SINGLE UNIVERSAL INPUT CONTROLLER
With CANopen®

USER MANUAL

P/N: AX031701

ACRONYMS

AI	Analog Input (Universal)
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
DI	Digital Input
EDS	Electronic Data Sheet
EMCY	Emergency
LSB	Least Significant Byte (or Bit)
LSS	Layer Settling Service
MSB	Most Significant Byte (or Bit)
NMT	Network Management
PID	Proportional-Integral-Derivative Control
RO	Read Only Object
RPDO	Received Process Data Object
RW	Read/Write Object
SDO	Service Data Object
TPDO	Transmitted Process Data Object
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 Single Universal Input Controller

The following User Manual describes the architecture and functionality a single universal input CANopen ® controller.

The Single Input Controller (1IN-CAN) is designed for continuous measurements of analog sensors and broadcast information onto a CANopen network bus. Its flexible circuit design allows it to measure different types of signals, including voltage, current, frequency/RPM, PWM and digital signals. The firmware control algorithms allow the ability of performing data decisions before broadcasting onto the CANopen network without the need for custom software.

The various function blocks supported by the 1IN-CAN 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.

1.2. Digital Input Function Block

The digital input (DI) function block only becomes applicable on the input when object 6112h, **AI Operation**, is set to a digital input response.
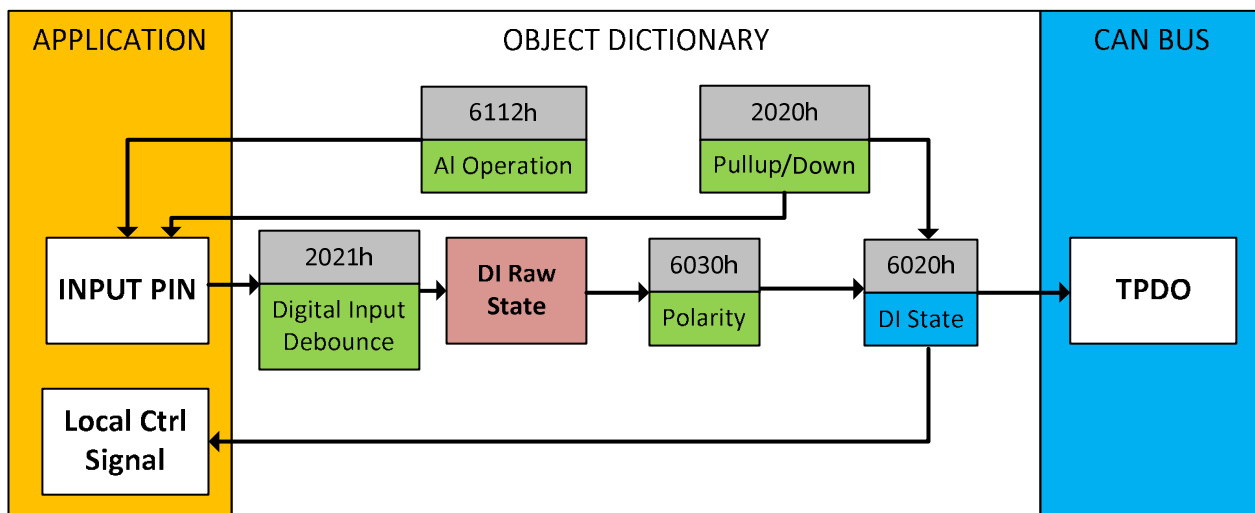
Figure 2 – Digital Input Objects

When 6112h is set to 10 = Digital Input, object 2020h **DI Pullup/Down Mode** will determine if the input signal is active high (10kΩ pulldown enabled, switch to +V) or active low (10kΩ pullup enabled, switched to GND) The options for object 2020h are shown in Table 1, with the default bolded.

Value	Meaning
0	Pullup/Down Disabled (high impedance input)
1	10kΩ Pullup Resistor Enabled
2	**10kΩ Pulldown Resistor Enabled**

Table 1: DI Pullup/Down Options

Figure 3 shows the hysteresis on the input when switching a discrete signal. A digital input can be switched up to +Vcc (48Vmax.)
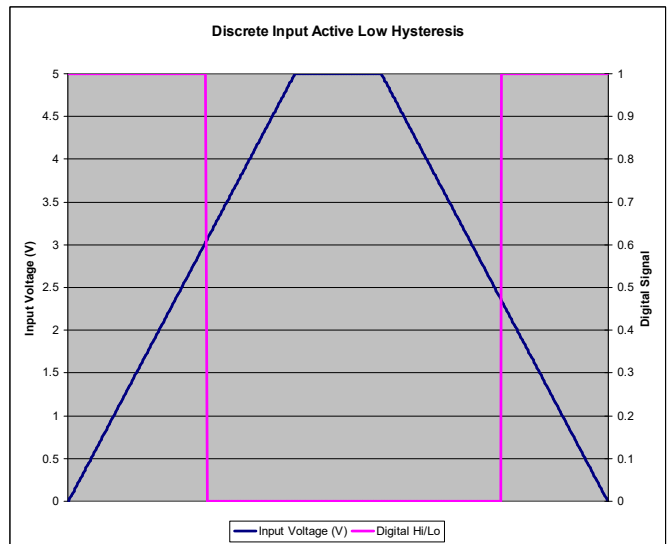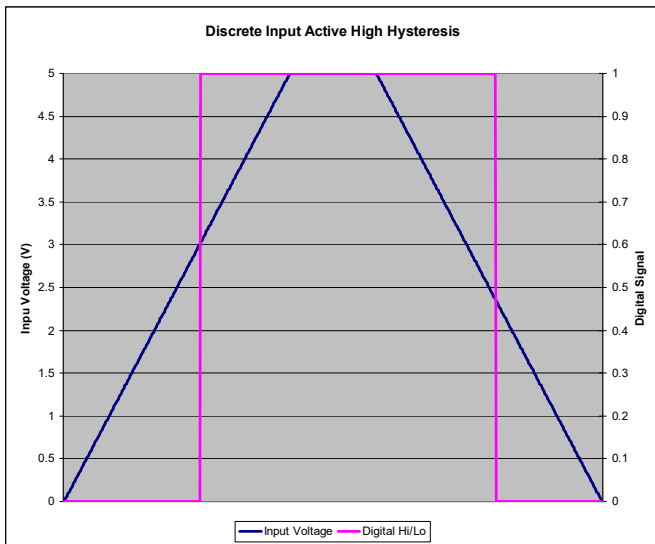
Figure 3 – Discrete Input Hysteresis

Object 2021h **DI Debounce Time** is applied to the input before the state is read by the processor. By default, the debounce time is 10ms.
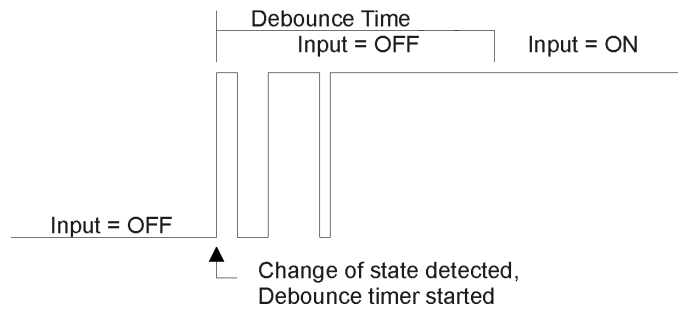
Figure 4 – Digital Input Debouncing

Once the raw state has been evaluated, the logical state of the input is determined by **object 6030h DI Polarity.** The options for object 6030h are shown in Table 3. The 'calculated' state of the DI that will be written to read-only object 6020h **DI Read State** will be a combination of active high/low and the polarity selected. By default, normal on/off logic is used.

Value	Meaning	Active High	Active Low	State
0	**Normal On/Off**	HIGH	LOW	ON
		LOW or Open	HIGH or Open	OFF
1	Inverse On/Off	HIGH	LOW	OFF
		LOW or Open	HIGH or Open	ON
2	Latched Logic	HIGH to LOW	LOW to HIGH	No Change
		LOW to HIGH	HIGH to LOW	State Change (i.e. OFF to ON)

Table 2: DI Polarity Options versus DI State

There is another type of 'digital' input that can be selected when 6112h is set to 20 = Analog On/Off. However, in this case, the input is still configured as an analog input, and therefore the objects from the Analog Input (AI) block are applied instead of those discussed above. Here, objects 2020h, 2030h and 6030h are ignored, and 6020h is written as per the logic shown in Figure 5. In this case, the MIN parameter is set by object 7120h **AI Scaling 1 FV**, and the MAX is set by 7122h **AI Scaling 2 FV**.

For all other operating modes, object 6020h will always be zero.
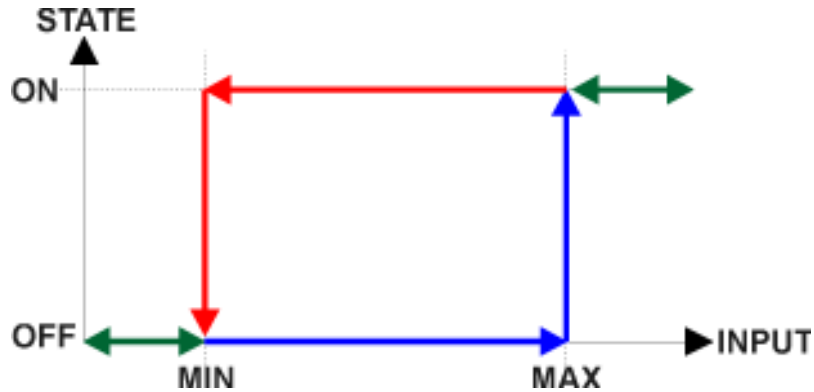
Figure 5 – Analog Input Read as Digital

