	VAR
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Motor Output Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	0h to FFFFh
Default Value	2

3.3.24 Object 2116h: MO 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	2116h
Name	MO Error Reaction Delay
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

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

3.3.25 Object 2117h: MO Auto Update

Object Description

Index	2117h
Name	MO Auto Update
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Motor Output Auto Update
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	1

3.3.26 Object 2201h: 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	2201h
Name	ENC Stationary Delay
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Time until direction becomes state Stationary
Access	RW
PDO Mapping	No
Value Range	0x0 to 0xFFFF
Default Value	100 [ms]

3.3.27 Object 2202h: ENC Debounce Filter

This object will debounce the input signal applied on a single input. The options for this object are listed in **Error! Reference source not found.**.

Object Description

Index	2202h
Name	ENC Debounce Filter
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Encoder Input Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See **Error! Reference source not found.**
Default Value	2

3.3.28 Object 2203h: 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	2203h
Name	ENC Error Detect Enable
Object Type	VAR
Data Type	BOOLEAN

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

3.3.29 Object 2204h: 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	2204h
Name	ENC Error Clear Hysteresis

Object Type	VAR
Data Type	INTEGER16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RW
PDO Mapping	No
Value Range	Depends on Input Type Selected
Default Value	1

3.3.30 Object 2205h: 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	2205h
Name	ENC Error Reaction Delay
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

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

3.3.31 Object 2206h: ENC Auto Update

This object allows the controller to update objects to defaults automatically when an Encoder Input type is changed. Thee default of this object is set to TRUE.

Object Description

Index	2206h
Name	ENC 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]

3.3.32 Object 22X0h: ENC X Input Logic Type

This object allows flexibility in the response of the input. Table 14 demonstrates all possible options for this parameter.

Object Description

Index	22X0h
Name	ENC X Input Logic Type
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Encoder X Input Logic Type
Access	RW
PDO Mapping	No
Value Range	See Table 14
Default Value	0

3.3.33 Object 22X1h: ENC 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	23X1h
Name	ENC X Active State
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Encoder Input Active State
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	0

3.3.34 Object 22X2h: ENC X Debounce Time

The Input Debounce Time parameter is a useful parameter in cases where the digital input signal coming into the controller is noisy. Please refer to 2.2.5 for more information.

Object Description

Index	22X2h
Name	ENC X Debounce Time
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

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

3.3.35 Object 2233h: ENC Z Debounce Filter

This object will debounce the input signal applied on a single input. The options for this object are listed in Table 18.

Object Description

Index	2233h
Name	ENC Z Debounce Filter
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Encoder Z Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 18**Error! Reference source not found.**
Default Value	0 [Debounce Filter OFF]

3.3.36 Object 2234h: ENC 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 7120h, 7122h, 7148h and 7149h will be interpreted as RPM data.

Object Description

Index	2234h
Name	ENC Number of Pulses Per Revolution
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Encoder Input Z Pulses per Revolution
Access	RW
PDO Mapping	No
Value Range	0x0 to 0xFFFFFFFF
Default Value	0

3.3.37 Object 2235h: ENC 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 **ENC Pulse Count to Reset Counter** value of object 2235h is reached at which the counter will reset back to 0.

Object Description

Index	2235h
Name	ENC Number to Reset Pulse Counter
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

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

3.3.38 Object 2236h: ENC 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	2236h
Name	ENC Pulse Counter Pulse Window
Object Type	VAR

Data Type	UNSIGNED32

Entry Description

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

3.3.39 Object 2237h: ENC Pulse Count Time Window

This object is a Counter input type in which object **ENC Pulse Count in Time Window** 2237h is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Object Description

Index	2237h
Name	ENC Pulse Counter Pulse Window
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

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

3.3.40 Object 2238h: ENC Overflow Value

Object Description

Index	2238h
Name	ENC Overflow Value
Object Type	VAR
Data Type	UNSIGEND32

Entry Description

Sub-Index	0h
Description	Encoder Input Z Overflow Value
Access	RW
PDO Mapping	No
Value Range	1 to 50
Default Value	1

3.3.41 Object 2300h: UNI Input Logic Type

This object allows flexibility in the response of the input. Table 14 demonstrates all possible options for this parameter.

Object Description

Index	2300h
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	2
Default Value	2

Sub-Index	1h to 2h
Description	Input Logic Type
Access	RW
PDO Mapping	No
Value Range	See Table 14
Default Value	0

3.3.42 Object 2301h: 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	2301h
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	2
Default Value	2

Sub-Index	1h to 2h
Description	Universal Input Active State
Access	RW
PDO Mapping	No
Value Range	0(Active High) to 1(Active Low)
Default Value	0(Active High)

3.3.43 Object 2302h: 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 2.2.5 for more information.

Object Description

Index	2302h
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	2
Default Value	2

Sub-Index	1h to 2h
Description	Universal Input Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 to 65,535
Default Value	250

3.3.44 Object 2303h: 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 18.

Object Description

Index	2203h
Name	UNI Debounce Filter 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 to 2h
Description	Universal Input Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 18
Default Value	0 [Debounce Filter OFF]

3.3.45 Object 2304h: UNI Overflow Value

Object Description

Index	2204h
Name	UNI Overflow Value
Object Type	ARRAY
Data Type	UNSIGNED32

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 2h
Description	Universal Input Overflow Value
Access	RW
PDO Mapping	No
Value Range	1 to 50
Default Value	1

3.3.46 Object 2305h: 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 2205h, 7120h, 7122h, 7148h and 7149h will be interpreted as RPM data.

Object Description

Index	2305h

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	2
Default Value	2

Sub-Index	1h to 2h
Description	Universal Input 1 Pulses per Revolution
Access	RW
PDO Mapping	No
Value Range	0 to 1000
Default Value	0

3.3.47 Object 2306h: 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 2206h is reached at which the counter will reset back to 0.

Object Description

Index	2306h
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	2
Default Value	2

Sub-Index	1h to 2h
Description	Universal Input Number to Reset Pulse Counter
Access	RW

PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

3.3.48 Object 2307h: 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	2307h
Name	UNI Pulse Counter Pulse Window
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 Pulse Count Pulse Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

3.3.49 Object 2308h: UNI Pulse Count Time Window

This object is a Counter input type in which object **UNI Pulse Count in Time Window** 2308h is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Object Description

Index	2308h
Name	UNI Pulse Counter Pulse Window
Object Type	ARRAY
Data Type	UNSIGEND32

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 2h
Description	Universal Input Pulse Count Time Window
Access	RW
PDO Mapping	No
Value Range	0h to FFFFFFFFh
Default Value	0

3.3.50 Object 2309h: 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 24, and the controller will adjust the input hardware according to what is specified.

Object Description

Index	2208h
Name	UNI 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 to 2h
Description	Universal Input Pullup/Down
Access	RW
PDO Mapping	No
Value Range	See Table 24
Default Value	0 (pullup/down disabled)

3.3.51 Object 220Ah: UNI Auto Update

This object allows flexibility in the response of the input. Table 14 demonstrates all possible options for this parameter.

