USER MANUAL UMAX031501

# MAGNETIC PICK-UP INPUT CONTROLLER

With CANopen®

## USER MANUAL

**P/N: AX031501**

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

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

## VERSION HISTORY

| Version | Date | Author | Modification |
|---|---|---|---|
| 1.0.0. | May 4, 2017 | Ilona Korpelainen | Initial Draft |
| - | June 7, 2017 | Amanda Wilkins | Marketing Review |
| 1.00A | Jan 30, 2020 | Ilona Korpelainen | Added 48V nominal to the power supply input technical specification |
| 1.00B | August 1, 2023 | Kiril Mojsov | Legacy Updates |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

**ACCRONYMS**

| | |
|---|---|
| BATT +/- | Battery positive (a.k.a. Vps) or Battery Negative (a.k.a. GND) |
| CAN | Control Area Network |
| CANopen® | CANopen® is a registered community trademark of CAN in Automation e.V. |
| CAN-ID | CAN 11-bit Identifier |
| COB | Communication Object |
| CTRL | Control |
| EDS | Electronic Data Sheet |
| EMCY | Emergency |
| GND | Ground reference (a.k.a. BATT-) |
| LSB | Least Significant Byte (or Bit) |
| LSS | Layer Settling Service |
| LUT | Lookup Table |
| MI | Magnetic Input |
| MSB | Most Significant Byte (or Bit) |
| NMT | Network Management |
| RO | Read Only Object |
| RPDO | Received Data Object |
| RPM | Rotations per Minute |
| RW | Read/Write Object |
| SDO | Service Data Object |
| TPDO | Transmitted Process Data Object |
| Vps | Voltage Power Supply (a.k.a. BATT+) |
| WO | Write Only Object |

## TABLE OF CONTENTS

## 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, at: http:/www.can-cia.org/

# 1. OVERVIEW OF CONTROLLER

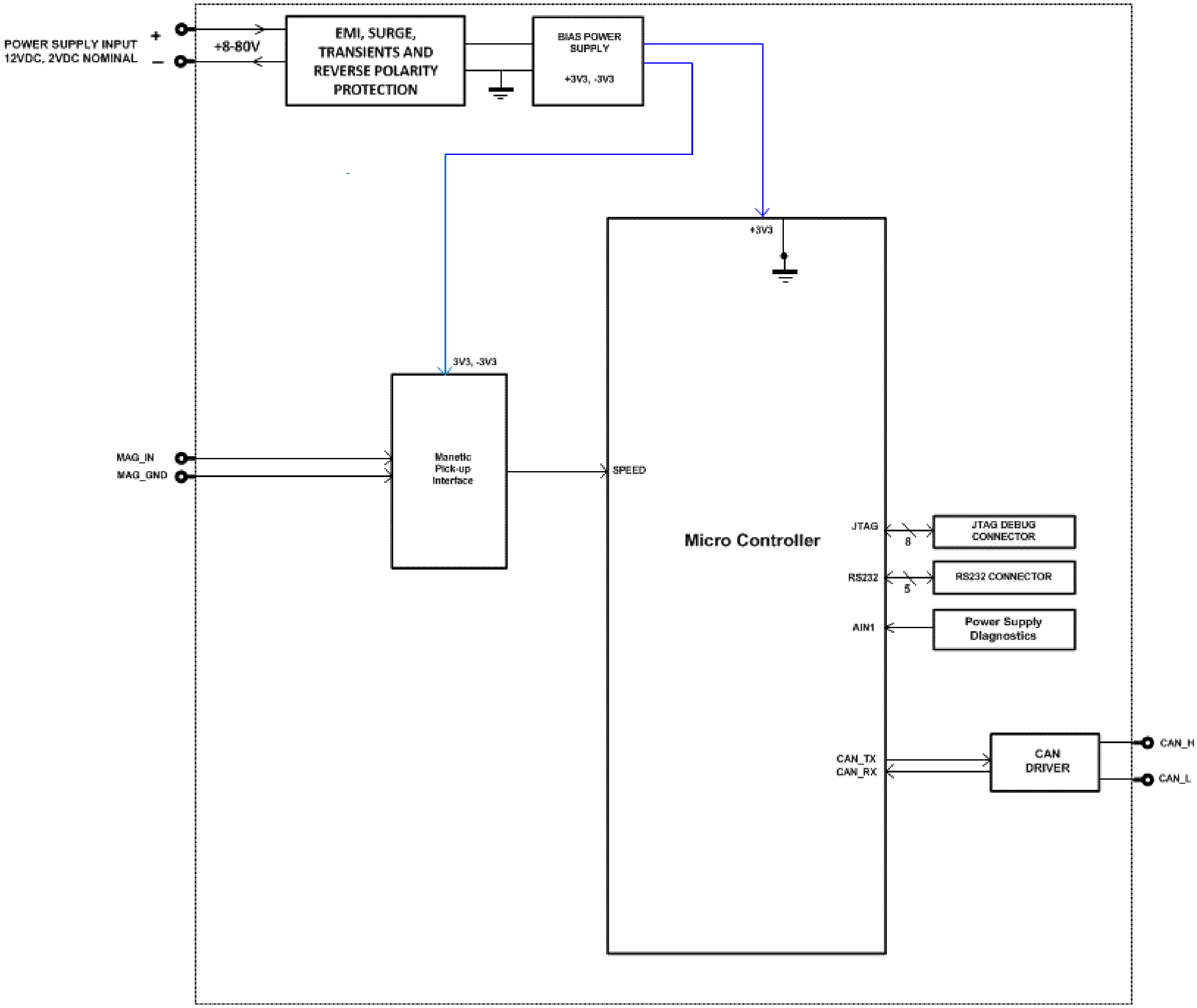



**Figure 1– Hardware Functional Block Diagram**

The Magnetic Pick-up Input CANopen® controller (MAG-IN- CO) is designed for measuring a single frequency input and providing the measurement result to CANopen® network. The controller is widely configurable through CANopen® Objects. With its sophisticated control algorithms, Input measurement can be adjusted to serve wide variety of custom applications.

The Magnetic input picks up frequencies from 0.5Hz to 50kHz or 30 to 3000000RPM. In addition to basic frequency measurement the MAG-IN-CO can be configured to measure the number of pulses received in defined time window or time elapsed to receive defined number of pulses.

Logical (software) function blocks are depicted in Figure 2. To produce desired control 6 Lookup Tables, 2 Programmable Control Logics and 4 Mathematical Function Blocks are provided. All functional blocks are independent from one another, but can be programmed to interact in numerous ways. The function blocks are outlined in following sections. All objects are user-configurable using standard commercially available tools that can interact with a CANopen® Object Dictionary via an .EDS file.

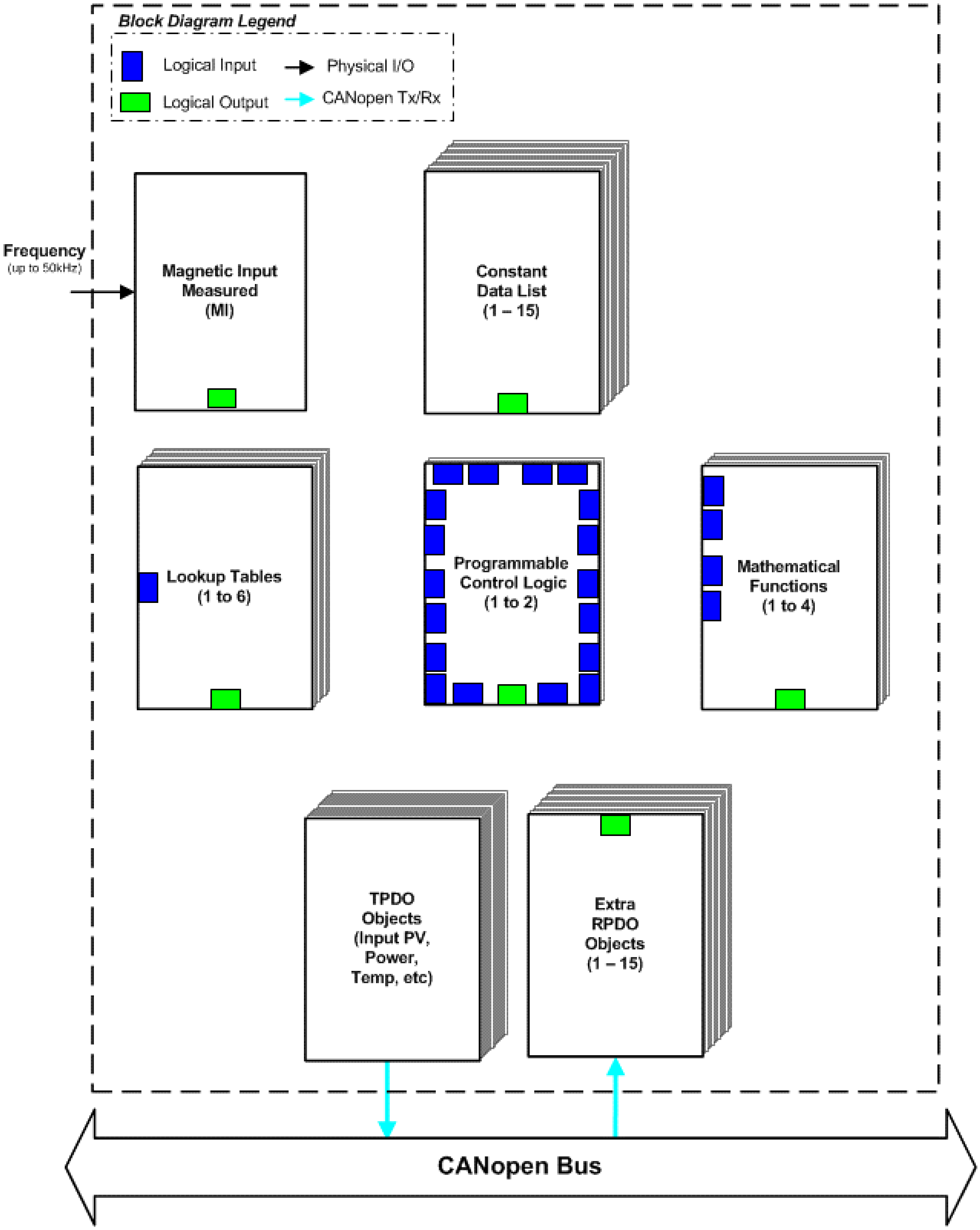



**Figure 2 – Logic Functional Block Diagram**

## 1.1. Input Function Blocks

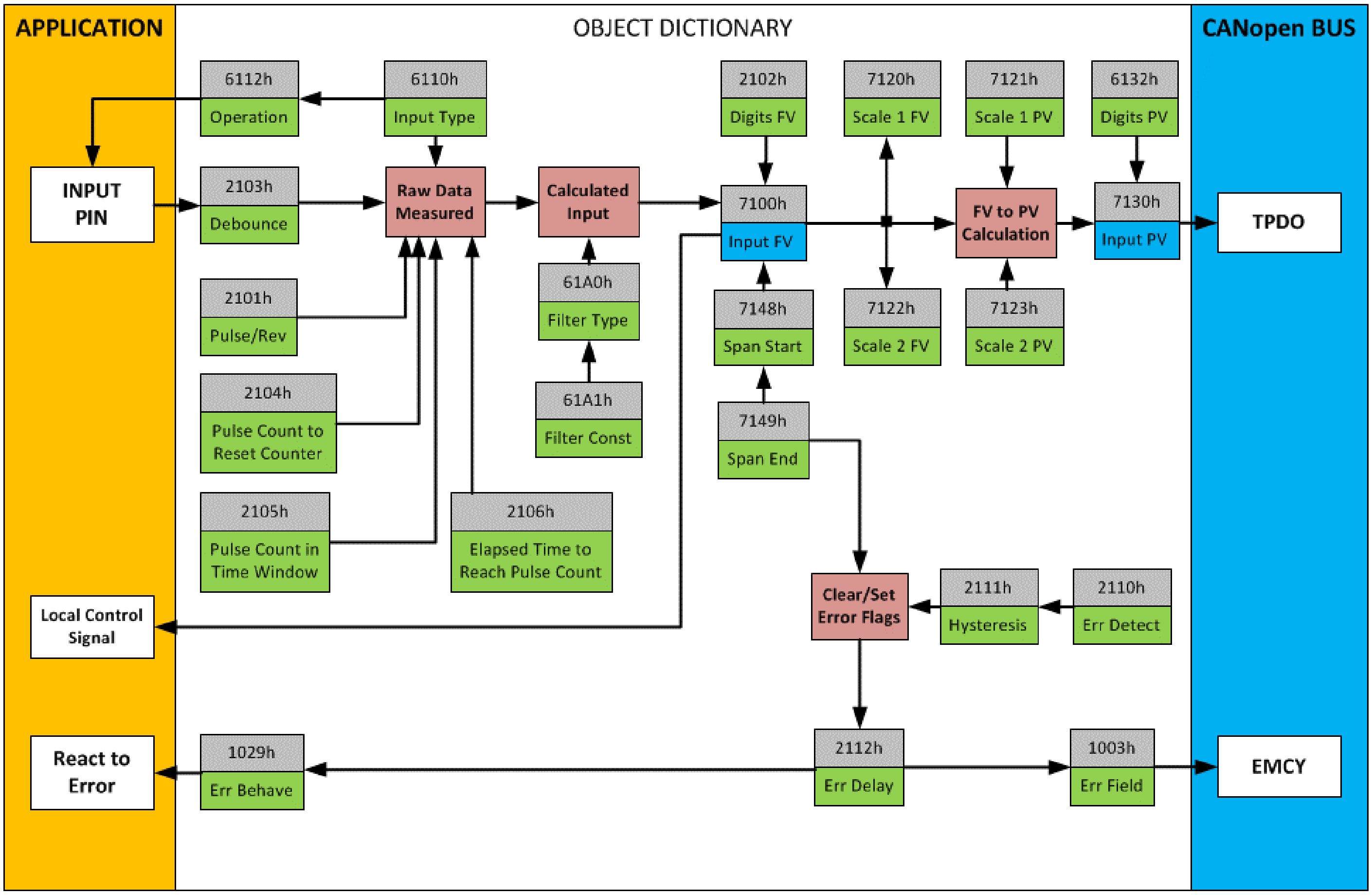



**Figure 3 – Magnetic Input Objects**

Object 6112h, **AI Operating Mode** determines whether the MI function block is associated with an input. The options for object 6112h are shown in Table 1. No values other than what are shown here will be accepted.

