USER MANUAL UMAX031301
Version 1.0.2

10 LED Outputs Controller

CANopen®

USER MANUAL

P/N: AX031301

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

ACRONYMS

AO	Analog Output
CAN	Controller Area Network
CANopen®	CANopen® is a registered community trademark of CAN in Automation e.V.
CAN-ID	CAN 11-bit Identifier
COB	Communication Object
CTRL	Control
EDS	Electronic Data Sheet
EMCY	Emergency
LED	Light-Emitting Diode
LSB	Least Significant Byte (or Bit)
LSS	Layer Settling Service
LUT	Lookup Table
MSB	Most Significant Byte (or Bit)
NMT	Network Management
PID	Proportional-Integral-Derivative Control
PWM	Pulse Width Modulation
RO	Read Only Object
RPDO	Received Process Data Object
RW	Read/Write Object
SDO	Service Data Object
TPDO	Transmitted Process Data Object
Vps	Voltage Power Supply (a.k.a. BATT+)
% DC	Percent Duty Cycle (Measured from a PWM input)
WO	Write Only Object

REFERENCES

[DS-301] CiA DS-301 V4.1 – CANopen® Application Layer and Communication Profile. CAN in Automation 2005

[DS-305] CiA DS-305 V2.0 – Layer Setting Service (LSS) and Protocols. CAN in Automation 2006

[DS-404] CiA DS-404 V1.2 – CANopen® profile for Measurement Devices and Closed Loop Controllers. CAN in Automation 2002

These documents are available from the CAN in Automation e.V. website: http://www.can-cia.org/.

TABLE OF CONTENTS

1. OVERVIEW OF CONTROLLER

1.1. Description of Controller

This User Manual describes the architecture and functionality of the CAN-10LEDOUT-8P-CO controller with CANopen®.

The Controller is a versatile and high-performance solution for managing LED lighting in automotive and industrial applications. This controller supports ten independent LED outputs, each capable of driving up to 600 mA, providing ample power for a variety of lighting configurations. It offers dual control modes, allowing for both direct current (DC) operation for steady illumination and pulse-width modulation (PWM) for dimming and dynamic lighting effects.

The controller is highly programmable, allowing the user to configure it for their application. Its sophisticated control algorithms/function blocks allow the user to configure the controller for a wide range of applications without the need for custom firmware. It can be operated as either a self-contained control system, driving the outputs directly from the on-board logical function blocks, and/or it can be integrated into a CANopen® network of controllers. Its flexibility and robust design make it ideal for demanding environments, delivering precise and reliable LED control across a wide range of applications. All outputs and logical function blocks on the unit are inherently independent from one another but can be programmed to interact in a large number of ways. Figure 1 shows the hardware features of the ECU.

The controller has some built-in protections features that can shut off the outputs in adverse conditions. These features are described in detail in subsequent sections, and they include hardware shutoffs to protect the circuits from being damaged as well as software shutdown features that can be enabled in safety critical systems when a CAN fault is detected.

The various function blocks supported by the ECU are outlined in the following sections. All objects are user-configurable using standard commercially available tools that can interact with a CANopen® Object Dictionary via an .EDS file.

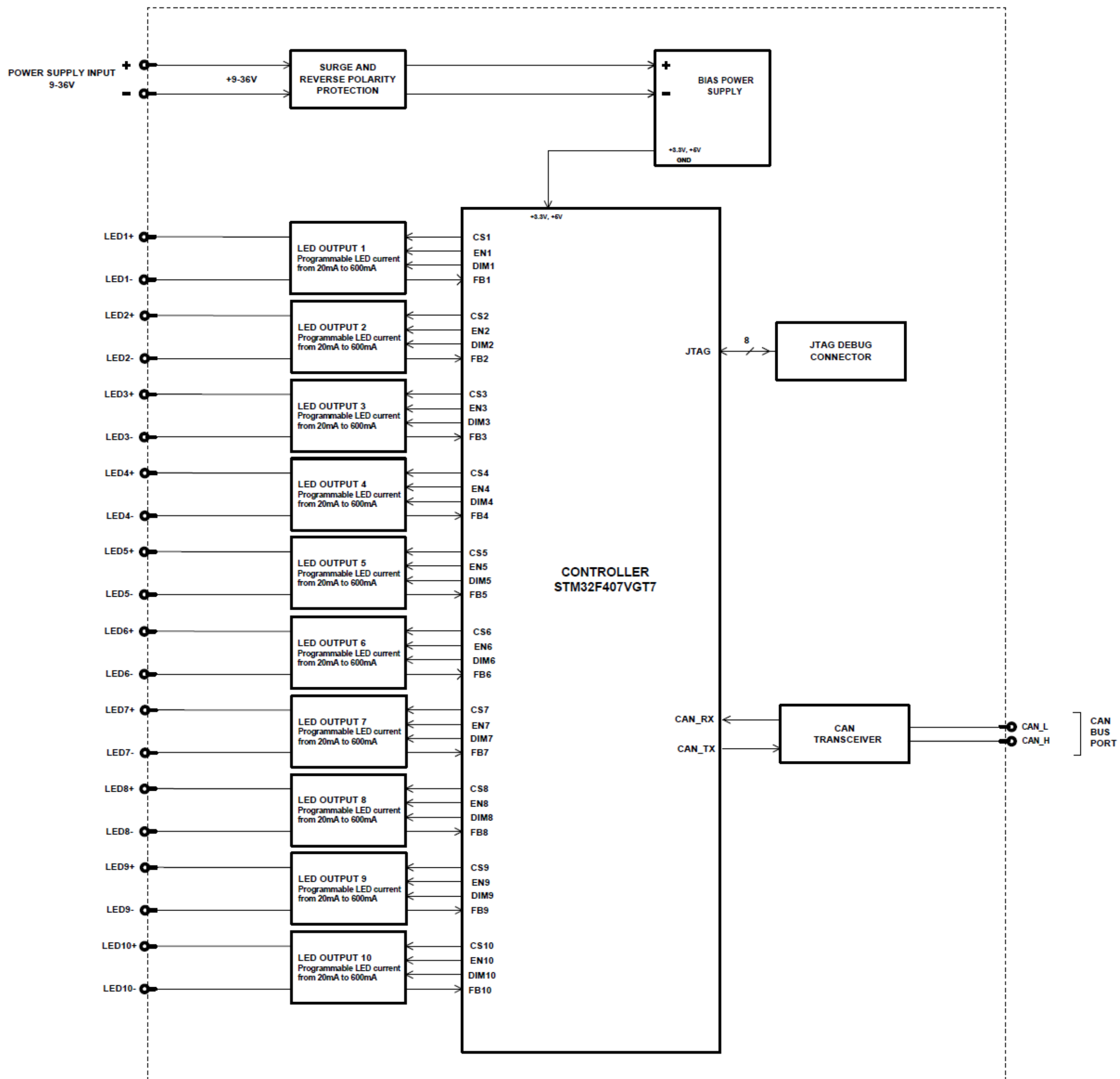


Figure 1 - Hardware Functional Block Diagram

1.2. LED Output Function Block

The 10LEDOUT function block is the default logic associated with output 1 to 10.

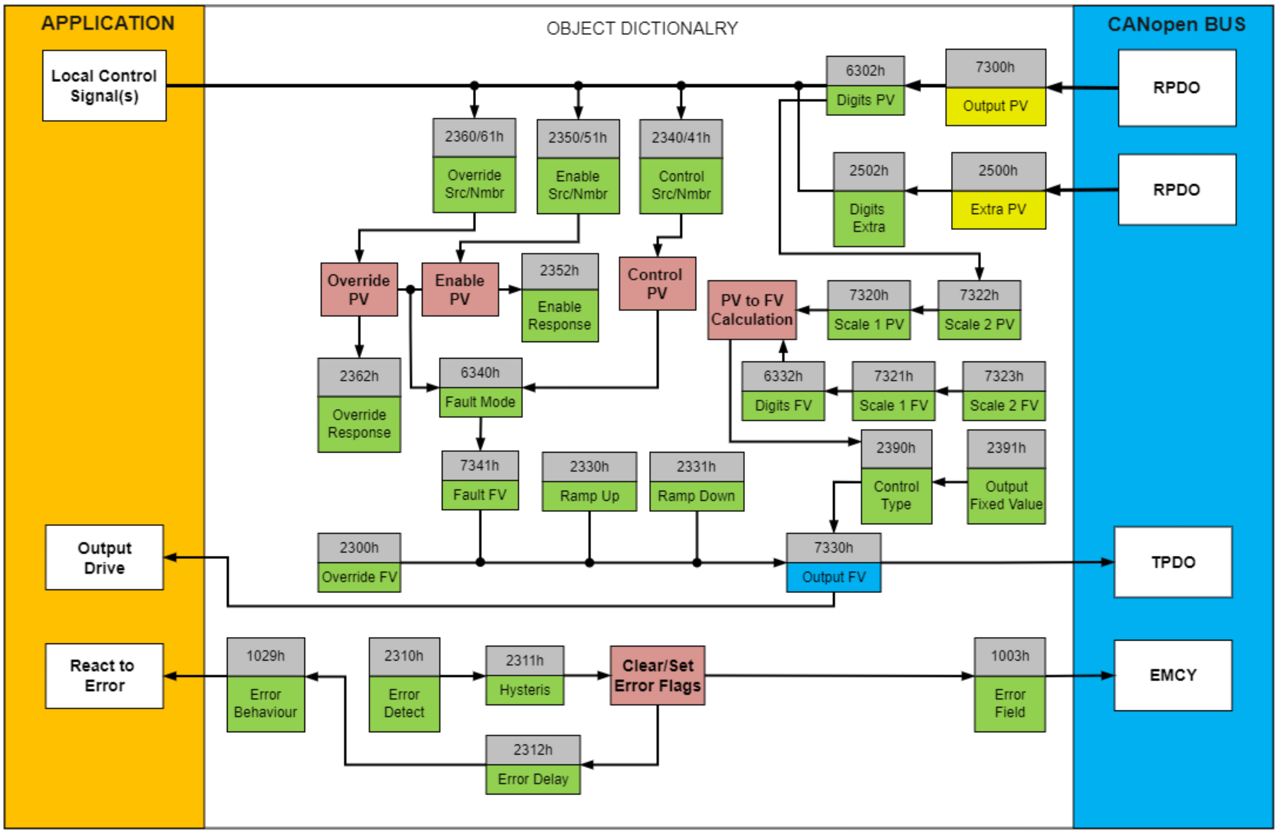


Figure 2 - LED Output Objects

There are many objects associated with the LED output function block, but not all of them apply to all output types or control conditions. To start with, object 6310h **LED Output Type** defines how the output drive circuitry will be configured as per Table 1. This table also shows the output unit and range for each type. By default, outputs are configured as '0, Output Disable'.

Value	Meaning	Output Range [Unit]
0	**Output Disabled**	**N/A**
20	Output LED	80 to 600 [mA]

Table 1 – Object 6310h - LED Output Type Options

Note that the objects relating to the Output Functional Block have sub-indices corresponding to the different outputs on this ECU as follows:

Sub-Index	Corresponding Output
1	LED Output 1
2	LED Output 2
3	LED Output 3
4	LED Output 4
5	LED Output 5
6	LED Output 6
7	LED Output 7
8	LED Output 8
9	LED Output 9
10	LED Output 10

Table 2 – LED Output Object Subindices

When the output type is changed, all objects related to the output (scaling PV, Decimal Digits PV, etc.) are automatically updated by default. Object 5550h enables/disables automatic updates. When disabled (set to False), the objects are to be manually configured.

By default, when the outputs are in LED mode, the output is actually a high frequency PWM signal that is being adjusted by the processor such that the average voltage would match the target FV, up to the supply voltage. *External filtering of the signal would have to be applied to get a true DC voltage*.

The relationship between the Process Value (input) and the Field Value (output) is a linear one, as shown in Figure 3. However, the output will actually use the AO Scaling FV objects as limits to the drive, such that the output will hold at the minimum and maximum FV points, as shown in the figure.

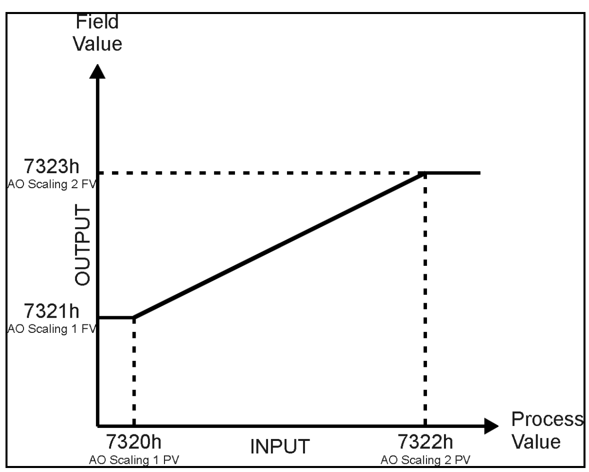


Figure 3 - LED Output Linear Scaling PV to FV

The CAN-10LEDOUT controller allows for the PV input 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 3. By default, LED outputs are set up to respond to the corresponding CANopen® RPDO message.

Note:

-When using CANopen Message (RPDO) as a control source for the outputs, only the corresponding subindex to that specific output may be used.

-While any CANopen Message can be used as a source to another logic block, or the output enable/override signal, it is **not** recommended to use the CANopen messages that control output 1 to output 10. This data is dependent on the output type (7300h for LEDOUT, and 2700h for LED PWM Duty Cycle), which may cause unintended behaviour when an output changes.

Value	Meaning
0	Control Source Not Used (Ignored)
1	CANopen® Message (RPDO)
2	Constant Function Block
3	PID Control Function Block
4	Lookup Table Function Block
5	Set-Reset Latch Function Block
6	Conditional Logic Function Block
7	Mathematical Function Block
8	Programmable Logic Function Block
9	Output Commanded Field Value
10	Power Supply Measured
11	Processor Temperature Measured

Table 3 – 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 4 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	7300h sub-index 1 (LED Output)
	2	7300h sub-index 2 (LED Output)
	3	7300h sub-index 3 (LED Output)
	4	7300h sub-index 4 (LED Output)
	5	7300h sub-index 5 (LED Output)
	6	7300h sub-index 6 (LED Output)
	7	7300h sub-index 7 (LED Output)
	8	7300h sub-index 8 (LED Output)
	9	7300h sub-index 9 (LED Output)
	10	7300h sub-index 10 (LED Output)
	11	2700h sub-index 1 (PWM Duty Cycle)
	12	2700h sub-index 2 (PWM Duty Cycle)
	13	2700h sub-index 3 (PWM Duty Cycle)
	14	2700h sub-index 4 (PWM Duty Cycle)
	15	2700h sub-index 5 (PWM Duty Cycle)
	16	2700h sub-index 6 (PWM Duty Cycle)
	17	2700h sub-index 7 (PWM Duty Cycle)
	18	2700h sub-index 8 (PWM Duty Cycle)
	19	2700h sub-index 9 (PWM Duty Cycle)
	20	2700h sub-index 10 (PWM Duty Cycle)
	21	2500h sub-index 1 (Extra Received PV 1)
	22	2500h sub-index 2 (Extra Received PV 2)
	23	2500h sub-index 3 (Extra Received PV 3)
	24	2500h sub-index 4 (Extra Received PV 4)
	25	2500h sub-index 5 (Extra Received PV 5)
	26	2500h sub-index 6 (Extra Received PV 6)
	27	2500h sub-index 7 (Extra Received PV 7)
	28	2500h sub-index 8 (Extra Received PV 8)
	29	2500h sub-index 9 (Extra Received PV 9)
	30	2500h sub-index 10 (Extra Received PV 10)
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)
	11	5010h sub-index 11 (Constant FV 11)
	12	5010h sub-index 12 (Constant FV 12)
	13	5010h sub-index 13 (Constant FV 13)
	14	5010h sub-index 14 (Constant FV 14)
	15	5010h sub-index 15 (Constant FV 15)

PID Control Function Block	1	2460h sub-index 1 (PID Output FV 1)
	2	2460h sub-index 2 (PID Output FV 2)
	3	2460h sub-index 3 (PID Output FV 3)
	4	2460h sub-index 4 (PID Output FV 4)
	5	2460h sub-index 5 (PID Output FV 5)
	6	2460h sub-index 6 (PID Output FV 6)
	7	2460h sub-index 7 (PID Output FV 7)
	8	2460h sub-index 8 (PID Output FV 8)
	9	2460h sub-index 9 (PID Output FV 9)
	10	2460h sub-index 10 (PID Output FV 10)
	11	2460h sub-index 11 (PID Output FV 11)
	12	2460h sub-index 12 (PID Output FV 12)
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)
Set-Reset Latch Function Block	1	3910h sub-index 1 (SR Latch 1 Output PV)
	2	3910h sub-index 2 (SR Latch 2 Output PV)
	3	3910h sub-index 3 (SR Latch 3 Output PV)
	4	3910h sub-index 4 (SR Latch 4 Output PV)
	5	3910h sub-index 5 (SR Latch 5 Output PV)
Conditional Logic Block	1	4B10h sub-index 1 (Cond Logic 1 Output PV)
	2	4B10h sub-index 2 (Cond Logic 2 Output PV)
	3	4B10h sub-index 3 (Cond Logic 3 Output PV)
	4	4B10h sub-index 4 (Cond Logic 4 Output PV)
	5	4B10h sub-index 5 (Cond Logic 5 Output PV)
	6	4B10h sub-index 6 (Cond Logic 6 Output PV)
	7	4B10h sub-index 7 (Cond Logic 7 Output PV)
	8	4B10h sub-index 8 (Cond Logic 8 Output PV)
	9	4B10h sub-index 9 (Cond Logic 9 Output PV)
	10	4B10h sub-index 10 (Cond Logic 10 Output 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)
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)
***NOTE**: The following options should be considered for diagnostic feedback, and should not be selected as a control source for logic inputs (i.e. output control or lookup table X-Axis)*		

Output Commanded Field Value	1	7330h sub-index 1
	2	7330h sub-index 2
	3	7330h sub-index 3
	4	7330h sub-index 4
	5	7330h sub-index 5
	6	7330h sub-index 6
	7	7330h sub-index 7
	8	7330h sub-index 8
	9	7330h sub-index 9
	10	7330h sub-index 10
PWM Duty Cycle Field Value	1	2730h sub-index 1
	2	2730h sub-index 2
	3	2730h sub-index 3
	4	2730h sub-index 4
	5	2730h sub-index 5
	6	2730h sub-index 6
	7	2730h sub-index 7
	8	2730h sub-index 8
	9	2730h sub-index 9
	10	2730h sub-index 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 4 – Control Number Options Depending on Source Selected

There are three inputs to the output function block, each one with a unique source and number object. For the control function (PV axis in Figure 3), objects 2340h **AO Control Source** and 2341h **AO Control Number** are used. For the enable function, objects 2350h **AO Enable Source** and 2351h **AO Enable Number** are used. Lastly, for the override function, objects 2360h **AO Override Source** and 2361h **AO Override Number** are used.

When using any control source as the X-Axis input to a function block, the corresponding scaling limits are defined as per Table 5. 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.