1.3. Analog Input Function Block

The analog input (AI) function block is the default logic associate with the universal input.
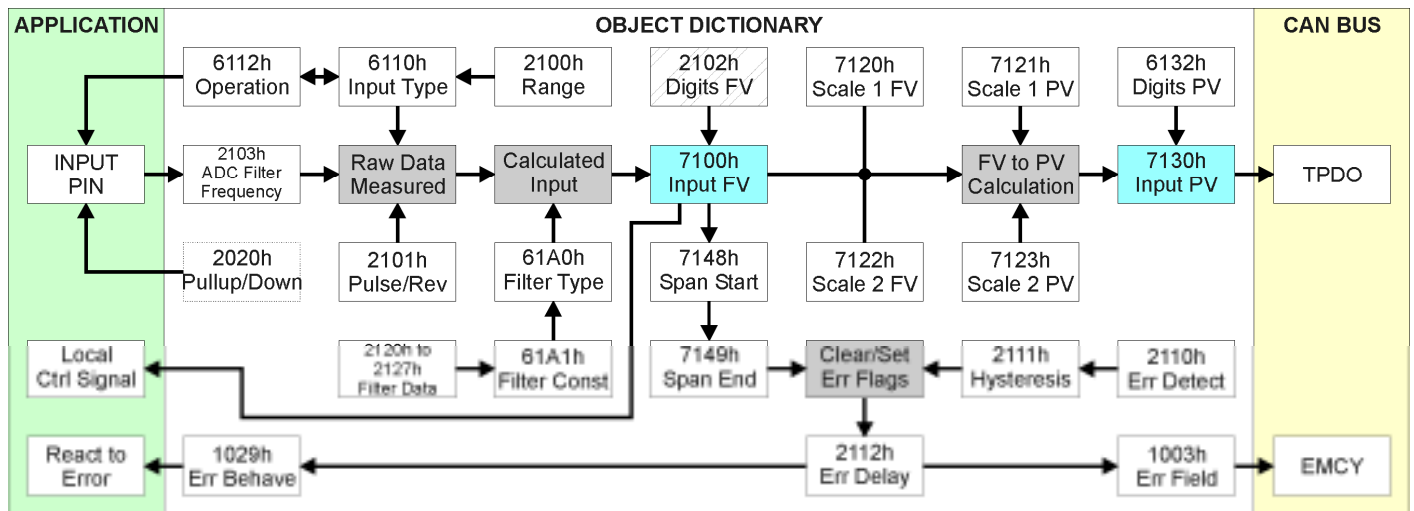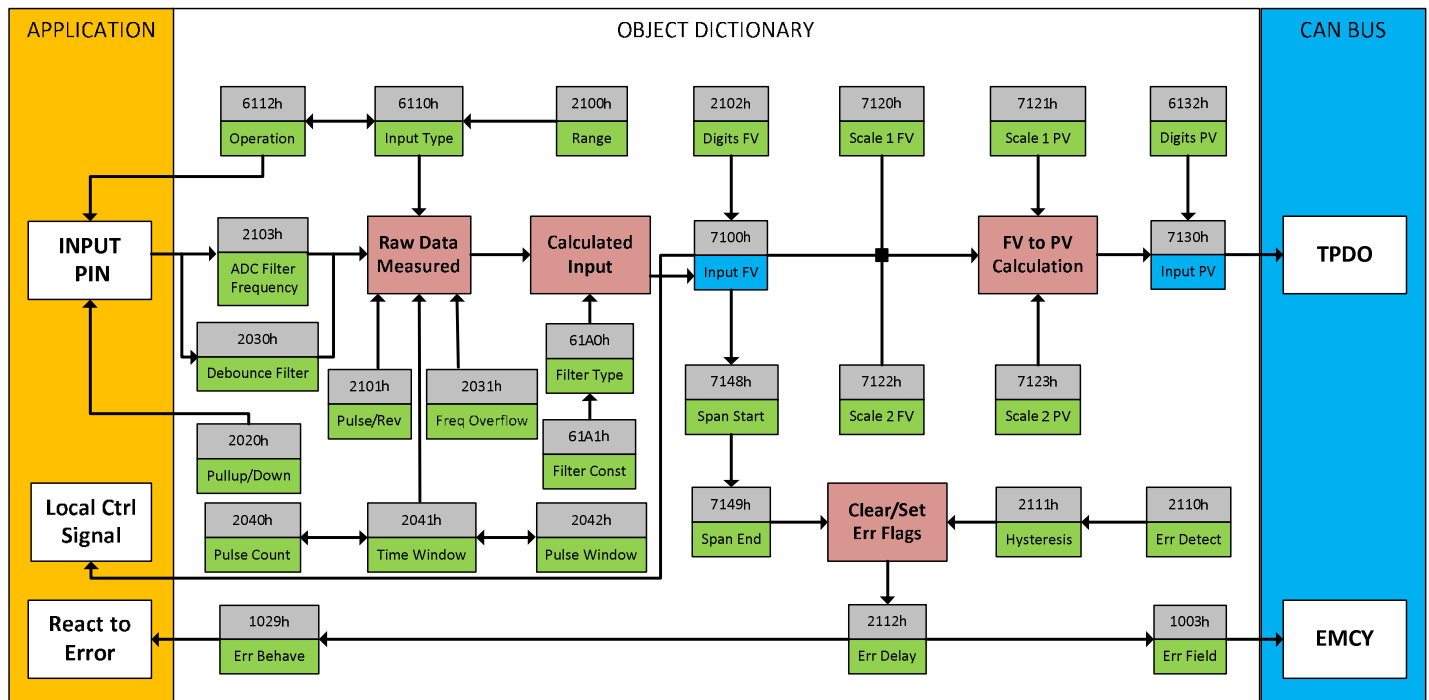
Figure 6 – Analog Input Objects

Object 6112h, **AI Operating Mode** determines whether the AI or DI function block is associated with input. The options for object 6112h are shown in Table 4. No values other than what are shown here will be accepted.

Value	Meaning
0	Channel Off
1	**Normal Operation (analog)**
10	Digital Input (on/off)
20	Analog and On/Off

Table 3: AI Operating Mode Options

The most important object associate with the AI function block is object 6110h **AI Sensor Type**. By changing this value, and associated with it object 2100h **AI Input Range**, other objects will be automatically updated by the controller. The options for object 6110h are shown in Table 5, and no values other than what are shown here will be accepted. The input is setup to measure voltage by default.

Value	Meaning
40	**Voltage Input**
50	Current Input
60	Frequency Input (or RPM)
10000	PWM Input
10010	Counter

Table 4: AI Sensor Type Options

The allowable ranges will depend on the input sensor type selected. Table 6 shows the relationship between the sensor type, and the associated range options. The default value for each range is bolded, and object 2100h will automatically be updated with this value when 6110h is changed. Grayed out cells mean that the associate value is not allowed for the range object when that sensor type has been selected.

Value	Voltage	Current	Frequency	PWM	Counter
0	**0 to 5V**	0 to 20mA	**0.5Hz to 20kHz**	**0.5Hz to 20kHz**	**Pulse Count**
1	0 to 10V	**4 to 20mA**			Time Window
2					Pulse Window

Table 5: AI Input Range Options Depending on Sensor Type

Not all objects apply to all input types. For example, object 2103h **AI Filter Frequency for ADC** is only applicable with a voltage, current or resistive input is being measured. In these cases, the ADC will automatically filter as per Table 7, and is set for 50Hz noise rejection by default.

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

Table 6: ADC Filter Frequency Options

Conversely, frequency and PWM inputs use object 2020h **DI Pullup/Down Mode** (see Table 1) while voltage, current and resistive inputs set this object to zero. Also, a frequency input can be automatically turned into an RPM measurement instead simply by setting object 2101h **AI Number of Pulses Per Revolution** to a non-zero value. All other input types ignore this object.

With Frequency/RPM and PWM input types, **AI Debounce Time**, object 2030h can be applied. The options for object 2030h are shown in Table 2, with the default bolded.

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

Table 7: AI Debounce Filter Options

Irregardless of type, however, all analog inputs can be further filtered once the raw data has been measured (either from ADC or Timer.) Object 61A0h **AI Filter Type** determines what kind of filter is used per Table 8. By default, additional software filtering is disabled.

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

Table 8: AI Filter Type Options

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

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

Calculation with the moving average filter:

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

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

Calculation with the repeating average filter:

$$\text{Value} = \frac{\sum \text{Input}_N}{N}$$

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

The value from the filter is shifted according to read-only object 2102h **AI Decimal Digits FV** and then written to read-only object 7100h **AI Input Field Value.**

The value of 2102h will depend on the AI Sensor Type and Input Range selected, and will be automatically updated per Table 9 when either 6110h or 2100h are changed. All other objects associated with the input field value also apply this object. These objects are 7120h **AI Scaling 1 FV**, 7122h **AI Scaling 2 FV**, 7148h AI **Span Start**, 7149h **AI Span End**, and 2111h **AI Error Clear Hysteresis**. These objects are also automatically updated when the Type or Range is changed.

Sensor Type and Range	Decimal Digits
Voltage: All Ranges	3 [mV]
Current: All Ranges	3 [uA]
Frequency: 0.5Hz to 20kHz	0 [Hz]
Frequency: RPM Mode	1 [0.1 RPM]
PWM: All Ranges	1 [0.1 %]
Digital Input	0 [On/Off]
Counter: Pulse Count	0 [pulses]
Counter: Time/Pulse Window	3 [ms]

Table 9: AI Decimal Digits FV Depending on Sensor Type

It is the AI Input FV which is used by the application for error detection, and as a control signal for other logic blocks (i.e. output control.) Object 7100h is mappable to a TPDO, and is mapped to TPDO1 by default.

Read-only object 7130h **AI Input Process Value** is also mappable. However, the default values for objects 7121h **AI Scaling 1 PV** and 7123h **AI Scaling 2 PV** are set to equal 7120h and 7122h respectively, while object 6132h **AI Decimal Digits PV** is automatically initialize to equal 2102h. This means that the default relationship between the FV and PV is one-to-one, so object 7130h is not mapped to a TPDO by default.

Should a different linear relationship between what is measured versus what is sent to the CANopen bus be desired, objects 6132h, 7121h and 7123h can be changed. The linear

relationship profile is shown in Figure 7 below. Should a non-linear response be desired, the lookup table function block can be used instead, as described in section 1.7.
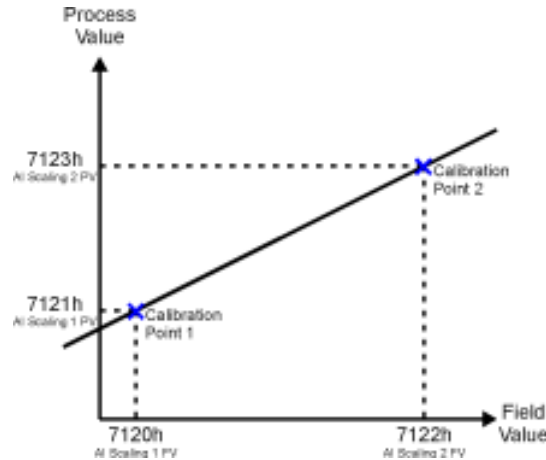

Figure 7 – Analog Input Linear Scaling FV to PV

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

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

Table 10 lists the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2111h for each Sensor Type and Input Range combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 9.

Sensor Type/ Input Range	7148h AI Span Start (i.e. Error Min)	7120h AI Scaling 1 FV (i.e. Input Min)	7122h AI Scaling 2 FV (i.e. Input Max)	7149h AI Span End (i.e. Error Max)	2111h Error Clear Hysteresis
Voltage: 0 to 5V	200 [mV]	500 [mV]	4500 [mV]	4800 [mV]	100 [mV]
Voltage: 0 to 10V	200 [mV]	500 [mV]	9500 [mV]	9800 [mV]	200 [mV]
Current: 0 to 20mA	0 [uA]	0 [uA]	20000 [uA]	20000 [uA]	250 [uA]
Current: 4 to 20mA	1000 [uA]	4000 [uA]	20000 [uA]	21000 [uA]	250 [uA]
Freq: 0.5Hz to 20kHz	100 [Hz]	150 [Hz]	2400 [Hz]	2500 Hz]	5 [Hz]
Freq: RPM Mode	500 [0.1RPM]	1000 [0.1RPM]	30000 [0.1RPM]	33000 [0.1RPM]	100 [0.1RPM]
PWM: 0 to 100%	10 [0.1%]	50 [0.1%]	950 [0.1%]	990 [0.1%]	10 [0.1%]
Digital Input	OFF	OFF	ON	ON	0
Counter Input	0	0	60000	60000	60000

Table 10: AI Object Defaults Based on Sensor Type and Input Range

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

Sensor Type/ Input Range	7148h	7120h	7122h	7149h	2111h

Voltage: 0 to 5V and 0 to 10V Current: 0 to 20mA RPM: 0 to 6000RPM PWM: 0 to 100%	0 to 7120h	7148h to 7122h	7120h to 7149h If(7149h>MAX) 7120h to MAX	7122h to 110% of MAX	10% of MAX
Current: 4 to 20mA	0 to 7120h	7148h to 7122h If(7148h<4mA) 4mA to 7122h			
Freq: 0.5Hz to 20kHz	0.1Hz to 7120h	7148h to 7122h If(7148h<0.5Hz) 0.5Hz to 7122h			

Table 11: AI Object Ranges Based on Sensor Type and Input Range

The last objects associated with the analog input block left to discuss are those associated with fault detection. Should the calculated input (after measuring and filtering) fall outside of the allowable range, as defined by the AI Span Start and AI Span End objects, an error flag will be set in the application if and only if object 2110h **AI Error Detect Enabled** is set to TRUE (1).

When (7100h AI Input FV < 7148h AI Span Start), an "Out of Range Low" flag is set. If the flag stays active for the 2112h **AI Error Reaction Delay** time, an Input Overload Emergency (EMCY) message will be added to object 1003h **Pre-Defined Error Field**. Similarly, when (7100h AI Input FV > 7149h AI Span End), an "Out of Range High" flag is set, and will create an EMCY message should it stay active throughout the delay period. In either case, the application will react to the EMCY message as defined by object 1029h **Error Behaviour** at the sub-index corresponding to an Input Fault. Refer to section 3.2.4 and 3.2.13 for more information about objects 1003h and 1029h.