| Value | Meaning |
|---|---|
| 0 | Channel Off |
| **1** | **Normal Operation** |

**Table 1 – Object 6112h - AI Operating Mode Options**

The most important object associate with the MI function block is object 6110h **AI Sensor Type**. By changing this value, other objects will be automatically updated by the controller. The options for object 6110h are shown in Table 2, and no values other than what are shown here will be accepted.

| Value | Meaning |
|---|---|
| 60 | Frequency Input (or RPM) |
| 10000 | Counter (Pulse Count Reset) |
| 10001 | Counter (Time Window) |
| 10002 | Counter (Pulse Window) |

**Table 2 – Object 6110h - AI Sensor Type Options**

By default, the input is setup to measure frequency in Hz. Frequency measurement can be changed to RPM, by setting object 2101h **AI Number of Pulses per Revolution** to a non-zero value. With other input types this object will be ignored.

There are three different types of a Counter input mode: Counter (Pulse Count Reset), Counter (Time Window), Counter (Pulse Window).

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

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

Object 2103h **AI Debounce Filter** is applied to the input before the state is read by the processor. The options for object 2103h are shown in Table 3, with the default bolded.

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

**Table 3 – Object 2103h Debounce Filter Options**

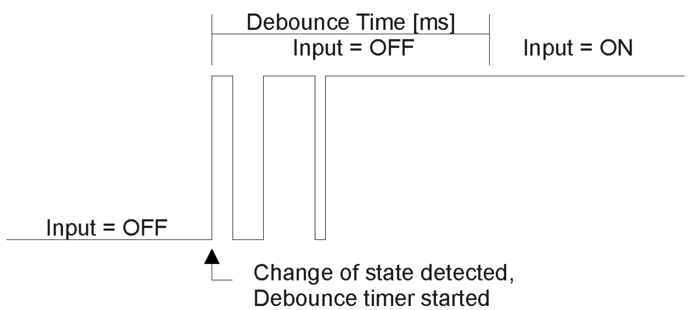



**Figure 4 – Input Debouncing**

Regardless of type, all analog inputs can be further filtered once the raw data has been measured. Object 61A0h **AI Filter Type** determines what kind of filter is used per Table 4. By default, additional software filtering is disabled.

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

**Table 4 – Object 61A0h - AI Filter Type Options**

Measured input data from magnetic input can be filtered to form desired CAN message data. Input filters are configured with "**Filter Type**" and "**Filter Constant**" setpoints.

"**Filter Type**" setpoint defines the type of software filter used. Setpoint options are '*No Filtering'*, '*Moving Average'* and '*Repeating Average'*. The '*No Filtering'* option applies no filtering to the measured input data. The '*Moving Average* option applies the transfer function below to the measured input data, where $Value_N$ is the current value of the CAN message data, $Value_{N-1}$ is the previous CAN message data and Filter Constant is the value of the "**Filter Constant setpoint**".

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

<u>Calculation with no filter:</u>

Value = Input

The data is simply a 'snapshot' of the latest value measured by the ADC or timer.

<u>Equation 1 - Moving Average Transfer Function:</u>

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

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

<u>Equation 2 - Repeating Average Transfer Function:</u>

$$Value = \frac{\sum_0^N Input_N}{N}$$

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 Pulses per Revolution selected, and will be automatically updated per Table 6 when either 6110h or 2101h 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 Pulses per Rev are changed.

| **Sensor Type** | **Decimal Digits** |
|---|---|
| Frequency: | 0 [1 Hz] |
| Frequency: RPM Mode | 0 [1 RPM] |
| Counter (Pulse Count Reset) | 0 [Pulses] |
| Counter (Time Window) | 0 [Pulses] |
| Counter (Pulse Window) | 3 [ms] |

**Table 5 – Object 2102h - 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 1.2.

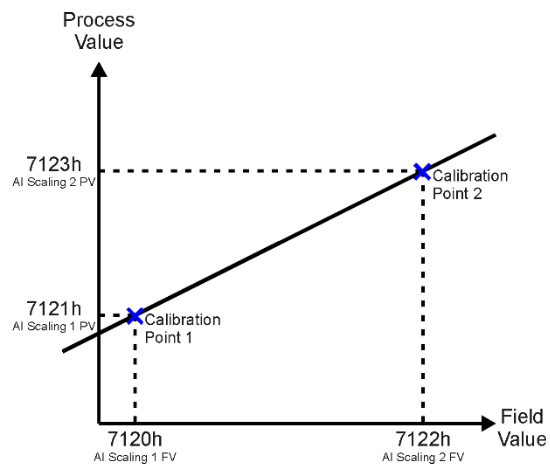



**Figure 5 – Analog Input Linear Scaling FV to PV**

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 6 lists the default values that are loaded into objects 7120h, 7122h, 7148h, 7149h, and 2111h for each Sensor Type and Pulses per Rev combination. Recall that these objects all have the decimal digits applied to them as outlined in Table 7.

| Sensor Type/<br>Input Range | 7148h<br>AI Span Start<br>(i.e. Error Min) | 7120h<br>AI Scaling 1 FV<br>(i.e. Input Min) | 7122h<br>AI Scaling 2 FV<br>(i.e. Input Max) | 7149h<br>AI Span End<br>(i.e. Error Max) | 2111h<br>Error Clear<br>Hysteresis |
|---|---|---|---|---|---|
| Frequency | 0.1 [Hz] | 1 [Hz] | 5000 [Hz] | 6000 [Hz] | 50 [Hz] |

| | | | | | |
|---|---|---|---|---|---|
| Freq: RPM Mode | 500[1RPM] | 100 [1RPM] | 3000 [1RPM] | 33000[1RPM] | 10[1RPM] |
| Counter<br>(Pulse Count Reset) | 0 | 0 | 30000 | 30000 | 0 |
| Counter (Time Window) | 0 | 0 | 30000 | 30000 | 0 |
| Counter (Pulse Window) | 0 | 0 | 30 | 30 | 0 |

**Table 6 – AI Object Defaults Based on Sensor Type and Input Range**

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

When changing these objects, Table 7 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. 50000Hz 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 7 uses the base unit of the input, but recall the limits will also have object 2102h apply to them as per Table 5.

| Sensor Type/<br>Pulses per Rev | 7148h | 7120h | 7122h | 7149h | 2111h |
|---|---|---|---|---|---|
| Frequency | 0.0Hz to<br>7120h | 7148h to 7122h<br>If(7148h<0.5Hz) 0.5Hz to 7122h | 7120h to 7149h<br>If(7149h>MAX)<br>7120h to MAX | 7122h to<br>110% of<br>MAX | 10% of<br>MAX |
| RPM | 0 to 7120h | 7148h to 7122h | | | |
| Counter | | | | | |

**Table 7 – AI Object Ranges Based on Sensor Type and Input Range**

The last objects associated with the magnetic 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.

## 1.2. Lookup Table Function Block

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

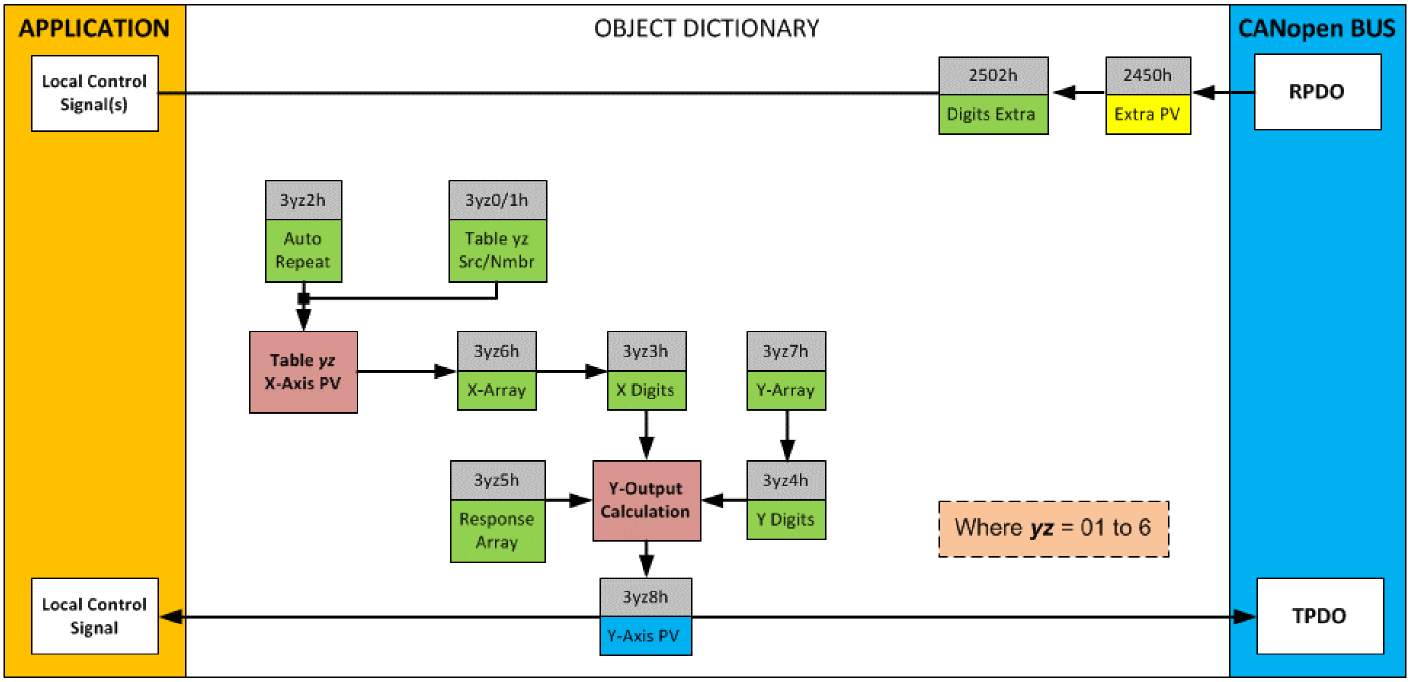



**Figure 6 – Lookup Table Objects**

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

Note: If more than 10 slopes are required, a Programmable Logic Block can be used to combine up to three tables to get 30 slopes, as is described in Section 1.3.

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

There are two (or three) other key parameters that will affect how this function block will behave depending on the "**X-Axis Type**" chosen. If chosen *'Data Response'*, then the objects 3yz0h **Lookup Table yz Input X-Axis Source** and 3yz1h **Lookup Table yz Input X-Axis Number** together define the control source for the function block. When it is changed, the table values in object 3yz6h need to be updated with new defaults based on the X-Axis source selected as described in Tables 15 and 16. If however, the "**X-Axis Type**" is chosen to be *'Time Response'*, an additional parameter is taken into consideration - object 3yz2h, **Lookup Table yz Auto Repeat**. These will be described in more detail in Section 1.2.4.

### 1.2.1. 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.

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

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

As stated earlier, MinInputRange and MaxInputRange will be determined by the scaling objects associated with X-Axis Source that has been selected, as outlined in Table 18.

### 1.2.2. 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 18.

However, the Y-Axis has no constraints on the data that it represents. This means that inverse, or increasing/decreasing or other responses can be easily established. The Y-Axis does not have to be a percentage output, but could represent full scale process values instead.

In all cases, the controller looks at the entire range of the data in the Y-Axis sub-indexes, and selects the lowest value as the MinOutRange and the highest value as the MaxOutRange. So long as they are not both within the 0 to 100 range, they are passed directly to other function blocks as the limits on the lookup table output. (i.e. Scaling 1 and Scaling 2 values in linear calculations.)

Even if some of the data points are *‘Ignored’* as described in Section1.2.3, 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.