Note that for the Outputs, the actual objects for the scaling (6302h, 7320h, 7322h) should be edited to match the objects defined in this table when the control source is changed.

Control Source	Scaling 1	Scaling 2	Dec Digits
CANopen® Message – Num 1 to 10	7320h	7322h	6302h
CANopen® Message – Num 11 to 20	2720h	2722h	2702h
CANopen® Message – Num 21 to 30	2520h	2522h	2502h
Constant Function Block	0.0	1.0	N/A (float)
PID Control Function Block	0%	100%	1 (fixed)
Lookup Table yz Function Block (where yz = 01 to 9)	0 or lowest from 3yz6h(*)	100 or highest from 3yz6h(**)	3yz3h
Set-Reset Latch	0 [OFF]	1 [ON]	0 (fixed)
Conditional Logic	0 [OFF]	1 [ON]	0 (fixed)
Mathematical Function	4021h	4023h	4032h
Programmable Logic Function	0%	100%	1 (fixed)
Output Commanded Field Value	7320h	7322h	6302h
PWM Duty Cycle Field Value	2720h	2722h	2702h
Power Supply Measured	N/A	N/A	1 (fixed)
Processor Temperature Measured	N/A	N/A	1 (fixed)

(*) - Whichever value is smaller; (**) - Whichever value is larger

Table 5 – Scaling Limits per Control Source

As shown in Figure 3, the Output FV will be calculated based on the FV scaling selected. Since 7321h represents the value at or below the lowest control input received, it represents the minimum field value that will be applied at the output. Similarly, 7323h represents the maximum FV that will be applied.

While (7320h < 7322h) must always hold true, in order to get an inverse response (i.e. output decreases as the input increased), simply set 7321h higher than 7323h.

In general, the profile shown in Figure 3 holds true. However, in some cases it may be desired that the minimum offset not be applied when the value is outside of the range, i.e. when using a joystick profile with a deadband. For this reason, object 2342h **AO Control Response** has the options shown in Table 6.

Value	Meaning
0	**Single Output Profile (Figure 3)**
1	Output OFF below Scaling 1 PV
2	Output OFF above Scaling 2 PV

Table 6 – Object 2342 - AO Control Response Options

Enable and Override inputs have been mentioned several times already. By default, neither input are used (control sources are set to 0=Ignore), but they can be activated for safety interlocks or other more complex applications. Table 7 shows the options for object 2352h **AO Enable Response.**

Note that the Enable and Override functionalities do not apply the output delay, only the Control Response uses the delay.

Value	Meaning
0	**Enable When ON, Else Shut OFF**
1	Enable When ON, Else Ramp OFF
2	Enable When ON, Else Keep Last Value
3	Enable When OFF, Else Shut OFF
4	Enable When OFF, Else Ramp OFF
5	Enable When OFF, Else Keep Last Value

Table 7 – Object 2352h - AO Enable Response Options

Table 8 shows options for object 2362h **AO Override Response** respectively. In both cases, the default responses are bolded. When the override is applied, the output is driven to the value defined in object 2300h, **AO Override FV.**

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

Table 8 – Object 2362h - AO Override Response Options

When an input to the output block goes into an error condition, object 6340 **AO Fault Mode** determines how the output will respond, per Table 9. By default, the output will be driven to the value defined in object 7341h **AO Fault FV**.

Value	Meaning
0	Shutoff
1	**Apply Pre-Defined FV**
2	Maintain Last State

Table 9 – Object 6340 - AO Fault Mode Options

The controller applies the logic shown in Figure 2 when evaluating what output FV to apply. Under normal conditions, i.e. when the control input is driving the output as shown in the green box, there are ramping objects that can be applied to soften the output response. Object 2330h **AO Ramp Up** and object 2331 **AO Ramp Down** are both millisecond numbers that define how long it will take to ramp from AO Scaling 1 FV to AO Scaling 2 FV.

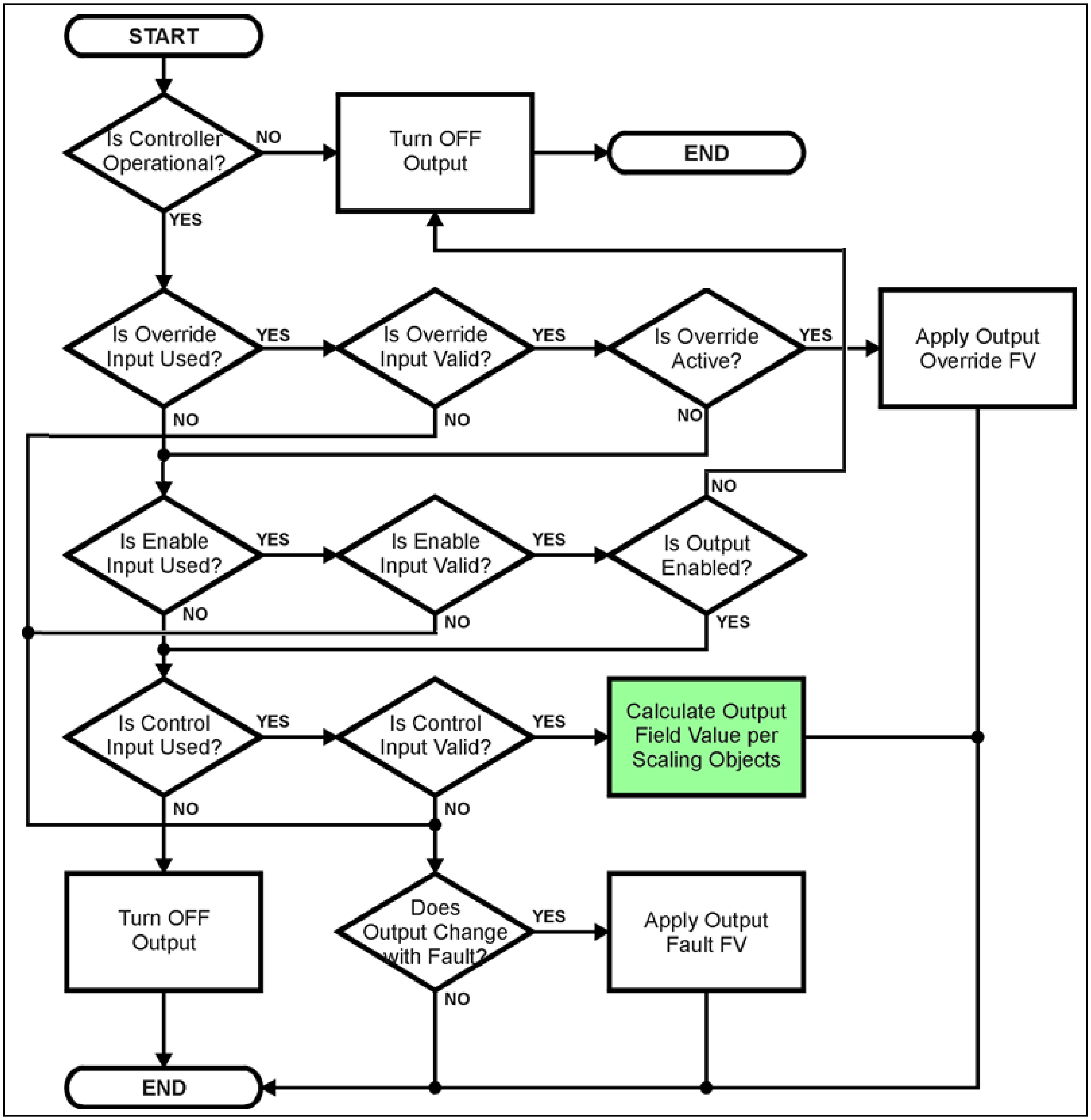


Figure 4 - LED Output Logic Flowchart

Object 7330h **AO Output Field Value** (as represented by the green box above) is a read-only mappable object. This object reflects the actual measured value at the output.

Object 2310h **AO Error Detect Enable** determines if the output open/short circuit errors are checked. In all output modes, open or short circuit errors can be detected.

If the controller will then flag an "Open Circuit" fault and the flag stays active for the 2312h **AO Error Reaction Delay** time, then an appropriate EMCY message will be added to object 1003h **Pre-Defined Error Field**. The application will react to the EMCY message as defined by object 1029h **Error Behaviour** at the sub-index corresponding to an Input Fault. Refer to section 2.2.4 and 2.2.13 for more information about objects 1003h and 1029h, including the complete list of EMCY messages.

1.2.1. Direct Current Control

The LED output supports DC dimming with a current range of 80 mA to 600 mA, providing flexible current control for various LED applications. Each outputs use the control logic defined on object 2330h by default. To change the control type and use a fixed value to command the output current, use object 2390h **AO Output Control Type** to set the desired configuration. The output fixed value is defined on object 2391h **AO Output Fixed Value**.

Value	Meaning
0	**Control Logic**
1	Fixed Value

Table 10. Output Control Type

The controller also supports a blinking mode, which operates on a fixed PWM Frequency value set to 5000 Hz. The blink interval is defined on object 2323h **AO Output Blink Rate** with a default value of 1000ms.

Note that it is recommended to avoid excessively fast blinking, as it may cause unintended effects.

1.2.2. PWM Duty Cycle Control

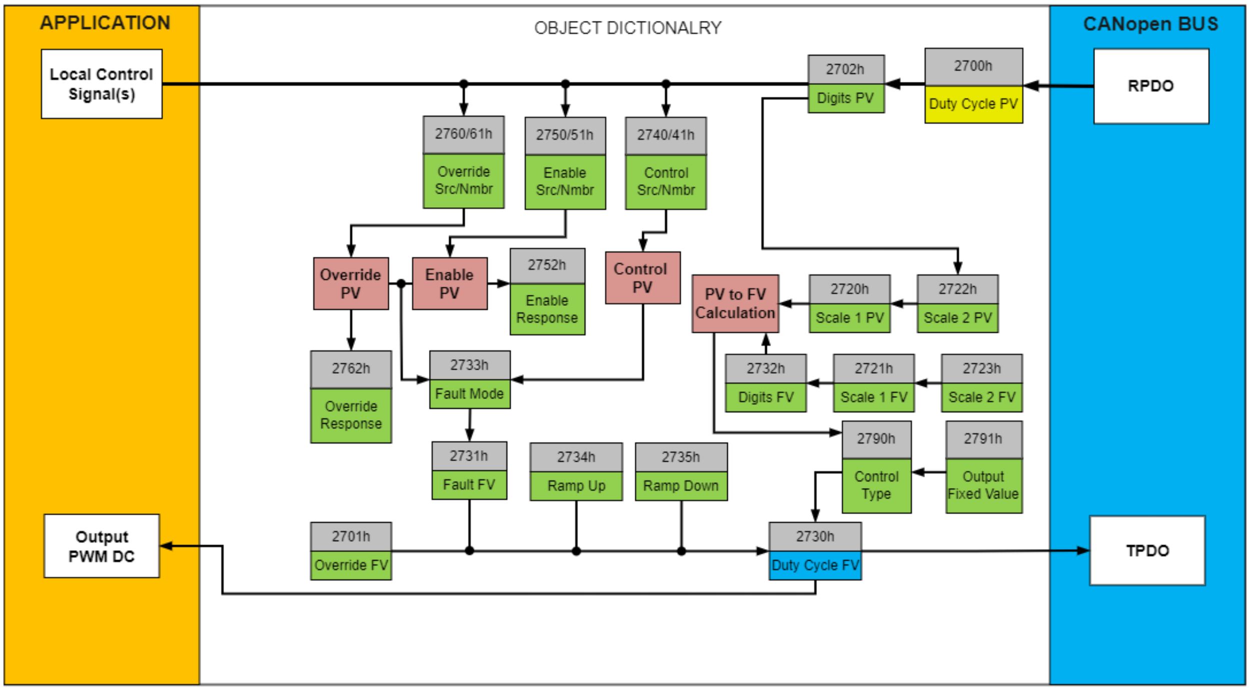


Figure 5 - PWM Duty Cycle Objects

For PWM dimming, the duty cycle can be adjusted from 0% to 100%. There are three inputs to the Output PWM Duty Cycle control block. For the Process Value control objects 2740h **Duty Cycle Control Source** and 2741h **Duty Cycle Control Number** are used. For the enable function, objects 2750h **Duty Cycle Enable Source** and 2751h **Duty Cycle Enable Number** are used. Lastly, for the override function, objects 2760h **Duty Cycle Override Source** and 2761h **Duty Cycle Override Number** are used.

To change the control type and use a fixed value to command the output Duty Cycle, use object 2790h **Duty Cycle Control Type** to set the desired configuration. The output fixed value is defined on object 2791h **Duty Cycle Fixed Value**.

Objects 2720h and 2722h should be edited to match the desired scaling when the control source is changed. Object 2702h sets the Process Value decimals digits. The Output PWM Duty Cycle will be calculated based on the FV scaling selected. Object 2721h represents the minimum field value that will be applied at the output. Similarly, 2723h represents the maximum FV that will be applied.

To control when the value is outside of the range, object 2742h **Duty Cycle Control Response** has the options. Enable and Override options are defined by object 2752h **Duty Cycle Enable Response,** and 2762h **Duty Cycle Override Response** respectively.

Object 2740 **Duty Cycle Fault Mode** determines how the output Duty Cycle will respond in case of fault. The output will be driven to the value defined in object 7341h **Duty Cycle Fault FV**.

Object 2734h **Duty Cycle Ramp Up** and object 2335 **Duty Cycle Ramp Down** are both millisecond numbers that define how long it will take to ramp from Duty Cycle Scaling 1 FV to Duty Cycle Scaling 2 FV.

1.3. PID Control Function Block

The PID control (PID) function blocks are not used by default.

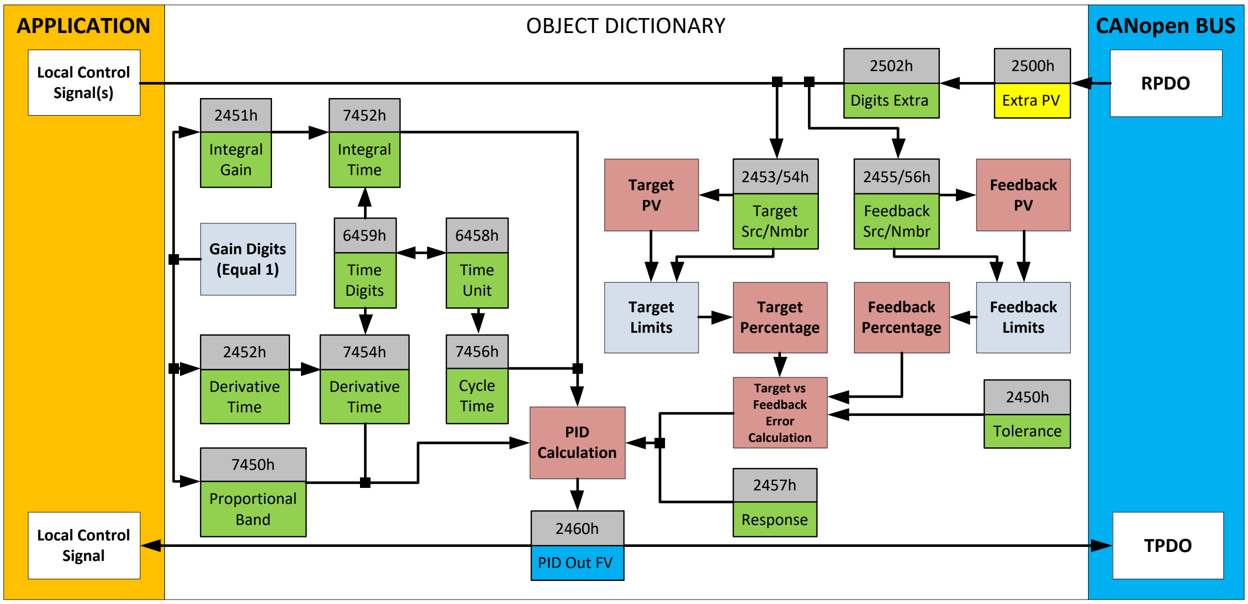


Figure 6 - PID Control Objects

As with the output function block, the PID control function has control inputs associate with it that can be mapped to the output from any other function block. Objects 2453h **PID Target Source** and 2454h **PID Target Number** define what value the PID loop will attempt to maintain. For example, in the case of a setpoint (fixed) control application, this input can be mapped to one of object 5010h, a Constant FV. In this case, since there is no pre-defined range associated with a constant (see Table 5), the scaling limits will be set equal to those of the feedback input. Otherwise, the target input units do not have to match the feedback units, so long as they are scaled relative to one another.

Objects 2355h **PID Feedback Source** and 2356h **PID Feedback Number** define the close-loop input. Both the target and feedback use Table 3 and Table 4 as the available options. Both inputs are normalized to a percentage based on the associated scaling limits as defined in Table 5.

Object 2450h **PID Tolerance** defines the acceptable difference between the target and feedback, as a percentage, whereby an absolute difference smaller than this is treated as a 0% error.

Unless both the target and feedback inputs have legitimate control sources selected, the PID loop is disabled. When active, however, the PID algorithm will be called every 7456h **PID Cycle Time,** the default being every 10ms.

Object 6458h **PID Physical Unit Timing** is a read-only value and is defined in Seconds. The default value for object 6459h **PID Decimal Digits Timing** is 3, which means the object 7456h, along with other PID timing objects, are interpreted in milliseconds. Other time objects associated with the PID control are 7452h **PID Integral Action Time (Ti)** and 7454h **PID Derivative Action Time (Td).**

None time related objects use a fixed resolution of 1 decimal digit. These objects include 7450h **PID Proportional Band (G)**, 2450h **PID Tolerance**, 2451h **PID Integral Gain (Ki)**, and 2454h **PID Derivative Gain (Kd).**

By default, the PID loop is assumed to be controlling a single output which will increase/decrease as the feedback over/undershoots the target. However, some systems may require a push-pull response where one output comes on when over target, and the other when under. Object 2457h **PID Control Response** allows the user to select the response profile as needed from Table 11.

Value	Meaning
0	**Single Output**
1	On When Over Target
2	On When Below Target

Table 11 – PID Control Response Options

The PID algorithm used is shown below, with names in red being the object variables. The result $PIDOutput_K$ is written to the read-only mappable object 2460h **PID Output Field Value and** is interpreted as a percentage value with 1 decimal place resolution. It can be used as the control source for another function block, i.e. one of the LED outputs.

```
T = Loop_Update_Rate*0.001

P_Gain = G
I_Gain = G*Ki*T/Ti
D_Gain = G*Kd*Td/T
Note: If Ti is zero, I_Gain = 0

Error_k     = Target – Feedback
ErrorSum_k  = ErrorSum_k-1 + Error_k

P_k = Error_k * P_Gain

I_k = ErrorSum_k * I_Gain

D_k = (Error_k-Error_k-1) * D_Gain

PIDOutput_k = P_k + I_k + D_k
```

Figure 7 - PID Control Algorithm

Each system will have to be tuned 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.