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

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

1.4. Lookup Table Function Block

The lookup table (LTz) function blocks are not used by default.
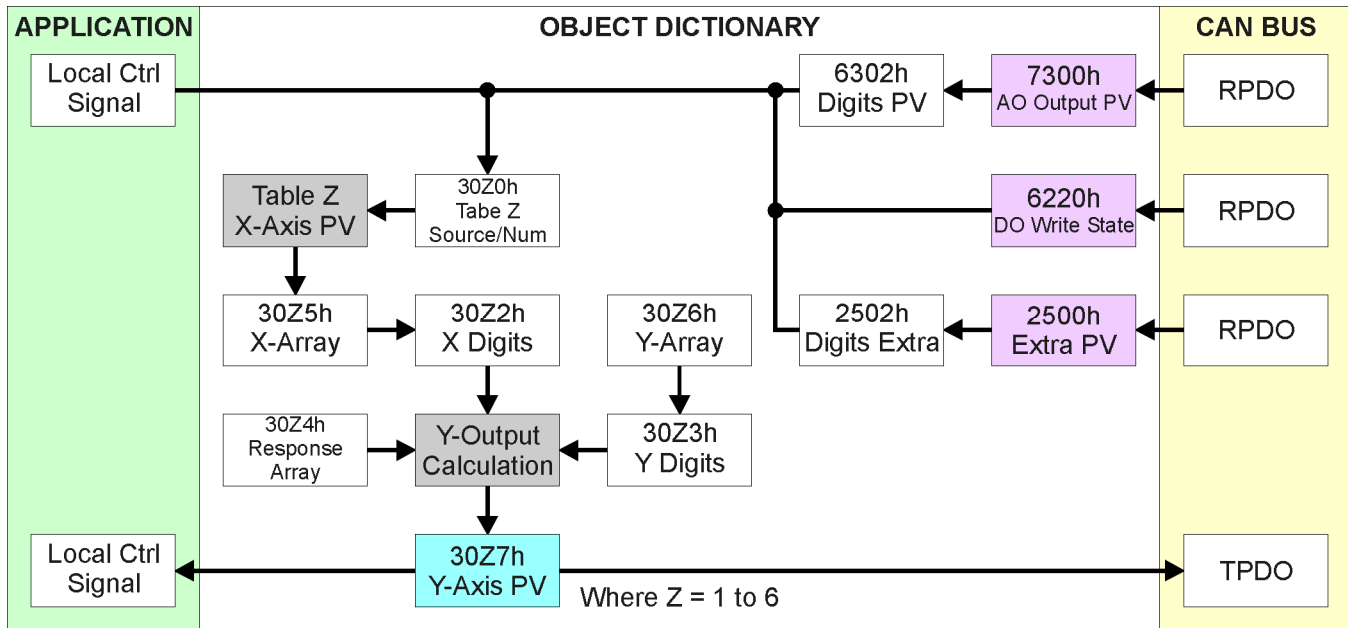
Figure 16 – Lookup Table Objects

Lookup tables are used to give an output response of up to 10 slopes per input. The array size of the objects 30z4h **LTz Point Response**, 30z5h **LTz Point X-Axis PV** and 30z6h **Point Y-Axis PV** shown in the block diagram above is therefore 11.

Note: If more than 10 slopes are required, a Logic Block can be used to combine up to three tables to get 30 slopes, as is described in Section 1.8.

There are two key parameters that will affect how this function block will behave. The objects 30z0h **Lookup Table z Input X-Axis Source** and 30z1h **Lookup Table z Input X-Axis Number** together define the control source for the function block. When it is changed, the table the values in object 30z5h need to be updated with new defaults based on the X-Axis source selected as described in Tables 15 and 16.

The second parameter that will affect the function block, is object **30z4h 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)*,* which is described later in Section 1.7.4.

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.

For example, if the control source is a Universal Input, setup as a 0-5V type, with an operating range of 0.5V to 4.5V. Object 30z2h **LTz X-Axis Decimal Digits PV** should be set to match that of object 2102 **AI Decimal Digits FV**. The X-Axis could be setup to have a "**LTz Point X-Axis PV sub-index 2**" of 500, and setpoint "**LTz Point X-Axis PV sub-index 11**" will be set to 4500. The first point "**LTz Point X-Axis PV sub-index 1**" should start from 0 in this case.
For most *'Data Responses'*, the default value at point (1,1) is [0,0].

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.

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

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 17.

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 17.

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.**

For example, should the X-Axis of a table be a resistive value (as read from an analog input), the output of the table could be temperature from an NTC sensor in the range $Y_1 = 125$ºC to $Y_{11} = -20$ºC. If this table is used as the control source for another function block (i.e. feedback to a PID control), then Scaling 1 would be -20 and Scaling 2 would be 125 when used in a linear formula.
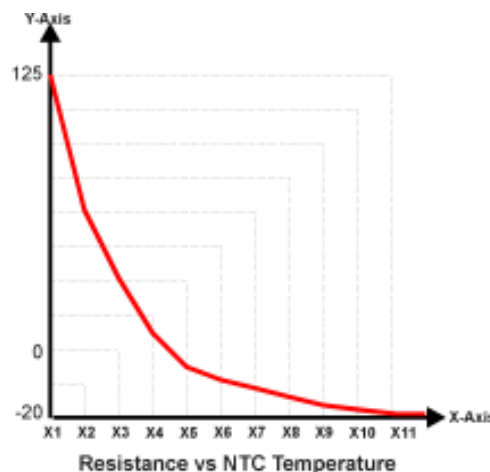
Figure 17 – Lookup Table Example Resistance vs. NTC Temperature

In all cases the controller looks at the **entire range** of the data in the Y-Axis sub-indices 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 1.7.3, they are still used in the Y-Axis range determination. If not all the data points are going to be used, it is recommended that Y10 be set to the minimum end of the range, and Y11 to the maximum first. This way, the user can get predictable results when using the table to drive another function block, such as an analog output.

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 30z4h **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 18 shows the difference between these two response profiles with the default settings.
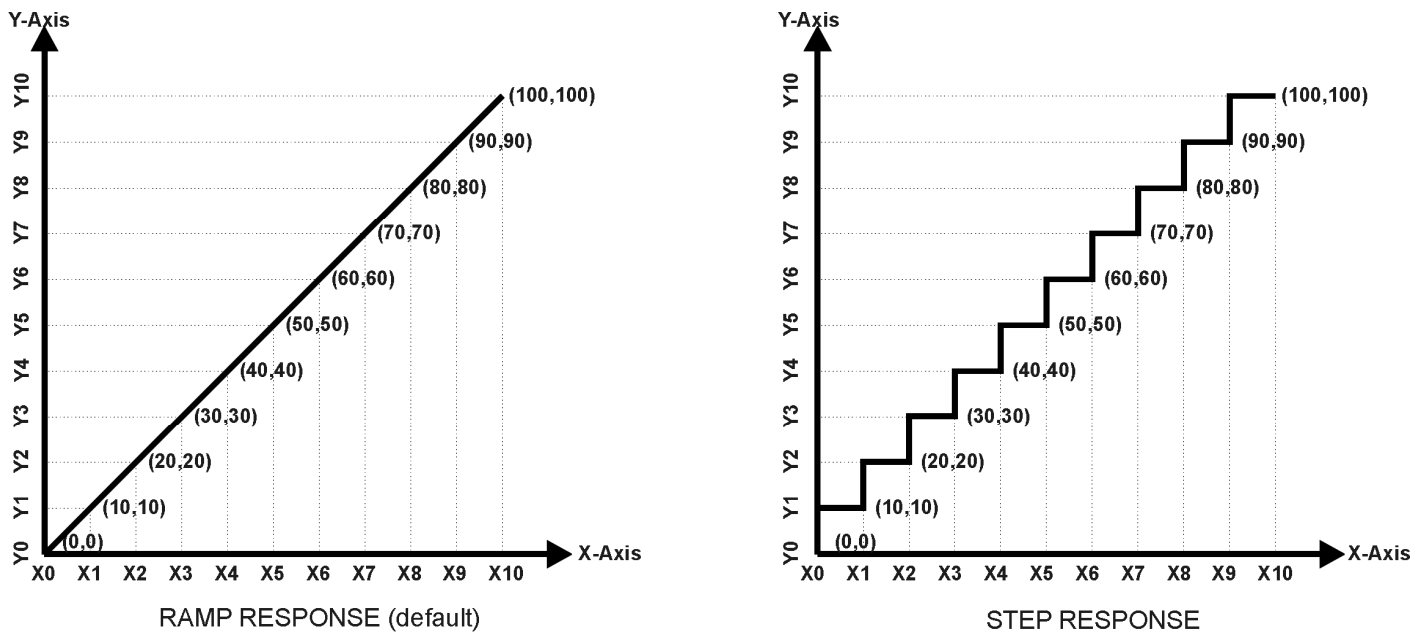
Figure 18 – Lookup Table Defaults with Ramp and Step Responses

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 19. 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.')*
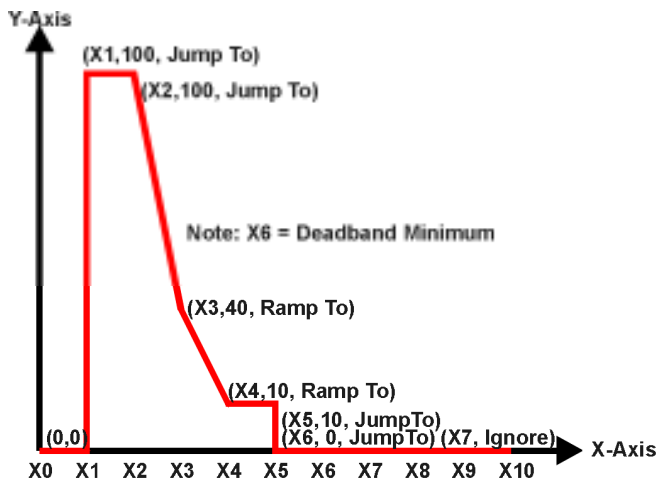
Figure A - Dual Slope "Profile B"
Joystick Deadband Response
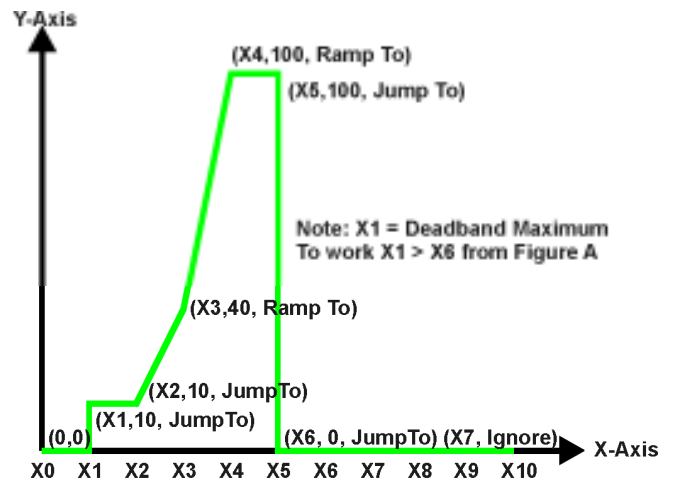
Figure B - Dual Slope "Profile A"
Joystick Deadband Response

Figure 19 – Lookup Table Examples to Setup for Dual-Slope Joystick Deadband Response

To summarize, Table 24 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: LTz Point Response Options

1.4.4. X-Axis, Time Response

As mentioned in Section 1.5, 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.

In this case, the X-Axis control source is treated as a digital input. If the signal is actually an analog input, it is interpreted like a digital input per Figure 5. 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.

When the control input is OFF, the output is always at zero. When the input come ON, the profile ALWAYS starts at position (X_1, Y_1) which is 0 output for 0ms.