### 1.2.3. Point to Point Response

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

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

Figure 7 shows the difference between these two response profiles with the default settings.

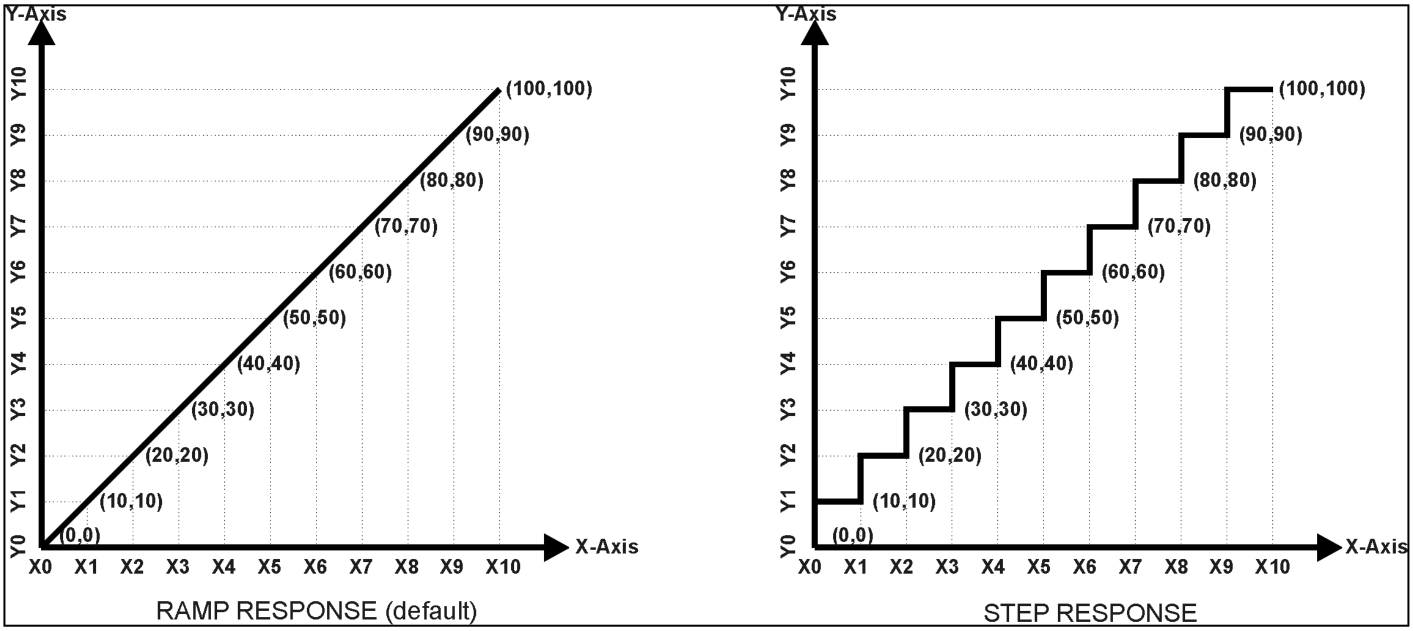



**Figure 7 – Lookup Table Defaults with Ramp and Step Responses**

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 8. 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.')*

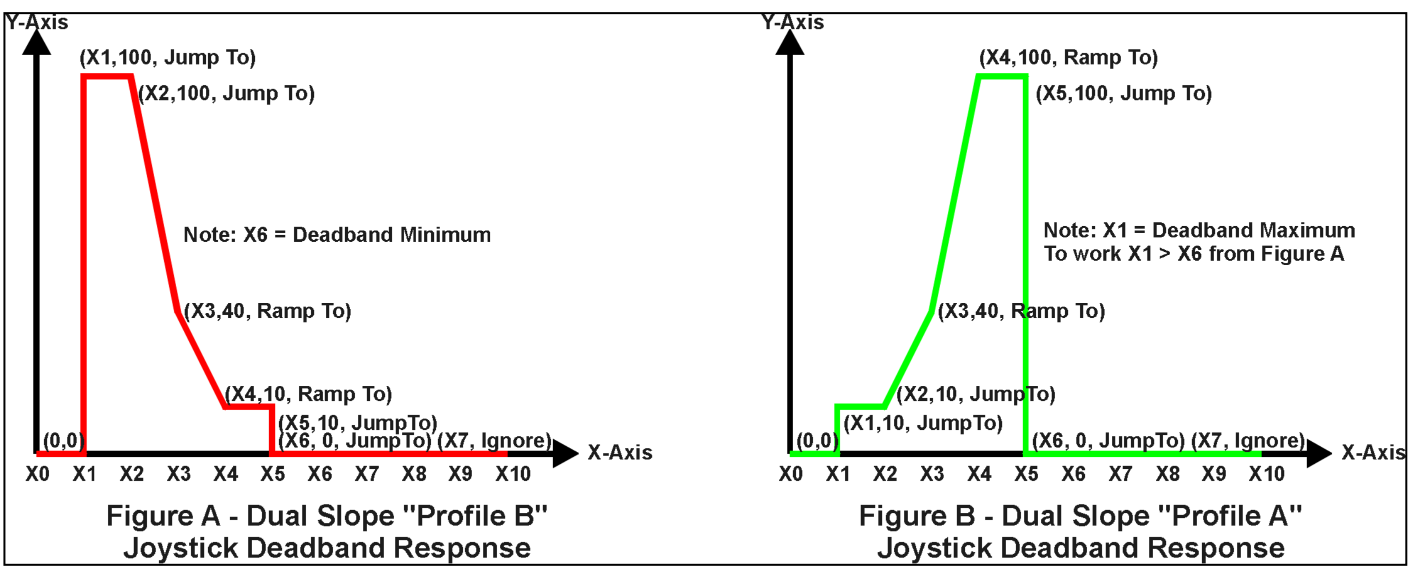



**Figure 8 – Lookup Table Examples to Setup for Dual-Slope Joystick Deadband Response**

To summarize, Table 8 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 8 – LTyz Point Response Options**

### 1.2.4. X-Axis, Time Response

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

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

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

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

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

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

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

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

In a time response, the data in object 30z6h **LTyz Point X-Axis PV** is measured in milliseconds, and object 3yz3h **LTyz X-Axis Decimal Digits PV** is automatically set to 0. A minimum value of 1ms must be selected for all points other than sub-index 1 which is automatically set to [0,0]. The interval time between each point on the X-axis can be set anywhere from 1ms to 24 hours. [86,400,000 ms]

## 1.3. Programmable Logic Function Block

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

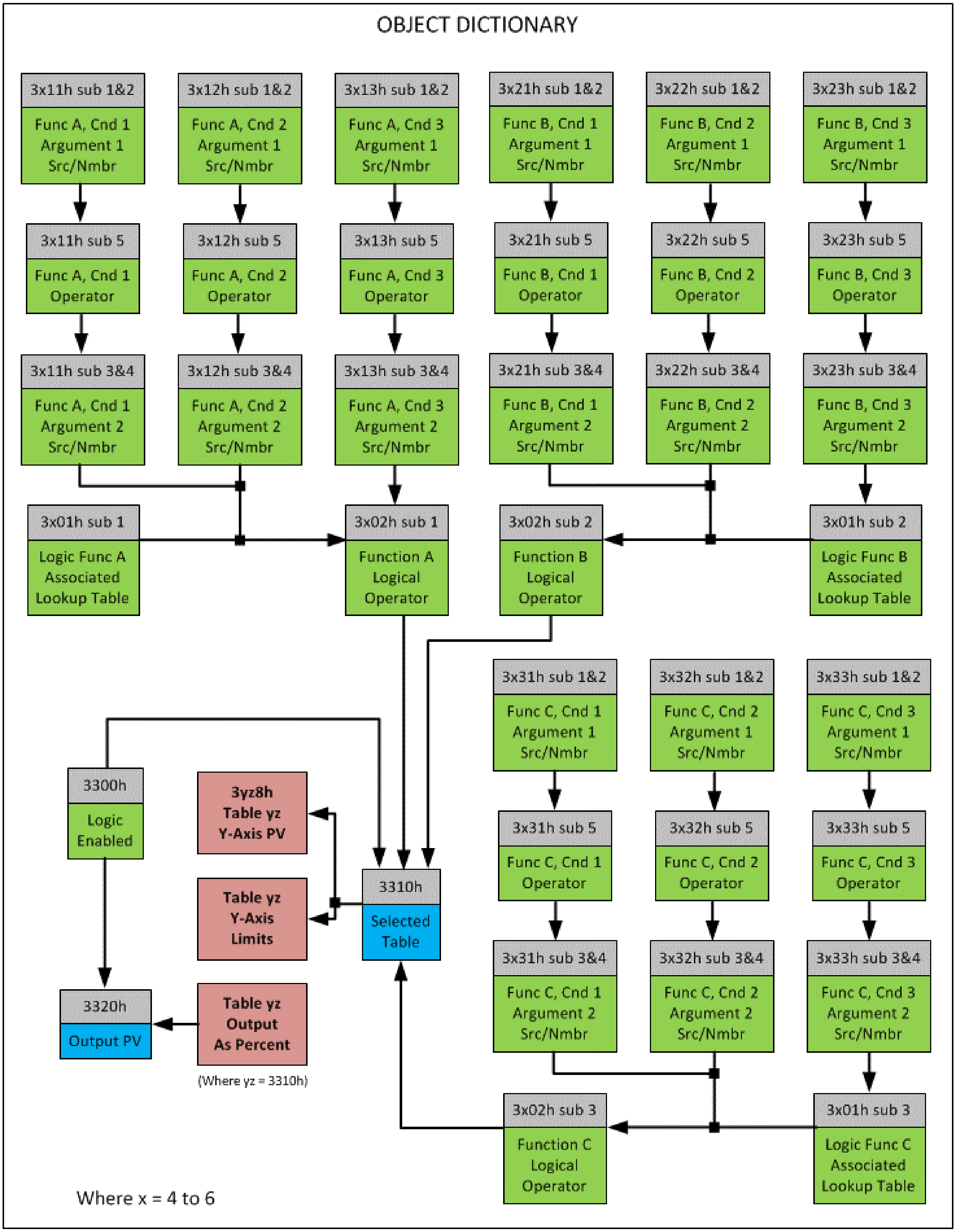



**Figure 9 – Logic Block Objects**

This function block is obviously the most complicated of them all, but very powerful. Any LBx (where X= 4 to 6) can be linked with up to three lookup tables, any one of which would be selected only under given conditions. Any three tables (of the available 6) can be associated with the logic, and which ones are used is fully configurable on object 3x01 LB(3-x) Lookup Table Number.

Should the conditions be such that a particular table (A, B or C) has been selected as described in Section 1.3.2, then the output from the selected table, at any given time, will be passed directly to LB(3-x)'s corresponding sub-index X in read-only mappable object 3320h Logic Block Output PV. The active table number can read from read-only object 3310h Logic Block Selected Table.

Note: In this document, the term LB(3-x) refers to Logic Blocks 1 to 2. Due to the CANopen® Object indices, Logic Block 1 begins at 3401h where x, in this case, is 4.

Therefore, an LBx allows up to three different responses to the same input, or three different responses to different inputs, to become the control for another function block. Here, the "Control Source" for the reactive block would be selected to be the 'Programmable Logic Function Block,' as described in Section 1.3.2.