1.4. Lookup Table Function Block

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

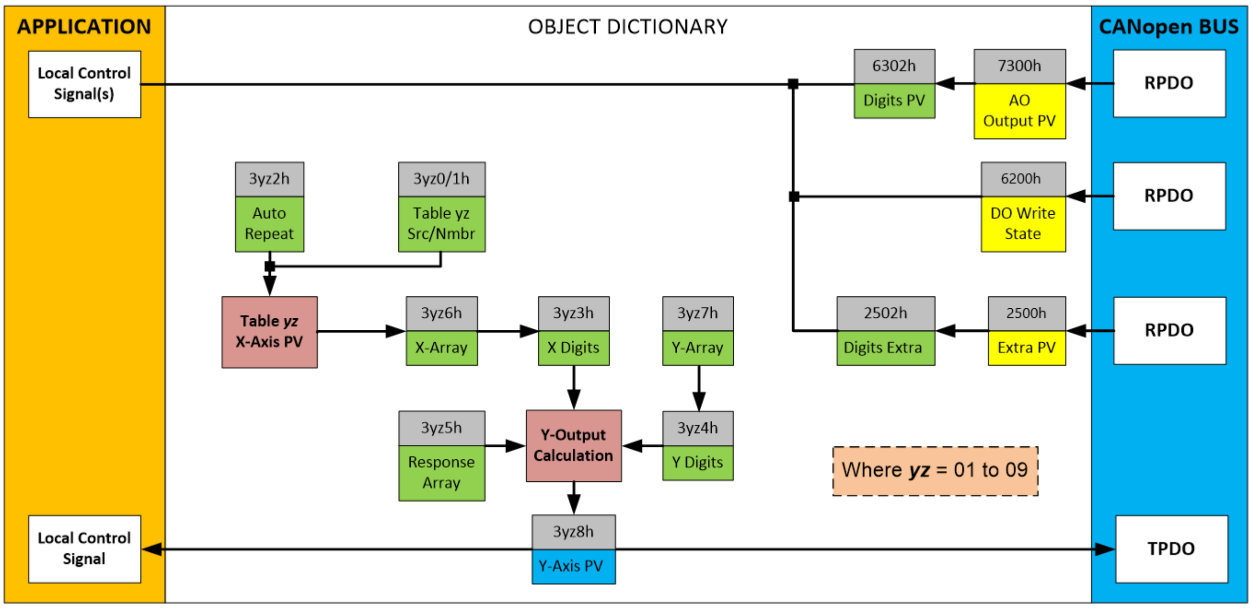


Figure 8 - Lookup Table Objects

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

Note: If more than 10 slopes are required, a Programmable Logic Block can be used to combine up to three tables to get 30 slopes, as is described in Section 1.5.

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 as described in Table 3 and Table 4. 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 Section 1.4.4.

1.4.1. 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 5.

1.4.2. 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, so 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 5.

However, the Y-Axis has no constraints on the data that it represents. This means that inverse, or increasing/decreasing or other responses can be easily established. **The Y-Axis does not have to be a percentage output, but 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-indexes, and selects the lowest value as the MinOutRange and the highest value as the MaxOutRange. So 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 Section 1.4.3, 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 LED output.

1.4.3. 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 9 shows the difference between these two response profiles with the default settings.

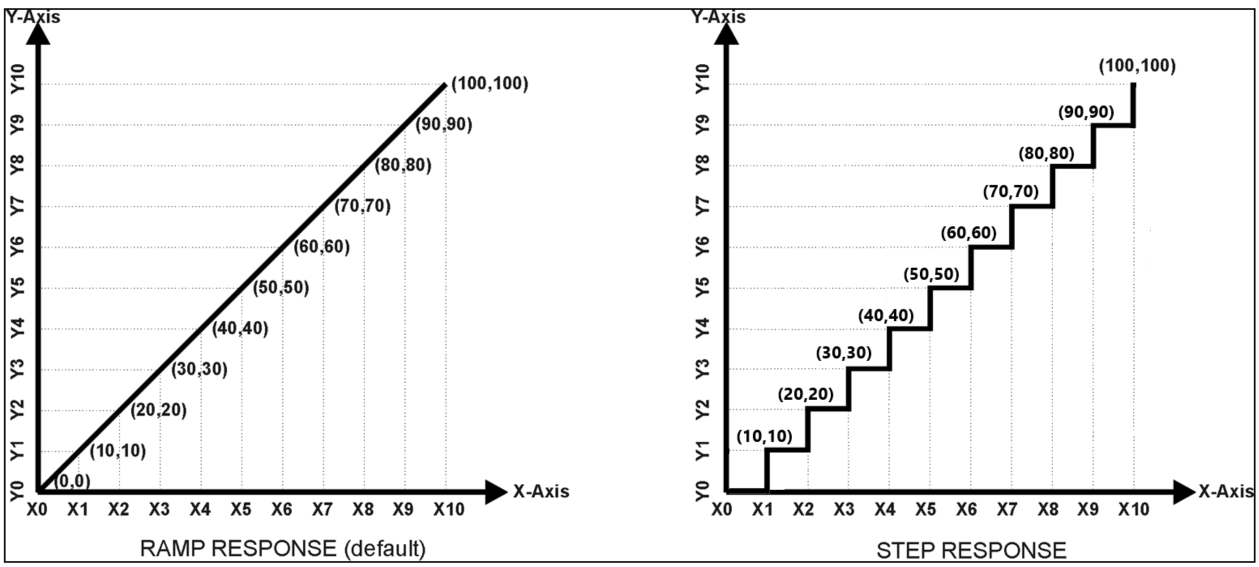


Figure 9 - Lookup Table Defaults with Ramp and Step Responses

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

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

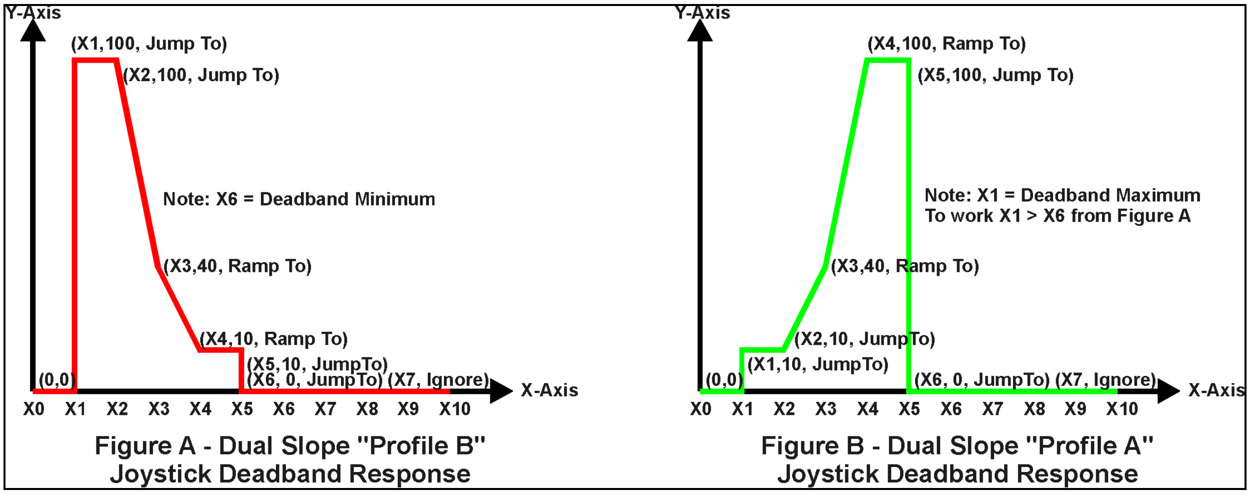


Figure 10 - Lookup Table Examples to Setup for Dual-Slope Joystick Deadband Response

To summarize, Table 12 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 12 – LTyz Point Response Options

1.4.4. X-Axis, Time Response

As mentioned in Section 1.4, 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 LED 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]

1.5. Programmable Logic Function Block

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

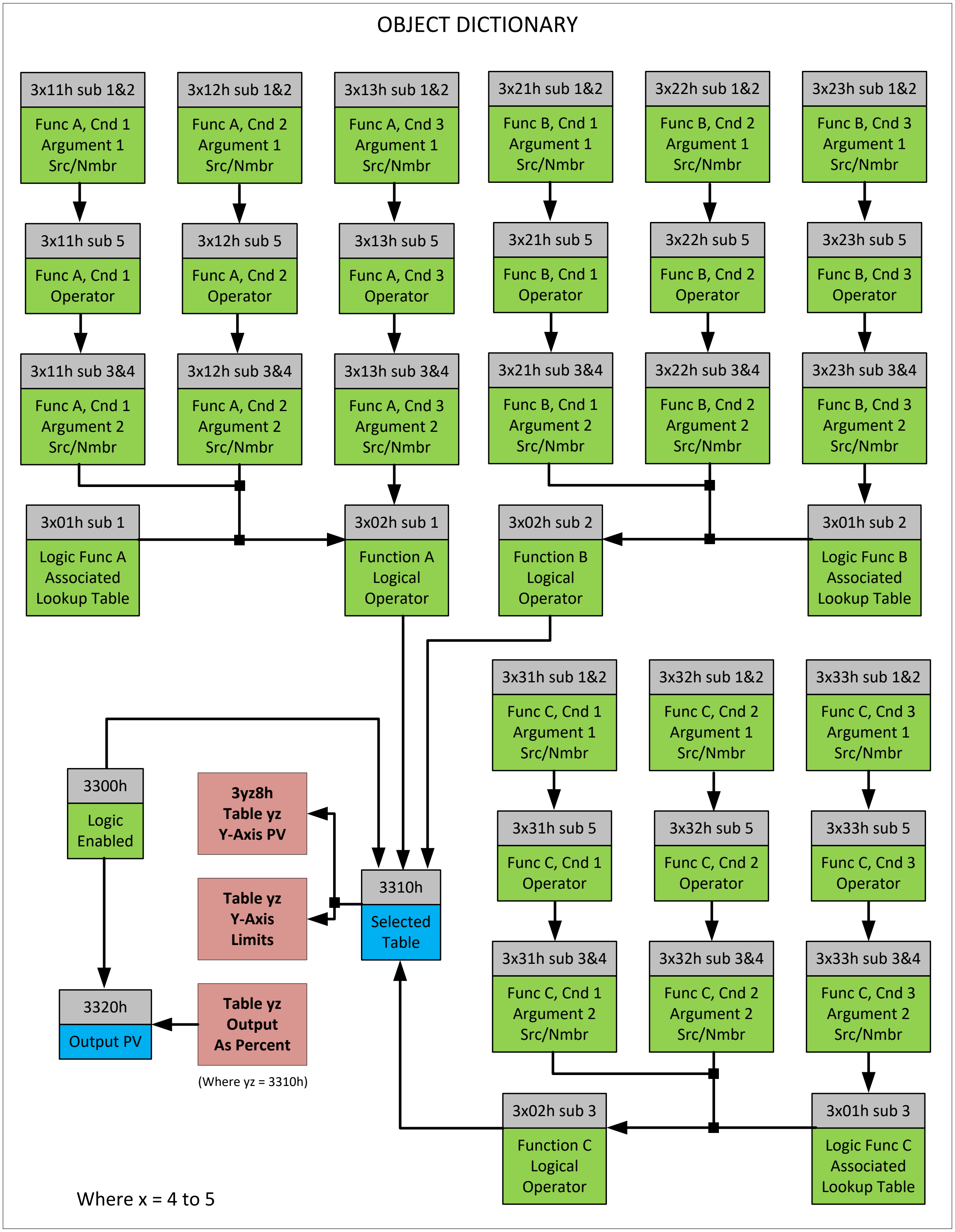


Figure 11 - Logic Block Objects

This function block is obviously the most complicated of them all, but very powerful. Any LBx (where X= 4 to 6) 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 9) can be associated with the logic, and which ones are used is fully configurable on object 3x01 **LB(x-3) Lookup Table Number**.

Should the conditions be such that a particular table (A, B or C) has been selected as described in Section 1.5.2, then the output from the selected table, at any given time, will be passed directly to LB(x-3)'s corresponding sub-index X in read-only mappable object 3320h **Logic Block Output PV**. The active table number can be read from read-only object 3310h **Logic Block Selected Table**.

Note: In this document, the term *LB(x-3)* 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 LED output. Here, the "**Control Source**" for the reactive block would be selected to be the *'Programmable Logic Function Block,'* as described in Section 1.2.

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

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

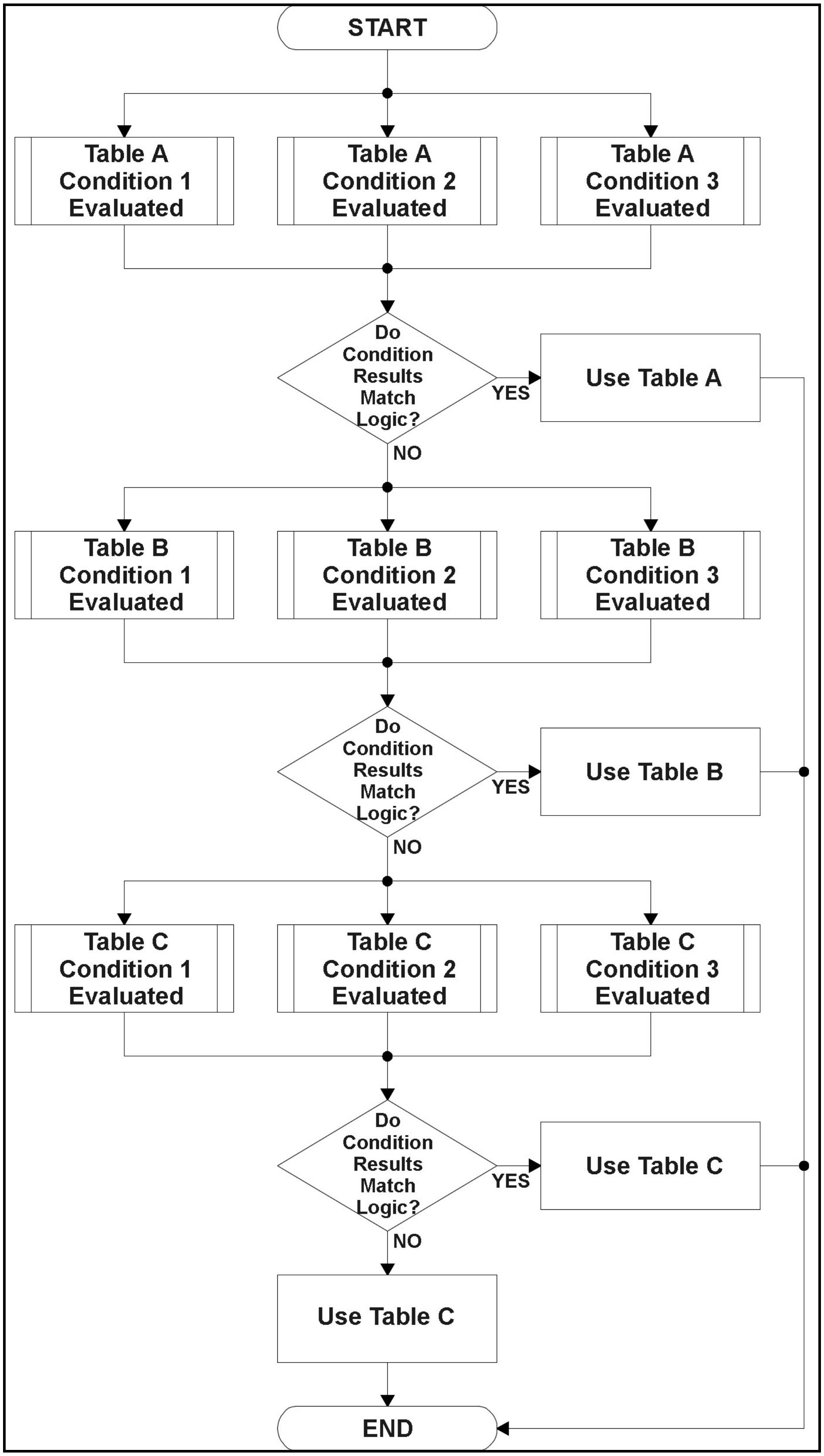


Figure 12 - Logic Block Flowchart

1.5.1. 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 13.

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 6, Y = A, B or C, and Z = 1 to 3*

Table 13 – LB(x-3) 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 listed in Table 15. 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 14, 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 14 – LB(x-3) 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 as the result. Although 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 15.

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 15 – LB(x-3) Condition Evaluation Results

1.5.2. 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 Section 1.5.1. There are several logical combinations that can be selected, as listed in Table 16. The default value for object 3x02h **LB(x-3) 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 16 – LB(x-3) 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 17.

Note: N/A Corresponds to input source Not Selected (ie, control source = 0).
Once a condition is N/A, the controller will consider the remaining conditions N/A as well. For this reason, the conditions should be set up in order from 1 to 3, to avoid undesired behaviour.

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 17 – LB(x-3) 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 set up 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 so as 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 3310h **Logic Block Selected Table.** This will change as different conditions result in different tables being used.

1.5.3. Logic Block Output

Recall that Table Y, where Y = A, B or C in the LB(x-3) function block does NOT mean lookup table 1 to 3. Each table has object 3x01h LB(x-3) **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 18.

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

Table 18 – LB(x-3) 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(x-3) 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(x-3) function block so long as that table is selected.

The LB(x-3) output is always setup as a percentage, based on the range of the Y-Axis for the associated table (see Section 1.4.2) It is written to sub-index X of read-only object 3320h **Logic Block Output PV** with a resolution of 1 decimal place.

1.6. Math Function Block

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

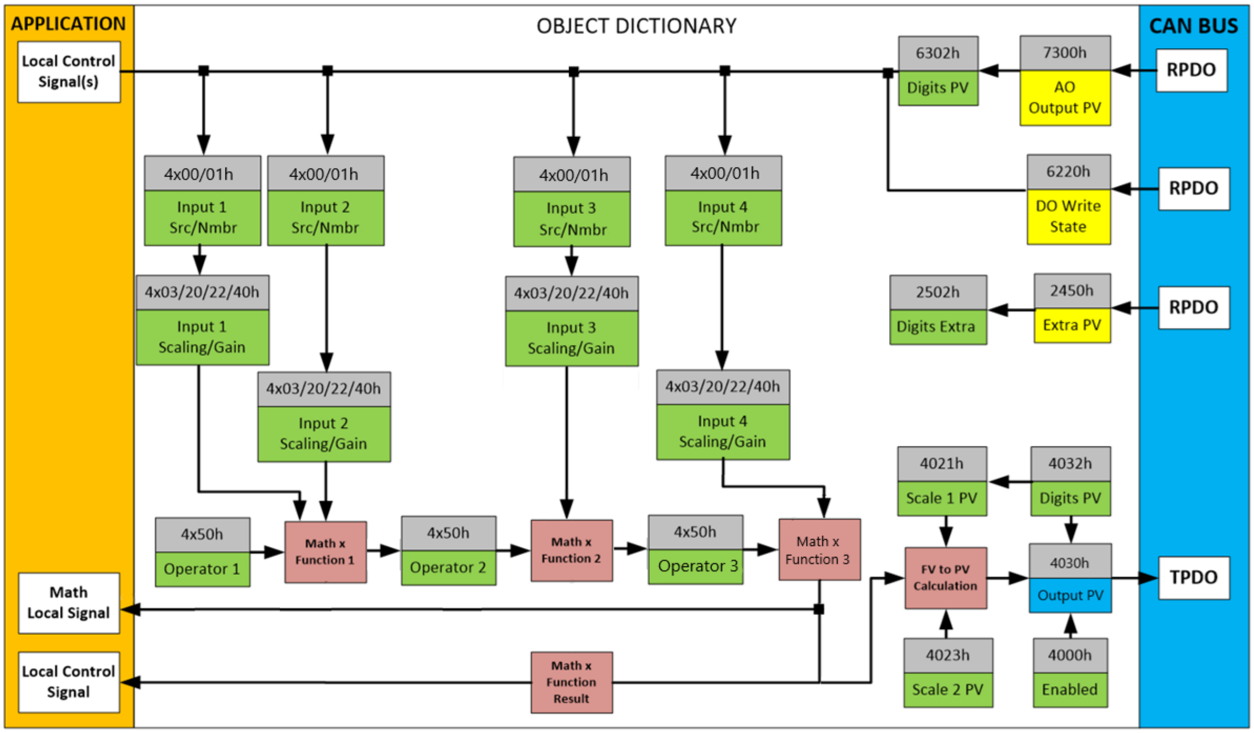


Figure 13 - Math Function Block Objects

A math function block can take up to four input signals, as listed in Table 3 in Section 1.2. Each input is then scaled according to the associated scaling and gain objects. A "Math Input X" is determined by the corresponding sub-index X = 1 to 4 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 4y03h **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 5.