When using the lookup table to drive an output based on **time**, it is mandatory that objects 2330h **Ramp Up** and 2331h **Ramp Down** in the analog output function block be set to **zero**. Otherwise, the output result will not match the profile as expected. Recall, also, that the AO scaling should be

set to match the Y-Axis scaling of the table in order to get a 1:1 response of AO Output FV versus LTz Output Y-Axis PV.

An application where the time response feature would be useful is filling a clutch when a transmission is engaged. An example of some fill profiles is shown in Figure 20.
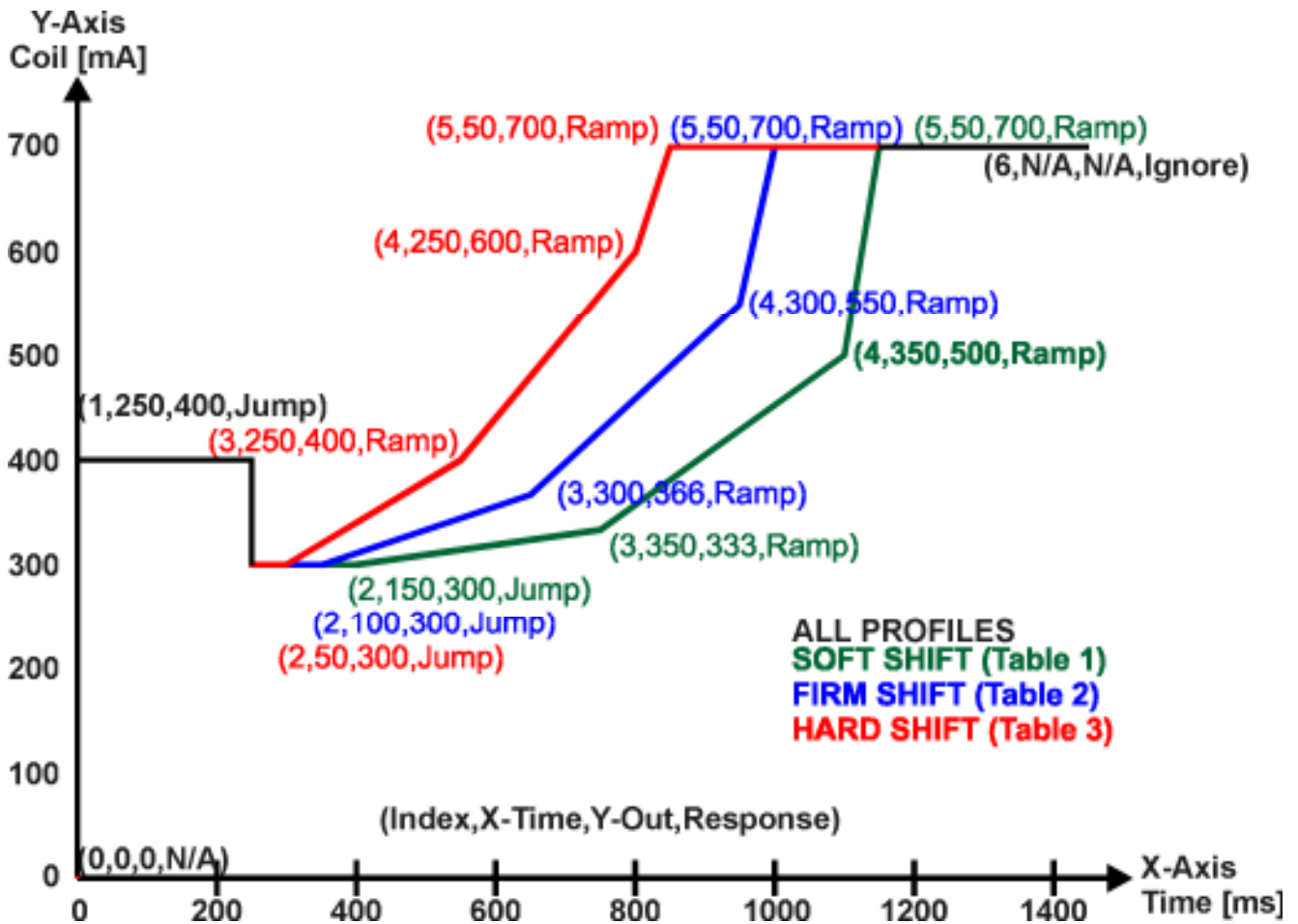
Figure 20 – Lookup Table Time Response Clutch Fill Profiles

In a time response, the data in object 30z5h **LTz Point X-Axis PV** is measured in milliseconds, and object 30z2h **LTz 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]

1.4.5. Lookup Table Final Note

One final note about the lookup tables is that if a digital input is selected as the control source for the X-Axis, only a 0 (Off) or 1 (On) will be measured. Ensure that the data range for the X-Axis on the table is updated appropriately in this condition.

1.5. Programmable Logic Function Block

The programmable logic block (LBx) functions are not used by default.
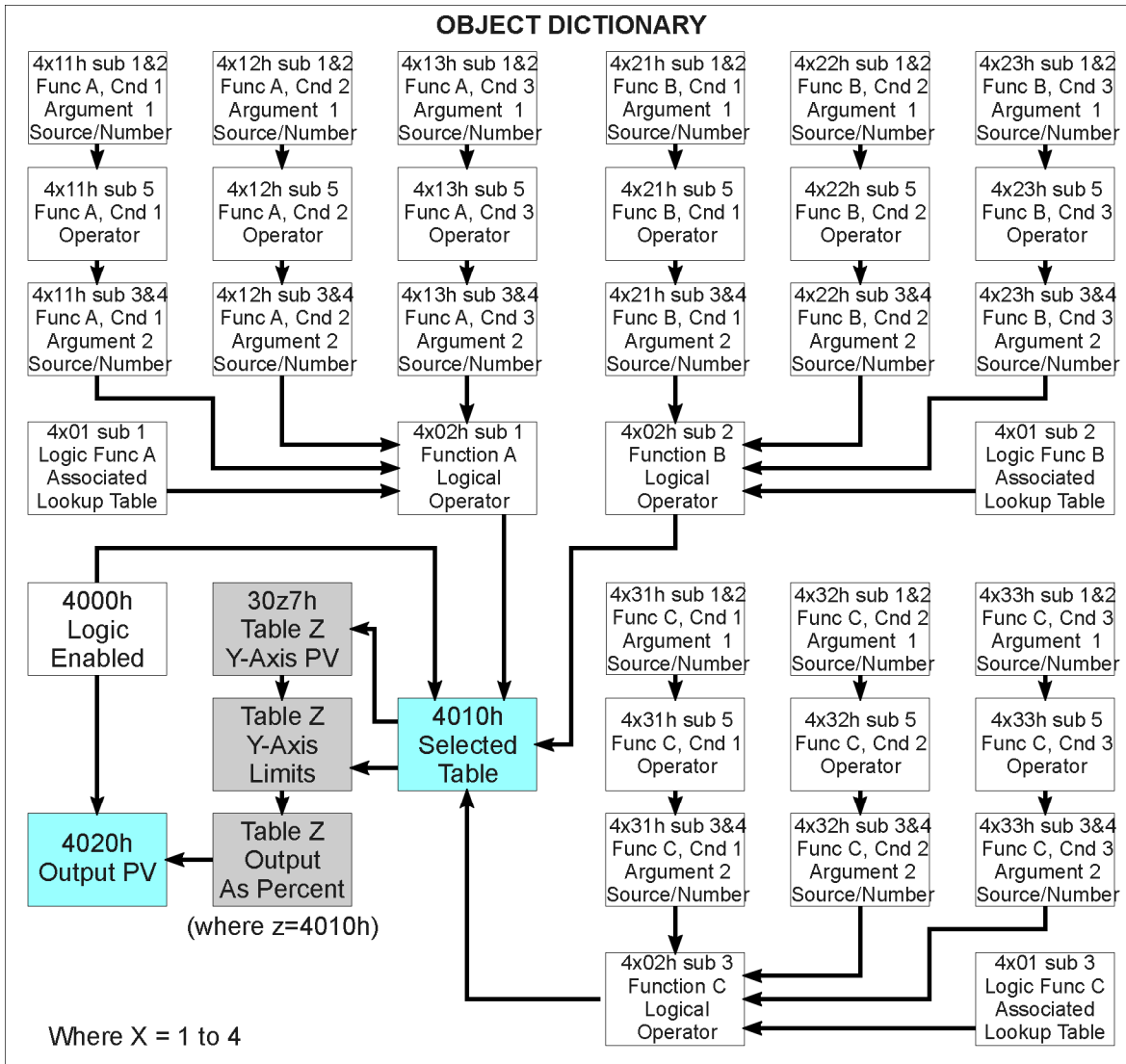
Figure 21 – Logic Block Objects

This function block is obviously the most complicated of them all, but very powerful. Any LBx (where X=1 to 4) 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 6) can be associated with the logic, and which ones are used is fully configurable on object 4x01 **LBx Lookup Table Number**.

Should the conditions be such that a particular table (A, B or C) has been selected as described in Section 1.8.2, then the output from the selected table, at any given time, will be passed directly to LBx's corresponding sub-index X in read-only mappable object 4020h **Logic Block Output PV**. The active table number can read from read-only object 4010h **Logic Block Selected Table**.

Therefore, an LBx allows up to three different responses to the same input, or three different responses to different inputs, to become the control for another function block, such as an analog

output. Here, the "**Control Source**" for the reactive block would be selected to be the *'Programmable Logic Function Block,'* as described in Section 1.5.

In order to enable any one of logic blocks, the corresponding sub-index in object 4000h **Logic Block Enable** must be set to TRUE. They are all disabled by default.

Logic is evaluated in the order shown in Figure 22. 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.**
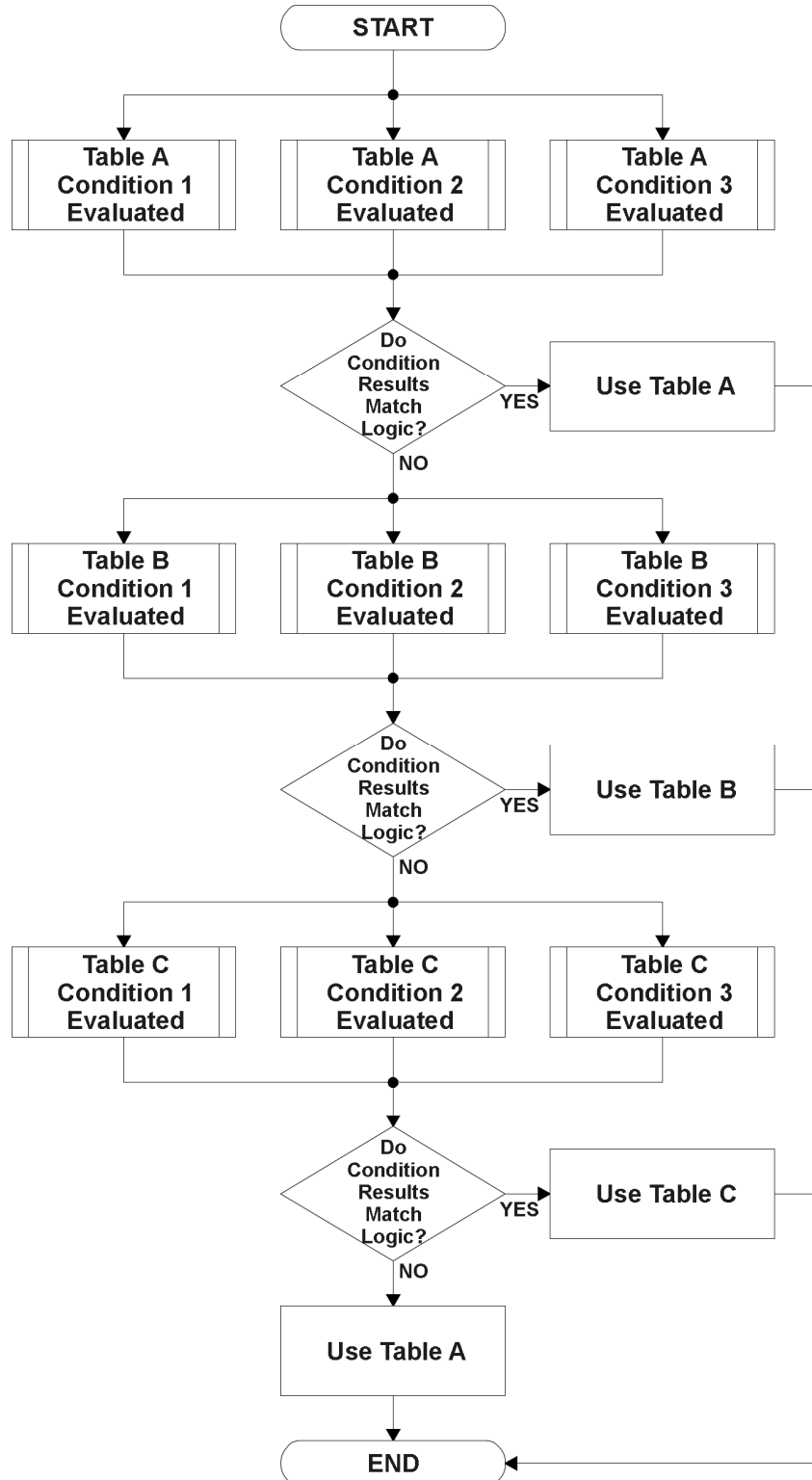
Figure 22 – Logic Block Flowchart

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 25.