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

Logic is evaluated in the order shown in Figure 10. 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.**

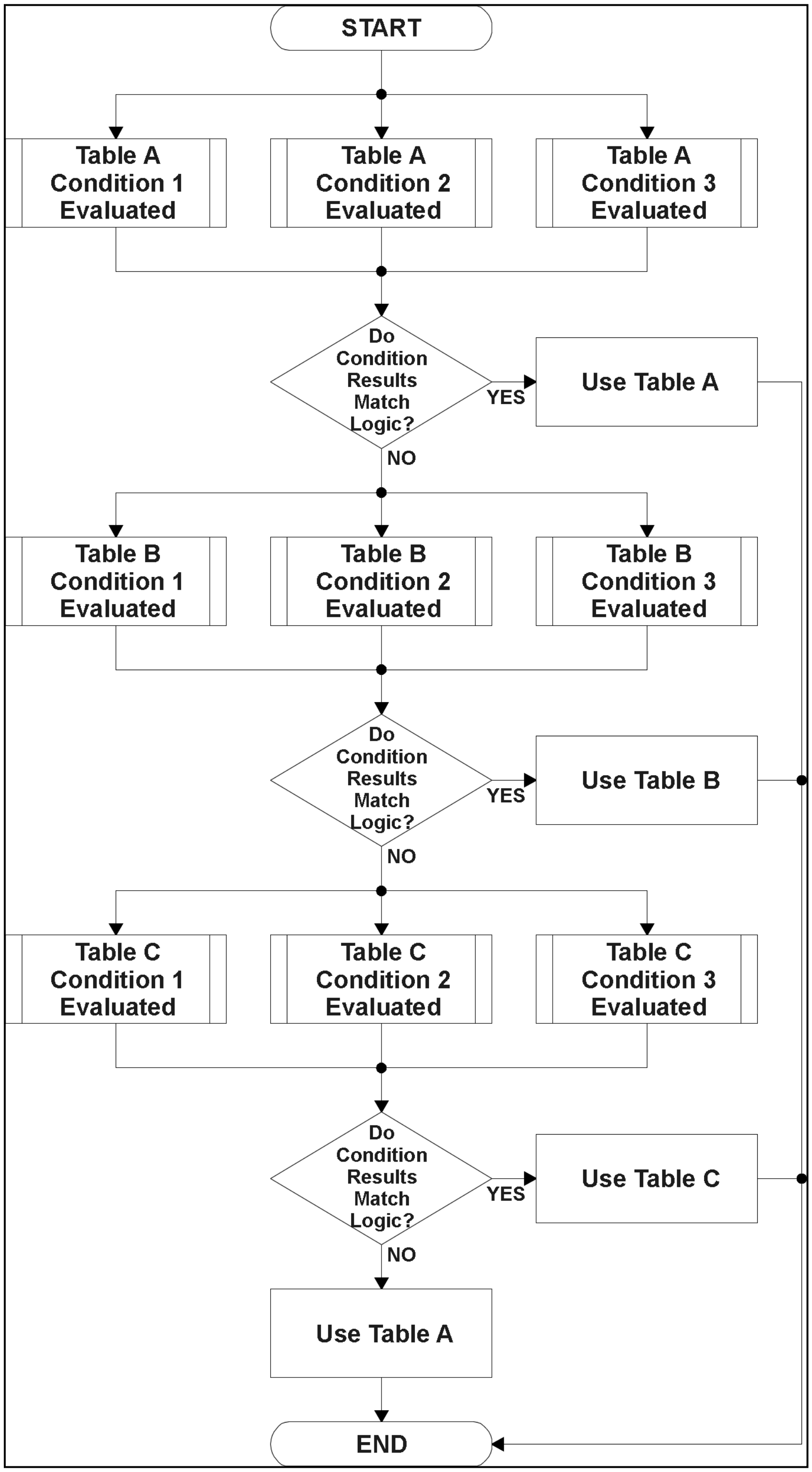



**Figure 10 – Logic Block Flowchart**

### 1.3.1. 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 8.

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

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

**Table 9 – LB(3-x) Condition Structure Definition**

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

Argument 1 is always a logical output from another function block, as listed in Table 11. As always, the input is a combination of the functional block objects 3xyzh sub-index 1 **"Argument 1 Source"** and **"Argument 1 Number."**

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

Argument 1 is evaluated against Argument 2 based on the **"Operator"** selected in sub-index 5 of the condition object. The options for the operator are listed in Table 10, 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 10 – LB(3-x) Condition Operator Options**

By default, both arguments are set to *'Control Source Not Used'* which disables the condition, and automatically results in a value of N/A 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 11.

| 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 11 – LB(3-x) Condition Evaluation Results**

### 1.3.2. 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 1.3.1. There are several logical combinations that can be selected, as listed in Table 12. The default value for object 3x02h **LB(3-x) Function Logical Operator** is dependent on the sub-index. For sub-index 1 (Table A) and 2 (Table B), the *'Cnd1 And Cnd2 And Cnd3'* operator is used, whereas sub-index 3 (Table C) is setup as the *'Default Table''* response.

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

**Table 12 – LB(3-x) Function Logical Operator Options**

Not every evaluation is going to need all three conditions. The case given in the earlier section, for example, only has one condition listed, i.e. that the Engine RPM be below a certain value. Therefore, it is important to understand how the logical operators would evaluate an Error or N/A result for a condition, as outlined in Table 13.

| 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.**<br><br>If any condition equals False or Error, the table is not selected.<br>An N/A is treated like a True.<br>If all three conditions are True (or N/A), the table is selected.<br><br>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.**<br><br>If any condition is evaluated as True, the table is selected.<br>Error or N/A results are treated as False<br><br>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.**<br><br>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<br><br>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<br><br>If( ((Cnd1==True)\|\|(Cnd2==True)) && (Cnd3==True) ) Then Use Table |

**Table 13 – LB(3-x) Conditions Evaluation Based on Selected Logical Operator**

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

### 1.3.3. Logic Block Output

Recall that Table Y, where Y = A, B or C in the LB(3-x) function block does NOT mean lookup table 1 to 3. Each table has object 3x01h LB(3-x) **Lookup Table Number** which allows the user to select which lookup tables they want associated with a particular logic block. The default tables associated with each logic block are listed in Table 14.

| 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 |

**Table 14 – LB(3-x) Default Lookup Tables**

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

The LB(3-x) output is always setup as a percentage, based on the range of the Y-Axis for the associated table (see Section 1.2.2) It is written to sub-index X of read-only object 3320h **Logic Block Output PV** with a resolution of 1 decimal place.

## 1.4. Math Function Block

There are four mathematic function blocks that allow the user to define basic algorithms. Math function block Z = 1 to 4 will be enabled based on sub-index Z in object 4000h **Math Enable**.

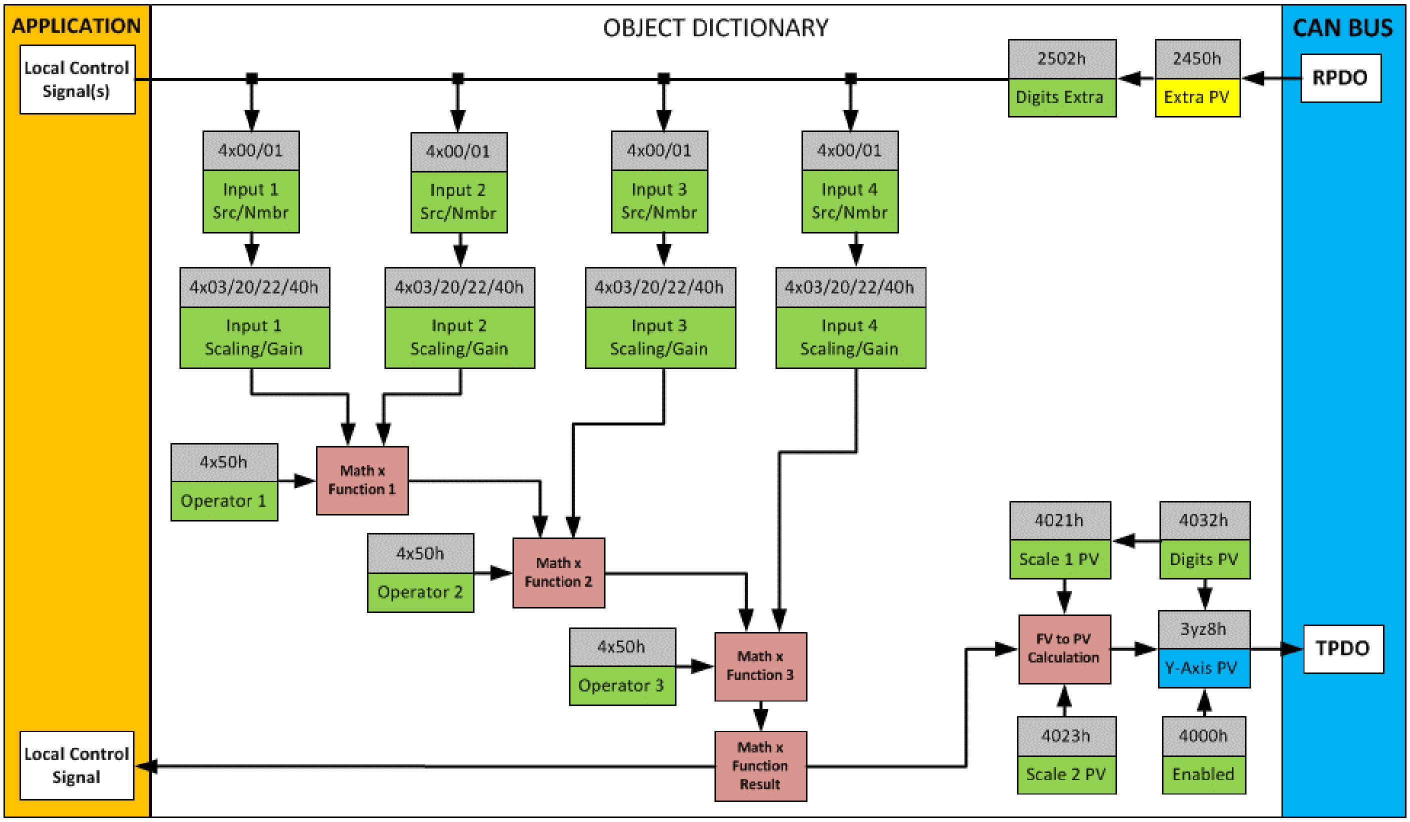



**Figure 11 – Math Function Block Objects**

A math function block can take up to four input signals, as listed in Table 16 in Section 1.6. Each input is then scaled according the associated scaling and gain objects. A “Math Input X” is determined by the corresponding sub-index X = 1 to 4 of the objects 4y00h **Math Y Input Source** and 4y01h **Math Y Input Number.** Here, y = 1 to 4; corresponding the Math 1- Math 4.

Inputs are converted into a percentage value based on objects 4y20h **Math Y Scaling 1 FV** and 4y22h **Math Y Scaling 2 FV**. Before being used in the calculation, these objects apply the resolution shift defined by object 4y02h **Math Y Decimal Digits FV**. As with any other function block using a control source for the X-Axis in a conversion, the scaling objects should be selected to match the values in the control’s corresponding objects as per Table 18.

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

For each input pair, the appropriate arithmetic or logical operation is performed on the two inputs, InA and InB, according the associated function in sub-index of InB in object 4y50h **Math Y Operator**. The list of selectable function operations is defined in Table 15.

| | | |
|---|---|---|
| 0 | = | True when InA Equals InB |
| 1 | != | True when InA Not Equal InB |
| 2 | > | True when InA Greater Than InB |
| 3 | >= | True when InA Greater Than or Equal InB |
| 4 | < | True when InA Less Than InB |
| 5 | <= | True when InA Less Than or Equal InB |
| 6 | OR | True when InA or InB is True |

| | | |
|---|---|---|
| 7 | AND | True when InA and InB are True |
| 8 | XOR | True when InA/InB is True, but not both |
| 9 | + | Result = InA plus InB |
| 10 | - | Result = InA minus InB |
| 11 | x | Result = InA times InB |
| 12 | / | Result = InA divided by InB |
| 13 | MIN | Result = Smallest of InA and InB |
| 14 | MAX | Result = Largest of InA and InB |

**Table 15 – Object 4y50h Math Function Operators**

For Function 1, InA and InB are Math Inputs 1 and 2, respectively.
For Function 2, InA is the result of Function 2 and InB is Math Input 3, respectively.
For Function 3, InA is the result of Function 3 and InB is Math Input 4, respectively.

For a valid result in each Function, both inputs must be non-zero value (other than *'Control Source Not Used'*). Otherwise, the corresponding Function is ignored, and the "Output Data" for the math function block is the result of the earlier function scaled according to the output objects. For example, if Math Input 4 is not used, the math output would be the result of the Function 3 operation.

For logical operators (6 to 8), any SCALED input greater than or equal to 0.5 is treated as a TRUE input. For logic output operators (0 to 8), the result of the calculation for the function will always be 0 (FALSE) or 1 (TRUE).

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

For the arithmetic functions (9 to 14), it is recommended to scale the data such that the resulting operation will not exceed full scale (0 to 100%) and saturate the output result.

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

The resulting final mathematical output calculation is in the appropriate physical units using object 4021h **Math Output Scaling 1 PV** and 4023h **Math Output Scaling 2 PV**. These objects are also considered the Min and Max values of the Math Block output and apply the resolution shift defined by object 4032h **Math Output Decimal Digits PV**. The result is written to read-only object 4030h **Math Output PV**. These scaling objects should also be taken into account when the Math Function is selected as the input source for another function block, as outlined in Table 18.