Calculations are performed in the order of the source/number that comes first, as shown in Figure 13. For example, if Input 1, 2 and 3 are used, and operator 1 is addition (**4y50 Math Y Operator** subindex 1 = 12), and operator 2 is multiplication (**4y50 Math Y Operator** subindex 1 = 14), the calculation would be carried out as follows:

Result = (Input1+Input2) * Input3

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 (resulting in a gain of -1.0 to 1.0, respectively). 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 to the associated function in sub-index of InB in object 4y50h **Math Y Operator**. The list of selectable function operations is defined in Table 19.

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 19 – 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 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, as outlined in Table 5.

1.7. Conditional Logic Block

The Conditional Block 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 14 demonstrates the connections between all parameters.

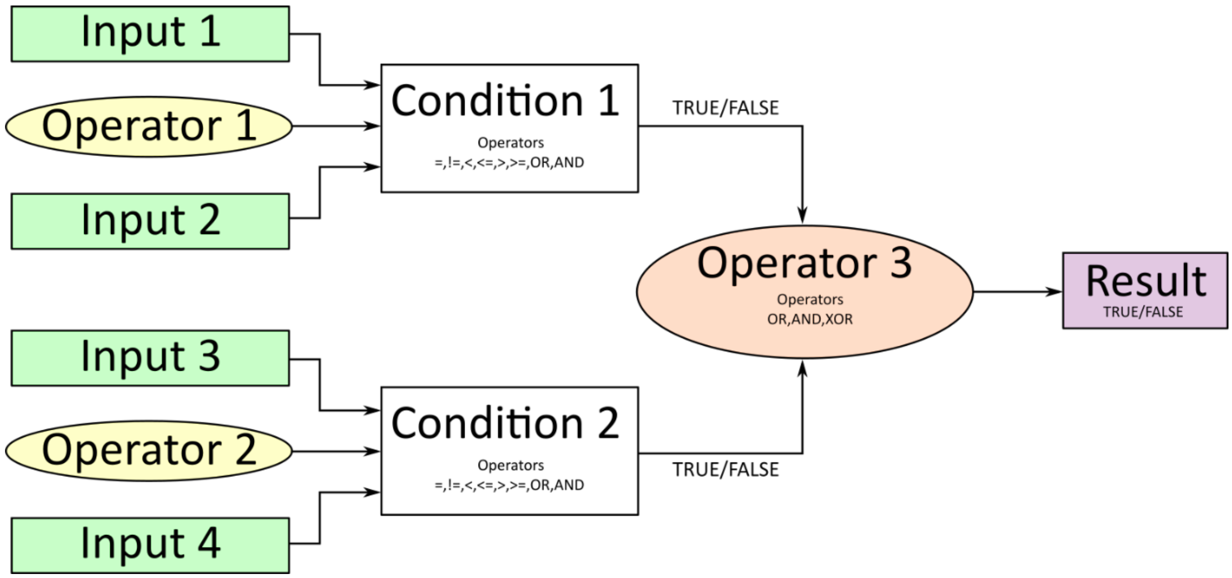


Figure 14 - Conditional Block Diagram

Each **Conditional Block [x]** offers two conditions, which are determined according to the parameters set in **Object 4Bx1h** (Condition 1) and **Object 4Bx2h** (Condition 2). Both use the defined operator to compare two inputs (argument 1 and argument 2), which can hold a logical value or an integer value. The output of the conditions can 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 output of the conditional blocks is held in the read-only object 4B10h **Conditional Logic Block Output PV**.

The value of each source will then be compared to each other with an operator of Table 20. If no source is selected, the output value of an Input will be zero. Operator 1 and Operator 2 are configured to OR by default.

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 or equal to Argument 2)
4	< (True when Argument 1 is less than Argument 2)
5	<= (True when Argument 1 is less than or equal to Argument 2)
6	OR (True when Argument 1 or Argument 2 is True)
7	NOR (True When Argument 1 and Argument 2 are False)
8	AND (True when Argument 1 and Argument 2 are True)
9	NAND (True when Argument 1 or Argument 2 are False)
10	XOR (True when Argument 1 or Argument 2 is True, but not both)
11	XNOR (True when Argument 1 is equal to Argument 2)

Table 20 – Input Operator Options

The Condition Operator used for the Conditional Logic Block is assigned using object 4B01h. The table above cannot be used for comparing the conditions because they can only be compared with logical operators, which are listed in Table 21.

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 21 – Condition Operator Options

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

1.8. Set-Reset Latch Function Block

The Set-Reset Latches are disabled by default and must first be enabled through object 3900h **SR Latch Enable** to be configured. The Set-Reset Blocks consist of only 2 control sources: Reset and Set. The purpose of these blocks is to simulate a modified latching function in which the **Reset** signal has more precedence. The latching function works as per the Table 22 below.

Set Signal	Reset Signal	SR Latch Block Output *(Initial State: OFF)*
OFF	OFF	Latched State
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF

Table 22 – Set-Reset Function Block Operation

The Reset and Set sources have minimum and maximum threshold values associated with them, which determine the ON/OFF state. Reset threshold values are assigned using object 39x3h **SR Latch [x] Reset Minimum Threshold** and object 39x4h **SR Latch [x] Reset Maximum Threshold**. Set threshold values are assigned using object 39x7h **SR Latch [x] Set Minimum Threshold** and 39x8h **SR Latch [x] Set Maximum Threshold**. These values are a configurable percentage ranging from 0% to 100%, corresponding to the selected input range. The threshold values can allow for a dead band in between the ON/OFF states.

The **Reset** signal has more precedence over the **Set** signal. If the state of the Reset signal is **ON**, the state of the SR Block Output will be **OFF** regardless of the Set signal. To create an ON output state, the Reset signal must be OFF while the Set signal is ON. Once set, the output will stay ON even if the Set signal is turned OFF, as long as the Reset signal is also OFF. As soon as the Reset signal turns ON the output will turn OFF, regardless of the state of the Set signal.

1.9. Miscellaneous Function Block

There are some other objects available which have not yet been discussed or mentioned briefly in passing (i.e. constants.) These objects are not necessarily associated with one another but are all discussed here.

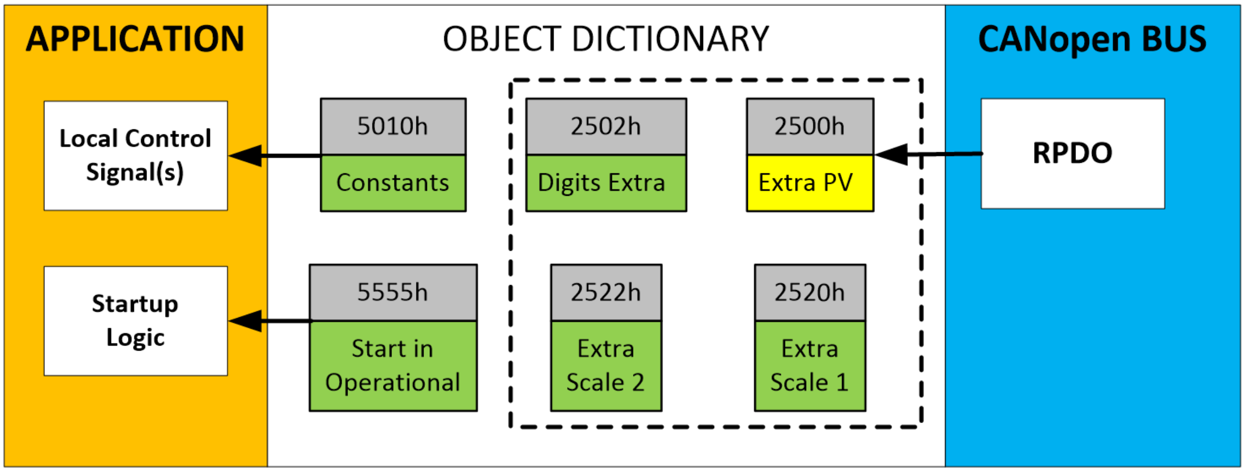


Figure 15 - Miscellaneous Objects

Extra RPDO Messages

Objects 2500h **Extra Control Received PV**, 2502h **EC Decimal Digits PV**, 2502h **EC Scaling 1 PV** and **EC Scaling 2 PV** have been mentioned in Section 1.2, Table 4. These objects allow for additional data received on a CANopen® RPDO to be mapped independently to various function blocks as a control source. For example, a PID loop must have two inputs (target and feedback), so one of them has to come from the CAN bus. The scaling objects are provided to define the limits of the data when it is used by another function block, as shown in Table 5.

Constant Values

Object 5010h **Constant Field Value** is provided to give the user the option for a fixed value that can be used by other function blocks. Sub-index 1 is fixed as FALSE (0) and sub-index 2 is always TRUE (1). There are 13 other sub-indexes provided for user selectable values.

The constants are read as 32-bit real (float) data, so no decimal digit object is provided. When setting up the constant, make sure to do it with the resolution of the object that will be compared with it.

The False/True constants are provided primarily to be used with the logic block. The variable constants are also useful with logic or math blocks, and they can also be used as a setpoint target for a PID control block.

Fault Detection Objects

Object 5040h **FD Field Value** is a read only object containing the field values of the over temperature, over and under voltage. Object 5041h **FD Set Threshold** sets the limit values for which the faults occur when reached. When any of these thresholds are reached, the faults will clear when the values have lowered to values set in object 5042h **FD Clear Threshold.**

For the I/O controller to begin monitoring fault detections, object 5050h **Error Check Detection** determines which Fault Detection is enabled through 1 byte data as bits. Once a fault is detected, object 5051h **Error Response Delay** will determine how long the fault needs to be present to flag and error.

Automatic Update of Objects

Object 5550h **Enable Automatic Updates** allows for the controller to automatically update the objects related to the output to defaults when it is changed. By default, this object is set to TRUE, in which case the objects are set to their default values depending on the type selected.

On the other hand, when this object is FALSE, the objects are not set to defaults and are left with the same values previous to changing the type. In this case, these are to be configured manually.

Startup

The last object 5555h **Start in Operational** is provided as a ‘cheat’ when the unit is not intended to work with a CANopen® network (i.e. a stand-alone control) or is working on a network comprised solely as slaves so the OPERATION command will never be received from a master. By default, this object is disabled (FALSE).

When using the ECU as a stand-alone controller where 5555h is set to TRUE, it is recommended to disable all TPDOs (set the Event Timer to zero) so that it does not run with a continuous CAN error when not connected to a bus.

1.10. Dimensions and Pinout

The LED Output Controller is packaged in a plastic housing from TE Deutsch. The assembly carries an IP67 rating.

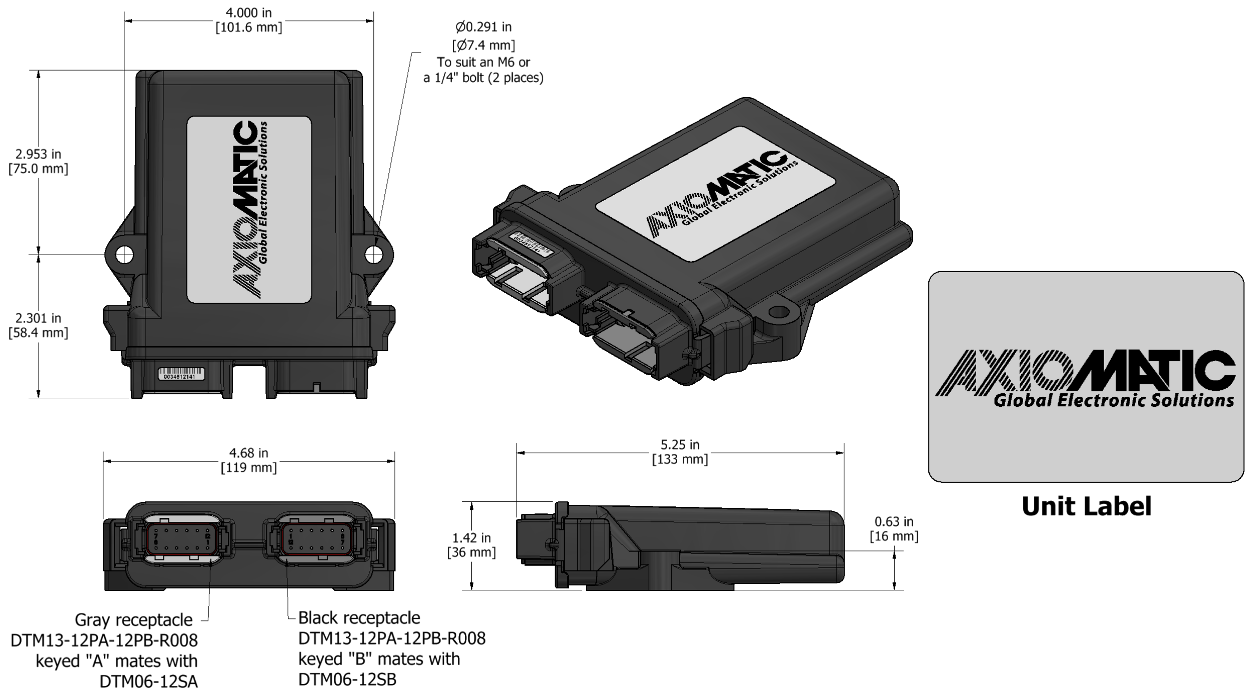
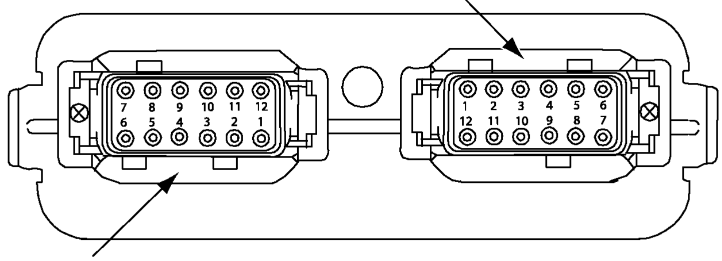


Figure 16 - Housing Dimensions



FRONT VIEW 24 PIN RECEPTACLE

Table 23 – Connector Pinout

24-pin TE Deutsch receptacle P/N: DTM13-12PA-12PB-R008			
Grey Connector		**Black Connector**	
Pin	**Function**	**Pin**	**Function**
1	LED Output 5 +	1	LED Output 4 -
2	LED Output 6 +	2	LED Output 3 -
3	LED Output 7 +	3	LED Output 2 -
4	LED Output 8 +	4	LED Output 1 -
5	LED Output 9 +	5	CAN Low
6	LED Output 10 +	6	Battery -
7	LED Output 10 -	7	Battery +
8	LED Output 9 -	8	CAN High
9	LED Output 8 -	9	LED Output 1 +
10	LED Output 7 -	10	LED Output 2 +
11	LED Output 6 -	11	LED Output 3 +
12	LED Output 5 -	12	LED Output 4 +

2. CANOPEN ® OBJECT DICTIONARY

The CANopen ® object dictionary of the CAN-10LEDOUT Controller is based on CiA device profile DS-404 V1.2 (device profile for Closed Loop Controllers). The object dictionary includes Communication Objects beyond the minimum requirements in the profile, as well as several manufacturer-specific objects for extended functionality.

2.1. NODE ID and BAUDRATE

By default, the Controller ships factory programmed with a Node ID = 127 (0x7F) and with Baud rate = 125 kbps.

2.1.1. 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.

2.1.1.1. 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)

2.1.1.2. Setting Baudrate

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

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

- Set the baudrate by **sending** the following message:

Item	***Value***
COB-ID	0x7E5
Length	3
Data 0	0x13 (cs=19 for configure bit timing parameters)
Data 1	0x00 (switches to waiting state)
Data 2	Index (select baudrate index per Table 24)

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

Table 24 – LSS Baudrate Indices

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

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

- Activate bit timing parameters by **sending** the following message:

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

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

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

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

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

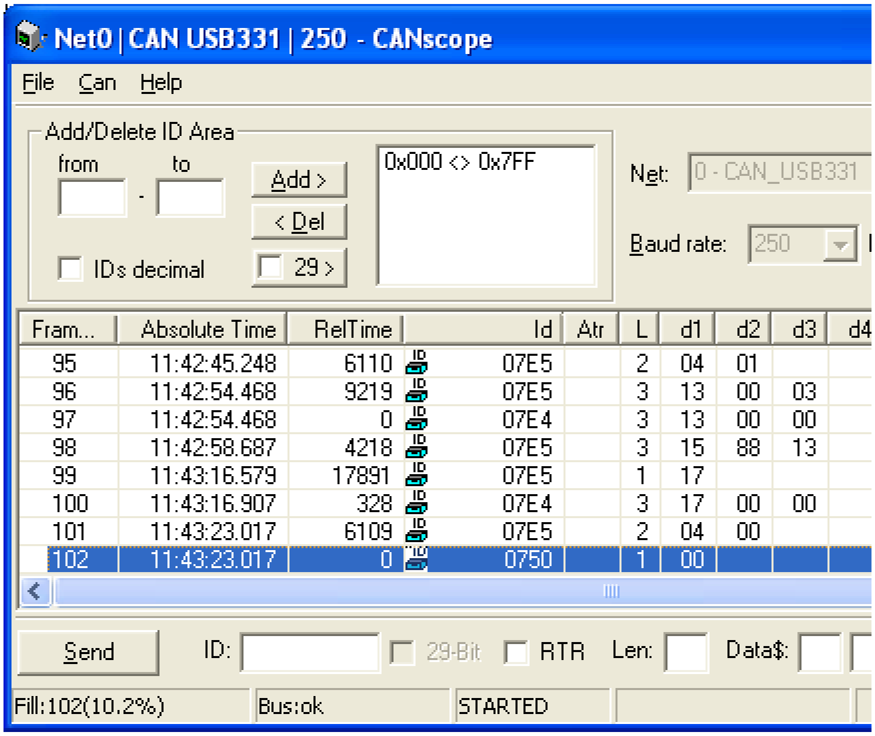
Item	***Value***
COB-ID	0x7E4
Length	3
Data 0	0x17 (cs=23 for store configuration)
Data 1	0x00
Data 2	0x00

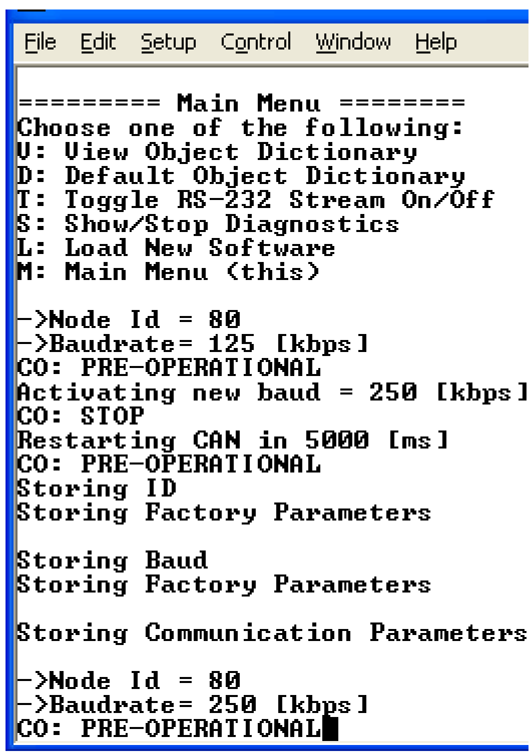
- Set the module state to LSS-operation by **sending** the following message:
  (Note, the module will reset itself back to the pre-operational state)

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

The following screen capture (left) shows the CAN data was sent (7E5h) and received (7E4h) by the tool when the baud rate was changed to 250 kbps using the LSS protocol. The other image (right) shows what was printed on an example debug RS-232 menu while the operation took place.