Index	Sub-Index	Name	Data Type
4xyz*	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 = 1 to 4, Y = A, B or C, and Z = 1 to 3
Table 13: LBx Condition Structure Definition

Objects 4x11h, 4x12h and 4x13h are the conditions evaluated for selecting Table A.
Objects 4x21h, 4x22h and 4x23h are the conditions evaluated for selecting Table B.
Objects 4x31h, 4x32h and 4x33h 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 4xyzh 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 26, 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: LBx Condition Operator Options

For example, a condition for a transmission control shift selection, as shown in Figure 20 in the previous section, could be that the Engine RPM be less than a certain value to select a Soft Fill profile. In this case, **"Argument 1 Source"** could be set to *'Analog Input Function Block'* (where the input is configured for RPM pickup)*,* **"Argument 2 Source"** to *'Constant Function Block'*, and the **"Operator"** to *'<, Less Than.'* Object 5010h Constant FV at sub-index **"Argument 2 Number"** would be set to whatever cutoff RPM the application required.

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 27.

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: LBx 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 1.8.1. There are several logical combinations that can be selected, as listed in Table 28. The default value for object 4x02h **LBx 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: LBx 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 29.

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: LBx Conditions Evaluation Based on Selected Logical Operator

If the result of the function logic is TRUE, then the associated lookup table (see object 4x01h) is immediately selected as the source for the logic output. No further conditions for other tables are evaluated. For this reason, the *'Default Table'* should always be setup as the highest letter table being used (A, B or C) If no default response has been setup, the Table A automatically becomes the default when no conditions are true for any table to be selected. This scenario should be avoided whenever possible 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 4010h **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 LBx function block does NOT mean lookup table 1 to 3. Each table has object 4x01h LBx **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 30.

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	1	2	3
4	4	5	6

Table 18: LBx Default Lookup Tables

If the associated Lookup Table Z (where Z equals 4010h sub-index X) does not have an **"X-Axis Source"** selected, then the output of LBx will always be "Not Available" so long as that table is selected. However, should LTz be configured for a valid response to an input, be it Data or Time, the output of the LTz function block (i.e. the Y-Axis data that has been selected based on the X-Axis value) will become the output of the LBx function block so long as that table is selected.

The LBx output is always setup as a percentage, based on the range of the Y-Axis for the associated table (see Section 1.7.2) It is written to sub-index X of read-only object 4020h **Logic Block Output PV** with a resolution of 1 decimal place.

1.5.4. Application Ideas

This section is not meant to be a comprehensive list of all the possibilities that the Logic Block offers. Rather, it is meant to show how some common, but widely diversified functions can be achieved by using it.

a) Dual Speed Application – Under certain conditions, an analog output could be driven between Min_A to Max_A while under others, the speed is limited by having the output respond to changes at the input between Min_B and Max _B.

b) Multi-Speed Transmission Control – By using a Forward input as the enable of one analog output, and the Reverse input as the other, different clutch fill profiles could be selected based on Engine Speed as discussed in earlier examples.

c) Getting better resolution (i.e. up to 30 slopes) on a resistive to temperature curve for an NTC sensor. The condition for Table A would be input resistance <= R1, Table B is input <= R2 and Table C as the default for high resistance values.

1.6. 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.
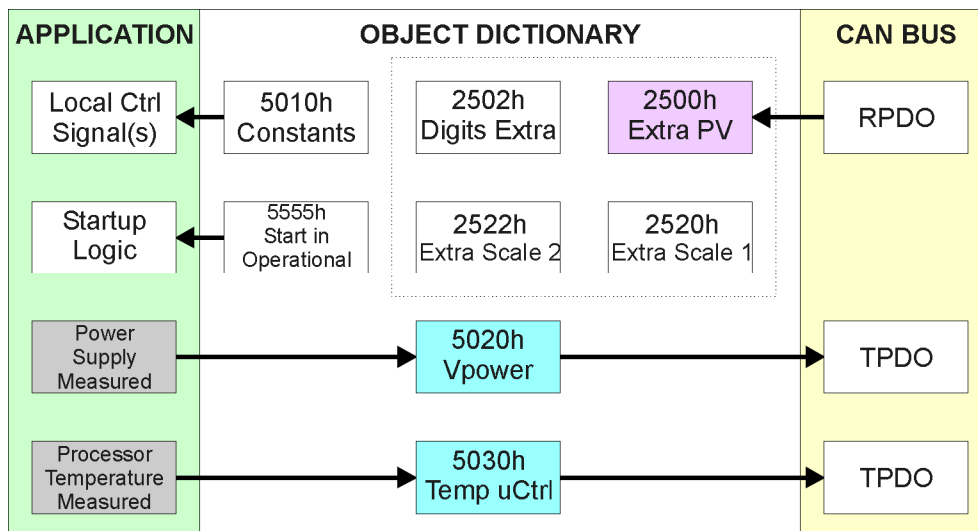
Figure 23 – Miscellaneous Objects

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 1.5, Table 16. 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 17.

Objects 5020h **Power Supply FV** and 5030h **Processor Temperature FV** are available as read-only feedback for additional diagnostics.

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 4 other sub-indexes provided for user selectable values. (Defaults 25, 50, 75 and 100)

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 the logic block, and they can also be used as a setpoint target for a PID control block.

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 1IN-CAN 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.

2. INSTALLATION INSTRUCTIONS

2.1. Dimensions and Pinout

The Single Input, Dual Output Valve Controller is packaged in an encapsulated aluminum enclosure, as shown in Figure 24. The assembly carries an IP67 rating.
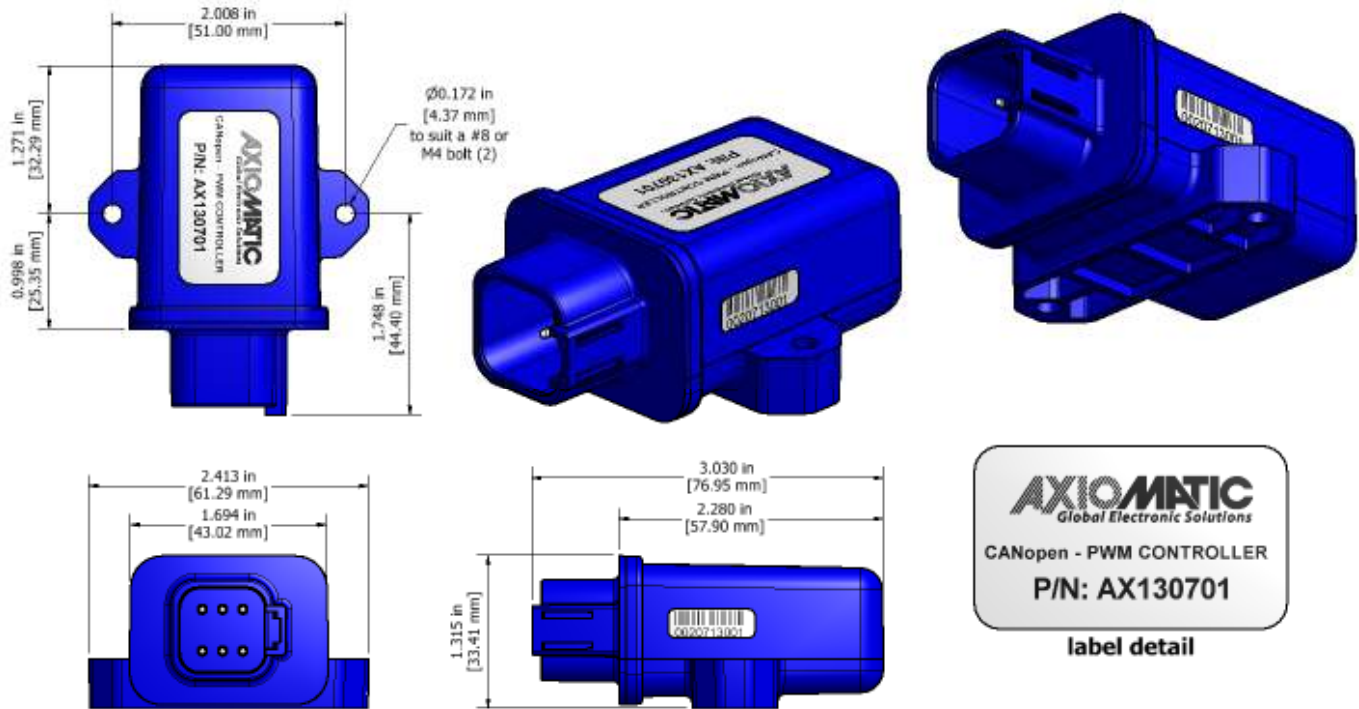
Figure 24 – Housing Dimensions

CAN and I/O Connector	
Pin #	Function
1	BATT+
2	Input+
3	CAN_L
4	CAN_H
5	Input-
6	BATT-

Table 19: Connector Pinout

6 pin Deutsch IPD connector P/N: DT04-6P
A mating plug kit is available as Axiomatic P/N: **AX070119**.

2.2. Installation Instructions

2.2.1. Notes & Warnings

- Do not install near high-voltage or high-current devices.
- Ground the chassis for safety purposes and proper EMI shielding.
- Note the operating temperature range. All field wiring must be suitable for that temperature range.
- Install the unit with appropriate space available for servicing and for adequate wire harness access (15 cm) and strain relief (30 cm).
- Do not connect or disconnect the unit while the circuit is live, unless the area is known to be non-hazardous.

2.2.2. Mounting

The module is designed for mounting on the valve block. If it is mounted without an enclosure, the controller should be mounted horizontally with connectors facing left or right, or with the connectors facing down, to reduce likelihood of moisture entry.

Mask all labels if the unit is to be repainted, so label information remains visible.

Mounting legs include holes sized for #10 or M4.5 bolts. The bolt length will be determined by the end-user's mounting plate thickness. Typically 20 mm (3/4 inch) is adequate.

If the module is mounted away from the valve block, no wire or cable in the harness should exceed 30 meters in length. The power input wiring should be limited to 10 meters.

2.2.3. Connections

Use the following Deutsch IPD mating plugs to connect to the integral receptacles. Wiring to these mating plugs must be in accordance with all applicable local codes. Suitable field wiring for the rated voltage and current must be used. The rating of the connecting cables must be at least 85°C. For ambient temperatures below –10°C and above +70°C, use field wiring suitable for both minimum and maximum ambient temperature.