## 1.5. 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.

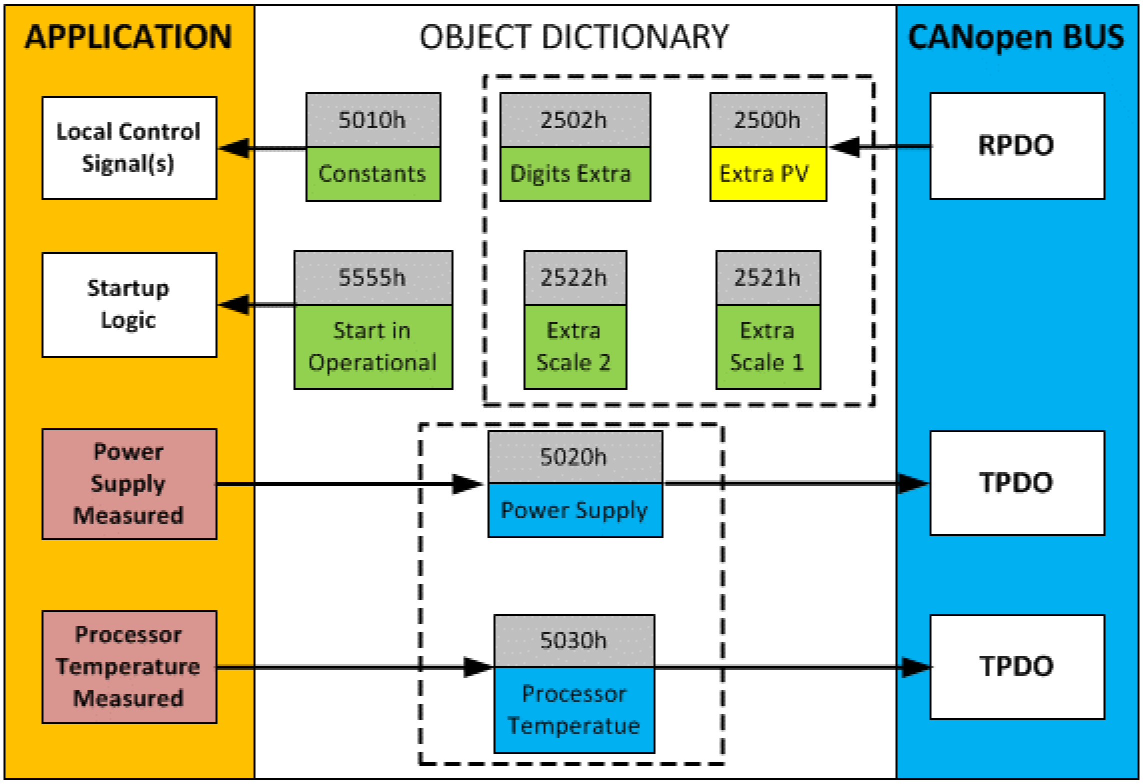



**Figure 12 – Miscellaneous Objects**

<u>Extra RPDO Messages</u>

Objects 2500h **Extra Control Received PV**, 2502h **EC Decimal Digits PV**, 2502h **EC Scaling 1 PV** and **EC Scaling 2 PV** are mentioned in Table 17. These objects allow for additional data received on a CANopen® RPDO to be mapped independently to various function blocks as a control source. The scaling objects are provided to define the limits of the data when it is used by another function block, as shown in Table 18.

<u>Constant Values</u>

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

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

The False/True constants are provided primarily to be used with the logic block. The variable constants are also useful with the logic or math blocks.

<u>Fault Detection Objects</u>

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

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

<u>Automatic Update of Objects</u>

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

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

<u>Startup</u>

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

## 1.6. Internal Function Block Control Sources

The Magnetic Pick-up Input controller allows for internal function block sources to be selected from the list of the logical function blocks supported by the controller. As a result, any output from one function block can be selected as the control source for another. Keep in mind that not all options make sense in all cases, but the complete list of control sources is shown in Table 16.

| Value | Meaning |
|---|---|
| 0 | Control Source Not Used (Ignored) |
| 1 | CANopen® Message (RPDO) |
| 2 | Input Function Block |
| 3 | Lookup Table Function Block |
| 4 | Programmable Logic Function Block |
| 5 | Mathematical Function Block |
| 6 | Constant Function Block |
| 7 | Power Supply Measured |
| 8 | Processor Temperature Measured |

**Table 16 – Control Source Options**

In addition to a source, each control also has a number which corresponds to the sub-index of the function block in question. Table 17 outlines the ranges supported for the number objects, depending on the source that had been selected.

| Control Source | Range | Object (Meaning) |
|---|---|---|
| **Control Source Not Used** | 0 | Ignored |
| **CANopen® Message (RPDO)** | 1 | 2500h sub-index 1 (Extra Received PV 1) |
| | 2 | 2500h sub-index 1 (Extra Received PV 2) |
| | 3 | 2500h sub-index 2 (Extra Received PV 3) |
| | 4 | 2500h sub-index 3 (Extra Received PV 4) |
| | 5 | 2500h sub-index 4 (Extra Received PV 5) |
| | 6 | 2500h sub-index 5 (Extra Received PV 6 ) |
| | 7 | 2500h sub-index 6 (Extra Received PV 7) |
| | 8 | 2500h sub-index 7 (Extra Received PV 8) |
| | 9 | 2500h sub-index 8 (Extra Received PV 9 ) |
| | 10 | 2500h sub-index 9 (Extra Received PV 10) |
| | 11 | 2500h sub-index 10 (Extra Received PV 11) |
| | 12 | 2500h sub-index 11 (Extra Received PV 12) |
| **Input Function Block** | 1 | 7100h sub-index 1 or 6000h sub-index 1 bit 0 |
| **Constant Function Block** | 1 | 5010h sub-index 1 (always FALSE) |
| | 2 | 5010h sub-index 2 (always TRUE) |
| | 3 | 5010h sub-index 3 (Constant FV 3) |
| | 4 | 5010h sub-index 4 (Constant FV 4) |
| | 5 | 5010h sub-index 5 (Constant FV 5) |
| | 6 | 5010h sub-index 6 (Constant FV 6) |
| | 7 | 5010h sub-index 7 (Constant FV 7) |
| | 8 | 5010h sub-index 8 (Constant FV 8) |
| | 9 | 5010h sub-index 9 (Constant FV 9) |
| | 10 | 5010h sub-index 10 (Constant FV 10) |
| | 11 | 5010h sub-index 11 (Constant FV 11) |
| | 12 | 5010h sub-index 12 (Constant FV 12) |
| | 13 | 5010h sub-index 13 (Constant FV 13) |
| | 14 | 5010h sub-index 14 (Constant FV 14) |
| | 15 | 5010h sub-index 15 (Constant FV 15) |
| **Lookup Table Function Block** | 1 | 3017h (Lookup Table 1 Output Y-Axis PV) |
| | 2 | 3027h (Lookup Table 2 Output Y-Axis PV) |
| | 3 | 3037h (Lookup Table 3 Output Y-Axis PV) |
| | 4 | 3047h (Lookup Table 4 Output Y-Axis PV) |
| | 5 | 3057h (Lookup Table 5 Output Y-Axis PV) |
| | 6 | 3067h (Lookup Table 6 Output Y-Axis PV) |
| **Mathematical Function Block** | 1 | 4350h sub-index 1 (Math Output PV 1) |
| | 2 | 4350h sub-index 2 (Math Output PV 2) |
| | 3 | 4350h sub-index 3 (Math Output PV 3) |
| | 4 | 4350h sub-index 4 (Math Output PV 4) |
| **Programmable Logic Function Block** | 1 | 3xy7h (Lookup Table Selected by Logic 1) |
| | 2 | 3xy7h (Lookup Table Selected by Logic 2) |
| **Processor Temperature Measured** | N/A | 5040h (Power Supply FV) sub-index 1 |
| **Power Supply Measured** | N/A | 5040h (Temperature FV) sub-index 2 |

**Table 17 – Control Number Options Depending on Source Selected**

When using any control source as the X-Axis input to a function block, the corresponding scaling limits are defined as per Table 18. It is the responsibility of the user to make sure that the scaling objects for any function block are setup appropriately depending on the source selected for the X-Axis input.

| Control Source | Scaling 1 | Scaling 2 | Dec Digits |
|---|---|---|---|
| CANopen® Message – Num 1 to 12 | 2520h | 2522h | 2502h |
| Input Function Block | 7120h | 7122h | 6132h |
| Constant Function Block | N/A | N/A | N/A (float) |
| Lookup Table yz Function Block (where yz = 01 to 6) | 0 or lowest from 3yz6h(*) | 100 or highest from 3yz6h(**) | 3yz3h |
| Mathematical Function | 4021h | 4023h | 4032h |
| Programmable Logic Function | 0% | 100% | 1 (fixed) |
| Power Supply Measured | N/A | N/A | 1 (fixed) |
| Processor Temperature Measured | N/A | N/A | 1 (fixed) |

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

**Table 18 – Scaling Limits per Control Source**

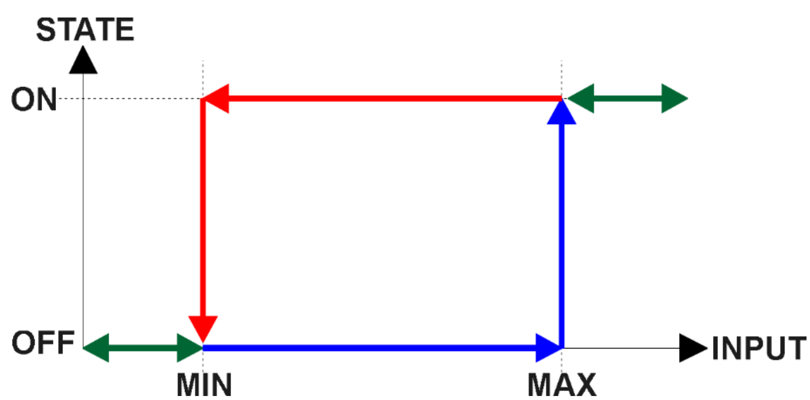



**Figure 13 – Analog source to Digital input**

## 2. INSTALLATION INSTRUCTIONS

### 2.1. Dimensions and Pinout

The MAG-IN-CO Controller is packaged in an ultra-sonically welded plastic housing. The assembly carries an IP67 rating.

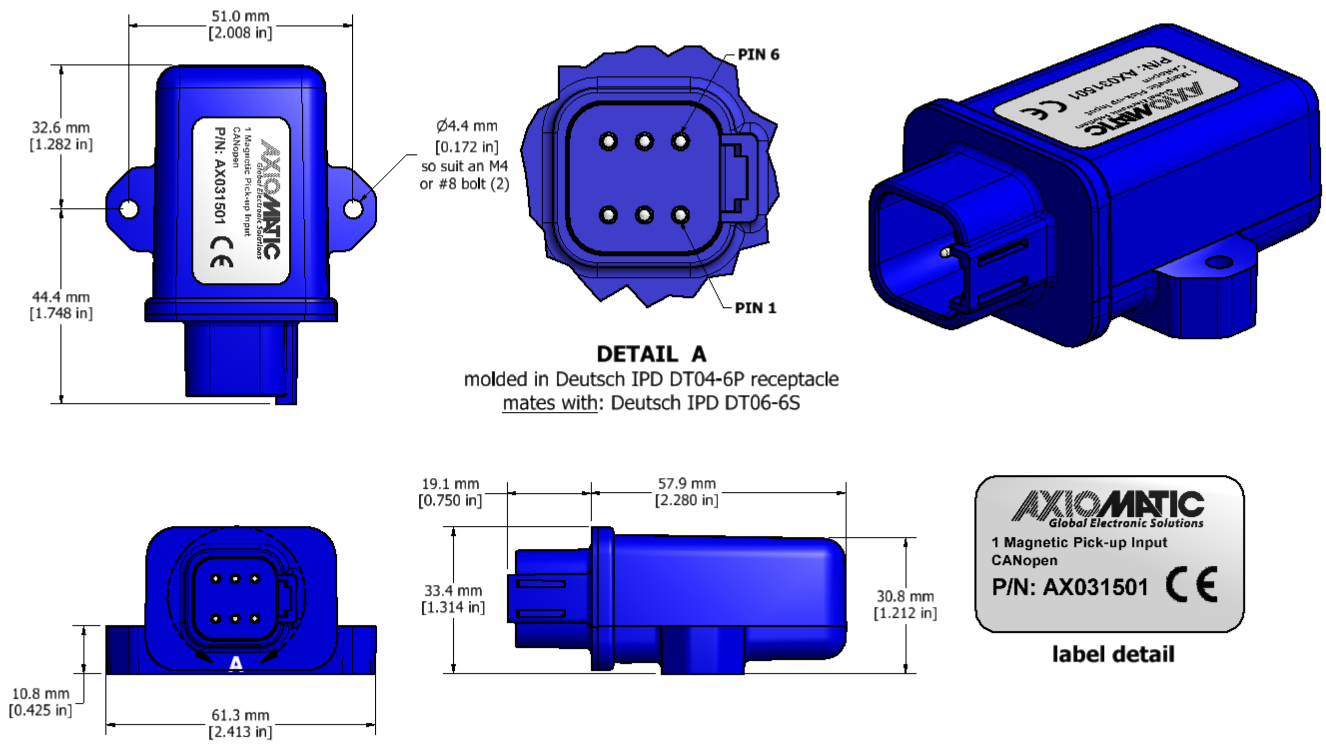



**Figure 14 – AX031501 Housing Dimensions**

| Pin # | Description |
|---|---|
| 1 | BATT+ |
| 2 | Input + |
| 3 | CAN_H |
| 4 | CAN_L |
| 5 | Input - |
| 6 | BATT- |

**Table 19 – AX031501 Connector Pinout**

## 2.2. Mounting Instructions

### NOTES & WARNINGS

- Do not install near high-voltage or high-current devices.
- 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.

**MOUNTING**

Mounting holes are sized for #8 or M4 bolts.  The bolt length will be determined by the end-user's mounting plate thickness.  The mounting flange of the controller is 0.425 inches (10.8 mm) thick.

If the module is mounted without an enclosure, it should be mounted vertically with connectors facing left or right to reduce likelihood of moisture entry.