Object Description

Index	220Ah
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 to 2h
Description	Input Logic Type
Access	RW
PDO Mapping	No
Value Range	See Table 14
Default Value	0

3.3.52 Object 220Bh: 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	220Bh
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 to 2h
Description	Universal Input 1 Error Detect Enable
Access	RW

PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

3.3.53 Object 220Ch: 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	220Ch
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	2
Default Value	2

Sub-Index	1h
Description	Universal Input 1 Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	Depends on Input Type selected**Error! Reference source not found.**
Default Value	10 [1%]

3.3.54 Object 220Dh: 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	220Dh

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	2
Default Value	2

Sub-Index	1h to 2h
Description	UNI Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

3.3.55 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

3.3.56 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)

3.3.57 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

3.3.58 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)

3.3.59 Object 260X: LED Control Source

This object defines the type of input that will be used to determine the LED input value for the LED indication. The available control sources on this controller are listed in Table 3. A selection of "Control Source Not Used" disables the associated LED Stage.

Object Description

Index	260Xh
Name	LED 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	5
Default Value	5

Sub-Index	1h to 5h
Access	RW
PDO Mapping	No
Value Range	See Table 41
Default Value	11 (LED Constant Data)

3.3.60 Object 261X: LED Control Number

Object Description

Index	261Xh
Name	LED 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	5

Default Value	5

Sub-Index	1h to 5h
Access	RW
PDO Mapping	No
Value Range	Depending on Control Source selected
Default Value	0 (LED Constant Data 1)

3.3.61 Object 262X: LED Response

This parameter can be used to set the behaviour of the LED in case the selected Control Source is been triggered.

Object Description

Index	262Xh
Name	LED Response
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
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	2 (LED Blinking)

3.3.62 Object 263X: LED Blink Rate

This parameter can be used to set the frequency of blinking in case object 262X is configured to 'LED Blinking'.

Object Description

Index	263Xh
Name	LED Blink Rate
Object Type	ARRAY
Data Type	UNSIGNED16

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
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 ms

3.3.63 Object 2700: EX Under Voltage

This object sets the minimum value for the VPS before triggering an Error in case Error detection is enabled.

Object Description

Index	2700h
Name	EX Under Voltage
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Under Voltage Value
Access	RW
PDO Mapping	No
Value Range	9V to 36V
Default Value	9 [V]

3.3.64 Object 2701: EX Over Voltage

This object sets the maximum value for the VPS before triggering an Error in case Error detection is enabled.

Object Description

Index	2701h
Name	EX Maximum Voltage
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Over Voltage Value
Access	RW
PDO Mapping	No
Value Range	9V to 36V
Default Value	36 [V]

3.3.65 Object 2702: EX Shut Down Temperature

This object sets the maximum value for the Temperature measured on the Microcontroller before triggering an Error in case Error detection is enabled.

Object Description

Index	2702h
Name	EX Shut Down Temperature
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Maximum Temperature Value
Access	RW
PDO Mapping	No
Value Range	9V to 36V
Default Value	36 [V]

3.3.66 Object 2703: EX Disable Output

In case a voltage supply error occurs during runtime and object 2704h is set to true, the Output will be set to 0 as soon as the Voltage error is been triggered.

Object Description

Index	2703h
Name	EX Disable Outputs
Object Type	VAR
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Disables the output when a VPS fault is triggered.
Access	RW

PDO Mapping	No
Value Range	0 to 1
Default Value	1 [TRUE]

3.3.67 Object 2704: EX Error React Enable

Object Description

Index	2704h
Name	EX Error React Enable
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
Description	Enable Voltage Fault
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	True (1)

Sub-Index	2h
Description	Enable Temperature Fault
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	True (1)

3.3.68 Object 2705: EX Error React Enable

Object Description

Index	2705h
Name	EX Maximum Voltage
Object Type	ARRAY
Data Type	UNSIGNED16

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	Enable Voltage Fault
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	True (1)

Sub-Index	2h
Description	Enable Temperature Fault
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	True (1)

3.3.69 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 25 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)

3.3.70 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 25. Once selected, the limits for the points on the X-Axis will be constrained by the scaling objects of the control source/number.

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 25
Default Value	0 (null control source)

3.3.71 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]

3.3.72 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 25.

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
Default Value	0

3.3.73 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

3.3.74 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 29. See Figure 4 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 29 (0, 1 or 2)
Default Value	1 (ramp to response)

3.3.75 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.

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

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

3.3.76 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]

3.3.77 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

3.3.78 Object 3300h: Logic Block Enable

This object defines whether or not the logic shown in Figure 6 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]

3.3.79 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 6 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

3.3.80 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

3.3.81 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

3.3.82 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 31. 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 31

Default Value	Function A = 1 (and all) Function B = 1 (and all) Function C = 0 (default)

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

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

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

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

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

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

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

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

3.3.91 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 25
Default Value	1 (CANopen® Message)

Sub-Index	2h
Description	Argument 1 Number
Access	RW
PDO Mapping	No
Value Range	See Table 26
Default Value	11 (EC Received PV 1)

Sub-Index	3h
Description	Argument 2 Source
Access	RW
PDO Mapping	No
Value Range	See Table 25
Default Value	5 (Constant PV)

Sub-Index	4h
Description	Argument 2 Number
Access	RW
PDO Mapping	No
Value Range	See Table 26
Default Value	3 (Constant FV 3)

Sub-Index	5h
Description	Operator
Access	RW
PDO Mapping	No
Value Range	See Table 33
Default Value	0 (Equals)

3.3.92 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]

3.3.93 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 4032h 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

3.3.94 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 4032h 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)

3.3.95 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

3.3.96 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)

3.3.97 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 25.

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 25
Default Value	0 (control source not used)

3.3.98 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 26. 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 26
Default Value	0 (null input)

3.3.99 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)

3.3.100 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)

3.3.101 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

3.3.102 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%)

3.3.103 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)

3.3.104 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 36.

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)

3.3.105 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

3.3.106 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]

3.3.107 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]

3.3.108 Object 55AAh: Start Bootloader

In case a firmware update is needed, this object resets the unit and starts the bootloader which can then be used to download new firmware.

Object Description

Index	55AAh
Name	Switch to Bootloader Mode
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]

3.3.109 Object 5B50h: Change Baud Rate

The baud rate of the unit can be changed with this object.