Receptacle	Mating Sockets as appropriate (Refer to www.laddinc.com for more information on the contacts available for this mating plug.)
Mating Connector	DT06-12SA and wedge W12S

2.2.4. Noise – Electrical Connections and Shielding

To reduce noise, separate all power and output wires from those of the input and CAN. Shielded wires will protect against injected noise. Shield wires should be connected at the power or input source, or at the output load.

The CAN shield can be connected at the controller using the CAN Shield pin provide on the connector. However the other end should not be connected in this case.

All wires used must be 16 or 18 AWG.

2.2.5. CAN Network Constructions

Axiomatic recommends that multi-drop networks be constructed using a "daisy chain" or "backbone" configuration with short drop lines.

2.2.6. CAN Termination

It is necessary to terminate the network; therefore an external CAN termination is required. No more than two network terminators should be used on any one single network. A terminator is a 121Ω, 0.25 W, 1% metal film resistor placed between CAN_H and CAN_L terminals at the end two nodes on a network.

3. CANOPEN ® OBJECT DICTIONARY

The CANopen object dictionary of the 1IN-CAN 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.

3.1. NODE ID and BAUDRATE

By default, the 1IN-CAN Controller ships factory programmed with a Node ID = 127 (0x7F) and with Baudrate = 125 kbps.

3.1.1. LSS Protocol to Update

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

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

3.1.2. Setting Node-ID

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

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

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

Item	Value	
COB-ID	0x7E5	
Length	2	
Data 0	0x11	(cs=17 for configure node-id)
Data 1	Node-ID	(set new Node-ID as a hexadecimal number)

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

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

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

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

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

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

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

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

3.1.3. Setting Baudrate

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

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

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

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

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 20: LSS Baudrate Indexes

- 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 baudrate 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 baudrate on the CAN Scope tool was changed from 125 to 250 kbps.

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

The communication objects supported by the 1IN-CAN 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	RW	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
1600	RPDO1 Mapping Parameter	RECORD		RO	No
1601	RPDO2 Mapping Parameter	RECORD		RO	No
1602	RPDO3 Mapping Parameter	RECORD		RO	No
1603	RPDO4 Mapping Parameter	RECORD		RO	No
1800	TPDO1 Communication Parameter	RECORD		RW	No
1801	TPDO2 Communication Parameter	RECORD		RW	No
1802	TPDO3 Communication Parameter	RECORD		RW	No
1803	TPDO4 Communication Parameter	RECORD		RW	No
1A00	TPDO1 Mapping Parameter	RECORD		RW	No
1A01	TPDO2 Mapping Parameter	RECORD		RW	No
1A02	TPDO3 Mapping Parameter	RECORD		RW	No
1A03	TPDO4 Mapping Parameter	RECORD		RW	No

3.2.1. 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 = 0x201F	**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	0xE01F0194
Default Value	0xE01F0194

3.2.2. Object 1001h: Error Register

This object is an error register for the device. Any time there is an error detected by the 1IN-CAN 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 1IN-CAN 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

3.2.3. Object 1002h: Manufacturer Status Register

This object is used for manufacturer debug purposes.

3.2.4. 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 in 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 1IN-CAN 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 input fault is detect as described in Section 1.3 or an analog output is not working as described in Section 1.5, 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 32 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)					
2001F001h	20h	Positive Overload (Out of Range High)	01h	Analog Input 1	F001h	Input Overload
4001F001h	40h	Negative Overload (Out of Range Low)	01h	Analog Input 1	F001h	Input Overload
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

Table 21: 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 4
Default Value	0

Sub-Index	1h to 4
Description	Standard error field
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

3.2.5. 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

3.2.6. 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

3.2.7. 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 1IN-CAN 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 1IN-CAN 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	4
Default Value	4

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

3.2.8. 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 1IN-CAN 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 1IN-CAN 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 1IN-CAN 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

3.2.9. Object 1016h: Consumer Heartbeat Time

The 1IN-CAN Controller can be a consumer of heartbeat objects for up to four modules. This object defines the expected heartbeat cycle time for those modules, and if set to zero, it is not used. When 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 1IN-CAN 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	4
Default Value	4

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

3.2.10. Object 1017h: Producer Heartbeat Time

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

Object Description

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

Entry Description

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

3.2.11. Object 1018h: Identity Object

The identity object indicates the data of the 1IN-CAN 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	0xAA031701
Default Value	0xAA031701

Sub-Index	3h
Description	Revision Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00010100

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

3.2.12. 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 0x10082010 would indicate that the software was compiled on August 10th, 2010. A time value of 0x00001620 would indicate it was compiled at 4:20pm.

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

3.2.13. Object 1029h: Error Behaviour

This object controls the state that the 1IN-CAN Controller will be set into in case of an error of the type associated with the sub-index.

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 3.2.14 for more information) or if a lifeguard or heartbeat message is not received as expected. Input faults are defined in Section 1.3, and output faults are defined in Section 1.5.

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	1 (No State Change)
Sub-Index	2h
Description	Digital Input Fault (not used)
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	3h
Description	Analog Input Fault (AI1)
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	4h
Description	Digital Output Fault (not used)
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

Sub-Index	5h
Description	Analog Output Fault (not used)
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	1 (No State Change)

3.2.14. 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 1IN-CAN Controller can support up to four RPDO messages. All RPDOs on the 1IN-CAN 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 four 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	4	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

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

Sub-Index	Value	Object
0	2	Number of mapped application objects in PDO
1	0x25000510	Extra Received 1 PV (i.e. PID Control Feedback 1 PV)
2	0x25000610	Extra Received 2 PV (i.e. PID Control Feedback 2 PV)
3	0	Not used by default
4	0	Not used by default

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

Sub-Index	Value	Object
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

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

Sub-Index	Value	Object
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

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 1403h
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		X	RPDOx ID
Description	COB-ID used by RPDO			
Access	RW		1	0200h
PDO Mapping	No		2	0300h
Value Range	See value definition in DS-301		3	0400h
Default Value	40000000h + RPDO1 + Node ID		4	0500h
	C0000000h + RPDOx + Node-ID			

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.

3.2.15. TPDO Behaviour

The 1IN-CAN Controller can support up to four TPDO messages. All TPDOs on the 1IN-CAN 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 four have valid default mappings defined (see below) only TPDO1 is enabled by default (i.e. TPDO exists).

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

Sub-Index	Value	Object
0	3	Number of mapped application objects in PDO
1	0x71000110	Analog Input 1 Field Value
2	0x71000210	Analog Input 1 Frequency Measured Field Value
3	0	Not used by default
4	0	Not used by default

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

Sub-Index	Value	Object
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

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

Sub-Index	Value	Object
0	2	Number of mapped application objects in PDO
1	0x24600110	PID Control Output 1 Field Value
2	0x24600210	PID Control Output 2 Field Value
3	0	Not used by default
4	0	Not used by default

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

Sub-Index	Value	Object
0	2	Number of mapped application objects in PDO
1	0x50200020	Power Supply Field Value (measured)
2	0x50300020	Processor Temperature Field Value (measured)
3	0	Not used by default
4	0	Not used by default

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

Object Description

Index	1800h to 1803h
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 + TPDO1 + Node-ID C0000000h + TPDOx + Node-ID

X	TPDOx ID
1	0180h
2	0280h
3	0380h
4	0480h

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	100ms (on TPDO1) 0ms (on TPDO2, TPDO3, TPDO4)

3.3. APPLICATION OBJECTS (DS-404)

Index (hex)	Object	Object Type	Data Type	Access	PDO Mapping
6020	DI Read State 1 Input Line	ARRAY	BOOLEAN	RO	Yes
6030	DI Polarity 1 Input Line	ARRAY	UNSIGNED8	RW	No
7100	AI Input Field Value	ARRAY	INTEGER16	RO	Yes
6110	AI Sensor Type	ARRAY	UNSIGNED16	RW	No
6112	AI Operating Mode	ARRAY	UNSIGNED8	RW	No
7120	AI Input Scaling 1 FV	ARRAY	INTEGER16	RW	No
7121	AI Input Scaling 1 PV	ARRAY	INTEGER16	RW	No
7122	AI Input Scaling 2 FV	ARRAY	INTEGER16	RW	No
7123	AI Input Scaling 2 PV	ARRAY	INTEGER16	RW	No
7130	AI Input Process Value	ARRAY	INTEGER16	RO	Yes
6132	AI Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
7148	AI Input Span Start	ARRAY	INTEGER16	RW	No
7149	AI Input Span End	ARRAY	INTEGER16	RW	No
61A0	AI Filter Type	ARRAY	UNSIGNED8	RW	No
61A1	AI Filter Constant	ARRAY	UNSIGNED16	RW	No

3.3.1. Object 6020h: DI Read State 1 Input Line

This read-only object represents the digital input state from a single input line. Refer to Section 1.2 for more information

Object Description

Index	6020h
Name	DI Read State 1 Input Line
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

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

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

3.3.2. Object 6030h: DI Polarity 1 Input Line

This object determines how the state read on the input pin corresponds to the logic state, in conjunction with manufacturer object 2020h, as defined in Table 3.

Object Description

Index	6030h
Name	DI Polarity 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	Digital Input 1 Polarity
Access	RW
PDO Mapping	No
Value Range	See Table 3

Default Value	0 (Normal On/Off)

3.3.3. Object 7100h: AI Input Field Value

This object represents the measured value of an analog input that has been scaled as per manufacturer object 2102h AI Decimal Digits PV. The base unit for each type of input is defined in Table 9, as well as the read-only resolution (decimal digits) associated with the FV.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 FV
Access	RO
PDO Mapping	Yes
Value Range	Data Type Specific, see Table 11
Default Value	No

3.3.4. Object 6110h: AI Sensor Type

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

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Sensor Type
Access	RW

PDO Mapping	No
Value Range	See Table 5
Default Value	40 (voltage)

3.3.5. Object 6112h: AI Operating Mode

This object enables special operating modes for the input.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Operating Mode
Access	RW
PDO Mapping	No
Value Range	See Table 4
Default Value	1 (normal operation)

3.3.6. Object 7120h: AI Input Scaling 1 FV

This object describes the field value of the first calibration point for the analog input channel, as shown in Figure 7. It also defines the "minimum" value of the analog input range when using this input as a control source for another function block, as described in Table 17 in Section 1.5. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

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

Entry Description

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

Sub-Index	1h

Description	AI1 Scaling 1 FV
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	500 [mV]

3.3.7. Object 7121h: AI Input Scaling 1 PV

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

Object Description

Index	7121h
Name	AI Input Scaling 1 PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h
Description	AI1 Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	500 [same as 7120h]

3.3.8. Object 7122h: AI Input Scaling 2 FV

This object describes the field value of the second calibration point for the analog input channel, as shown in Figure 7. It also defines the "maximum" value of the analog input range when using this input as a control source for another function block, as described in Table 17 in Section 1.5. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Scaling 2 FV
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	4500 [mV]

3.3.9. Object 7123h: AI Input Scaling 2 PV

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

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	4500 [same as 7122h]

3.3.10. Object 7130h: AI Input Process Value

This object represents the result of the input scaling applied per Figure 7, and gives the measured quantity scaled in the physical unit of the process value (i.e. °C, PSI, RPM, etc) with the resolution defined in object 6132h AI Decimal Digits PV.

Object Description

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

Entry Description

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

Value Range	1
Default Value	1

Sub-Index	1h
Description	AI1 Process Value
Access	RO
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

3.3.11. Object 6132h: AI Decimal Digits PV

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

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

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	3 [Volt to mV]

3.3.12. Object 7148h: AI Span Start

This value specifies the lower limit where field values are expected. Field values which are lower than this limit are marked as negative overload. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

Index	7148h
Name	AI Span Start

Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h
Description	AI1 Span Start (Error Min)
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	200 [mV]

3.3.13. Object 7149h: AI Span End

This value specifies the upper limit where field values are expected. Field values which are higher than this limit are marked as positive overload. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Span End (Error Max)
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	4800 [mV]

3.3.14. Object 61A0h: AI Filter Type

This object defines the type of data filter that will be applied to the raw input data, as read from the ADC or Timer, before it is passed to the field value object. The types of data filters are defined in Table 8, and how they are used is outlined in Section 1.3.