The CAN wiring is considered intrinsically safe.  The power wires are not considered intrinsically safe and so in hazardous locations they need to be located in conduit or conduit trays at all times. The module must be mounted in an enclosure in hazardous locations for this purpose.

No wire or cable harness should exceed 30 meters in length. The power input wiring should be limited to 10 meters.

All field wiring should be suitable for the operating temperature range.

Install the unit with appropriate space available for servicing and for adequate wire harness access (6 inches or 15 cm) and strain relief (12 inches or 30 cm).

**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.

**CONNECTIONS**

Use the following TE Deutsch 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.

Refer to the respective TE Deutsch datasheets for usable insulation diameter ranges and other instructions.

| | |
|---|---|
| Receptacle Contacts | Mating Sockets as appropriate<br>(Refer to www.laddinc.com for more information on the contacts available for this mating plug.) |
| Mating Connector | DT06-06S, 1 W6S, 6 0462-201-16141 |

# 3. CANOPEN® OBJECT DICTIONARY

The CANopen® object dictionary of the MAG-IN-CO Controller is based on CiA device profile DS-404 V1.2 (device profile for measurement devices and 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 Baud rate

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

### 3.1.1. LSS Protocol to Update

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

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

#### 3.1.1.1. Setting Node-ID

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

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

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

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

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

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

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

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

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

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

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

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

#### 3.1.1.2. Setting Baudrate

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

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

- Set the baudrate by sending the following message:

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

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

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

- Activate bit timing parameters by sending the following message:

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

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

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

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

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

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

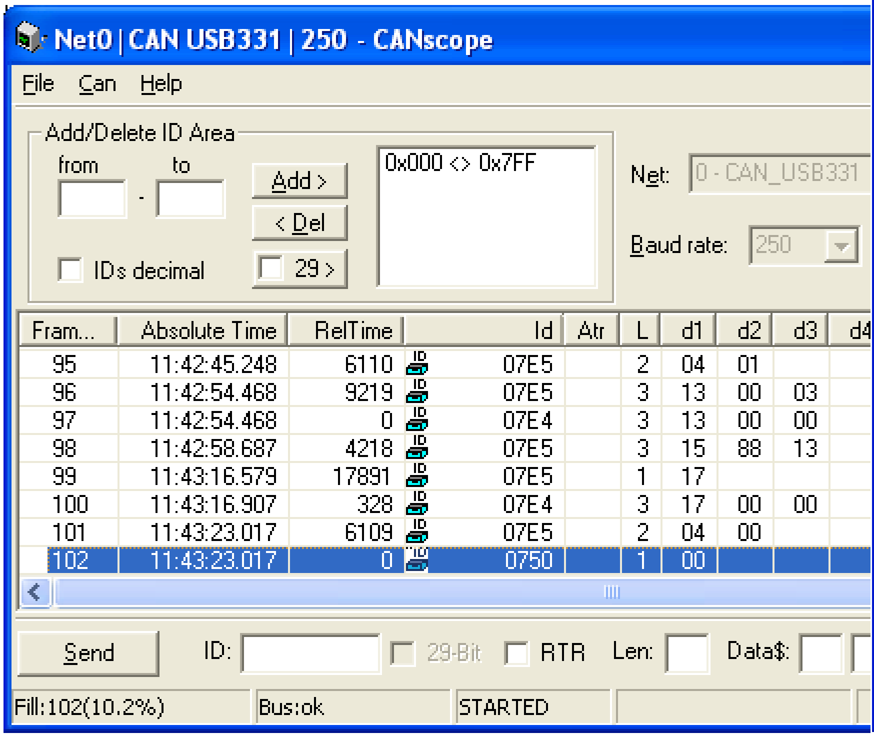
- Set the module state to LSS-operation by sending the following message: (Note, the module will reset itself back to the pre-operational state)

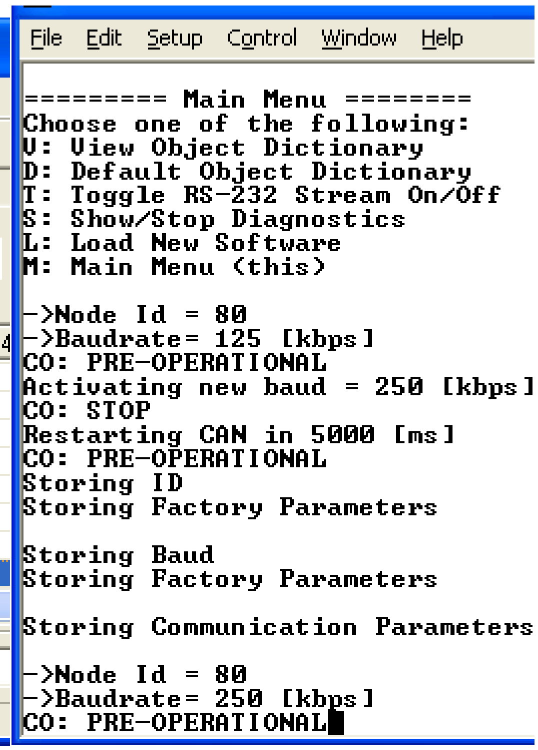
| ***Item*** | ***Value*** |
|---|---|
| COB-ID | 0x7E5 |
| Length | 2 |

| | | |
|---|---|---|
| Data 0 | 0x04 | (cs=4 for switch state global) |
| Data 1 | 0x00 | (switches to waiting state) |

The following screen capture (left) shows the CAN data was sent (7E5h) and received (7E4h) by the tool when the 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 MAG-IN-CO Controller are listed in the following table. A more detailed description of some of the objects is given in the following subchapters. Only those objects that have device-profile specific information are described. For more information on the other objects, refer to the generic CANopen® protocol specification DS-301.

| ***Index (hex)*** | ***Object*** | ***Object Type*** | ***Data Type*** | ***Access*** | ***PDO Mapping*** |
|---|---|---|---|---|---|
| 1000 | Device Type | VAR | UNSIGNED32 | RO | No |
| 1001 | Error Register | VAR | UNSIGNED8 | RO | No |
| 1002 | Manufacturer Status Register | VAR | UNSIGNED32 | RO | No |
| 1003 | Pre-Defined Error Field | ARRAY | UNSIGNED32 | RO | No |
| 100C | Guard Time | VAR | UNSIGNED16 | RW | No |
| 100D | Life Time Factor | VAR | UNSIGNED8 | RW | No |
| 1010 | Store Parameters | ARRAY | UNSIGNED32 | RW | No |

| | | | | | |
|---|---|---|---|---|---|
| 1011 | Restore Default Parameters | ARRAY | UNSIGNED32 | RW | No |
| 1016 | Consumer Heartbeat Time | ARRAY | UNSIGNED32 | RW | No |
| 1017 | Producer Heartbeat Time | VAR | UNSIGNED16 | RW | No |
| 1018 | Identity Object | RECORD | | RO | No |
| 1020 | Verify Configuration | ARRAY | UNSIGNED32 | RO | No |
| 1029 | Error Behaviour | ARRAY | UNSIGNED8 | RW | No |
| 1400 | RPDO1 Communication Parameter | RECORD | | RW | No |
| 1401 | RPDO2 Communication Parameter | RECORD | | RW | No |
| 1402 | RPDO3 Communication Parameter | RECORD | | RW | No |
| 1403 | RPDO4 Communication Parameter | RECORD | | RW | No |
| 1404 | RPDO5 Communication Parameter | RECORD | | RW | No |
| 1405 | RPDO6 Communication Parameter | RECORD | | RW | No |
| 1600 | RPDO1 Mapping Parameter | RECORD | | RO | No |
| 1601 | RPDO2 Mapping Parameter | RECORD | | RO | No |
| 1602 | RPDO3 Mapping Parameter | RECORD | | RO | No |
| 1603 | RPDO4 Mapping Parameter | RECORD | | RO | No |
| 1604 | RPDO5 Mapping Parameter | RECORD | | RO | No |
| 1605 | RPDO6 Mapping Parameter | RECORD | | RO | No |
| 1800 | TPDO1 Communication Parameter | RECORD | | RW | No |
| 1801 | TPDO2 Communication Parameter | RECORD | | RW | No |
| 1802 | TPDO3 Communication Parameter | RECORD | | RW | No |
| 1803 | TPDO4 Communication Parameter | RECORD | | RW | No |
| 1804 | TPDO5 Communication Parameter | RECORD | | RW | No |
| 1805 | TPDO6 Communication Parameter | RECORD | | RW | No |
| 1A00 | TPDO1 Mapping Parameter | RECORD | | RW | No |
| 1A01 | TPDO2 Mapping Parameter | RECORD | | RW | No |
| 1A02 | TPDO3 Mapping Parameter | RECORD | | RW | No |
| 1A03 | TPDO4 Mapping Parameter | RECORD | | RW | No |
| 1A04 | TPDO5 Mapping Parameter | RECORD | | RW | No |
| 1A05 | TPDO6 Mapping Parameter | RECORD | | RW | No |

### 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** = 0xE020 | **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 | 0xE0200194 |
| Default Value | 0xE0200194 |

### 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 MAG-IN-CO Controller, the Generic Error Bit (bit 0) is set. Only if there are no errors in the module will this bit will be cleared. No other bits in this register are used by the MAG-IN-CO 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 Object

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 behavior state if at least one error is still active.

The MAG-IN-CO 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.

If an RPDO is not received within the expected "Event Timer" period, an RPDO timeout will be flagged. Table 21 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) | | | | | |
| 1001F001h | 10h | Positive Overload (Out-of-range High) | 01h | Magnetic Pickup | F001h | Input Overload |
| 2001F001h | 20h | Negative Overload (Out-of-range Low) | 01h | Magnetic Pickup | F001h | Input Overload |
| 10003100h | 10h | Positive Overload (Vps Overvoltage) | 00h | Unspecified | 3100h | Device Voltage |
| 20003100h | 20h | Negative Overload (Vps Undervoltage) | 00h | Unspecified | 3100h | Device Voltage |
| 10004200h | 10h | Positive Overload (Over Temp) | 00h | Unspecified | 4200h | Device Temperature |
| 00008100h | 00h | RPDO Timeout | 00h | Unspecified | 8100h | Communication - generic |
| 10008130h | 10h | Lifeguard Event | 00h | Unspecified | 8130h | Lifeguard/Heartbeat Error |
| 80nn8130h | 80h | Heartbeat Timeout | nn | Node-ID | 8130h | Lifeguard/Heartbeat Error |
| 00008140h | 00h | Bus OFF Event | 00h | Unspecified | 8400h | Bus OFF Recovery |

**Table 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 15 |
| Default Value | 0 |

| | |
|---|---|
| Sub-Index | 1h to 15 |
| 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 MAG-IN-CO 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 MAG-IN-CO Controller saves parameters on command. **This means that if power is removed before the Store object is written, changes to the Object Dictionary will NOT have been saved in the non-volatile memory, and will be lost on the next power cycle.**

***Object Description***

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

***Entry Description***

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

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

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

| | |
|---|---|
| Sub-Index | 3h |
| Description | Save application parameters |

| | |
|---|---|
| Access | RW |
| PDO Mapping | No |
| Value Range | 0x65766173 (write access)<br>1h (read access) |
| Default Value | 1h |

| | |
|---|---|
| Sub-Index | 4h |
| Description | Save manufacturer parameters |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0x65766173 (write access)<br>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 MAG-IN-CO 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 MAG-IN-CO 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 MAG-IN-CO 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 MAG-IN-CO Controller can be a consumer of heartbeat objects for a single module. This object defines the expected heartbeat cycle time for that module, and if set to zero, it is not used. When the object is non-zero, the time is a multiple of 1ms, and monitoring will start after the reception of the first heartbeat from the module. If the MAG-IN-CO Controller fails to receive a heartbeat from a node in the expected timeframe, it will indicate a communication error, and respond as per object 1029h.