Object Description

Index	5B50h
Name	Change Baud Rate
Object Type	VARIABLE
Data Type	UNSIGEND8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0: 1000 kbit/s 1: 800 kbit/s 2: 500 kbit/s 3: 250 kbit/s 4: 125 kbit/s 5: 100 kbit/s 6: 50 kbit/s 7: 20 kbit/s

	8: 10 kbit/s
Default Value	4 (125 kbit/s)

3.3.110 Object 5B51h: Change Node ID

Object Description

Index	5B51h
Name	Change Node ID
Object Type	VARIABLE
Data Type	BOOLEAN

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0x00 to 0x7F
Default	0x7F

3.4 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
7450	PID Proportional Band	ARRAY	INTEGER16	RW	No
7452	PID Integral Time	ARRAY	INTEGER16	RW	No
7454	PID Derivative Time	ARRAY	INTEGER16	RW	No
7456	PID Cycle Time	ARRAY	INTEGER16	RW	No
6458	PID Time Unit	ARRAY	UNSIGNED32	RW	No
6459	PID Time Decimal Digits	ARRAY	UNSIGNED8	RW	No
6302	AO Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
6310	AO Output Type	ARRAY	UNSIGNED8	RW	No
7300	AO Output Process Value	ARRAY	INTEGER16	RW	Yes
7320	AO Output Scaling 1 PV	ARRAY	INTEGER16	RW	No

7321	AO Output Scaling 1 FV	ARRAY	INTEGER16	RW	No
7322	AO Output Scaling 2 PV	ARRAY	INTEGER16	RW	No
7323	AO Output Scaling 2 FV	ARRAY	INTEGER16	RW	No
7330	AO Output Field Value	ARRAY	INTEGER16	RO	Yes
6340	AO Fault Mode	ARRAY	UNSIGNED16	RW	No
7341	AO Fault FV	ARRAY	INTEGER16	RW	No

3.4.1 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 2.2.7 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

3.4.2 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 17
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 17
Default Value	0 (Normal On/Off)

3.4.3 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 19, Table 6**Error! Reference source not found.**, 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	5
Default Value	5

Sub-Index	1h to 5h
Description	Analog Input X FV
Access	RO
PDO Mapping	Yes
Value Range	Data Type Specific
Default Value	No

3.4.4 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Sensor Type
Access	RW
PDO Mapping	No
Value Range	
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)

3.4.5 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Operating Mode
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	1 (normal operation)

3.4.6 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	5
Default Value	5

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

3.4.7 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	same as 7120h

3.4.8 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Scaling 2 FV
Access	RW

PDO Mapping	No
Value Range	See Table 27
Default Value	Encoder Input 1-3: 2 Universal Input: 4500 Digital Input 1,2,4,5: 9000 Digital Input 3 and 6: 1

3.4.9 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	same as 7122h

3.4.10 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Process Value
Access	RO
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

3.4.11 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	5
Default Value	5

Sub-Index	1h to 5h
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

3.4.12 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Span Start (Error Min)
Access	RW
PDO Mapping	No
Value Range	See **Error! Reference source not found.**
Default Value	Encoder Input 1-3: 0 Universal Input: 200 Digital Input 1,2,4,5: 10 Digital Input 3 and 6: 0

3.4.13 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	5

Default Value	5

Sub-Index	1h to 5h
Description	AIx Span End (Error Max)
Access	RW
PDO Mapping	No
Value Range	See **Error! Reference source not found.**
Default Value	Encoder Input 1-3: 2 Universal Input: 4800 Digital Input 1,2,4,5: 10000 Digital Input 3 and 6: 1

3.4.14 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Input Offset
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

3.4.15 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Autozero
Access	WO
PDO Mapping	No
Value Range	0x6F72657A
Default Value	No

3.4.16 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 **Error! Reference source not found.**, 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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Filter Type
Access	RW
PDO Mapping	No
Value Range	See **Error! Reference source not found.**
Default Value	0 (no filter)

3.4.17 Object 61A1h: AI Filter Constant

This object defines the number of steps used in the various filters, as defined in **Error! Reference source n ot found.**

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	5
Default Value	5

Sub-Index	1h to 5h
Description	AIx Filter Constant
Access	RW
PDO Mapping	No
Value Range	1 to 1000
Default Value	10

3.4.18 Object 7450h: PID Proportional Band

Object Description

Index	7450h
Name	PID Proportional Band
Object Type	ARRAY
Data Type	INTEGER16

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 2h
Description	Additional PID controller P gain
Access	RW
PDO Mapping	No
Value Range	1 to 100
Default Value	1

3.4.19 Object 7452h: PID Integral Time

Object Description

Index	7452h
Name	PID Integral Time
Object Type	ARRAY
Data Type	INTEGER16

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 2h
Description	Additional PID controller integral action time
Access	RW
PDO Mapping	No
Value Range	1 to 1000
Default Value	5

3.4.20 Object 7454h: PID Derivative Action Time

Object Description

Index	7454h
Name	PID Derivative Time
Object Type	ARRAY
Data Type	INTEGER16

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 2h
Description	Additional PID controller derivative action time
Access	RW
PDO Mapping	No
Value Range	1 to 1000
Default Value	1

3.4.21 Object 7456h: PID Cycle Time

Object Description

Index	7456h
Name	PID Cycle Time
Object Type	ARRAY
Data Type	INTEGER16

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 2h
Description	Additional PID controller cycle time
Access	RW
PDO Mapping	No
Value Range	1 to 1000
Default Value	10

3.4.22 Object 6458h: PID Physical Unit Timing

Object Description

Index	6458h
Name	PID Physical Unit Timing
Object Type	ARRAY
Data Type	UNSIGNED32

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 2h
Description	Additional PID controller cycle time
Access	RW
PDO Mapping	No
Value Range	12288
Default Value	12288

3.4.23 Object 6459h: PID Decimal Digits Timing

Object Description

Index	6459h
Name	PID Decimal Digits Timing
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 to 2h
Description	Additional PID controller cycle time
Access	RW
PDO Mapping	No
Value Range	0 - 4
Default Value	3

3.4.24 Object 7300h: AO Output Process Value

This object is the process value that is fed into the analog output function block. The process value can be in any physical unit (bar, rpm, etc.) or in any custom format. The output value is scaled into a field value in object 7330h using the scaling coefficients defined in objects 7320h to 7323h.

Object Description

Index	7300h
Name	AO Output Process Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	0h
Description	Process Value
Access	RW
PDO Mapping	Yes
Value Range	INTEGER16
Default Value	0

3.4.25 Object 6302h: AO Decimal Digits PV

This object sets the number of decimal digits included in the process value.

Object Description

Index	6302h

Name	AO Decimal Digits PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Subindex	0h
Description	Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 9
Default Value	0

3.4.26 Object 6310h: AO Output Type

This object defines the type of the output. The following values are allowed for this object.

- Output Type 0 = voltage output
- Output Type 1 = current output

Only the mentioned output types are supported. Writing a value other than those listed above will result in SDO abort download error, and the previous value will remain unchanged. The default value is 0, voltage output.

Object Description

Index	6310h
Name	AO Output Type
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Subindex	1h
Description	Output Type, OUT1
Access	RW
PDO Mapping	No
Value Range	0, 1
Default Value	0 (voltage)

3.4.27 Object 7320h: AO Output Scaling 1 PV

This object defines the process value for the first calibration point for the output channel. The value is scaled in the physical unit of the output process value, object 7300h.

Object Description

Index	7320h
Name	AO Output Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	1h
Description	Scaling 1 PV, CAN input 1
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

3.4.28 Object 7321h: AO Output Scaling 1 FV

This object defines the field value for the first calibration point for the output channel. The value is scaled in the physical unit of the output field value, object 7330h.