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Filter Type
Access	RW
PDO Mapping	No
Value Range	See Table 8
Default Value	0 (no filter)

3.3.15. Object 61A1h: AI Filter Constant

This object defines the number of steps used in the various filters, as defined in Section 1.3

Object Description

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

Entry Description

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

Sub-Index	1h
Description	AI1 Filter Constant
Access	RW
PDO Mapping	No
Value Range	1 to 1000

Default Value	10

3.4. MANUFACTURER OBJECTS

Index (hex)	Object	Object Type	Data Type	Access	PDO Mapping
2020	DI Pull Up/Down Mode 1 Input Line	ARRAY	UNSIGNED8	RW	No
2021	DI Debounce Time	ARRAY	UNSIGNED16	RW	No
2030	DI Debounce Filter 1 Input Line	ARRAY	UNSIGNED8	RW	No
2031	DI Frequency Debounce Time	ARRAY	UNSIGNED8	RW	No
2040	DI Reset Pulse Count	ARRAY	UNSIGNED32	RW	No
2041	DI Time Window	ARRAY	UNSIGNED32	RW	No
2031	DI Pulse Window	ARRAY	UNSIGNED32	RW	No
2100	AI Input Range	ARRAY	UNSIGNED8	RW	No
2101	AI Number of Pulses Per Revolution	ARRAY	UNSIGNED16	RW	No
2102	AI Decimal Digits FV	ARRAY	UNSIGNED8	RW	No
2103	AI Filter Frequency for ADC	ARRAY	UNSIGNED8	RW	No
2110	AI Error Detect Enable	ARRAY	BOOLEAN	RW	No
2111	AI Error Clear Hysteresis	ARRAY	INTEGER16	RW	No
2112	AI Error Reaction Delay	ARRAY	UNSIGNED16	RW	No
2500	EC Extra Received Process Value	ARRAY	INTEGER16	RW	Yes
2502	EC Decimal Digits PV	ARRAY	UNSIGNED8	RW	No
2520	EC Scaling 1 PV	ARRAY	INTEGER16	RW	No
2522	EC Scaling 2 PV	ARRAY	INTEGER16	RW	No
30z0	LTz Input X-Axis Source	VAR	UNSIGNED8	RW	No
30z1	LTz Input X-Axis Number	VAR	UNSIGNED8	RW	No
30z2	LTz X-Axis Decimal Digits PV	VAR	UNSIGNED8	RW	No
30z3	LTz Y-Axis Decimal Digits PV	VAR	UNSIGNED8	RW	No
30z4	LTz Point Response	ARRAY	UNSIGNED8	RW	No
30z5	LTz Point X-Axis PV	ARRAY	INTEGER16	RW	No
30z6	LTz Point Y-Axis PV	ARRAY	INTEGER16	RW	No
30z7	LTz Output Y-Axis PV	ARRAY	INTEGER16	RO	Yes
4000	Logic Block Enable	ARRAY	UNSIGNED8	RW	No
4010	Logic Block Selected Table	ARRAY	UNSIGNED8	RW	No
4020	Logic Output Process Value	ARRAY	INTEGER16	RO	Yes
4x01	LBx Lookup Table Number	ARRAY	UNSIGNED8	RW	No
4x02	LBx Function Logical Operator	ARRAY	UNSIGNED8	RW	No
4x11	Logic Block A Function A Condition 1	ARRAY	RECORD	RW	No
4x12	Logic Block A Function A Condition 2	ARRAY	RECORD	RW	No
4x13	Logic Block A Function A Condition 3	ARRAY	RECORD	RW	No
4x21	Logic Block A Function B Condition 1	ARRAY	RECORD	RW	No
4x22	Logic Block A Function B Condition 2	ARRAY	RECORD	RW	No
4x23	Logic Block A Function B Condition 3	ARRAY	RECORD	RW	No
4x31	Logic Block A Function C Condition 1	ARRAY	RECORD	RW	No
4x32	Logic Block A Function C Condition 2	ARRAY	RECORD	RW	No
4x33	Logic Block A Function C Condition 3	ARRAY	RECORD	RW	No
5010	Constant Field Value	ARRAY	FLOAT32	RW	No

5020	Power Supply Field Value	VAR	FLOAT32	RO	Yes
5030	Processor Temperature Field Value	VAR	FLOAT32	RO	Yes
5555	Start in Operational Mode	VAR	BOOLEAN	RW	No

Where z = 1 to 6 and x = 1 to 4

3.4.1. Object 2020h: DI Pullup/Down Mode 1 Input Line

This object determines how the state read on the input pin corresponds to the logic state, in conjunction with application object 6020h, as defined in Table 3. The options for this object are listed in Table 1, and the controller will adjust the input hardware according to what is specified.

Object Description

Index	2020h
Name	DI Pullup/Down Mode 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	Digital Input 1 Pullup/Down
Access	RW
PDO Mapping	No
Value Range	See Table 1
Default Value	0 (pullup/down disabled)

3.4.2. Object 2020h: DI Debounce Time 1 Input Line

This object determines the debounce time applied when the input is configured as digital input type. The options for this object are listed below.

Object Description

Index	2021h
Name	DI Debounce Time 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Sub-Index	1h
Description	Digital Input Debounce Time
Access	RW
PDO Mapping	No
Value Range	0 – 60000

Default Value	10 (ms)

3.4.3. Object 2030h: DI Debounce Filter 1 Input Line

This object determines the debounce time of a digital signal when the input is configured as Frequency/RPM or PWM input types. The options for this object are listed in Table 2.

Object Description

Index	2020h
Name	DI Debounce Filter 1 Input Line
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	Digital Input Debounce Filter
Access	RW
PDO Mapping	No
Value Range	See Table 2
Default Value	2 [Filter 1.78 us]

3.4.4. Object 2031h: AI Frequency Overflow Value

This object determines the debounce time of a digital signal when the input is configured as Frequency/RPM or PWM input types.

Object Description

Index	2031h
Name	AI Frequency Overflow Value
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	Frequency Overflow Value
Access	RW
PDO Mapping	No

Value Range	0-50
Default Value	50 (Hz)

3.4.5. Object 2040h: AI Reset Pulse Count Value

This object determines the value (in pulses) which will reset the Counter input type to start count from 0 again. This value is considered when the input is selected as Counter Input type.

Object Description

Index	2040h
Name	AI Reset Pulse Count Value
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

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

Sub-Index	1h
Description	AI Reset Pulse Count Value
Access	RW
PDO Mapping	No
Value Range	0-0xFFFFFFFF
Default Value	1000 (pulses)

3.4.6. Object 2041h: AI Counter Time Window

This object determines the value (in milliseconds) which will be used as a time window to count the pulses detected within it. This value is considered when the input is selected as Counter Input type.

Object Description

Index	2041h
Name	AI Counter Time Window
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

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

Sub-Index	1h
Description	AI Counter Time Window

Access	RW
PDO Mapping	No
Value Range	0-0xFFFFFFFF
Default Value	500 (milliseconds)

3.4.7. Object 2041h: AI Counter Pulse Window

This object determines the value (in pulses) which will be used as a target count for the controller to detect and provide a time (in milliseconds) required to reach such count. This value is considered when the input is selected as Counter Input type.

Object Description

Index	2041h
Name	AI Counter Pulse Window
Object Type	ARRAY
Data Type	UNSIGNED32

Entry Description

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

Sub-Index	1h
Description	AI Counter Pulse Window
Access	RW
PDO Mapping	No
Value Range	0-0xFFFFFFFF
Default Value	1000 (pulses)

3.4.8. Object 2100h: AI Input Range

This object, in conjunction with 6110h AI Sensor Type, defines the analog input defaults (Table 10) and allowable ranges (Table 11) for objects 2111h, 7120h, 7122h, 7148h and 7149h. The number and types of ranges will vary according to what type of sensor is connected to the input, as described in Table 6.

Object Description

Index	2100h
Name	AI Input Range
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Value Range	1
Default Value	1

Sub-Index	1h
Description	AI1 Range
Access	RW
PDO Mapping	No
Value Range	See Table 6
Default Value	2 [0-5V]

3.4.9. Object 2101h: AI Number of Pulses Per Revolution

This object is only used when a "Frequency" input type has been selected by object 6110h. The controller will automatically convert frequency measurement from Hz to RPM when a non-zero value is specified. In this case, objects 2111h, 7120h, 7122h, 7148h and 7149h will be interpreted as RPM data. Object 2100h AI Input Range must still be specified in Hertz, and should be selected according to the expected frequencies that the RPM sensor will operate in.

Object Description

Index	2101h
Name	AI Number of Pulses Per Revolution
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1
Sub-Index	1h
Description	AI1 Pulses per Revolution
Access	RW
PDO Mapping	No
Value Range	0 to 1000
Default Value	1

3.4.10. Object 2102h: AI Decimal Digits FV

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

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

In addition to the FV object 7100h, objects 2111h, 7120h, 7122h, 7148h and 7149h will also be specified with this resolution. This object is read-only, and will be automatically adjusted by the controller as per Table 9 depending on the analog input type and range that has been selected.

Object Description

Index	2102h
Name	AI Decimal Digits FV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	AI1 Decimal Digits FV
Access	RO
PDO Mapping	No
Value Range	See Table 9
Default Value	3 [Volt to mV]

3.4.11.   Object 2103h: AI Filter Frequency for ADC

This object is used to specify the cutoff filter frequency for the ADC peripheral on the processor. The analog-to-digital converter is used with analog input types: voltage; current; and resistive. It is also used to measure: analog output current feedback; power supply voltage, and processor temperature. The available filters are listed in Table 7.

Object Description

Index	2104h
Name	AI Filter Frequency for ADC
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h
Description	ADC Filter Frequency
Access	RW
PDO Mapping	No
Value Range	See Table 7
Default Value	1 [Filter 50Hz]

3.4.12.   Object 2110h: AI Error Detect Enable

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

Object Description

Index	2110h
Name	AI Error Detect Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

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

Sub-Index	1h
Description	AI1 Error Detect Enable
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	1 [TRUE]

3.4.13. Object 2111h: AI Error Clear Hysteresis

This object is used to prevent rapid activation/clearing of an input fault flag, and sending of object 1003h to the CANopen ® network. Once the input has gone above/below the thresholds that define the valid operating range, it must come back into range minus/plus this value to clear the fault. It is scaled in the physical unit of the FV, i.e. object 2102h applies to this object.

Object Description

Index	2111h
Name	AI Error Clear Hysteresis
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h
Description	AI1 Error Clear Hysteresis
Access	RW
PDO Mapping	No
Value Range	See Table 11
Default Value	100 [mV]

3.4.14. Object 2112h: AI 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	2112h
Name	AI Error Reaction Delay
Object Type	ARRAY
Data Type	UNSIGNED16

Entry Description

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

Default Value	1

Sub-Index	1h
Description	AI1 Error Reaction Delay
Access	RW
PDO Mapping	No
Value Range	0 to 60,000
Default Value	1000 [ms]

3.4.15. 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 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 (x = 1 to 6)
Description	ECx Received PV
Access	RW
PDO Mapping	Yes
Value Range	Integer16
Default Value	No

3.4.16. 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	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	ECx Decimal Digits PV
Access	RW
PDO Mapping	No
Value Range	0 to 4
Default Value	1 (0.1 resolution)

3.4.17. 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 11. There is no physical unit associate with the data, but it uses the same resolution as the received PV as defined in object 2502h, EC Decimal Digits PV. This object must always be smaller than object 2522h EC Scaling 2 PV.