Between CAN Frame 98 and 99, the baud rate on the CAN Scope tool was changed from 125 to 250 kbps.


Net0 | CAN USB331 | 250 - CANscope
File Can Help
Add/Delete ID Area
from to
Add >
< Del
IDs decimal
29 >
0x000 <> 0x7FF
Net: 0 - CAN_USB331
Baud rate: 250
Fram... Absolute Time RelTime Id Atr L d1 d2 d3 d4
95 11:42:45.248 6110 07E5 2 04 01
96 11:42:54.468 9219 07E5 3 13 00 03
97 11:42:54.468 0 07E4 3 13 00 00
98 11:42:58.687 4218 07E5 3 15 88 13
99 11:43:16.579 17891 07E5 1 17
100 11:43:16.907 328 07E4 3 17 00 00
101 11:43:23.017 6109 07E5 2 04 00
102 11:43:23.017 0 0750 1 00
Send ID: 29-Bit RTR Len: Data$:
Fill:102(10.2%) Bus:ok STARTED


File Edit Setup Control Window Help
========= Main Menu ========
Choose one of the following:
V: View Object Dictionary
D: Default Object Dictionary
T: Toggle RS-232 Stream On/Off
S: Show/Stop Diagnostics
L: Load New Software
M: Main Menu (this)
->Node Id = 80
->Baudrate= 125 [kbps]
CO: PRE-OPERATIONAL
Activating new baud = 250 [kbps]
CO: STOP
Restarting CAN in 5000 [ms]
CO: PRE-OPERATIONAL
Storing ID
Storing Factory Parameters
Storing Baud
Storing Factory Parameters
Storing Communication Parameters
->Node Id = 80
->Baudrate= 250 [kbps]
CO: PRE-OPERATIONAL

2.2. COMMUNICATION OBJECTS (DS-301 and DS-404)

The communication objects supported by the CAN-10LEDOUT 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
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 Behaviour	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
1404	RPDO5 Communication Parameter	RECORD		RW	No
1405	RPDO6 Communication Parameter	RECORD		RW	No
1406	RPDO5 Communication Parameter	RECORD		RW	No
1407	RPDO6 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

1604	RPDO5 Mapping Parameter	RECORD		RO	No
1605	RPDO6 Mapping Parameter	RECORD		RO	No
1606	RPDO5 Mapping Parameter	RECORD		RO	No
1607	RPDO6 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
1804	TPDO5 Communication Parameter	RECORD		RW	No
1805	TPDO6 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
1A04	TPDO5 Mapping Parameter	RECORD		RW	No
1A05	TPDO6 Mapping Parameter	RECORD		RW	No

2.2.1. Object 1000h: Device Type

This object contains information about the device type as per device profile DS-404. The 32-bit parameter is divided into two 16-bit values, showing General and Additional information as shown below.

MSB	LSB
Additional Information = 0xE01C	**General Information** = 0x0194 (404)

DS-404 defines the Additional Information field in the following manner:
0000h = reserved
0001h = digital input block
0002h = analog input block
0004h = digital output block
0008h = analog output block
0010h = controller block (aka PID)
0020h = alarm block
0040h … 0800h = reserved
1000h = reserved
2000h = lookup table block (manufacturer-specific)
4000h = programmable logic block (manufacturer-specific)
8000h = miscellaneous block (manufacturer-specific)

Object Description

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

Entry Description

Access	RO
PDO Mapping	No
Value Range	0xE0180194
Default Value	0xE0180194

2.2.2. 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 is no errors in the module will this bit will be cleared. No other bits in this register are used by the Controller.

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

2.2.3. Object 1002h: Manufacturer Status Register

This object is used for manufacturer debugging purposes.

2.2.4. Object 1003h: Pre-Defined Error Field

This object 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 sub-index 1, with sub-index 0 containing the number of errors currently on the list. When the device is in an error-free state, the value of sub-index 0 is zero.

The error list may be cleared by writing a zero to sub-index 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 4 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 on which the error occurred.

MSB LSB

Error Description	Channel-ID	EMCY Error Code

If node-guarding is used (not recommended per the latest standard) and a lifeguard event occurs, the manufacturer-specific field will be set to 0x1000. On the other hand, if a heartbeat consumer fails to be received within the expected timeframe, the Error Description will be set to 0x80, and the Channel-ID (nn) will reflect the Node-ID of the consumer channel that was not producing. In this case, the manufacturer-specific field will therefore be 0x80nn. In both cases, the corresponding EMCY Error Code will be the Guard Error 0x8130.

When an analog output is not working as described in Section 1.2, then the Error Description will reflect what channel(s) is at fault using the following table. Also, if an RPDO is not received within the expected “Event Timer” period, an RPDO timeout will be flagged. Table 25 outlines the resulting Error Field Codes and their meanings.

Error Field Code	Error Description	Meaning	ID	Meaning	EMCY Code	Meaning
00000000h	EMCY Error Reset (fault no longer active)					
400yF001h	40h	Positive Overload (Out-of-range High)	0yh	Universal Input y	F001h	Input Overload
500yF001h	50h	Negative Overload (Out-of-range Low)	0yh	Universal Input y	F001h	Input Overload
100yF002h	10h	Sensor Break (Open Circuit on AO)	0yh	Analog Output y	F002h	Output Overload
000y2310h	00h	Short Circuit on AO	0yh	Analog Output y	2310h	Current at Output too High (short to GND or Vcc)
40003100h	40h	Positive Overload (Vps Overvoltage)	00h	Unspecified	3100h	Device Voltage
50003100h	50h	Negative Overload (Vps Undervoltage)	00h	Unspecified	3100h	Device Voltage
40004200h	40h	Positive Overload (Over Temp)	00h	Unspecified	4200h	Device Temperature
00008100h	00h	RPDO Timeout	00h	Unspecified	8100h	Communication - generic
10008130h	10h	Lifeguard Event	00h	Unspecified	8130h	Lifeguard/ Heartbeat Error
80nn8130h	80h	Heartbeat Timeout	nn	Node-ID	8130h	Lifeguard/ Heartbeat Error
00008140h	00h	Bus OFF Event	00h	Unspecified	8400h	Bus OFF Recovery

Table 25 – Pre-Defined Error Field Codes

Object Description

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

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RW
PDO Mapping	No
Value Range	0 to 15
Default Value	0

Sub-Index	1h to 15
Description	Standard error field
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

2.2.5. Object 100Ch: Guard Time

The objects at index 100Ch and 100Dh shall indicate the configured guard time respective to the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol described in DS-301. The Guard Time value shall be given in multiples of ms, and a value of 0000h shall disable the life guarding.

It should be noted that this object, and that of 100Dh are only supported for backwards compatibility. The standard recommends that newer networks do not use the life guarding protocol, but rather heartbeat monitoring instead. Both life guarding and heartbeats can NOT be active simultaneously.

Object Description

Index	100Ch
Name	Guard Time
Object Type	VAR
Data Type	UNSIGNED16

Entry Description

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

2.2.6. Object 100Dh: Lifetime Factor

The life time factor multiplied with the guard time gives the life time for the life guarding protocol. A value of 00h shall disable life guarding.

Object Description

Index	100Dh
Name	Life time factor
Object Type	VAR
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 255
Default Value	0

2.2.7. 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 sub-index. 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 sub-index, 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. For all sub-indexes, this value is 1h, indicating that the Controller saves parameters on command. **This means that if power is removed before the Store object is written, changes to the Object Dictionary will NOT have been saved in the non-volatile memory and will be lost on the next power cycle.**

Object Description

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

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	8
Default Value	8

Sub-Index	1h
Description	Save all parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

Sub-Index	2h
Description	Save communication parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

Sub-Index	3h
Description	Save application parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

Sub-Index	4h
Description	Save manufacturer parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

2.2.8. Object 1011h: Restore 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 sub-index. 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 sub-index, the Controller will restore the defaults in non-volatile memory, and then confirm the SDO transmission. **The default values are set valid only after the device is reset or power-cycled.** This means that the Controller will NOT start using the default values right away, but rather continue to run from whatever values were in the Object Dictionary prior to the restore operation.

By read access, the object provides information about the module's default parameter restoring capabilities. For all sub-indexes, this value is 1h, indicating that the Controller restores defaults on command.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	Restore all default parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

Sub-Index	2h
Description	Restore default communication parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

Sub-Index	3h
Description	Restore default application parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

Sub-Index	4h
Description	Restore default manufacturer parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

2.2.9. Object 1016h: Consumer Heartbeat Time

The Controller can be a consumer of heartbeat objects for a single module. This object defines the expected heartbeat cycle time for that module, and if set to zero, it is not used. When the object 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

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
Description	Consumer heartbeat time
Access	RW
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

2.2.10. 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

2.2.11. 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 sub-index 3, the format of the data is as shown below

MSB		LSB
Major revision number (object dictionary)	Hardware Revision	Software Version

Object Description

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

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

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

Sub-Index	2h
Description	Product Code
Access	RO
PDO Mapping	No
Value Range	0xAA130761
Default Value	0xAA130761

Sub-Index	3h
Description	Revision Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00000000

Sub-Index	4h
Description	Serial Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

2.2.12. Object 1020h: Verify Configuration

This object can be read to see what date the software (version identified in object 1018h) was compiled. The date is represented as a hexadecimal value showing day/month/year as per the format below. The time value at sub-index 2 is a hexadecimal value showing the time in a 24 hour clock

MSB LSB

Day (in 1-Byte Hex)	Month (in 1-Byte Hex)	Year (in 2-Byte Hex)
00	00	Time (in 2-Byte Hex)

For example, a value of 0x30042014 would indicate that the software was compiled on April 30th, 2014. A time value of 0x00001842 would indicate it was compiled at 6:42pm.

Object Description

Index	1020h
Name	Verify configuration
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1h
Description	Configuration date
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2h
Description	Configuration time
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

2.2.13. 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 sub-index.

Note: Object 1029h Error Behaviour can ONLY be changed if there are NO errors present (1001h = 1). Ensure errors are cleared or disabled to change this object.

A network fault is flagged when an RPDO is not received within the expected time period defined in the "Event Timer" of the associated communication objects, (see Section 2.2.14 for more information) or if a lifeguard or heartbeat message is not received as expected. Output faults are defined in Section 1.2. Power Supply faults are described in Section 1.9.

For all sub-indexes, the following definitions hold true:

0 = Pre-Operational (node reverts to a pre-operational state when this fault is detected)
1 = No State Change (node remains in the same state it was in when the fault occurred)
2 = Stopped (node goes into stopped mode when the fault occurs)

Object Description

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

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	Communication Fault
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	0 (Pre-Operational)

Sub-Index	2h
Description	Digital Input Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	3h
Description	Analog Input Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	4h
Description	Digital Output Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	5h
Description	Analog Output Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	6h
Description	Fault Detection Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

2.2.14. RPDO Behaviour

Per the CANopen® standard DS-301, the following procedure shall be used for re-mapping, and is the same for both RPDOs and TPDOs.

a) Destroy the PDO by setting bit **exists** (most significant bit) of sub-index 01h of the according PDO communication parameter to 1b
b) Disable mapping by setting sub-index 00h of the corresponding mapping object to 0
c) Modify the mapping by changing the values of the corresponding sub-indices
d) Enable mapping by setting sub-index 00h to the number of mapped objects
e) Create the PDO by setting bit **exists** (most significant bit) of sub-index 01h of the according PDO communication parameter to 0b

The Controller can support up to six RPDO messages. All RPDOs on the Controller use the similar default communication parameters, with the PDO IDs set according to the pre-defined connection set described in DS-301. Most RPDOs do not exist, there is no RTR allowed, they use 11-bit CAN-IDs (base frame valid) and they are all event-driven. While all six have valid default mappings defined (see below) only RPDO1 is enabled by default (i.e. RPDO exists).

RPDO1 Mapping at Object 1600h: Default ID 0x200 + Node ID

Sub-Index	***Value***	***Object***
0	1	Number of mapped application objects in PDO
1	0x73000110	Analog Output 1 Process Value
2	0x73000210	Analog Output 2 Process Value
3	0x73000310	Analog Output 3 Process Value
4	0x73000410	Analog Output 4 Process Value

RTPDO2 Mapping at Object 1601h: Default ID 0x300 + Node ID

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x73000510	Analog Output 5 Process Value
2	0x73000610	Analog Output 6 Process Value
3	0x73000710	Analog Output 7 Process Value
4	0x73000810	Analog Output 8 Process Value

RPDO3 Mapping at Object 1602h: Default ID 0x400 + Node ID

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x73000910	Analog Output 9 Process Value
2	0x73000A10	Analog Output 10 Process Value
3	0x27000110	PWM Duty Cycle Control 1 PV
4	0x27000210	PWM Duty Cycle Control 2 PV

RPDO4 Mapping at Object 1603h: Default ID 0x500 + Node ID

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x27000310	PWM Duty Cycle Control 3 PV
2	0x27000410	PWM Duty Cycle Control 4 PV
3	0x27000510	PWM Duty Cycle Control 5 PV
4	0x27000610	PWM Duty Cycle Control 6 PV

RPDO5 Mapping at Object 1604h: Default ID 0x200 + Node ID + 1

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x27000710	PWM Duty Cycle Control 7 PV
2	0x27000810	PWM Duty Cycle Control 8 PV
3	0x27000910	PWM Duty Cycle Control 9 PV
4	0x27000A10	PWM Duty Cycle Control 10 PV

RPDO6 Mapping at Object 1605h: Default ID 0x300 + Node ID + 1

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

RPDO7 Mapping at Object 1606h: Default ID 0x400 + Node ID + 1

Sub-Index	***Value***	***Object***
0	3	Number of mapped application objects in PDO
1	0x25000510	Extra Received 5 PV
2	0x25000610	Extra Received 6 PV
3	0x25000710	Extra Received 7 PV
4	0x25000810	Extra Received 8 PV

RPDO8 Mapping at Object 1607h: Default ID 0x500 + Node ID + 1

Sub-Index	***Value***	***Object***
0	3	Number of mapped application objects in PDO
1	0x25000910	Extra Received 9 PV
2	0x25000A10	Extra Received 10 PV
3	0	
4	0	

None of them have the timeout feature enabled, i.e. the “Event Timer” on sub-index 5 is set to zero. When this is changed to a non-zero value, if the RPDO has not been received from another node within the time period defined (while in Operational mode), a network fault is activated, and the controller will go to the operational state define in Object 1029h sub-index 4.

Object Description

Index	1400h to 1407h
Name	RPDO communication parameter
Object Type	RECORD
Data Type	PDO Communication Record

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	COB-ID used by RPDO
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	40000000h + RPDO1 + Node ID C0000000h + RPDOx + Node-ID

X	***RPDOx ID***
1	0200h
2	0300h
3	0400h
4	0500h
5	0201h
6	0301h

Node-ID = Node-ID of the module. The RPDO COB-IDs are automatically updated if the Node-ID is changed by LSS protocol.
80000000h in the COB-ID indicates that the PDO does not exist (destroyed)
04000000h in the COB-ID indicates that there is no RTR allowed on the PDO

Sub-Index	2h
Description	Transmission type
Access	RO
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	255 (FFh) = Event Driven

Sub-Index	3h
Description	Inhibit Time
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

Sub-Index	4h
Description	Compatibility entry
Access	RW
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	5
Description	Event-timer
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

Recall: A non-zero event timer for an RPDO means that it will result in a network fault being flagged if it has not been received within this timeframe while in Operational mode.

2.2.15. TPDO Behaviour

The Controller can support up to eight TPDO messages. All TPDOs on the Controller use the similar default communication parameters, with the PDO IDs set according to the pre-defined connection set described in DS-301. Most TPDOs do not exist, there is no RTR allowed, they use 11-bit CAN-IDs (base frame valid) and they are all time-driven. While all have valid default mappings defined (see below) only TPDO1 to TPDO3 are enabled by default (i.e. TPDO exists).

TPDO1 Mapping at Object 1A00h: Default ID 0x180 + Node ID

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x73300110	Analog Output 1 Field Value
2	0x73300210	Analog Output 2 Field Value
3	0	
4	0	

TPDO2 Mapping at Object 1A01h: Default ID 0x280 + Node ID

Sub-Index	***Value***	***Object***
0	2	Number of mapped application objects in PDO
1	0x50400110	Processor Temperature Measured
2	0x50400210	Power Supply Measured
3	0	
4	0	

TPDO3 Mapping at Object 1A02h: Default ID 0x380 + Node ID

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0x27300110	PWM Duty Cycle 1 Field Value
2	0x27300210	PWM Duty Cycle 2 Field Value
3	0	
4	0	

TPDO4 Mapping at Object 1A03h: Default ID 0x480 + Node ID

Sub-Index	Value	Object
0	4	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

TPDO5 Mapping at Object 1A04h: Default ID 0x180 + Node ID + 1

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

TPDO6 Mapping at Object 1A05h: Default ID 0x280 + Node ID + 1

Sub-Index	***Value***	***Object***
0	4	Number of mapped application objects in PDO
1	0	
2	0	
3	0	
4	0	

Since only TPDO1 has a non-zero value transmission rate (i.e. Event Timer in sub-index 5 of communication object), only these TPDOs will be automatically broadcasted when the unit goes into OPERATIONAL mode.