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

***Object Description***

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

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Description | Number of entries |
| Access | RO |
| PDO Mapping | No |
| Value Range | 1 to 4 |
| Default Value | 4 |

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

### 3.2.10.Object 1017h: Producer Heartbeat Time

The MAG-IN-CO 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 MAG-IN-CO 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 | 0xAA031501 |
| Default Value | 0xAA031501 |

| | |
|---|---|
| Sub-Index | 3h |
| Description | Revision Number |
| Access | RO |
| PDO Mapping | No |
| Value Range | UNSIGNED32 |
| Default Value | 0x00010000 |

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

***Object Description***

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

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Description | Number of entries |
| Access | RO |
| PDO Mapping | No |

| | |
|---|---|
| Value Range | 2 |
| Default Value | 2 |

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

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

### 3.2.13.Object 1029h: Error Behavior

This object controls the state that the MAG-IN-CO 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 1.1, and Power Supply faults are described 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 | 0 to 6 |
| Default Value | 6 |

| | |
|---|---|
| Sub-Index | 1h |
| Description | Communication Fault |
| Access | RW |
| PDO Mapping | No |

| | |
|---|---|
| Value Range | See above |
| Default Value | 0 (Pre-Operational) |

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

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

### 3.2.14.RPDO Behavior

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

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

| ***Sub-Index*** | ***Value*** | ***Object*** |
|---|---|---|
| 0 | 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 | 4 | Number of mapped application objects in PDO |
| 1 | 0x25000510 | Extra Received 5 PV |

| | | |
|---|---|---|
| 2 | 0x25000610 | Extra Received 6 PV |
| 3 | 0x25000710 | Extra Received 7 PV |
| 4 | 0x25000810 | Extra Received 8 PV |

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

| ***Sub-Index*** | ***Value*** | ***Object*** |
|---|---|---|
| 0 | 4 | Number of mapped application objects in PDO |
| 1 | 0x25000910 | Extra Received 9 PV |
| 2 | 0x25000A10 | Extra Received 10 PV |
| 3 | 0x25000B10 | Extra Received 11 PV |
| 4 | 0x25000C10 | Extra Received 12 PV |

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

| ***Sub-Index*** | ***Value*** | ***Object*** |
|---|---|---|
| 0 | 4 | Number of mapped application objects in PDO |
| 1 | 0x25000D10 | Extra Received 13 PV |
| 2 | 0x25000E10 | Extra Received 14 PV |
| 3 | 0 | |
| 4 | 0 | |

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

***Object Description***

| | |
|---|---|
| Index | 1400h to 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 |
| Description | COB-ID used by RPDO |
| Access | RW |
| PDO Mapping | No |
| Value Range | See value definition in DS-301 |
| Default Value | 40000000h + RPDO1 + Node ID |

| ***X*** | ***RPDOx ID*** |
|---|---|
| 1 | 0200h |
| 2 | 0300h |
| 3 | 0400h |

| | | | | |
|---|---|---|---|---|
| | C0000000h + RPDOx + Node-ID | | 4 | 0500h |

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 Behavior

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

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

| ***Sub-Index*** | ***Value*** | ***Object*** |
|---|---|---|
| 0 | 3 | Number of mapped application objects in PDO |

| | | |
|---|---|---|
| 1 | 0x71000110 | Magnetic Input Field Value |
| 2 | 0 | Not used by default |
| 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 | 3 | Number of mapped application objects in PDO |
| 1 | 0x50200020 | Processor Temperature Field Value (measured) |
| 2 | 0x50300020 | Power Supply Field Value (measured) |
| 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 | 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 |

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

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

***Object Description***

| | |
|---|---|
| Index | 1800h to 1805h |
| Name | TPDO communication parameter |
| Object Type | RECORD |
| Data Type | PDO Communication Record |

***Entry Description***

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

| | |
|---|---|
| Default Value | 5 |

| | |
|---|---|
| Sub-Index | 1h |
| Description | COB-ID used by TPDO |
| Access | RW |
| PDO Mapping | No |
| Value Range | See value definition in DS-301 |
| Default Value | 40000000h + TPDOx + Node-ID<br>C0000000h + TPDOy + 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, TPDO2)<br>0ms (on TPDO3, TPDO4) |

## 3.3. Application Objects (DS-404)

| *Index (hex)* | *Object* | *Object Type* | *Data Type* | *Access* | *PDO Mapping* |
|---|---|---|---|---|---|
| 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 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 5, 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 | Analog Input X FV |
| Access | RO |
| PDO Mapping | Yes |
| Value Range | Data Type Specific, see Table 11 |
| Default Value | No |

### 3.3.2. 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 | AI Sensor Type |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 2 |
| Default Value | 60 (frequency) |

### 3.3.3. 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 | AI Operating Mode |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 4 |
| Default Value | 1 (normal mode) |

### 3.3.4. 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 5. 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 18 in Section 1.6. 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 | AI Scaling 1 FV |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 7 |
| Default Value | 1[Hz] |

### 3.3.5. 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 5. 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 | AI Scaling 1 PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | Integer16 |
| Default Value | 1 [same as 7120h] |

### 3.3.6. 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 5. 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 18 in Section 1.6. 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 | AI Scaling 2 FV |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 7 |
| Default Value | 5000 [Hz] |

### 3.3.7. 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 5. 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 | AI Scaling 2 PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | Integer16 |
| Default Value | 5000 [same as 7122h] |

### 3.3.8. Object 7130h: AI Input Process Value

This object represents the result of the input scaling applied per Figure 5, 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 | AI Process Value |
| Access | RO |
| PDO Mapping | Yes |
| Value Range | Integer16 |
| Default Value | No |

### 3.3.9. 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 | AIx Decimal Digits PV |
| Access | RW |

| | |
|---|---|
| PDO Mapping | No |
| Value Range | 0 to 3 |
| Default Value | 0 [Hz] |

### 3.3.10.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 | AI Span Start (Error Min) |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 11 |
| Default Value | 0 [Hz] |

### 3.3.11.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 | AI Span End (Error Max) |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 11 |
| Default Value | 6000 [mV] |

### 3.3.12.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 4, and how they are used is outlined in Section 1.1.

***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 | 1 |
| Description | AI Filter Type |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 8 |
| Default Value | 0 (no filter) |

### 3.3.13.Object 61A1h: AI Filter Constant

This object defines the number of steps used in the various filters, as defined in Section 1.1.

***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 | AI 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*** |
|---|---|---|---|---|---|
| 2101 | AI Number of Pulses Per Revolution | ARRAY | UNSIGNED16 | RW | No |
| 2102 | AI Decimal Digits FV | ARRAY | UNSIGNED8 | RW | No |
| 2103 | AI Debounce FIlter | ARRAY | UNSIGNED8 | RW | No |
| 2104 | AI Pulse Count to Reset Counter | ARRAY | FLOAT32 | RW | No |
| 2105 | AI Pulse Count in Time Window | ARRAY | FLOAT32 | RW | No |
| 2106 | AI Elapsed Time to Reach Pulse Count | ARRAY | FLOAT32 | RW | No |
| 2110 | AI Error Detect Enable | ARRAY | BOOLEAN | RW | No |
| 2112 | 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 |
| 3yz0 | LTyz Input X-Axis Source | VAR | UNSIGNED8 | RW | No |
| 3yz1 | LTyz Input X-Axis Number | VAR | UNSIGNED8 | RW | No |
| 3yz2 | LTyz Auto Repeat | VAR | UNSIGNED8 | RW | No |
| 3yz3 | LTyz X-Axis Decimal Digits PV | VAR | UNSIGNED8 | RW | No |
| 3yz4 | LTyz Y-Axis Decimal Digits PV | VAR | UNSIGNED8 | RW | No |
| 3yz5 | LTyz Point Response | ARRAY | UNSIGNED8 | RW | No |
| 3yz6 | LTyz Point X-Axis PV | ARRAY | INTEGER32 | RW | No |
| 3yz7 | LTyz Point Y-Axis PV | ARRAY | INTEGER16 | RW | No |
| 3yz8 | LTyz Output Y-Axis PV | VAR | INTEGER16 | RO | Yes |
| 3300 | Logic Block Enable | ARRAY | BOOLEAN | RW | No |
| 3310 | Logic Block Selected Table | ARRAY | UNSIGNED8 | RO | Yes |
| 3320 | Logic Output Process Value | ARRAY | INTEGER16 | RO | Yes |
| 3x01 | LB(3-x) Lookup Table Number | ARRAY | UNSIGNED8 | RW | No |
| 3x02 | LB(3-x) Function Logical Operator | ARRAY | UNSIGNED8 | RW | No |
| 3x11 | LB(3-x) Function A Condition 1 | RECORD | UNSIGNED8 | RW | No |
| 3x12 | LB(3-x) Function A Condition 2 | RECORD | UNSIGNED8 | RW | No |
| 3x13 | LB(3-x) Function A Condition 3 | RECORD | UNSIGNED8 | RW | No |
| 3x21 | LB(3-x) Function B Condition 1 | RECORD | UNSIGNED8 | RW | No |

| | | | | | |
|---|---|---|---|---|---|
| 3x22 | LB(3-x) Function B Condition 2 | RECORD | UNSIGNED8 | RW | No |
| 3x23 | LB(3-x) Function B Condition 3 | RECORD | UNSIGNED8 | RW | No |
| 3x31 | LB(3-x) Function C Condition 1 | RECORD | UNSIGNED8 | RW | No |
| 3x32 | LB(3-x) Function C Condition 2 | RECORD | UNSIGNED8 | RW | No |
| 3x33 | LB(3-x) Function C Condition 3 | RECORD | UNSIGNED8 | RW | No |
| 4500 | Math Block Enable | ARRAY | BOOLEAN | RW | No |
| 4521 | Math Output Scaling 1 PV | ARRAY | INTEGER16 | RW | No |
| 4523 | Math Output Scaling 2 PV | ARRAY | INTEGER16 | RW | No |
| 4530 | Math Output Process Value | ARRAY | INTEGER16 | RO | Yes |
| 4532 | Math Output Decimal Digits PV | ARRAY | UNSIGNED8 | RW | No |
| 4y00 | Math Y Input Source | ARRAY | UNSIGNED8 | RW | No |
| 4y01 | Math Y Input Number | ARRAY | UNSIGNED8 | RW | No |
| 4y03 | Math Y Input Decimal Digits FV | ARRAY | UNSIGNED8 | RW | No |
| 4y20 | Math Y Input Scaling 1 FV | ARRAY | INTEGER16 | RW | No |
| 4y22 | Math Y Input Scaling 2 FV | ARRAY | INTEGER16 | RW | No |
| 4y40 | Math Y Input Gain | ARRAY | INTEGER8 | RW | No |
| 4y50 | Math Y Operator | ARRAY | UNSIGNED8 | RW | No |
| 5010 | Constant Field Value | ARRAY | FLOAT32 | RW | No |
| 5040 | Fault Detection Field Value | ARRAY | UNSIGNED16 | RO | Yes |
| 5041 | Fault Detection Set Threshold | ARRAY | UNSIGNED16 | RW | No |
| 5042 | Fault Detection Clear Threshold | ARRAY | UNSIGNED16 | RW | No |
| 5050 | Fault Detection Enable Err Check 8 Faults | ARRAY | UNSIGNED8 | RW | No |
| 5041 | Fault Detection Error Response Delay | ARRAY | UNSIGNED16 | RW | No |
| 5555 | Start in Operational Mode | VAR | BOOLEAN | RW | No |

*Where **yz** = 01 to 06 (LUT 1 to 06) and **x** = 4 to 5(Logic 1 to 2) and **y** = 1 to 4 (Math 1 to 4)*

### 3.4.1. 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 | Input Pulses per Revolution |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 1000 |
| Default Value | 0 |

### 3.4.2. 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 0.

In addition to the FV object 7100h, objects 2111h, 7120h, 7122h, 7148h and 7149h will also be specified with this resolution. This object is normally read-only, and will be automatically adjusted by the controller as per Table 5 depending on the analog input type and range that has been selected. When object 5550h is set to FALSE (disables automatic updates), this object becomes writeable.

***Object Description***

| | |
|---|---|
| Index | 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 | Input Decimal Digits FV |
| Access | RW (only when object 5550h is false) |
| PDO Mapping | No |
| Value Range | See Table 5 |
| Default Value | 0 [1 Hz] |

### 3.4.3. Object 2103h: AI Debounce Filter

This object will debounce the input signal applied on a single digital input as shown in Figure 4. The options for this object are listed in Table 2.

***Object Description***

| | |
|---|---|
| Index | 2103h |

| | |
|---|---|
| Name | AI Debounce Filter |
| Object Type | ARRAY |
| Data Type | UNSIGNED8 |

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h |
| Description | Input Debounce Filter |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 3 |
| Default Value | 2 [Filter 1.78 us] |

### 3.4.4. Object 2104h: AI Pulse Count to Reset Counter

This object is only used when a "Counter (Pulse Count Reset)" input type has been selected by object 6110h. When, input pulse count reaches value selected by object 2104h, the counter will reset back to 0.

***Object Description***

| | |
|---|---|
| Index | 2104h |
| Name | Pulse Count to Reset Counter |
| Object Type | ARRAY |
| Data Type | UNSIGNED32 |

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h |
| Description | Pulse Count to Reset Counter |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 1000 000 |
| Default Value | 1000 |

### 3.4.5. Object 2105h: AI Pulse Count in Time Window

This object is only used when a “Counter (Time Window)” input type has been selected by object 6110h. Object 2105h selects the time window (in milliseconds) within the amount of pulses is measured.

***Object Description***

| | |
|---|---|
| Index | 2105h |
| Name | Pulse Count in 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 | Pulse Count in Time Window |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 8.64e7 |
| Default Value | 500 |

### 3.4.6. Object 2106h: AI Elapsed Time to Reach Pulse Count

This object is only used when a “Counter (Pulse Window)” input type has been selected by object 6110h. Object 2106h defines the amount of input pulses (in number of pulses) to be received.