Object Description

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

Entry Description

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

Sub-Index	1h to 6h (x = 1 to 6)
Description	ECx Scaling 1 PV
Access	RW
PDO Mapping	No
Value Range	-32768 to 2522h sub-index X
Default Value	0

3.4.18. 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 11. 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	6
Default Value	6

Sub-Index	1h to 6h (x = 1 to 6)
Description	ECx Scaling 2 PV
Access	RW
PDO Mapping	No
Value Range	2520h sub-index X to 32767
Default Value	1000 (100.0)

3.4.19. Object 30z0h: LTz Input X-Axis Source

This object defines the type of input that will be used to determine the X-Axis input process value for the lookup table function. The available control sources on the 1IN-CAN controller are listed in Table 15. 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	30z0h (where z = 1 to 6)
Name	LTz Input X-Axis Source
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	See Table 15
Default Value	0 (control not used, PID disabled)

3.4.20. Object 30z1h: LTz 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 16. 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 17.

Object Description

Index	30z1h (where z = 1 to 6)
Name	LTz Input X-Axis Number
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	See Table 16
Default Value	0 (null control source)

3.4.21. Object 30z2h: LTz 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 17.

Object Description

Index	30z2h (where z = 1 to 6)
Name	LTz 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 17)
Default Value	0

3.4.22. Object 30z3h: LTz 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	30z3h (where z = 1 to 6)
Name	LTz Y-Axis Decimal Digits PV
Object Type	VARIABLE
Data Type	UNSIGNED8

Entry Description

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

3.4.23. Object 30z4h: LTz 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 24. See Figure 18 for an example of the difference between a step and ramp response.

Object Description

Index	30z4h (where z = 1 to 6)
Name	LTz 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 24 (0 or 1)
Default Value	0 (x-axis data response)

Sub-Index	2h to 11h (x = 2 to 11)
Description	LTz Point X Response
Access	RW
PDO Mapping	No
Value Range	See Table 24 (0, 1 or 2)
Default Value	1 (ramp to response)

3.4.24. Object 30z5h: LTz 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 30z4), this object is constrained such that X1 cannot be less than the Scaling 1 value of the selected control source/number, and X11 cannot be more than the Scaling 2 value. The rest of the points are constrained by the formula below. The physical unit associate with the data will be that of the selected input, and it will use the resolution defined in object 30z2h, LTz X-Axis Decimal Digits PV.

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

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	30z5h (where z = 1 to 6)
Name	LTz 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 11h (x = 1 to 11)	
Description	LTz Point X-Axis PVx	
Access	RW	
PDO Mapping	No	
Value Range	See above (data)	1 to 86400000 (time)
Default Value	10*(x-1)	No

3.4.25. Object 30z6h: LTz Point Y-Axis PV

This object defines the Y-Axis data for the 11 calibration points on the lookup table, resulting in 10 different output slopes. The data is unconstrained and has no physical unit associate with it. It will use the resolution defined in object 30z3h, LTz Y-Axis Decimal Digits PV.

Object Description

Index	30z6h (where z = 1 to 6)
Name	LTz Point Y-Axis PV
Object Type	ARRAY
Data Type	INTEGER16

Entry Description

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

Sub-Index	1h to 11h (x = 1 to 11)
Description	LTz Point Y-Axis PVx
Access	RW
PDO Mapping	No
Value Range	Integer16
Default Value	10*(x-1) [i.e. 0, 10, 20, 30, … 100]

3.4.26. Object 30z7h: LTz 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 30z3h, LTz Y-Axis Decimal Digits PV.

Object Description

Index	30z7h (where z = 1 to 6)
Name	LTz Output Y-Axis PV
Object Type	VARIABLE
Data Type	INTEGER16

Entry Description

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

3.4.27. Object 4000h: Logic Block Enable

This object defines whether or not the logic shown in Figure 22 will be evaluated.

Object Description

Index	4000h
Name	Logic Block Enable
Object Type	ARRAY
Data Type	BOOLEAN

Entry Description

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

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

3.4.28. Object 4010h: 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 22 has been performed.

Object Description

Index	4010h
Name	Logic Block Selected Table
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to 4h (x = 1 to 4)
Description	LBx Selected Table
Access	RO
PDO Mapping	Yes
Value Range	1 to 6
Default Value	No

3.4.29. Object 4020h: 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.

Object Description

Index	4020h
Name	Logic Block Output PV
Object Type	ARRAY
Data Type	UNSIGNED8

Entry Description

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

Sub-Index	1h to 4h (x = 1 to 4)
Description	LBx Output PV
Access	RO
PDO Mapping	Yes
Value Range	Dependent on Selected Table

Default Value	No

3.4.30. Object 4x01h: LBx Lookup Table Numbers

This object determines which of the six lookup tables supports on the 1IN-CAN 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	4x01h (where x = 1 to 4)
Name	LBx 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	LBx Lookup Table Y Number
Access	RW
PDO Mapping	No
Value Range	1 to 6
Default Value	See Table 30

3.4.31. Object 4x02h: LBx 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 28. See Section 1.8 for more information about how this object is used.

Object Description

Index	4x02h (where x = 1 to 4)
Name	LBx 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	LBx Function Y Logical Operator
Access	RW
PDO Mapping	No
Value Range	See Table 28
Default Value	Function A = 1 (and all) Function B = 1 (and all) Function C = 0 (default)

3.4.32.   Object 4x11h: LBx Function A Condition 1
3.4.33.   Object 4x12h: LBx Function A Condition 2
3.4.34.   Object 4x13h: LBx Function A Condition 3
3.4.35.   Object 4x21h: LBx Function B Condition 1
3.4.36.   Object 4x22h: LBx Function B Condition 2
3.4.37.   Object 4x23h: LBx Function B Condition 3
3.4.38.   Object 4x31h: LBx Function C Condition 1
3.4.39.   Object 4x32h: LBx Function C Condition 2
3.4.40.   Object 4x33h: LBx Function C Condition 3

These objects, 4xyzh, represent Logic Block z, Function y, Condition z, where x = 1 to 4, y = A to C, and z = 1 to 3. All of these objects are a special type of record, defined in Table 25. Information on how to use these objects is defined in Section 1.8.

Object Description

Index	4xyzh
Name	LBx 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 15
Default Value	1 (CANopen Message)

Sub-Index	2h
Description	Argument 1 Number

Access	RW
PDO Mapping	No
Value Range	See Table 16
Default Value	3 (EC Received PV 1)
Sub-Index	3h
Description	Argument 2 Source
Access	RW
PDO Mapping	No
Value Range	See Table 15
Default Value	3 (Constant PV)

Sub-Index	4h
Description	Argument 2 Number
Access	RW
PDO Mapping	No
Value Range	See Table 16
Default Value	3 (Constant FV 3)

Sub-Index	5h
Description	Operator
Access	RW
PDO Mapping	No
Value Range	See Table 26
Default Value	0 (Equals)

3.4.41. 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 four 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	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1h
Description	Constant False
Access	RO

PDO Mapping	No
Value Range	0
Default Value	0 (false)

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

Sub-Index	3h
Description	Constant FV 3
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	25.0

Sub-Index	4h
Description	Constant FV 4
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	50.0

Sub-Index	5h
Description	Constant FV 5
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	75.0

Sub-Index	6h
Description	Constant FV 6
Access	RW
PDO Mapping	No
Value Range	Float32
Default Value	100.0

3.4.42.   Object 5020h: Power Supply Field Value

This read-only object is available for diagnostic feedback purposes. It reflects the measured voltage powering the controller. The physical unit for this object is volts.

Object Description

Index	5020h
Name	Power Supply Field Value
Object Type	VARIABLE
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	0 to 70 [V]
Default Value	No

3.4.43.   Object 5030h: Processor Temperature Field Value

This read-only object is available for diagnostic feedback purposes. It reflects the measured temperature of the processor, which will always run approximately 10°C to 20°C above ambient. The physical unit for this object is degrees Celsius.

Object Description

Index	5030h
Name	Processor Temperature Field Value
Object Type	VARIABLE
Data Type	FLOAT32

Entry Description

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	-50 to 150 [°C]
Default Value	No

3.4.44. 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 1IN-CAN 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]

4. TECHNICAL SPECIFICATIONS

4.1. Power Supply

Power Supply Input	12, 24 VDC nominal (8…36VDC power supply range)
Protection	Reverse polarity protection is provided. Power supply input section protects against transient surges and shorts. Overvoltage protection up to 38V is provided. Overvoltage (undervoltage).

4.2. Inputs

Analog Input Functions	Voltage [V], Current [mA], PWM [%], Frequency [Hz], RPM, Counter
Voltage Input	0-5V (Impedance 204 KΩ) 0-10V (Impedance 136 KΩ)
Current Input	0-20mA (Impedance 124 Ω) 4-20mA (Impedance 124 Ω)
PWM Input	0 to 100% (at 0.5Hz to 20kHz) Selectable 10kΩ pullup to +5V or pulldown to GND resistor
Frequency Input	0.5Hz to 20kHz Selectable 10kΩ pullup to +5V or pulldown to GND resistor
Counter Input	Pulse Count, Measuring Window, Pulses in Window
Digital Input Function	5V CMOS, Active High or Active Low Selectable 10kΩ pullup to +5V or pulldown to GND resistor Normal, Inverse or Latched (push-button) response
Input Accuracy	<1% full scale error (all types)
Analog Input Resolution	12-bit ADC
Digital Input Resolution	16-bit timer
Error Detection/Reaction	Out of Range High and Low detection EMCY code generation (object 1003h) and fault reaction possible (1029h).

4.3. Communication

CAN	1 CAN 2.0B port, protocol CiA CANopen ® By default, the 1IN-CAN Controller transmits measured input (FV object 7100h) and output current feedback (FV object 2370h) on TPDO1
Network Termination	According to the CAN standard, it is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.

4.4. General Specifications

Microprocessor	STM32F103CBT7, 32-bit, 128 Kbytes Flash Program Memory
Quiescent Current	Contact Axiomatic.
Control Logic	User programmable functionality using Electronic Assistant®
Communications	1 CAN port (CANopen®), SAE J1939 is available on request.
Operating Conditions	-40 to 85 °C (-40 to 185 °F)
Protection	IP67
EMC Compliance	CE marking
Vibration	MIL-STD-202G, Test 204D and 214A (Sine and Random) 10 g peak (Sine); 7.86 Grms peak (Random) (Pending)
Shock	MIL-STD-202G, Test 213B, 50 g (Pending)
Approvals	CE marking
Electrical Connections	6 pin Deutsch IPD connector P/N: DT04-6P A mating plug kit is available as Axiomatic P/N: **AX070119**. | Pin #	Description
1	BATT+
2	Input +
3	CAN_H
4	CAN_L
5	Input -
6	BATT-	

5. VERSION HISTORY

Version	Date	Author	Modifications
1	May 31st, 2016	Gustavo Del Valle	Initial Draft

OUR PRODUCTS

Actuator Controls

Battery Chargers

CAN bus Controls, Gateways

CAN/Wifi, CAN/Bluetooth

Current Converters

DC/DC Power Converters

DC Voltage/Current Signal
Converters

Engine Temperature Scanners

Ethernet/CAN Converters

Fan Drive Controllers

Hydraulic Valve Controllers

I/O Controls

LVDT Simulators

Machine Controls

Motor Controls

PID Controls

Position Sensors, Angle
Measurement Inclinometers

Power Supplies

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

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

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

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

QUALITY DESIGN AND MANUFACTURING

Axiomatic is an ISO 9001:2008 registered facility.

SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#).

Please provide the following information when requesting an RMA number:
- Serial number, part number
- Axiomatic invoice number and date
- Hours of operation, description of problem
- Wiring set up diagram, application
- Other comments as needed

When preparing the return shipping paperwork, please note the following. The commercial invoice for customs (and packing slip) should state the harmonized international HS (tariff code), valuation and return goods terminology, as shown in italics below. The value of the units on the commercial invoice should be identical to their purchase price.

Goods Made In Canada (or Finland)
Returned Goods for Warranty Evaluation, HS: 9813.00
Valuation Identical Goods
Axiomatic RMA#

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

CONTACTS

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

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