Object Description

Index	7321h
Name	AO Output Scaling 1 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	1h
Description	Scaling 1 FV, CAN input 1
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

3.4.29 Object 7322h: AO Output Scaling 2 PV

This object defines the process value for the second calibration point for the output channel. The value is scaled in the physical unit of the output process value, object 7300h.

Object Description

Index	7322h
Name	AO Output Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	1h
Description	Scaling 2 PV, CAN input 1
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	1000

3.4.30 Object 7323h: AO Output Scaling 2 FV

This object defines the field value for the second calibration point for the output channel. The value is scaled in the physical unit of the output field value, object 7330h.

Object Description

Index	7323h
Name	AO Output Scaling 2 FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	1h
Description	Scaling 2 FV, CAN input 1
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	5000

3.4.31 Object 7330h: AO Output Field Value

This object defines the field value of the proportional outputs. The FV is automatically updated if object 7300h [AO Output Process Value] is changed, either by a direct write to object 7300h or by using objects 7F50h and 6F52h. The value is defined as a multiple of 1mA / 1mV. The value range of this object depends on the selected output type.

Object Description

Index	7330h
Name	AO Output Field Value
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

Subindex	1h
Description	Field Value, CAN input 1
Access	RO
PDO Mapping	Yes
Value Range	-20000 to 20000 [uA], -10000 to 10000 [mV]
Default Value	0

3.4.32 Object 6340h: AO Fault Mode

This object defines the fault mode response for proportional outputs. It determines whether the output shall continue to operate normally when the 4 Analog Output CAN Controller detects an error, or if the output shall be driven to the value defined in object 7341h. Note, that these subindices map directly into the corresponding Analog Output Blocks (1 to 4).

- Value 0 = output continues to operate normally in controller fault state
- Value 1 = write the value defined in object 7341h in object 7330h, in controller fault state

Object Description

Index	6340h
Name	AO Fault Mode
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Subindex	1h
Description	Fault Mode, Output 1
Access	RW
PDO Mapping	No
Value Range	0, 1
Default Value	0

3.4.33 Object 7341h: AO Fault FV

This object defines the value that proportional outputs will be driven to in fault situations. Value range of this object depends on the selected output type. Note, that these subindices map directly into the corresponding Analog Output Blocks (1 to 4).

Object Description

Index	7341h
Name	AO Fault FV
Object Type	ARRAY
Data Type	INTEGER16

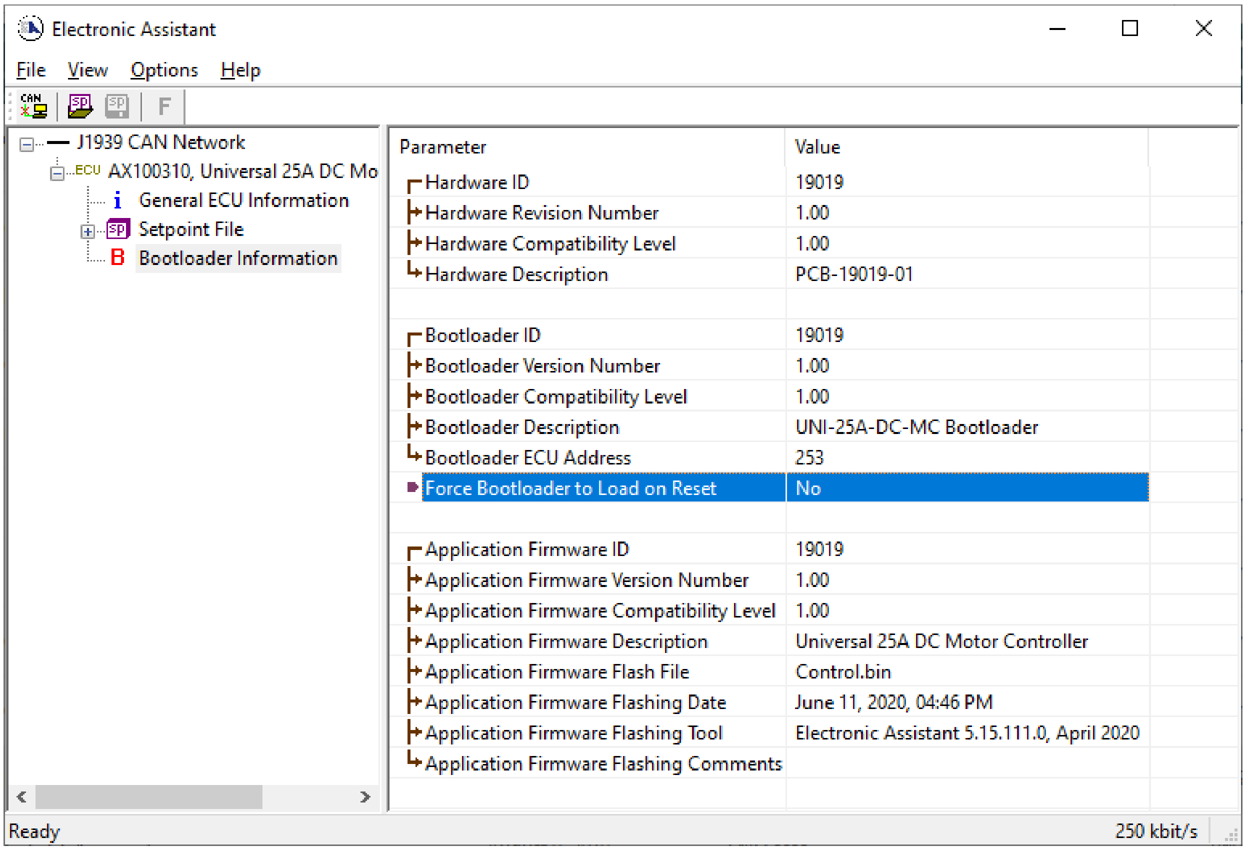
Entry Description

Subindex	0h
Description	Fault FV, OUT1
Access	RW
PDO Mapping	No
Value Range	-20000 to 20000 [uA], to [mV]
Default Value	0

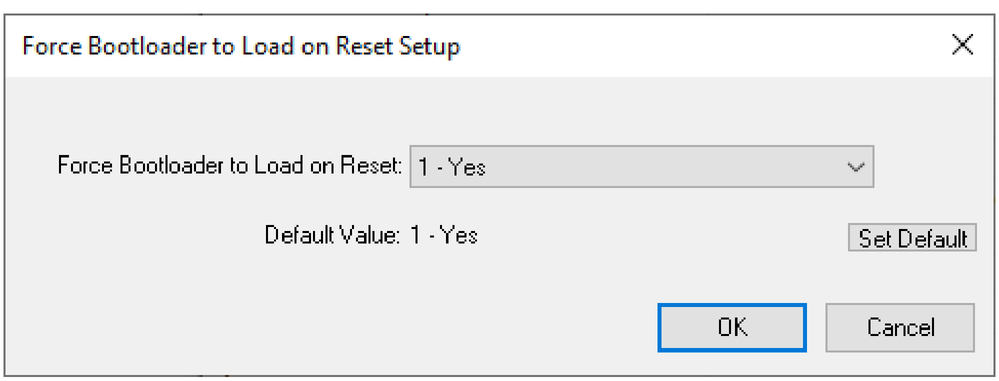
4 REFLASHING OVER CAN WITH EA BOOTLOADER

The AX100311 can be upgraded with new application firmware using the **Bootloader Information** section. This section details the simple step-by-step instructions to upload new firmware provided by Axiomatic onto the unit via CAN, without requiring it to be disconnected from the J1939 network.