Object Description

Index	1800h to 1807h
Name	TPDO communication parameter
Object Type	RECORD
Data Type	PDO Communication Record

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	COB-ID used by TPDO
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	40000000h + TPDOx + Node-ID C0000000h + TPDOy + Node-ID

X	***TPDOx ID***	***Y***	***TPDOy ID***
1	0180h	4	0480h
2	0280h	5	0181h
3	0380h	6	0281h
		7	0381h
		8	0481h

Node-ID = Node-ID of the module. The TPDO COB-IDs are automatically updated if the Node-ID is changed by LSS protocol.
80000000h in the COB-ID indicates that the PDO does not exist (destroyed)
04000000h in the COB-ID indicates that there is no RTR allowed on the PDO

Sub-Index	2h
Description	Transmission type
Access	RO
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	254 (FEh) = Event Driven

Sub-Index	3h
Description	Inhibit Time
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

Sub-Index	4h
Description	Compatibility entry
Access	RW
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	5
Description	Event-timer
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	250ms (on TPDO1, TPDO2, TPDO3) 0ms (on TPDO4 to TPDO8)

2.3. APPLICATION OBJECTS (DS-404)

Index (hex)	*Object*	*Object Type*	*Data Type*	*Access*	*PDO Mapping*
7300	AO Output Process Value	ARRAY	INTEGER32	RW	Yes
6302	AO Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
6310	AO Output Type	ARRAY	UNSIGNED16	RW	No
7320	AO Output Scaling 1 PV	ARRAY	INTEGER32	RW	No
7321	AO Output Scaling 1 FV	ARRAY	INTEGER32	RW	No
7322	AO Output Scaling 2 PV	ARRAY	INTEGER32	RW	No
7323	AO Output Scaling 2 FV	ARRAY	INTEGER32	RW	No
7330	AO Output Field Value	ARRAY	INTEGER32	RO	Yes
6332	AO Decimal Digits FV	ARRAY	UNSIGNED8	RW	No
6340	AO Fault Mode	ARRAY	UNSIGNED8	RW	No
7341	AO Fault Field Value	ARRAY	INTEGER32	RW	No
7450	PID Proportional Band	ARRAY	INTEGER16	RW	No
7452	PID Integral Action Time	ARRAY	INTEGER16	RW	No
7454	PID Derivative Action Time	ARRAY	INTEGER16	RW	No
7456	PID Cycle Time	ARRAY	INTEGER16	RW	No
6458	PID Physical Unit Timing	ARRAY	UNSIGNED32	RO	No
6459	PID Decimal Digits Timing	ARRAY	UNSIGNED8	RW	No

2.3.1. Object 7300h: AO Output Process Value

This object represents the process value of the output direct current. It can be used as an input to the analog output function block when the input has been selected as controlled by a CANopen® Message (per Table 3).

Object Description

Index	7300h
Name	Analog Output Process Value
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Process Value
Access	RW
PDO Mapping	Yes
Value Range	Integer32
Default Value	No

2.3.2. Object 6302h: AO Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the output control data, which is interpreted with data type Unsigned Integer8 in the process value object.

Object Description

Index	6302h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	0

2.3.3. Object 6310h: AO Output Type

This object specifies the type of analog output, as defined in Table 1.

Object Description

Index	6310h
Name	AO Output Type
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Type
Access	RW
PDO Mapping	No
Value Range	See Table 1
Default Value	20 (Current)

2.3.4. Object 7320h: AO Output Scaling 1 PV

This object defines the minimum value of the input and should be specified to equal the corresponding scaling object of the control source, as outlined in Table 5. It will be scaled in the physical unit of the control source. The resolution will ALWAYS be dependent on object 6302h AO Decimal Digits PV, even when the output is not being controlled directly by the AO Output PV object 7300h. This object must always be smaller than object 7322h AO Output Scaling 2 PV.

Object Description

Index	7320h
Name	AO Output Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	See Table 5
Default Value	80 [mA]

2.3.5. Object 7321h: AO Output Scaling 1 FV

This object defines the output field value when the input data is at or below the AO Output Scaling 1 PV value as shown in Figure 3. It will be scaled in the physical unit of the output, dependent on type, with the resolution defined in object 6332h AO Decimal Digits FV. The value can be set anywhere within the allowable output range as outlined in Table 1. This value can be set higher than object 7323h AO Output Scaling 2 FV for an inverse response (i.e. decreasing) to an increasing input.

Object Description

Index	7321h
Name	AO Output Scaling 1 FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	80 [mA]

2.3.6. Object 7322h: AO Output Scaling 2 PV

This object defines the maximum value of the input and should be specified to equal the corresponding scaling object of the control source, as outlined in Table 5. It will be scaled in the physical unit of the control source. The resolution will ALWAYS be dependent on object 6302h AO Decimal Digits PV, even when the output is not being controlled directly by the AO Output PV object 7300h. This object must always be larger than object 7322h AO Output Scaling 2 PV.

Object Description

Index	7322h
Name	AO Output Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	See Table 5
Default Value	600 [mA]

2.3.7. Object 7323h: AO Output Scaling 2 FV

This object defines the output field value when the input data is at or above the AO Output Scaling 2 PV value as shown in Figure 3. It will be scaled in the physical unit of the output, dependent on type, with the resolution defined in object 6332h AO Decimal Digits FV. The value can be set anywhere within the allowable output range as outlined in Table 1. This value can be set lower than object 7321h AO Output Scaling 1 FV for an inverse response (i.e. decreasing) to an increasing input.

Object Description

Index	7323h
Name	AO Output Scaling 2 FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	600 [mA]

2.3.8. Object 7330h: AO Output Field Value

This object represents the target output drive field value as a result of the output logic described in Section 1.2, and the scaling applied as shown in Figure 3. It is defined in the physical unit of the output dependent on type, as outlined in Table 1. The resolution of the object is defined in object 6332h AO Decimal Digits FV.

Object Description

Index	7330h
Name	Analog Output Field Value
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Field Value
Access	RO
PDO Mapping	Yes
Value Range	Integer32
Default Value	No

2.3.9. Object 6332h: AO Decimal Digits FV

This object describes the number of digits following the decimal point (i.e. resolution) of the output data, which is interpreted with data type Unsigned Integer8 in the field value object.

Object Description

Index	6332h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Decimal Digits FV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	0

2.3.10. Object 6340h: AO Fault Mode

This object defines how an output shall respond when a fault condition is detected on any control input, as described in Table 9.

Object Description

Index	6340h
Name	AO Fault Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Fault Mode
Access	RW
PDO Mapping	No
Value Range	See Table 9
Default Value	1 (apply pre-defined FV)

2.3.11. Object 7341h: AO Fault Field Value

This object contains the pre-defined field value of an analog output when a fault condition is present, and the corresponding sub-index in object 6340h is enabled. It will be scaled in the physical unit of the output, dependent on type, with the resolution defined in object 6332h AO Decimal Digits FV.

Object Description

Index	7341h
Name	AO Fault Field Value
Object Type	ARRAY
Data Type	INTEGER32

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 Ah (x=1 to 10)
Description	AOx Fault Field Value
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	0 mA

2.3.12. Object 7450h: PID Proportional Band

This object describes the proportional band gain (G in Figure 7) of the PID algorithm. The value is always interpreted as having a resolution of one digit after the decimal place.

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	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	PIDx Proportional Gain
Access	RW
PDO Mapping	No
Value Range	0 to 100 (0 to 10.0)
Default Value	5 [0.5]

2.3.13. Object 7452h: PID Integral Action Time

This object describes the integral time (Ti in Figure 7) of the PID algorithm. The physical unit is always seconds, as defined in object 6458h, with the decimal digits (resolution) given in object 6459h. To prevent instability, it is recommended to never set this less than three times higher than object 7454h (Td in Figure 7).

Object Description

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

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	PIDx Integral Time
Access	RW
PDO Mapping	No
Value Range	0.001 [sec] to 1000.00 [sec]
Default Value	5 [ms or 0.005 sec]

2.3.14. Object 7454h: PID Derivative Action Time

This object describes the derivative time (Td in Figure 7) of the PID algorithm. The physical unit is always seconds, as defined in object 6458h, with the decimal digits (resolution) given in object 6459h. To prevent instability, it is recommended to never set this more than three times smaller than object 7452h (Ti in Figure 7).

Object Description

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

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	PIDx Derivative Time
Access	RW
PDO Mapping	No
Value Range	0.001 [sec] to 1000.00 [sec]
Default Value	1 [ms or 0.001sec]

2.3.15. Object 7456h: PID Cycle Time

This object defines how frequently the PID loop is called. The physical unit is always seconds, as defined in object 6458h, with the decimal digits (resolution) given in object 6459h.

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	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	PIDx Cycle Time
Access	RW
PDO Mapping	No
Value Range	0.001 [sec] to 1000.00 [sec]
Default Value	10 [ms or 0.010sec]

2.3.16. Object 6458h: PID Physical Unit Timing

This read-only object defines the physical unit of objects 7452h, 7454h and 7456h. It represents “seconds” as the unit used in all cases.

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	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	PIDx Physical Unit Timing
Access	RO
PDO Mapping	No
Value Range	0003 0000h
Default Value	0003 0000h (seconds)

2.3.17. Object 6459h: PID Decimal Digits Timing

This object describes the number of digits following the decimal point (i.e. resolution) of the PID timing data, which is interpreted with data type Integer16 in objects 7452h, 7454h and 7456h.

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	4
Default Value	4

Sub-Index	1h to 4h (x = 1 to 4)
Description	PIDx Decimal Digits Timing
Access	RW
PDO Mapping	No
Value Range	0 to 2
Default Value	3 [default ms]

2.4. MANUFACTURER OBJECTS

Index (hex)	***Object***	***Object Type***	***Data Type***	***Access***	***PDO Mapping***
2300	AO Override Field Value	ARRAY	INTEGER32	RW	No
2310	AO Error Detect Enable	ARRAY	BOOLEAN	RW	No
2311	AO Error Clear Hysteresis	ARRAY	INTEGER16	RW	No
2312	AO Error Reaction Delay	ARRAY	UNSIGNED16	RW	No
2323	AO Blink Rate	ARRAY	UNSIGNED16	RW	No
2330	AO Ramp Up	ARRAY	UNSIGNED16	RW	No
2331	AO Ramp Down	ARRAY	UNSIGNED16	RW	No
2340	AO Control Input Source	ARRAY	UNSIGNED8	RW	No
2341	AO Control Input Number	ARRAY	UNSIGNED8	RW	No
2342	AO Control Response	ARRAY	UNSIGNED8	RW	No
2350	AO Enable Input Source	ARRAY	UNSIGNED8	RW	No
2351	AO Enable Input Number	ARRAY	UNSIGNED8	RW	No
2352	AO Enable Response	ARRAY	UNSIGNED8	RW	No
2360	AO Override Input Source	ARRAY	UNSIGNED8	RW	No
2361	AO Override Input Number	ARRAY	UNSIGNED8	RW	No
2362	AO Override Response	ARRAY	UNSIGNED8	RW	No
2390	AO Control Type	ARRAY	UNSIGNED16	RW	No
2391	AO Fixed Value	ARRAY	INTEGER32	RW	No
2450	PID Tolerance	ARRAY	INTEGER16	RW	No
2451	PID Integral Gain	ARRAY	INTEGER16	RW	No
2452	PID Derivative Gain	ARRAY	INTEGER16	RW	No
2453	PID Target Source	ARRAY	UNSIGNED8	RW	No
2454	PID Target Number	ARRAY	UNSIGNED8	RW	No
2455	PID Feedback Source	ARRAY	UNSIGNED8	RW	No
2456	PID Feedback Number	ARRAY	UNSIGNED8	RW	No
2457	PID Control Response	ARRAY	UNSIGNED8	RW	No
2460	PID Output Field Value	ARRAY	INTEGER16	RO	Yes
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

2700	PWM Duty Cycle Process Value	ARRAY	INTEGER32	RW	Yes
2701	PWM Duty Cycle Override Field Value	ARRAY	INTEGER32	RW	No
2702	PWM Duty Cycle Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
2720	PWM Duty Cycle Output Scaling 1 PV	ARRAY	INTEGER32	RW	No
2721	PWM Duty Cycle Output Scaling 1 FV	ARRAY	INTEGER32	RW	No
2722	PWM Duty Cycle Output Scaling 2 PV	ARRAY	INTEGER32	RW	No
2723	PWM Duty Cycle Output Scaling 2 FV	ARRAY	INTEGER32	RW	No
2730	PWM Duty Cycle Field Value	ARRAY	INTEGER32	RO	Yes
2731	PWM Duty Cycle Fault FV	ARRAY	INTEGER32	RW	No
2732	PWM Duty Cycle Decimal Digits FV	ARRAY	UNSIGNED8	RW	No
2733	PWM Duty Cycle Fault Mode	ARRAY	UNSIGNED8	RW	No
2734	PWM Duty Cycle Ramp Up	ARRAY	UNSIGNED16	RW	No
2735	PWM Duty Cycle Ramp Down	ARRAY	UNSIGNED16	RW	No
2740	PWM Duty Cycle Control Input Source	ARRAY	UNSIGNED8	RW	No
2741	PWM Duty Cycle Control Input Number	ARRAY	UNSIGNED8	RW	No
2742	PWM Duty Cycle Control Response	ARRAY	UNSIGNED8	RW	No
2750	PWM Duty Cycle Enable Input Source	ARRAY	UNSIGNED8	RW	No
2751	PWM Duty Cycle Enable Input Number	ARRAY	UNSIGNED8	RW	No
2752	PWM Duty Cycle Enable Response	ARRAY	UNSIGNED8	RW	No
2760	PWM Duty Cycle Override Input Source	ARRAY	UNSIGNED8	RW	No
2761	PWM Duty Cycle Override Input Number	ARRAY	UNSIGNED8	RW	No
2762	PWM Duty Cycle Override Response	ARRAY	UNSIGNED8	RW	No
2790	PWM Duty Cycle Control Type	ARRAY	UNSIGNED8	RW	No
2791	PWM Duty Cycle Fixed Value	ARRAY	INTEGER32	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(x-3) Lookup Table Number	ARRAY	UNSIGNED8	RW	No
3x02	LB(x-3) Function Logical Operator	ARRAY	UNSIGNED8	RW	No
3x11	LB(x-3) Function A Condition 1	RECOR D	UNSIGNED8	RW	No
3x12	LB(x-3) Function A Condition 2	RECOR D	UNSIGNED8	RW	No
3x13	LB(x-3) Function A Condition 3	RECOR D	UNSIGNED8	RW	No
3x21	LB(x-3) Function B Condition 1	RECOR D	UNSIGNED8	RW	No
3x22	LB(x-3) Function B Condition 2	RECOR D	UNSIGNED8	RW	No
3x23	LB(x-3) Function B Condition 3	RECOR D	UNSIGNED8	RW	No
3x31	LB(x-3) Function C Condition 1	RECOR D	UNSIGNED8	RW	No
3x32	LB(x-3) Function C Condition 2	RECOR D	UNSIGNED8	RW	No
3x33	LB(x-3) Function C Condition 3	RECOR D	UNSIGNED8	RW	No
4500	Math Block Enable	ARRAY	BOOLEAN	RW	No
4521	Math Output Scaling 1 PV	ARRAY	INTEGER16	RW	No
4523	Math Output Scaling 2 PV	ARRAY	INTEGER16	RW	No
4530	Math Output Process Value	ARRAY	INTEGER16	RO	Yes
4532	Math Output Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
4y00	Math Y Input Source	ARRAY	UNSIGNED8	RW	No
4y01	Math Y Input Number	ARRAY	UNSIGNED8	RW	No
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
5040	Fault Detection Field Value	ARRAY	UNSIGNED16	RO	Yes
5041	Fault Detection Set Threshold	ARRAY	UNSIGNED16	RW	No
5042	Fault Detection Clear Threshold	ARRAY	UNSIGNED16	RW	No
5050	Fault Detection Enable Err Check 8 Faults	ARRAY	UNSIGNED8	RW	No
5041	Fault Detection Error Response Delay	ARRAY	UNSIGNED16	RW	No
5555	Start in Operational Mode	VAR	BOOLEAN	RW	No

*Where **yz** = 01 to 09 (LUT 1 to 09) and **x** = 4 to 5(Logic 1 to 2) and **y** = 1 to 6 (Math 1 to 6)*

2.4.1. Object 2300h: AO Override Field Value

This object contains the pre-defined field value of an analog output when an override condition is active. It will be scaled in the physical unit of the output, dependent on type, with the resolution defined in object 6332h AO Decimal Digits FV.

Object Description

Index	2300h
Name	AO Override FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Override FV
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	0 [mA]

2.4.2. Object 2310h: AO Error Detect Enable

This object enables error detection and reaction associated with the analog output function block. When disabled, the input will not generate an EMCY code in object 1003h Pre-Defined Error Field should the control detect an open/short circuit at the load.

Object Description

Index	2310h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	POx Error Detect Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	1 [TRUE]

2.4.3. Object 2311h: AO Error Clear Hysteresis

This object is used to define the absolute difference that can be tolerated between the target output (as commanded by the control input) and the measured feedback. Any difference outside of the value will flag an open or short circuit fault. It is scaled in the physical unit of the output FV, i.e. object 6332h applies to this object.

Object Description

Index	2311h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	0 to 10% of 7321h or 7323h, whichever is larger
Default Value	200 [V]

2.4.4. Object 2312h: AO 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	2312h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.5. Object 2323h: AO Blink Rate

This object defines how frequently the output will flash. The blinking mode operates with a PWM Frequency Fixed Value set above 5000 Hz. The physical unit is always milliseconds.

Object Description

Index	2323h
Name	AO 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Blink rate
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.6. Object 2330h: AO Ramp Up

This object defines the time it will take to ramp from the minimum output PV to the maximum as defined by objects 7321h and 7323h. It can be used to soften the response to a step change at the input. The physical unit for this object is milliseconds.

Object Description

Index	2330h
Name	AO Ramp Up
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Ramp Up
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.7. Object 2331h: AO Ramp Down

This object defines the time it will take to ramp from the maximum output PV to the minimum as defined by objects 7321h and 7323h. It can be used to soften the response to a step change at the input. The physical unit for this object is milliseconds.

Object Description

Index	2331h
Name	AO Ramp Down
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 Ah (x=1 to 10)
Description	AOx Ramp Down
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.8. Object 2340h: AO Control Source

This object defines the type of input that will be used to control the analog output as shown in the logic flowchart in Figure 4. The available control sources on the controller are listed in Table 3. Not all sources would make sense to control the AO, and it is the user’s responsibility to select a source that makes sense for the application.

Object Description

Index	2340h
Name	AO Control 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Control Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	1 (CANopen® RPDO)

2.4.9. Object 2341h: AO Control Number

This object defines the number of the source that will be used to control the analog output as shown in the logic flowchart in Figure 4. The available control numbers are dependent on the source selected, as shown in Table 4. Once selected, the control represents the process value (X-Axis input) in Figure 3. Objects 6302h, 7320h, 7322h should therefore be updated to match the scaling limits defined by the control source/number, as listed in Table 5.

Object Description

Index	2341h
Name	AO Control 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Control Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	1 (CANopen® Message 1)

2.4.10. Object 2342h: AO Control Response

This object defines the response profile of the analog output FV with respect to the input PV (as selected by objects 2340h/2341h.) Normally it will follow the profile shown in Figure 3. However, in some cases the offset will be disabled (i.e. output at 0) when the PV is below 7320h Scaling 1 PV or alternatively above the 7322h Scaling 2 PV. The options for this object are listed in Table 6. When an output Control Type is configured as Fixed Value using object 2390h then this object is ignored.

Object Description

Index	2342h
Name	AO Control Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 2)
Description	Outx Control Response
Access	RW
PDO Mapping	No
Value Range	See Table 6
Default Value	1 (Output OFF below Scaling 1 PV)

2.4.11. Object 2350h: AO Enable Source

This object defines the type of input that will be used to enable/disable the analog output as shown in the logic flowchart in Figure 4. The available control sources on the controller are listed in Table 3. Not all sources would make sense to enable the AO, and it is the user’s responsibility to select a source that makes sense for the application.

Object Description

Index	2350h
Name	AO Enable 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Enable Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used)

2.4.12. Object 2351h: AO Enable Number

This object defines the number of the source that will be used to enable/disable the analog output as shown in the logic flowchart in Figure 4. The available control numbers are dependent on the source selected, as shown in Table 4.

Object Description

Index	2351h
Name	AO Enable 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Enable Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0 (null control source)

2.4.13. Object 2352h: AO Enable Response

This object determines if the input will act as an enable or safety interlock (i.e. input must be ON to engage the output) or a disabled signal (i.e. the output will shut off when the input is ON.) The options for this object for analog output 1 to 4 are listed in Table 7.

Object Description

Index	2352h
Name	AO Enable Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Enable Response
Access	RW
PDO Mapping	No
Value Range	See Table 7
Default Value	3 (Enable When Off, Else Shut Off)

2.4.14. Object 2360h: AO Override Source

This object defines the type of input that will be used to active the override value for the analog outputs as shown in the logic flowchart in Figure 4. The available control sources are listed in Table 3. Not all sources would make sense to enable the AO, and it is the user's responsibility to select a source that makes sense for the application.

Object Description

Index	2360h
Name	AO Override 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Override Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used)

2.4.15. Object 2361h: AO Override Number

This object defines the number of the source that will be used to override the analog outputs as shown in the logic flowchart in Figure 4. The available control numbers are dependent on the source selected, as shown in Table 4.

Object Description

Index	2361h
Name	AO Override 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Override Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0 (null control source)

2.4.16. Object 2362h: AO Override Response

This object determines how the override command will respond to the input state. The options for this object are listed in Table 8.

Object Description

Index	2362h
Name	AO Override Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Override Response
Access	RW
PDO Mapping	No
Value Range	See Table 8
Default Value	0 (Override When On)

2.4.17. Object 2390h: AO Control Type

This object is used to set the process values control type for the output direct current. When Control Logic is enabled, any of the control source can be selected as the input source. Alternatively, when in Fixed Value is enabled, it uses Object 2391h to set the output current.

Object Description

Index	2390h
Name	AO Control Type
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	AOx Control Type
Access	RW
PDO Mapping	No
Value Range	See Table 10
Default Value	0 (Control Logic)

2.4.18. Object 2391h: AO Fixed Value

This object is used to set a constant value to LED output signal when object 2390h is set to Fixed Value control. It determines the output direct current. If Object 2390h is set to Control Logic, then this object cannot be modified. The physical unit is always milliamps with no decimal digits.

Object Description

Index	2391h
Name	AO Fixed Value
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	Outx Fixed Value
Access	RW
PDO Mapping	No
Value Range	80 to 600
Default Value	80 [mA]

2.4.19. Object 2450h: PID Tolerance

This object defines the allowable absolution difference between the target and the feedback, below which the error will be interpreted as zero (i.e. PID output stops changing). The physical unit for this object is percentage, and the value is always interpreted as having a resolution of one digit after the decimal place.

Object Description

Index	2450h
Name	PID Tolerance
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PID1 Tolerance
Access	RW
PDO Mapping	No
Value Range	0 to 100 (0% to 10%)
Default Value	10 [1%]

2.4.20. Object 2451h: PID Integral Gain

This object describes the integral gain (Ki in Figure 7) of the PID algorithm. The value is always interpreted as having a resolution of one digit after the decimal place.

Object Description

Index	2451h
Name	PID Integral Gain
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h to 4h (x=1 to 4)
Description	PIDx Integral Gain
Access	RW
PDO Mapping	No
Value Range	0 to 100
Default Value	10 [1.0]

2.4.21. Object 2452h: PID Derivative Gain

This object describes the derivative gain (Kd in Figure 7) of the PID algorithm. The value is always interpreted as having a resolution of one digit after the decimal place.

Object Description

Index	2452h
Name	PID Derivative Gain
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PIDx Derivative Gain
Access	RW
PDO Mapping	No
Value Range	0 to 100
Default Value	10 [1.0]

2.4.22. Object 2453h: PID Target Source

This object defines the type of input that will be used to determine the target process value for the PID control loop. The available control sources are listed in Table 3. Not all sources would make sense to use as a PID target source, 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 PID control function block.

Object Description

Index	2453h
Name	PID Target Source
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 Ah (x=1 to 10)
Description	PIDx Target Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used, PID disabled)

2.4.23. Object 2454h: PID Target Number

This object defines the number of the source that will be used as the target PV for the PID control loop. The available control numbers are dependent on the source selected, as shown in Table 4. Once selected, the control will convert the commanded target into a percentage value using the scaling limits of the control source/number as defined in Table 5.

Object Description

Index	2454h
Name	PID Target Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PIDx Target Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0

2.4.24. Object 2455h: PID Feedback Source

This object defines the type of input that will be used to determine the feedback process value for the PID control loop. The available control sources are listed in Table 3. Not all sources would make sense to use as a PID feedback source, 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 PID control function block.

Object Description

Index	2455h
Name	PID Feedback Source
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PIDx Feedback Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used, PID disabled)

2.4.25. Object 2456h: PID Feedback Number

This object defines the number of the source that will be used as the feedback PV for the PID control loop. The available control numbers are dependent on the source selected, as shown in Table 4. Once selected, the control will convert the measured/received feedback into a percentage value using the scaling limits of the control source/number as defined in Table 5.

Object Description

Index	2456h
Name	PID Feedback Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PIDx Feedback Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0

2.4.26. Object 2457h: PID Control Response

This object defines the output profile for the PID control function block in a push-pull dual output system. The options for this object are listed in Table 11.

Object Description

Index	2457h
Name	PID Control Response
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 Ah (x=1 to 10)
Description	PIDx Control Response
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	0 (single output)

2.4.27. Object 2460h: PID Output Field Value

This read-only output contains the PID control function block FV (as a percentage) that can be used as the input source for another function block (i.e. analog output.) It will be a value between 0 to 100% as per the algorithm defined in Figure 7. The physical unit for this object is percentage, and the value is always interpreted as having a resolution of one digit after the decimal place.

Object Description

Index	2460h
Name	PID Output FV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PID1 Output FV
Access	RO
PDO Mapping	Yes
Value Range	0 to 1000 (0 to 100%)
Default Value	No

2.4.28. 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	10
Default Value	10

Sub-Index	1h to Ah (x = 1 to 10)
Description	ECx Received Process Value
Access	RW
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

2.4.29. 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	10
Default Value	10

Sub-Index	1h to Ah (x = 1 to 10)
Description	ECx Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	1 (0.1 resolution)

2.4.30. 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, i.e. as seen in Figure 3. There is no physical unit associated 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	10
Default Value	10

Sub-Index	1h to Ah (x = 1 to 10)
Description	ECx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	-32768 to 2522h sub-index X
Default Value	0

2.4.31. 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 data, i.e. as seen in Figure 3. 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 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	10
Default Value	10

Sub-Index	1h to Ah (x = 1 to 10)
Description	ECx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	2520h sub-index X to 32767
Default Value	1000 (100.0)

2.4.32. Object 2700h: PWM Duty Cycle Control Process Value

This object represents the process value of the output PWM Duty Cycle. It can be used as an input to the PWM Duty Cycle function block when the input has been selected as controlled by a CANopen® Message (per Table 3).

Object Description

Index	2700h
Name	PWM Duty Cycle Process Value
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Process Value
Access	RW
PDO Mapping	Yes
Value Range	Integer32
Default Value	No

2.4.33. Object 2701h: PWM Duty Cycle Override FV

This object contains the pre-defined field value of the output PWM Duty Cycle when an override condition is active. It will be scaled in the physical unit of the output, dependent on the resolution defined in object 2732h PWM Duty Cycle Decimal Digits FV.

Object Description

Index	2701h
Name	PWM Duty Cycle Override FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Override FV
Access	RW
PDO Mapping	No
Value Range	0 to 1000 (0% to 100%)
Default Value	0 [0%]

2.4.34. Object 2702h: PWM Duty Cycle Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the PWM Duty Cycle control data, which is interpreted with data type Unsigned Integer8 in the process value object. If Object 2790h is set to Fixed Value, then this object cannot be modified.

Object Description

Index	2702h
Name	PWM Duty Cycle 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	1

2.4.35. Object 2720h: PWM Duty Cycle Scaling 1 PV

This object defines the minimum value of the input and should be specified to equal the corresponding scaling object of the control source, as outlined in Table 5. It will be scaled in the physical unit of the control source. The resolution will ALWAYS be dependent on object 2702h PWM Duty Cycle Decimal Digits PV, even when the Duty Cycle is not being controlled directly by the PWM Duty Cycle PV object 2700h. This object must always be smaller than object 2722h PWM Duty Cycle Scaling 2 PV.

Object Description

Index	2720h
Name	PWM Duty Cycle Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	See Table 5
Default Value	0 [%]

2.4.36. Object 2721h: PWM Duty Cycle Scaling 1 FV

This object defines the output Duty Cycle field value when the input data is at or below the PWM Duty Cycle Scaling 1 PV value as shown in Figure 3. It will be scaled in the physical unit of the PWM Duty Cycle, dependent on the resolution defined in object 2732h PWM Duty Cycle Decimal Digits FV. The value can be set anywhere within the allowable Duty Cycle range as outlined in Section **Error! Reference source not found.**. This value can be set higher than object 2723h PWM Duty Cycle Scaling 2 FV for an inverse response (i.e. decreasing) to an increasing input.

Object Description

Index	2721h
Name	PWM Duty Cycle Scaling 1 FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	0 [%]

2.4.37. Object 2722h: PWM Duty Cycle Scaling 2 PV

This object defines the maximum value of the input and should be specified to equal the corresponding scaling object of the control source, as outlined in Table 5. It will be scaled in the physical unit of the control source. The resolution will ALWAYS be dependent on object 2702h PWM Duty Cycle Decimal Digits PV, even when the output Duty Cycle is not being controlled directly by the PWM Duty Cycle PV object 2700h. This object must always be larger than object 2722h PWM Duty Cycle Scaling 2 PV.

Object Description

Index	2722h
Name	PWM Duty Cycle Scaling 2 PV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	See Table 5
Default Value	1000 [100.0 %]

2.4.38. Object 2723h: PWM Duty Cycle Scaling 2 FV

This object defines the output Duty Cycle field value when the input data is at or above the PWM Duty Cycle Scaling 2 PV value as shown in Figure 3. It will be scaled in the physical unit of the PWM Duty Cycle, dependent on the resolution defined in object 2732h PWM Duty Cycle Decimal Digits FV. The value can be set anywhere within the allowable output range as outlined in Section **Error! Reference source not found.**. This value can be set lower than object 2721h PWM Duty Cycle Output Scaling 1 FV for an inverse response (i.e. decreasing) to an increasing input.

Object Description

Index	2723h
Name	PWM Duty Cycle Scaling 2 FV
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	1000 [100.0 %]

2.4.39. Object 2730h: PWM Duty Cycle Control FV

This object represents the target output PWM Duty Cycle field value as a result of the output logic described in Section 1.2, and the scaling applied as shown in Figure 3. It is defined in the physical unit of the output dependent on type, as outlined in Table 1. The resolution of the object is defined in object 2732h PWM Duty Cycle Decimal Digits FV.

Object Description

Index	2730h
Name	PWM Duty Cycle Field Value
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Field Value
Access	RO
PDO Mapping	Yes
Value Range	Integer32
Default Value	No

2.4.40. Object 2732h: PWM Duty Cycle Decimal Digits FV

This object describes the number of digits following the decimal point (i.e. resolution) of the output Duty Cycle data, which is interpreted with data type Unsigned Integer8 in the field value object.

Object Description

Index	2732h
Name	PWM Duty Cycle 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	10
Default Value	10

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Decimal Digits FV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	1

2.4.41. Object 2733h: PWM Duty Cycle Fault Mode

This object defines how an output Duty Cycle shall respond when a fault condition is detected on any control input, as described in Table 9.

Object Description

Index	2733h
Name	PWM Duty Cycle Fault Mode
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	10
Default Value	2

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Fault Mode
Access	RW
PDO Mapping	No
Value Range	See Table 9
Default Value	1 (apply pre-defined FV)

2.4.42. Object 2731h: PWM Duty Cycle Fault Field Value

This object contains the pre-defined field value of the output Duty Cycle when a fault condition is present, and the corresponding sub-index in object 2733h is enabled. It will be scaled in the physical unit of the output Duty Cycle, dependent on the resolution defined in object 2732h PWM Duty Cycle Decimal Digits FV.

Object Description

Index	2731h
Name	PWM Duty Cycle Fault Field Value
Object Type	ARRAY
Data Type	INTEGER32

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Fault Field Value
Access	RW
PDO Mapping	No
Value Range	Dependent on type (see Table 1)
Default Value	0 %

2.4.43. Object 2734h: PWM Duty Cycle Ramp Up

This object defines the time it will take to ramp from the minimum Duty Cycle PV to the maximum as defined by objects 2721h and 2723h. It can be used to soften the response to a step change at the input. The physical unit for this object is milliseconds.

Object Description

Index	2734h
Name	PWM Duty Cycle Ramp Up
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Ramp Up
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.44. Object 2735h: PWM Duty Cycle Ramp Down

This object defines the time it will take to ramp from the maximum Duty Cycle PV to the minimum as defined by objects 2721h and 2723h. It can be used to soften the response to a step change at the input. The physical unit for this object is milliseconds.

Object Description

Index	2735h
Name	PWM Duty Cycle Ramp Down
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Ramp Down
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

2.4.45. Object 2740h: PWM Duty Cycle Control Source

This object defines the type of input that will be used to control the output Duty Cycle as shown in the logic flowchart in Figure 5. The available control sources on the controller are listed in Table 3. Not all sources would make sense to control the PWM Duty Cycle, and it is the user's responsibility to select a source that makes sense for the application.

Object Description

Index	2740h
Name	PWM Duty Cycle Control Source
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Control Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	1 (CANopen® RPDO)

2.4.46. Object 2741h: PWM Duty Cycle Control Number

This object defines the number of the source that will be used to control the output Duty Cycle as shown in the logic flowchart in Figure 5. The available control numbers are dependent on the source selected, as shown in Table 4. Once selected, the control represents the process value (X-Axis input) in Figure 3. Objects 2702h, 2720h, 2722h should therefore be updated to match the scaling limits defined by the control source/number, as listed in Table 5.

Object Description

Index	2741h
Name	PWM Duty Cycle Control Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Control Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	5 (CANopen® Message 5)

2.4.47. Object 2742h: PWM Duty Cycle Control Response

This object defines the response profile of the output Duty Cycle FV with respect to the input PV (as selected by objects 2740h/2741h.) Normally it will follow the profile shown in Figure 3. However, in some cases the offset will be disabled (i.e. output at 0) when the PV is below 2720h Scaling 1 PV or alternatively above the 2722h Scaling 2 PV. The options for this object are listed in Table 6. When the Duty Cycle Control is configured as Fixed Value using object 2790h then this object is ignored.

Object Description

Index	2742h
Name	PWM Duty Cycle Control Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Control Response
Access	RW
PDO Mapping	No
Value Range	See Table 6
Default Value	1 (DC OFF below Scaling 1 PV)

2.4.48. Object 2750h: PWM Duty Cycle Enable Source

This object defines the type of input that will be used to enable/disable the output Duty Cycle as shown in the logic flowchart in Figure 5. The available control sources on the controller are listed in Table 3. Not all sources would make sense to enable the PWM Duty Cycle, and it is the user’s responsibility to select a source that makes sense for the application.

Object Description

Index	2750h
Name	PWM Duty Cycle Enable Source
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Enable Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used)

2.4.49. Object 2751h: PWM Duty Cycle Enable Number

This object defines the number of the source that will be used to enable/disable the output Duty Cycle as shown in the logic flowchart in Figure 5. The available control numbers are dependent on the source selected, as shown in Table 4.

Object Description

Index	2751h
Name	PWM Duty Cycle Enable Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Enable Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0 (null control source)

2.4.50. Object 2752h: PWM Duty Cycle Enable Response

This object determines if the input will act as an enable or safety interlock (i.e. input must be ON to engage the output) or a disabled signal (i.e. the output will shut off when the input is ON.) The options for this object for output 1 to 4 are listed in Table 7.

Object Description

Index	2752h
Name	PWM Duty Cycle Enable Response
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Enable Response
Access	RW
PDO Mapping	No
Value Range	See Table 7
Default Value	3 (Enable When Off, Else Shut Off)

2.4.51. Object 2760h: PWM Duty Cycle Override Source

This object defines the type of input that will be used to active the override value for the output Duty Cycle as shown in the logic flowchart in Figure 5. The available control sources are listed in Table 3. Not all sources would make sense to enable the PWM Duty Cycle, and it is the user's responsibility to select a source that makes sense for the application.

Object Description

Index	2760h
Name	PWM Duty Cycle Override Source
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 Ah (x=1 to 10)
Description	PWM DC x Override Input Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control not used)

2.4.52. Object 2761h: PWM Duty Cycle Override Number

This object defines the number of the source that will be used to override the output PWM Duty Cycle as shown in the logic flowchart in Figure 5. The available control numbers are dependent on the source selected, as shown in Table 4.

Object Description

Index	2761h
Name	PWM Duty Cycle Override Number
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 Ah (x=1 to 10)
Description	PWM DC x Override Input Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0 (null control source)

2.4.53. Object 2762h: PWM Duty Cycle Override Response

This object determines how the override command will respond to the input state. The options for this object are listed in Table 8.

Object Description

Index	2762h
Name	PWM Duty Cycle Override Response
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 Ah (x=1 to 10)
Description	PWM DC x Override Response
Access	RW
PDO Mapping	No
Value Range	See Table 8
Default Value	0 (Override When On)

2.4.54. Object 2790h: PWM Duty Cycle Control Type

This object is used to set the process value control type for the output Duty Cycle. When Control Logic is enabled, any of the control source can be selected as the input source. Alternatively, when in Fixed Value is enabled, it uses Object 2791h to set the output Duty Cycle.

Object Description

Index	2790h
Name	PWM Duty Cycle Control Type
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 Ah (x=1 to 10)
Description	PWM DC x Control Type
Access	RW
PDO Mapping	No
Value Range	See Table 10
Default Value	0 (Control Logic)

2.4.55. Object 2791h: PWM Duty Cycle Fixed Value

This object is used to set a constant value to output Duty Cycle when object 2790h is set to Fixed Value control. It determines the output PWM Duty Cycle. If Object 2790h is set to Control Logic, then this object cannot be modified. The value is always interpreted as having a resolution of one digit after the decimal place.