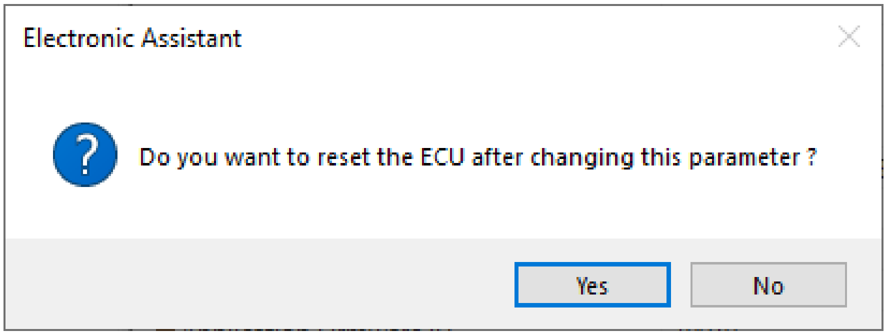
1. When EA first connects to the ECU, the **Bootloader Information** section will display the following information.



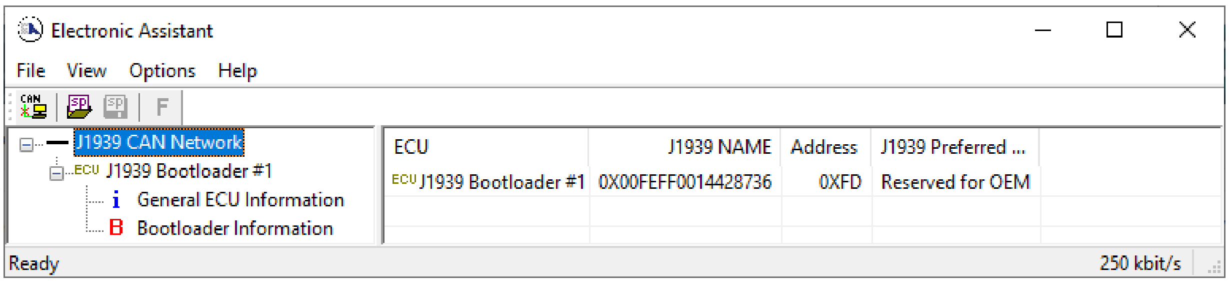
2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **"Force Bootloader To Load on Reset**" to *Yes.*



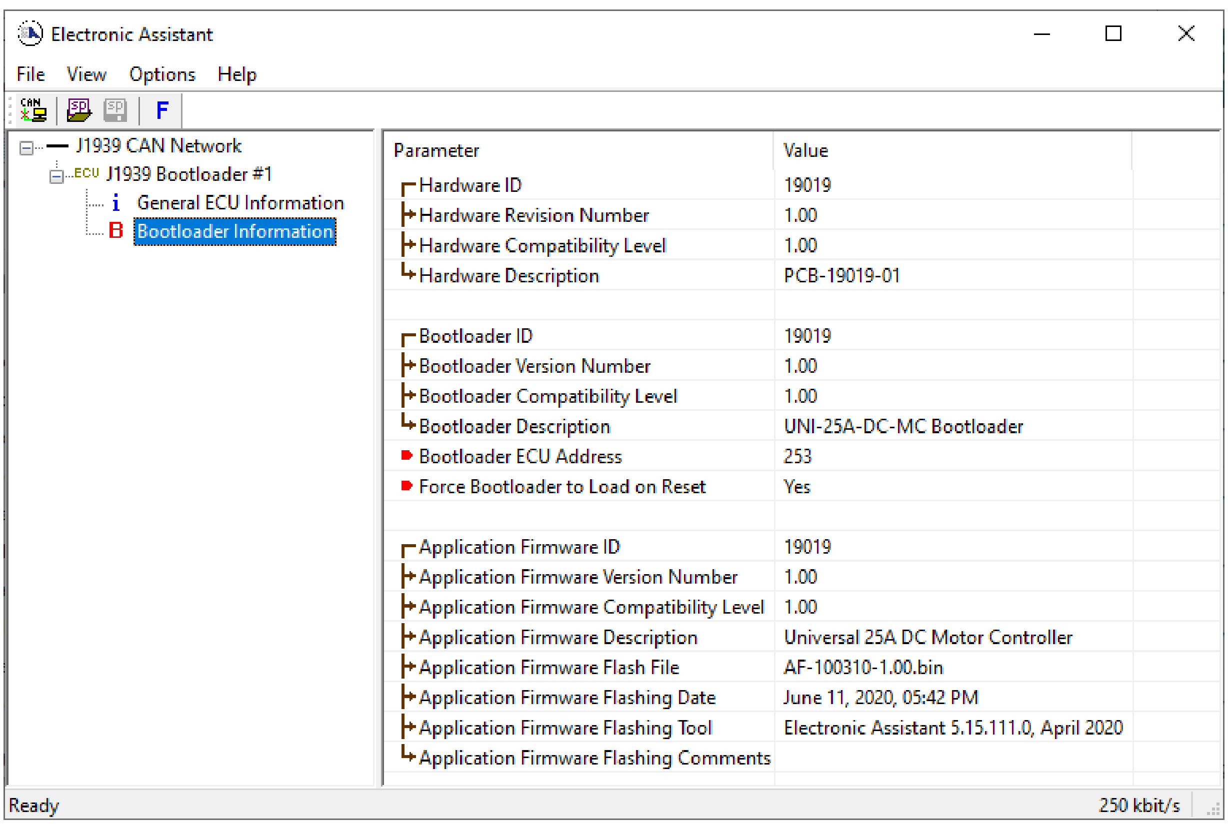
3. When the prompt box asks if you want to reset the ECU, select *Yes*.



4. Upon reset, the ECU will no longer show up on the J1939 network as an AX100310 but rather as **J1939 Bootloader #1**.

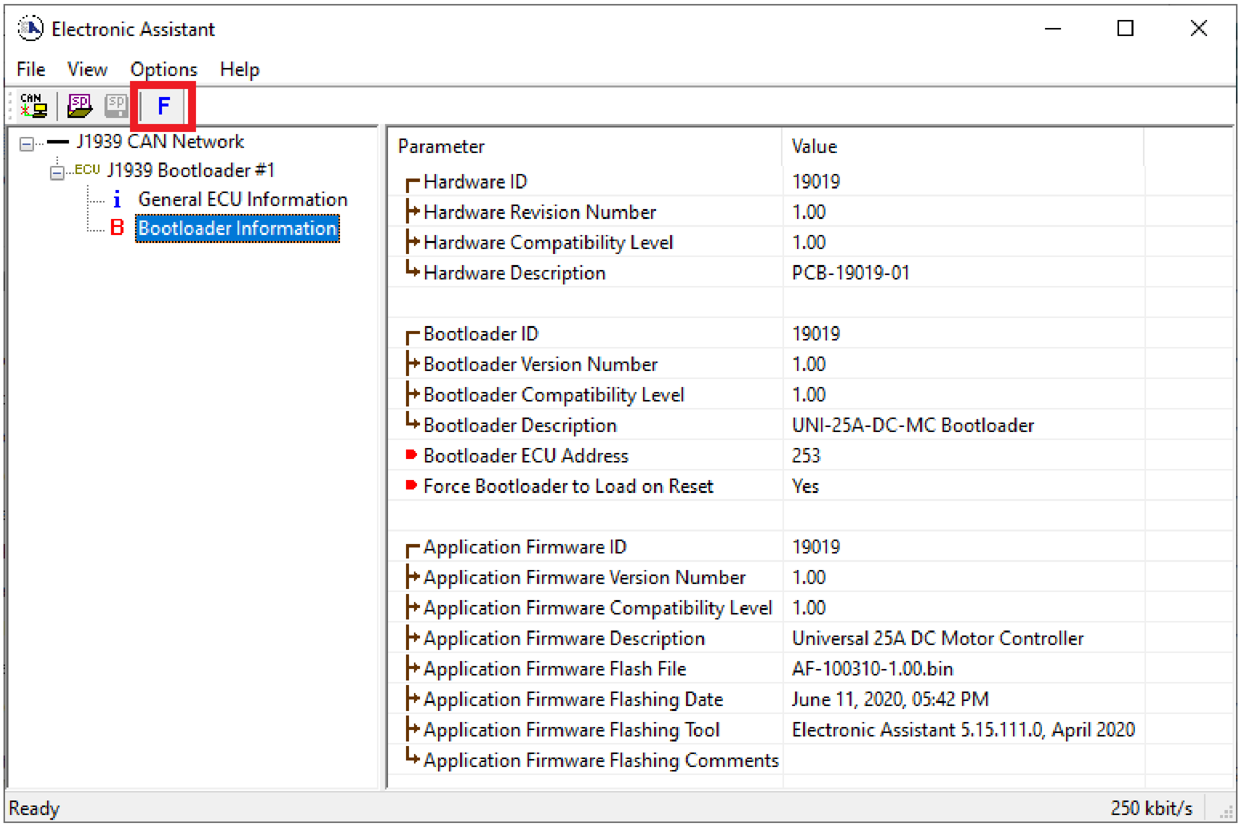


Note: It is recommended to **Close the CAN Port** after the unit was set to Bootloader Mode and afterwards, to **Open the CAN Port again**.



Note that the bootloader is NOT Arbitrary Address Capable. This means that if you want to have multiple bootloaders running simultaneously (not recommended) you would have to manually change the address for each one before activating the next, or there will be address conflicts, and only one ECU would show up as the bootloader. Once the 'active' bootloader returns to regular functionality, the other ECU(s) would have to be power cycled to re-activate the bootloader feature.

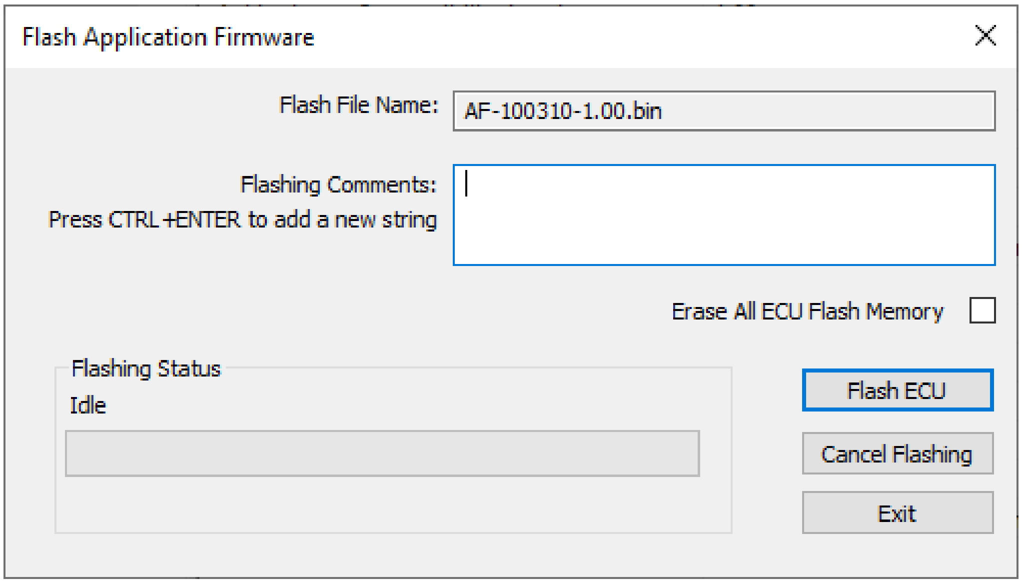
5. When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX100310 firmware, but in this case the **F**lashing feature has been enabled.



6. Select the **F**lashing button and navigate to where you had saved the **AF-100310-x.yy.bin** file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the EA tool)

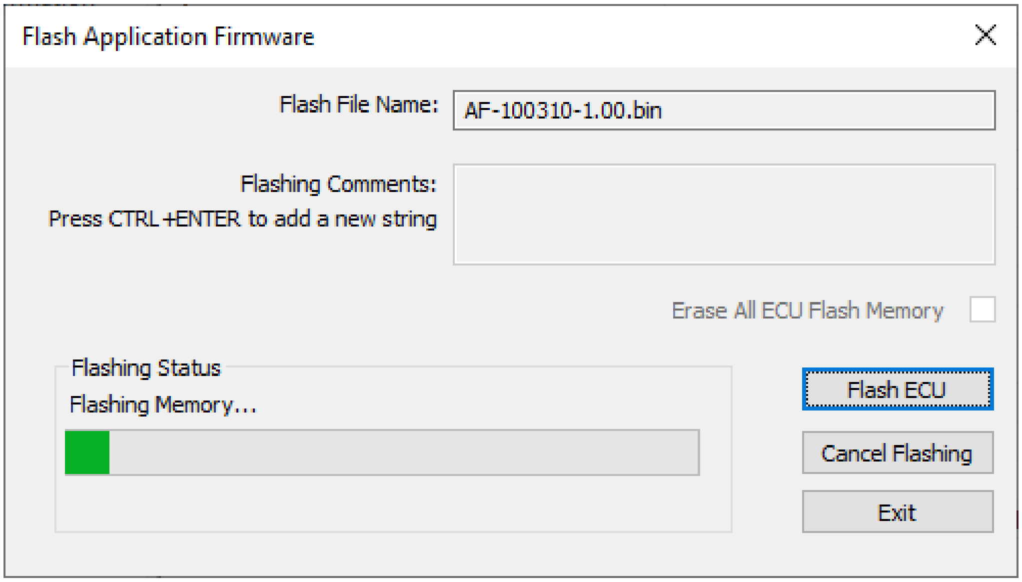
7. Once the Flash Application Firmware window opens, you can enter comments such as "Firmware upgraded by [Name]" if you so desire. This is not required, and you can leave the field blank if you do not want to use it.