Object Description

Index	2791h
Name	PWM Duty Cycle Fixed Value
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (x=1 to 10)
Description	PWM DC x Fixed Value
Access	RW
PDO Mapping	No
Value Range	0 to 1000 (0% to 100%)
Default Value	500 [50.0%]

2.4.56. 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 are listed in Table 3. 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 3
Default Value	0 (control not used)

2.4.57. 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 4. Once selected, the limits for the points on the X-Axis will be constrained by the scaling objects of the control source/number as defined in Table 5.

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 4
Default Value	0 (null control source)

2.4.58. 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 1.4.4

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]

2.4.59. 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 5.

Object Description

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

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 4 (see Table 5)
Default Value	0

2.4.60. 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

2.4.61. 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 12. See Figure 9 for an example of the difference between a step and ramp response.

Object Description

Index	3yz5h (where yz = 01 to 09)
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 12(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 12(0, 1 or 2)
Default Value	1 (ramp to response)

2.4.62. 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 associated with the data will be that of the selected input, and it will use the resolution defined in object 3yz3h, LTz X-Axis Decimal Digits PV.

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

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

Object Description

Index	3yz6h (where yz = 01 to 09)
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 Bh (x = 1 to 11)	
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

2.4.63. 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 associated 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 09)
Name	LTyz 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 Bh (x = 1 to 11)
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]

2.4.64. 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

2.4.65. Object 3300h: Logic Block Enable

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

Object Description

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

Entry Description

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

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

2.4.66. Object 3310h: Logic Block Selected Table

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

Object Description

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

Entry Description

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

Sub-Index	1h to 3h (x = 1 to 3)
Description	LBx Selected Table
Access	RO
PDO Mapping	Yes
Value Range	1 to 9
Default Value	No

2.4.67. 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 as shown in Table 17. 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	INTEGER16

Entry Description

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

Sub-Index	1h to 3h (x = 1 to 3)
Description	LBx Output PV
Access	RO
PDO Mapping	Yes
Value Range	Dependent on Selected Table
Default Value	No

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

This object determines which of the six lookup tables are 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 6)
Name	LB(x-3) 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	3
Default Value	3

Sub-Index	1h to 3h (y = A to C)
Description	LB(x-3) Lookup Table Y Number
Access	RW
PDO Mapping	No
Value Range	1 to 9
Default Value	See Table 18

2.4.69. Object 3x02h: LB(x-3) 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 16. See Section 1.5 for more information about how this object is used.

Object Description

Index	3x02h (where x = 4 to 6)
Name	LB(x-3) 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	3
Default Value	3

Sub-Index	1h to 3h (y = A to C)
Description	LB(x-3) Function Y Logical Operator
Access	RW
PDO Mapping	No
Value Range	See Table 16
Default Value	Function A = 1 (and all) Function B = 1 (and all) Function C = 0 (default)

2.4.70. Object 3x11h: LB(x-3) Function A Condition 1
2.4.71. Object 3x12h: LB(x-3) Function A Condition 2
2.4.72. Object 3x13h: LB(x-3) Function A Condition 3
2.4.73. Object 3x21h: LB(x-3) Function B Condition 1
2.4.74. Object 3x22h: LB(x-3) Function B Condition 2
2.4.75. Object 3x23h: LB(x-3) Function B Condition 3
2.4.76. Object 3x31h: LB(x-3) Function C Condition 1
2.4.77. Object 3x32h: LB(x-3) Function C Condition 2
2.4.78. Object 3x33h: LB(x-3) Function C Condition 3

These objects, 3xyzh, represent Logic Block z, Function y, Condition z, where x = 4 to 6, y = 1 (A) to 3 (C), and z = 1 to 3. All these objects are a special type of record, defined in Table 13. Information on how to use these objects is defined in Section 1.5.

Object Description

Index	3xyzh
Name	LB(x-3) 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 3
Default Value	1 (CANopen® Message)

Sub-Index	2h
Description	Argument 1 Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	11 (EC Received PV 1)

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

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

Sub-Index	5h
Description	Operator
Access	RW
PDO Mapping	No
Value Range	See Table 14
Default Value	0 (Equals)

2.4.79. Object 3900h: Set-Reset Latch Enable

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

Object Description

Index	3900h
Name	SR Latch Enable
Object Type	ARRAY
Data Type	BOOLEAN

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 (Y = 1 to 5)
Description	SR Latch Y Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]

2.4.80. Object 3910h: Set-Reset Latch Output Process Value

Object Description

Index	3910h
Name	Set-Reset Latch 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	5
Default Value	5

Sub-Index	1h to 5h (Y = 5 to 5)
Description	SR Latch Y Output PV
Access	RO
PDO Mapping	Yes
Value Range	0/1 (False/True)
Default Value	0 (False)

2.4.81. Object 39x1h: Set-Reset Latch [x] Reset Signal Source

Object Description

Index	39x1h (where x = 1 to 5)
Name	Set-Reset Latch Reset Signal Source
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	SR Latch x Reset Signal Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0

2.4.82. Object 39x2h: Set-Reset Latch [x] Reset Signal Number

Object Description

Index	39x2h (where x = 1 to 5)
Name	SR Latch Reset Signal Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	SR Latch x Reset Signal Number
Access	RW
PDO Mapping	No
Value Range	Input Dependent, see Table 3
Default Value	0

2.4.83. Object 39x3h: Set-Reset Latch [x] Reset Signal OFF Threshold

Object Description

Index	39x3h (where x = 1 to 5)
Name	SR Latch Reset Signal OFF Threshold
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	SR Latch x Reset Signal OFF Threshold
Access	RW
PDO Mapping	No
Value Range	0.0 - 100.0 [%]
Default Value	0.0 [%]

2.4.84. Object 39x4h: Set-Reset Latch [x] Reset Signal ON Threshold

Object Description

Index	39x4h (where x = 1 to 5)
Name	SR Latch Reset Signal ON Threshold
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	SR Latch x Reset Signal ON Threshold
Access	RW
PDO Mapping	No
Value Range	0.0-100.0 [%]
Default Value	100.0 [%]

2.4.85. Object 39x5h: Set-Reset Latch [x] Set Signal Source

Object Description

Index	39x5h (where x = 1 to 5)
Name	Set-Reset Latch Set Signal Source
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	SR Latch x Set Signal Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0

2.4.86. Object 39x6h: Set-Reset Latch [x] Set Signal Number

Object Description

Index	39x6h (where x = 1 to 5)
Name	SR Latch Set Signal Number
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	SR Latch x Set Signal Number
Access	RW
PDO Mapping	No
Value Range	Input Dependent, see Table 3
Default Value	0

2.4.87. Object 39x7h: Set-Reset Latch [x] Set Signal OFF Threshold

Object Description

Index	39x7h (where x = 1 to 5)
Name	SR Latch Set Signal OFF Threshold
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	SR Latch x Set Signal OFF Threshold
Access	RW
PDO Mapping	No
Value Range	0.0-100.0 [%]
Default Value	0.0 [%]

2.4.88. Object 39x8h: Set-Reset Latch [x] Set Signal ON Threshold

Object Description

Index	39x8h (where x = 1 to 5)
Name	SR Latch Set Signal ON Threshold
Object Type	ARRAY
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Description	SR Latch x Set Signal ON Threshold
Access	RW
PDO Mapping	No
Value Range	0.0-100.0 [%]
Default Value	100.0 [%]

2.4.89. 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]

2.4.90. Object 4021h: Math Output Scaling 1 PV

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

Object Description

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

Entry Description

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

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

2.4.91. Object 4023h: Math Output Scaling 2 PV

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

Object Description

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

Entry Description

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

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

2.4.92. 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

2.4.93. 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)

2.4.94. 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 are listed in Table 3.

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	4
Default Value	4

Sub-Index	1h to 4h (X = 1 to 4)
Description	Math Y Input X Source
Access	RW
PDO Mapping	No
Value Range	See Table 3
Default Value	0 (control source not used)

2.4.95. 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 4. Once selected, the input value will be used in the corresponding calculation as described in Section 1.6.

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	4
Default Value	4

Sub-Index	1h to 4h (X = 1 to 4)
Description	Math Y Input X Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	0 (null input)

2.4.96. 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	4
Default Value	4

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

2.4.97. 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	4
Default Value	4

Sub-Index	1h to 4h (X = 1 to 4)
Description	Math Y Input X Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

2.4.98. 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	4
Default Value	4

Sub-Index	1h to 4h (X = 1 to 4)
Description	Math Y Input X Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	INTEGER16
Default Value	10000 (100.00%)

2.4.99. 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	4
Default Value	4

Sub-Index	1h to 4h (X = 1 to 4)
Description	Math Y Input X Gain
Access	RW
PDO Mapping	No
Value Range	-100 to 100
Default Value	100 (1.0)

2.4.100. 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 1.6. The options for this object are listed in Table 19.

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 19
Default Value	12 (Plus)

2.4.101. Object 4B00h: Conditional Logic Block Enable

Object Description

Index	4B00h
Name	Conditional Logic Block Enable
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (X = 1 to 10)
Description	Logic Block X Enable
Access	RW
PDO Mapping	No
Value Range	0/1 (Disabled/Enabled)
Default Value	0 (False)

2.4.102. Object 4B01h: Conditional Logic Result Operator

This object defines the result operator (operator 3) that will be used in the Conditional Logic, as described in Section 1.7.

Object Description

Index	4B01h
Name	Conditional Logic Result Operator
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to Ah (X = 1 to 10)
Description	Cond Logic Block X Result Operator
Access	RW
PDO Mapping	No
Value Range	See Table 21
Default Value	0 (OR)

2.4.103. Object 4B10h: Conditional Logic Output Process Value

Object Description

Index	4B10h
Name	Conditional 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	10
Default Value	10

Sub-Index	1h to Ah (X = 1 to 10)
Description	Logic Block X Output PV
Access	RO
PDO Mapping	Yes
Value Range	0/1 (False/True)
Default Value	0 (False)

2.4.104. Object 4Bxyh: Conditional Logic Block [x] Condition [y] Parameters

These objects represent Conditional Logic Block x, Condition y, where x = 1 to 10, y = 1 (A) to 2 (B). All of these objects are a special type of record, defined in Table 13. Information on how to use these objects is defined in Section 1.5.

Object Description

Index	4Bxyh (x=1 to 10) (y=1 to 2)
Name	Cond Logic x Condition y
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 3
Default Value	1 (CANopen® Message)

Sub-Index	2h
Description	Argument 1 Number
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	2 (EC Received PV 1)

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

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

Sub-Index	5h
Description	Operator
Access	RW
PDO Mapping	No
Value Range	See Table 20
Default Value	0 (Equals)

2.4.105. 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	15
Default Value	15

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	3.141593

Sub-Index	4
Description	Constant FV 4
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	2.718282

Sub-Index	5
Description	Constant FV 5
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	1.414214

Sub-Index	6
Description	Constant FV 6
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	1.732051

Sub-Index	7
Description	Constant FV 7
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	2.236068

Sub-Index	8
Description	Constant FV 8
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	2.50

Sub-Index	9
Description	Constant FV 9
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	5.00

Sub-Index	10
Description	Constant FV 10
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	10.00

Sub-Index	11
Description	Constant FV 11
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	20.00

Sub-Index	12
Description	Constant FV 12
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	40.00

Sub-Index	13
Description	Constant FV 13
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	60.00

Sub-Index	14
Description	Constant FV 14
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	80.00

Sub-Index	15
Description	Constant FV 15
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	100.00

2.4.106. Object 5040h: Fault Detection Field Value

This read-only object is available for diagnostic feedback purposes. It reflects the measured over/under voltage powering the controller as well as the internal microcontroller temperature. The physical unit for this object is volts and ºC, respectively.

Object Description

Index	5040h
Name	Power Supply Field Value
Object Type	VARIABLE
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	3
Default Value	3

Sub-Index	1h
Description	Over Temperature Field Value
Access	RO
PDO Mapping	Yes
Value Range	0 to 1250 [ºC x 10]
Default Value	0

Sub-Index	2h
Description	Over Voltage Field Value
Access	RO
PDO Mapping	Yes
Value Range	0 to 500 [V x 10]
Default Value	0

Sub-Index	3h
Description	Under Voltage Field Value
Access	RO
PDO Mapping	Yes
Value Range	0 to 500 [V x 10]
Default Value	0

2.4.107. Object 5041h: FD Set Threshold

This object sets the value that will flag a fault detection error in the ECU if the measured field value (5040h) goes above (FD 1 and FD 2) or below (FD 3) this limit. If error checking on the fault is enabled by object 5050h, then the module will flag an appropriate error on that channel. This value must be in the same units as the field value for the fault, as determined by the sub-index.

Object Description

Index	5041h
Name	FD Set Threshold
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h
Description	Over Temperature Set Threshold
Access	RW
PDO Mapping	No
Value Range	5042h at sub-index 1 to 1250 [ºC x 10]
Default Value	1100 (110.0ºC)

Sub-Index	2h
Description	Over Voltage Set Threshold
Access	RW
PDO Mapping	No
Value Range	5042h at sub-index 2 to 1000 [V x 10]
Default Value	500 (50.0V)

Sub-Index	3h
Description	Under Voltage Set Threshold
Access	RW
PDO Mapping	No
Value Range	80 to 5042h at sub-index 3 [V x 10]
Default Value	90 (9.0V)

2.4.108. Object 5042h: FD Clear Threshold

This object sets the value that will clear a fault detection error in the ECU if the measured field value (5040h) goes below (FD 1 and FD 2) or above (FD 3) this threshold. This value must be in the same units as the field value for the fault, as determined by the sub-index.

Object Description

Index	5042h
Name	FD Clear Threshold
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h
Description	Over Temperature Clear Threshold
Access	RW
PDO Mapping	No
Value Range	500 to 5041h at subindex 1 [ºC x 10]
Default Value	850 (85.0ºC)

Sub-Index	2h
Description	Over Voltage Clear Threshold
Access	RW
PDO Mapping	No
Value Range	5042h at subindex 3 to 5041h at subindex 2 [V x 10]
Default Value	480 (48.0V)

Sub-Index	3h
Description	Under Voltage Clear Threshold
Access	RW
PDO Mapping	No
Value Range	5041h at subindex 3 to 5042h at subindex 2 [V x 10]
Default Value	120 (12.0V)

2.4.109. Object 5050h: FD Enable Error Check 8 Faults

This object enables or disables the fault detection error-checking feature for each fault detectable by the controller. The bitmap for this object at sub-index 1 is:

- Bit 0: Over Temperature Detection
- Bit 1: Over Voltage Detection
- Bit 2: Under Voltage Detection

Object Description

Index	5050h
Name	FD Enable Error Checking 8 Faults
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1h to 2h (x=1 to 2)
Description	Error Check for FDx
Access	RW
PDO Mapping	No
Value Range	Bit Value 0 = Error Check Disabled Bit Value 1 = Error Check Enabled
Default Value	00h (all error check disabled)

2.4.110. Object 5051h: FD Error Response Delay

This object is used to prevent intermittent faults from overloading the bus with error messages. The value is defined as a multiple of 1ms. If a fault has been present during the entirety of the delay time, the Controller will flag an error of the detected fault once the timer has expired. The object can be set to zero, in which case a fault will immediately trigger an error response.

Object Description

Index	5051h
Name	FD Error Response Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

Sub-Index	1h to 3h (x=1 to 3)
Description	Error Delay FDx
Access	RW
PDO Mapping	No
Value Range	0 to 10000 [ms]
Default Value	5000 [ms]

2.4.111. Object 5550h: Enable Automatic Updates

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

Object Description

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

Entry Description

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

2.4.112. Object 5555h: Start in Operational Mode

This object allows the unit to start in Operational mode without requiring the presence of a CANopen® Master on the network. It is intended to be used only when running the controller as a stand-alone module. This should always be set FALSE whenever it is connected to a standard master/slave network.

Object Description

Index	5555h
Name	Start in Operational 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 [FALSE]

3. Technical Specifications

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 Limitations & Return Materials Process as described on https://www.axiomatic.com/service/.

Power

Power Supply	12 or 24 Vdc nominal 9 to 36 Vdc power supply range
Quiescent Current	39.85 mA @ 12 Vdc; 21.85 mA @ 24 Vdc typical
Protection	Surge and transient protection Reverse polarity protection Undervoltage protection provided. Hardware shutdown at 6 Vdc. Overvoltage protection provided. Hardware shutdown at 38 V.

Outputs

Outputs	10 outputs to power LEDs LED dimming option is provided via CAN. Voltage: 28 Vdc max. Current: 80 to 600 mA per output PWM Dimming: Frequency range of 25 Hz to 100 kHz Duty cycle from 0 to 100 %
Protection	Overcurrent and overvoltage protection provided

General Specifications

Microcontroller	STM32F407VGT7 32-bit, 1 MB flash memory
Communication	1 CAN port (CANopen®) Supported baud rates: 10 kbit/s, 20 kbit/s, 50 kbit/s, 125 kbit/s (default), 250 kbit/s, 500 kbit/s, 800 kbit/s, and 1 Mbit/s
Network Termination	It is necessary to terminate the network with external termination resistors. The resistors are 120 Ω, 0.25 W minimum, metal film or similar type. They should be placed between CAN High and CAN Low terminals at both ends of the network.
User Interface	EDS file is provided to interface with the device using standard CANopen® tools.
Compliance	RoHS
Operating Conditions	-40 to 85 °C (-40 to 185 °F)
Storage Temperature	-50 to 125 °C (-58 to 257 °F)
Protection	IP67; Unit is conformal coated within the housing.
Weight	0.65 lb. (0.295 kg)
Enclosure	High Temperature Nylon housing, TE Deutsch P/N: EEC-325X4B Flammability rating: UL 94 HB 4.677 in. x 5.254 in. x 1.416 in. (119 mm x 134 mm x 36 mm) Note: W x L x H excluding mating plug Refer to dimensional drawing.
Mating Plugs	A mating plug kit is available under P/N: **PL-DTM06-12SA-12SB** (includes 1x plug DTM06-12S, 1x plug DTM06-12SB, 2x wedgelocks W12S, 18x sealing plugs 0413-204-2005, and 24x contacts 0462-201-20141)
Installation	Mounting holes are sized for four 5/16 in. or M8 bolts. The bolt length will be determined by the end-user's mounting plate thickness. The mounting flange of the controller is 0.47 in. (12 mm) thick. It should be mounted with connectors facing left or right to reduce likelihood of moisture entry. All field wiring should be suitable for the operating temperature range. Install the unit with appropriate space available for servicing and for adequate wire harness access (6 in. or 15 cm) and strain relief (12 in. or 30 cm).

4. VERSION HISTORY

Version	Date	Author	Modifications
1.0.0	Apr. 15, 2025	Juan Casanova	Initial Draft
1.0.1	Jul. 31, 2025	M Ejaz	Added technical specifications and pin out
1.0.2	Aug. 5, 2025	M Ejaz	Added block and dimensional drawing Updated technical specifications

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 rma@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