   Note: You do not have to date/time-stamp the file, as this is done automatically by the EA tool when you upload the new firmware.

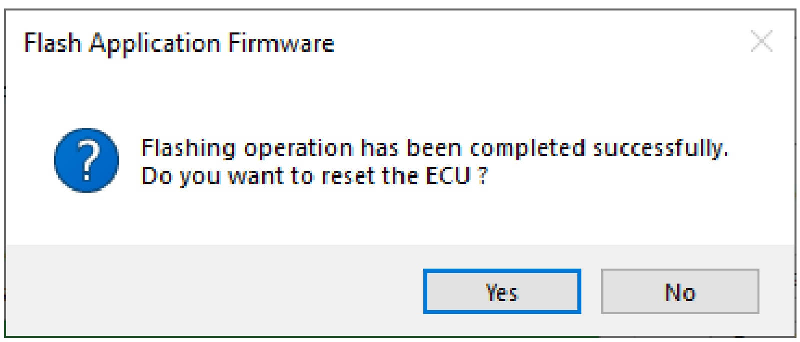


> WARNING: Do not check the "Erase All ECU Flash Memory" box unless instructed to do so by your Axiomatic contact. Selecting this will erased ALL data stored in non-volatile flash, including the calibration done by Axiomatic during factory testing. It will also erase any configuration of the setpoints that might have been done to the ECU and reset all setpoints to their factory defaults. By leaving this box unchecked, none of the setpoints will be changed when the new firmware is uploaded.

8. A progress bar will show how much of the firmware has been sent as the upload progresses. The more traffic there is on the J1939 network, the longer the upload process will take.



9. Once the firmware has finished uploading, a message will popup indicating the successful operation. If you select to reset the ECU, the new version of the AX100310 application will start running, and the ECU will be identified as such by EA. Otherwise, the next time the ECU is power-cycled, the AX100310 application will run rather than the bootloader function.



Note: If at any time during the upload the process is interrupted, the data is corrupted (bad checksum) or for any other reason the new firmware is not correct, i.e. bootloader detects that the file loaded was not designed to run on the hardware platform, the bad or corrupted application will not run. Rather, when the ECU is reset or power-cycled the **J1939 Bootloader** will continue to be the default application until valid firmware has been successfully uploaded into the unit.

5 TECHNICAL SPECIFICATIONS

All specifications are typical at nominal input voltage and 25 degrees C unless otherwise specified.

5.1 Power Input

Power Supply Input - Nominal	12V or 24V nominal 9…36 VDC power supply range Transient surge protection is provided. WARNING: This unit is designed for a battery power source. Back EMF from a motor could damage the controller if a bench power supply without voltage regulation is used.
Surge Protection	Provided
Reverse Polarity Protection	Provided
Quiescent Current	61mA @ 12V; 31mA @ 24V Typical

5.2 Input Specifications

CAN Interface CAN_HI CAN_LO	The user can select to control the motor using either a command received from the CAN bus or a wired command input.
Command Signal Input	Two signal inputs Speed command input is universal and user selectable through software configuration. Choose from one of the following signals: ▪ a digital input from a switch or other source (10 kΩ pull-up/pull down – normal, inversed or latched); ▪ potentiometer, 0-1V; 0-2.5V; 0-5V; 0-10Vdc, 0-20 mA or 4-20 mA analog input for machine controls; or ▪ PWM signal input (0.5 Hz to 50 Hz; 10 Hz to 1 kHz; 100 Hz to 10 kHz, 0-100%) for interface to a PLC or engine control module (ECM).
Input GND	Isolated GND connection provided
Quadrature Encoder Input	A, B, Z, +5V and GND connections Maximum Input Frequency 100 kHz *(Input 'Encoder Z", which measures the speed, is not able to measure a frequency higher than 60KHz.)* Input Amplitude: 0 to Vps. 20kΩ pulldown resistor Rising/Falling edge threshold 4.0V/1.0V +/- 0.1V Number or pulses per revolution are configurable. Custom scaler for encoder steps counter value. There is a configurable polarity of direction of rotation.
Minimum and Maximum Ratings	See Table 1.0 below

Table 1.0. Absolute Maximum and Minimum Ratings

Characteristic	Min	Max	Units
Power Supply	8	60	V dc
Voltage Input	0	43	V dc
Current Input	0	21	mA
Current Input – Voltage Level	0	12	Vdc
Digital Type Input – Voltage Level	0	43	Vdc
PWM Duty Cycle	0	100	%
PWM Frequency	50	20 000	Hz
PWM Voltage pk - pk	0	43	V dc
RPM Frequency	50	20 000	Hz

Input Accuracy	**Table 2.0. Input Accuracy** **Input Type** / **Accuracy** / **Resolution** Voltage / +/- 1% / 1 [mV] Current / +/- 1% / 1 [uA] PWM / +/- 1% (<5kHz) +/- 2% (>5kHz) / 0.1 [%] Frequency/RPM / +/- 1% / 0.01 [Hz]
Input Impedance	0-5V: 1 MOhm 0-10V: 170 kOhm 0(4)-20mA: 249 Ohm Frequency/Digital Input: Pull Up/Pull Down 1 MOhm
Scan Rate	Each input is scanned in 100uS. A complete scan of 10 inputs occurs with new measured values every 1mS.
Analog GND	1 Analog GND connections is provided.

Table 2.0. Input Accuracy

Input Type	Accuracy	Resolution
Voltage	+/- 1%	1 [mV]
Current	+/- 1%	1 [uA]
PWM	+/- 1% (<5kHz) +/- 2% (>5kHz)	0.1 [%]
Frequency/RPM	+/- 1%	0.01 [Hz]

5.3 Output Specifications

Drive Output	Outputs to a unidirectional, brushed DC motor (other DC motors, contact manufacturer) Half H-Bridge Driver Standard model – 12V or 24V, Up to 25 Amps continuous
Resolution	15-bit PWM
Accuracy	+/- 1% (Duty Cycle)
Protection	Overcurrent and short circuit protection are provided.
+5V Reference	+5V, 100 mA, 0.5%

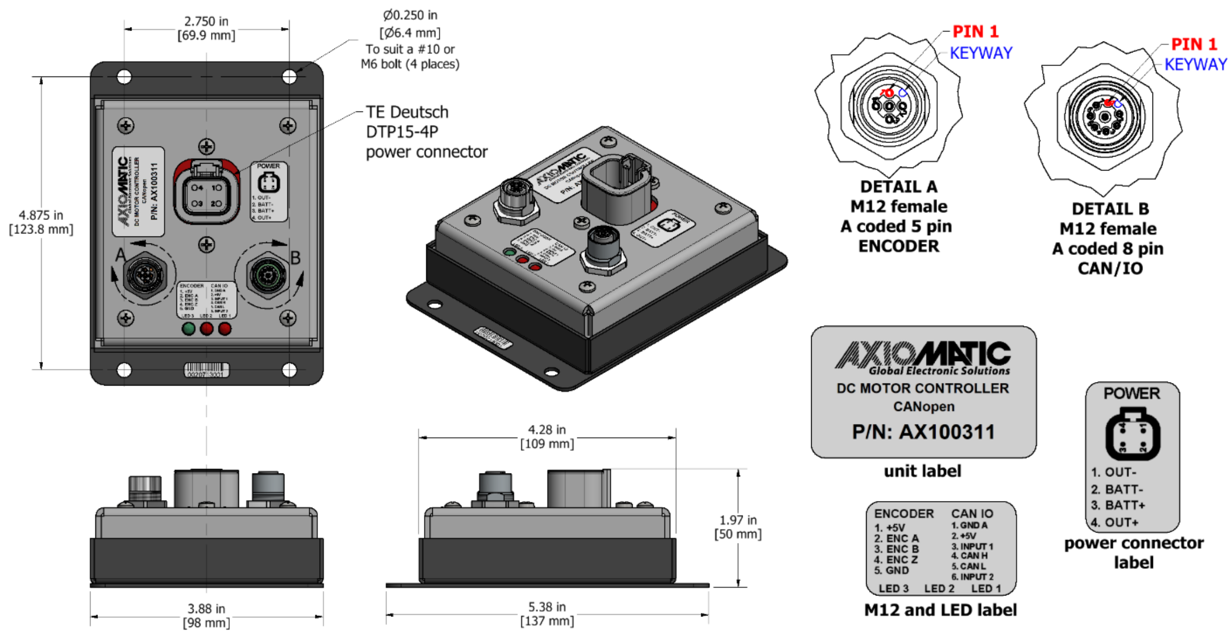
5.4 General Specifications

Microprocessor	STM32F405RG
CAN Interface	Model AX100311: CANopen®
Monitoring (options)	The controller can monitor the current drawn by the motor and transmits this information at a constant interval over the CAN bus. Other monitoring options include input measured and % Duty Cycle applied to the motor.
Quiescent Current	61 mA @ 12Vdc; 31 mA @ 24Vdc typical
LED Indication	LED1 – Red LED Flashing LED2 – Red LED Flashing Heartbeat – Green LED (Flashing)
Control Logic	Configurable properties of the controller are divided into function blocks. They include: input function block; diagnostic function block; lookup table function block; programmable logic function block; math function block; constant data block; conditional block; CAN receive message function block; and CAN transmit message function block. The Input function block includes properties used to select input sensor functionality. The Diagnostic function block properties are used to configure fault detection and reaction functionalities. The Math function block gives user an opportunity to process inputs with basic mathematics of the logical functions. The CAN transmit message function block configures properties of the messages sent to the CAN bus. All logic blocks are implemented. Refer to the user manual for details.
CAN Termination	120Ω should be connected between CAN_HI and CAN_LO in a mating wire harness assembly (not supplied).

Electrical Connections	1 8-pin M12, A coded connector 1 5-pin M12, A coded connector 1 4-pin connector (equivalent TE Deutsch P/N: DTP15-4P) See pin out below in Table 2.0.
Mating Wire Harness	The mating plug kits for the M12 connectors are not supplied. 1 4-pin TE Deutsch connector (equivalent TE Deutsch P/N: DTP15-4P) mates with wire harness **AX070149** (The mating plug assembly is equivalent to the TE Deutsch P/Ns: DTP06-4S, WP4S and four contact sockets 0462-203-12141 with 2 meters (6.5 ft.) of 12 AWG lead wire, unterminated.) Wire Harness Pin out: 1 Output - Black/White 2 Battery – Black 3 Battery + Red 4 Output + Red/White
Enclosure and Dimensions	Encapsulated unit enclosed in a stainless-steel casing with powder coated steel base *Refer to the dimensional drawing.*
Operating Conditions	-40 to +75°C (-40 to 167°F) 0 to 95% relative humidity
Storage Temperature	-40 to 100°C (-40 to 212°F)
Weight	1.80 lb. (0.816 kg)
Vibration	Vibration compliance is suitable for mobile equipment applications.
Protection	IP67

Specifications are indicative and subject to change. Actual performance will vary depending on the application and operating conditions. 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 https://www.axiomatic.com/service/.

5.5 Dimensions and Typical Connections



Connector	PIN#	Function
Command and CAN 8-pin M12 Female, A coded	1	Input GND
	2	+5V Reference
	3	Input 1
	4	CAN_H
	5	CAN_L
	6	Input 2
	7	Not Used
	8	Not Used
Power and Output Equivalent TE Deutsch P/N: DTP15-4P	1	Output -
	2	Battery -
	3	Battery +
	4	Output +
Encoder Input 5-pin M12, A coded P/N: 09-3442-126-05		
	1	+5V
	2	ENC_A
	3	ENC_B
	4	ENC_Z
	5	GND

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

Global Electronic Solutions

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

Axiomatic provides electronic machine control components to the off-highway, commercial vehicle, electric vehicle, power generator set, material handling, renewable energy and industrial OEM markets. ***We innovate with engineered and off-the-shelf machine controls that add value for our customers.***

QUALITY DESIGN AND MANUFACTURING

We have an ISO9001:2015 registered design/manufacturing facility in Canada.

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 at https://www.axiomatic.com/service/.

COMPLIANCE

Product compliance details can be found in the product literature and/or on axiomatic.com. Any inquiries should be sent to sales@axiomatic.com.

SAFE USE

All products should be serviced by Axiomatic. Do not open the product and perform the service yourself.

This product can expose you to chemicals which are known in the State of California, USA to cause cancer and reproductive harm. For more information go to www.P65Warnings.ca.gov.

SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#) from sales@axiomatic.com. Please provide the following information when requesting an RMA number:

- Serial number, part number
- Runtime hours, description of problem
- Wiring set up diagram, application and other comments as needed

DISPOSAL

Axiomatic products are electronic waste. Please follow your local environmental waste and recycling laws, regulations and policies for safe disposal or recycling of electronic waste.

CONTACTS

Axiomatic Technologies Corporation
1445 Courtneypark Drive E.
Mississauga, ON
CANADA L5T 2E3
TEL: +1 905 602 9270
FAX: +1 905 602 9279
www.axiomatic.com
sales@axiomatic.com

Axiomatic Technologies Oy
Höytämöntie 6
33880 Lempäälä
FINLAND
TEL: +358 103 375 750
www.axiomatic.com
salesfinland@axiomatic.com