***Object Description***

| | |
|---|---|
| Index | 2106h |
| Name | Elapsed Time to Reach Pulse Count |
| 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 | Elapsed Time to Reach Pulse Count |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 1000 000 |
| Default Value | 1000 |

### 3.4.7. 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 | Input Error Detect Enable |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 (FALSE) or 1 (TRUE) |
| Default Value | 0 [FALSE] |

### 3.4.8. 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 | Input Error Clear Hysteresis |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 11 |
| Default Value | 50 [Hz] |

### 3.4.9. 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 | AI Error Reaction Delay |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 60,000 |
| Default Value | 1000 [ms] |

### 3.4.10.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.

***Object Description***

| | |
|---|---|
| Index | 2500h |
| Name | EC Extra Received Process Value |
| Object Type | ARRAY |

| | |
|---|---|
| Data Type | INTEGER16 |

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Description | Largest sub-index supported |
| Access | RO |
| PDO Mapping | No |
| Value Range | 14 |
| Default Value | 14 |

| | |
|---|---|
| Sub-Index | 1h to 14h (x = 1 to 14) |
| Description | ECx Received Process Value |
| Access | RW |
| PDO Mapping | Yes |
| Value Range | Integer16 |
| Default Value | No |

### 3.4.11.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 | 15 |
| Default Value | 15 |

| | |
|---|---|
| Sub-Index | 1h to 15h (x = 1 to 15) |
| Description | ECx Decimal Digits PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 3 |
| Default Value | 2 (0.01 resolution) |

### 3.4.12.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 5.

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 | 15 |
| Default Value | 15 |

| | |
|---|---|
| Sub-Index | 1h to 15h (x = 1 to 15) |
| Description | ECx Scaling 1 PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | -32768 to 2522h sub-index X |
| Default Value | 0 |

### 3.4.13.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 5. 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 | 15 |
| Default Value | 15 |

| | |
|---|---|
| Sub-Index | 1h to 15h (x = 1 to 15) |
| Description | ECx Scaling 2 PV |
| Access | RW |

| | |
|---|---|
| PDO Mapping | No |
| Value Range | 2520h sub-index X to 32767 |
| Default Value | 1000 (10.00) |

### 3.4.14.Object 3yz0h: LTyz Input X-Axis Source

This object defines the type of input that will be used to determine the X-Axis input process value for the lookup table function. The available control sources on the CAN-IN-CO controller are listed in Table 16. Not all sources would make sense to use as an X-Axis input, and it is the user's responsibility to select a source that makes sense for the application. A selection of "Control Source Not Used" disables the associated lookup table function block.

***Object Description***

| | |
|---|---|
| Index | 3yz0h (where yz = 01 to 06) |
| Name | LTyz Input X-Axis Source |
| Object Type | VARIABLE |
| Data Type | UNSIGNED8 |

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 16 |
| Default Value | 0 (control not used) |

### 3.4.15.Object 3yz1h: LTyz Input X-Axis Number

This object defines the number of the source that will be used as the X-Axis input PV for the lookup table function. The available control numbers are dependent on the source selected, as shown in Table 17. 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 18.

***Object Description***

| | |
|---|---|
| Index | 3yz1h (where yz = 01 to 06) |
| Name | LTyz Input X-Axis Number |
| Object Type | VARIABLE |
| Data Type | UNSIGNED8 |

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 17 |
| Default Value | 0 (null control source) |

### 3.4.16.Object 3yz3h: LTyz X-Axis Decimal Digits PV

This object describes the number of digits following the decimal point (i.e. resolution) of the X-Axis input data and the points in the lookup table. It should be set equal to the decimal digits used by the PV from the control source/number as defined in Table 18.

***Object Description***

| | |
|---|---|
| Index | 3yz3h (where yz = 01 to 06) |
| Name | LTyz X-Axis Decimal Digits PV |
| Object Type | VARIABLE |
| Data Type | UNSIGNED8 |

***Entry Description***

| | |
|---|---|
| Sub-Index | 0h |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 4 (see Table 17) |
| Default Value | 0 |

### 3.4.17.Object 3yz4h: LTyz Y-Axis Decimal Digits PV

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

***Object Description***

| | |
|---|---|
| Index | 3yz4h (where yz = 01 to 06) |
| Name | LTyz Y-Axis Decimal Digits PV |
| Object Type | VARIABLE |
| Data Type | UNSIGNED8 |

***Entry Description***

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

### 3.4.18.Object 3yz5h: LTyz Point Response

This object determines the Y-Axis output response to changes in the X-Axis input. The value set in sub-index 1 determines the X-Axis type (i.e. data or time), while all other sub-indexes determine the response (ramp, step, ignore) between two points on the curve. The options for this object are listed in Table 8. See Figure 7 for an example of the difference between a step and ramp response.

***Object Description***

| | |
|---|---|
| Index | 3yz5h (where yz = 01 to 6) |
| Name | LTyz Point Response |
| Object Type | ARRAY |
| Data Type | UNSIGNED8 |

***Entry Description***

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

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

| | |
|---|---|
| Sub-Index | 2h to Bh (x = 2 to 11) |
| Description | LTyz Point X Response |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 23 (0, 1 or 2) |
| Default Value | 1 (ramp to response) |

### 3.4.19.Object 3yz6h: LTyz Point X-Axis PV

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

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

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

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

***Object Description***

| | |
|---|---|
| Index | 3yz6h (where yz = 01 to 6) |
| Name | LTyz Point X-Axis PV |
| Object Type | ARRAY |
| Data Type | INTEGER32 |

***Entry Description***

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

| | | |
|---|---|---|
| Sub-Index | 1h to Bh (x = 1 to 11) | |
| Description | LTyz Point X-Axis PVx | |
| Access | RW | |
| PDO Mapping | No | |
| Value Range | See above (data) | 1 to 86400000 (time) |
| Default Value | 10*(x-1) | No |

### 3.4.20.Object 3yz7h: LTyz Point Y-Axis PV

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

***Object Description***

| | |
|---|---|
| Index | 3yz7h (where yz = 01 to 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 Bh (x = 1 to 11) |
| Description | LTyz Point Y-Axis PVx |
| Access | RW |
| PDO Mapping | No |
| Value Range | Integer16 |
| Default Value | 10*(x-1) [i.e. 0, 10, 20, 30, … 100] |

### 3.4.21.Object 3yz8h: LTyz Output Y-Axis PV

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

***Object Description***

| | |
|---|---|
| Index | 3yz8h (where yz = 01 to 06) |
| Name | LTyz Output Y-Axis PV |
| Object Type | VARIABLE |
| Data Type | INTEGER16 |

***Entry Description***

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

### 3.4.22.Object 3300h: Logic Block Enable

This object defines whether or not the logic shown in Figure 10 will be evaluated.

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 2h (x = 1 to 2) |
| Description | LBx Enable |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 (FALSE) or 1 (TRUE) |
| Default Value | 0 [FALSE] |

### 3.4.23.Object 3310h: Logic Block Selected Table

This read-only object reflects what table has been selected as the output source for the logic block after the evaluation shown in Figure 10 has been performed.

***Object Description***

| | |
|---|---|
| Index | 3310h |
| Name | Logic Block Selected Table |
| Object Type | ARRAY |

| | |
|---|---|
| Data Type | UNSIGNED8 |

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 2h (x = 1 to 2) |
| Description | LBx  Selected Table |
| Access | RO |
| PDO Mapping | Yes |
| Value Range | 1 to 6 |
| Default Value | No |

### 3.4.24.Object 3320h: Logic Block Output PV

This read-only object reflects the output from the selected table, interpreted as a percentage. The limits for the percentage conversion are based on the range of the lookup tables Y-Axis Output PV as shown in Table 13. This value has a fixed decimal digit value of 1 giving a resolution of 0.1%.

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 2h (x = 1 to 2) |
| Description | LBx Output PV |
| Access | RO |
| PDO Mapping | Yes |

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

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (y = A to C) |
| Description | LB(3-x) Lookup Table Y Number |
| Access | RW |
| PDO Mapping | No |
| Value Range | 1 to 6 |
| Default Value | See Table 18 |

### 3.4.26.Object 3x02h: LB(3-x) Function Logical Operator

This object determines how the results of the three conditions for each function are to be compared to one another to determine the overall state of the function output. There are up to three functions that can be evaluated in each logic block. The options for this object are defined in Table 12. See Section 1.3 for more information about how this object is used.

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (y = A to C) |
| Description | LB(3-x) Function Y Logical Operator |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 12 |
| Default Value | Function A = 1 (and all)<br>Function B = 1 (and all)<br>Function C = 0 (default) |

### 3.4.27. Object 3x11h: LB(3-x) Function A Condition 1
### 3.4.28. Object 3x12h: LB(3-x) Function A Condition 2
### 3.4.29. Object 3x13h: LB(3-x) Function A Condition 3
### 3.4.30. Object 3x21h: LB(3-x) Function B Condition 1
### 3.4.31. Object 3x22h: LB(3-x) Function B Condition 2
### 3.4.32. Object 3x23h: LB(3-x) Function B Condition 3
### 3.4.33. Object 3x31h: LB(3-x) Function C Condition 1
### 3.4.34. Object 3x32h: LB(3-x) Function C Condition 2
### 3.4.35. Object 3x33h: LB(3-x) Function C Condition 3

These objects, 3xyzh, represent Logic Block x, Function y, Condition z, where x = 4 to 5, y = 1 (A) to 3 (C), and z = 1 to 3. All of these objects are a special type of record, defined in Table 9. Information on how to use these objects is defined in Section 1.3.

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h |
| Description | Argument 1 Source |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 15 |
| Default Value | 1 (CANopen® Message) |

| | |
|---|---|
| Sub-Index | 2h |
| Description | Argument 1 Number |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 17 |
| Default Value | 1 (EC Received PV 1) |

| | |
|---|---|
| Sub-Index | 3h |
| Description | Argument 2 Source |
| Access | RW |
| PDO Mapping | No |

| | |
|---|---|
| Value Range | See Table 16 |
| Default Value | 6 (Constant PV) |

| | |
|---|---|
| Sub-Index | 4h |
| Description | Argument 2 Number |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 17 |
| Default Value | 3 (Constant FV 3) |

| | |
|---|---|
| Sub-Index | 5h |
| Description | Operator |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 16 |
| Default Value | 0 (Equals) |

### 3.4.36.Object 4000h: Math Function Enable

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

***Object Description***

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

***Entry Description***

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

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

### 3.4.37.Object 4021h: Math Output Scaling 1 PV

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (Y = 1 to 4) |
| Description | Math Y Output Scaling 1 PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | -32768 to 32767 |
| Default Value | 0 |

### 3.4.38.Object 4023h: Math Output Scaling 2 PV

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (Y = 1 to 4) |
| Description | Math Y Output Scaling 2 PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | -32768 to 32767 |
| Default Value | 10000 (100.00) |

### 3.4.39.Object 4030h: Math Output Process Value

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (Y = 1 to 4) |
| Description | Math Y Output Process Value |
| Access | RO |
| PDO Mapping | Yes |
| Value Range | -32768 to 32767 |
| Default Value | No |

### 3.4.40.Object 4032h: Math Output Decimal Digits PV

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

***Object Description***

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

***Entry Description***

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

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

### 3.4.41.Object 4y00h: Math Y Input Source

This object defines the input sources that will be used in the mathematical calculations. Here, y = 1 to 4 – representing Math Block 1 to Math Block 4. If a control source is not used, the associate mathematical calculation would be ignored. The available control sources on the MAG-IN-CO controller are listed in Table 15.

***Object Description***

| | |
|---|---|
| Index | 4y00h (y = 1 to 4) |
| Name | Math Y Input Source |
| Object Type | ARRAY |
| Data Type | UNSIGNED8 |

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (X = 1 to 4) |
| Description | Math Y Input X Source |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 16 |
| Default Value | 0 (control source not used) |

### 3.4.42.Object 4y01h: Math Y Input Number

This object defines the number of the input source that will be used in the math calculation. The available control numbers are dependent on the source selected, as shown in Table 17. Once selected, the input value will be used in the corresponding calculation as described in Section 1.4.

***Object Description***

| | |
|---|---|
| Index | 4y01h (y = 1 to 4) |
| Name | Math Y Input Number |

| | |
|---|---|
| Object Type | ARRAY |
| Data Type | UNSIGNED8 |

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (X = 1 to 4) |
| Description | Math Y Input X Number |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 17 |
| Default Value | 0 (null input) |

### 3.4.43.Object 4y03h: Math Y Input Decimal Digits FV

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (X = 1 to 4) |
| Description | Math Y Input X Decimal Digits PV |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to4 |
| Default Value | 2 (0.01) |

### 3.4.44.Object 4y20h: Math Y Input Scaling 1 FV

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

***Object Description***

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

***Entry Description***

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

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

### 3.4.45.Object 4y22h: Math Y Input Scaling 2 FV

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Sub-Index | 1h to 4h (X = 1 to 4) |
| Description | Math Y Input X Scaling 2 FV |
| Access | RW |

| | |
|---|---|
| PDO Mapping | No |
| Value Range | INTEGER16 |
| Default Value | 10000 (100.00%) |

### 3.4.46.Object 4y40h: Math Y Input Gain

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

***Object Description***

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

***Entry Description***

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

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

### 3.4.47.Object 4y50h: Math Y Operator

This object defines the actual operators that will be used in each stage of a math calculation, as described in Section 1.4. The options for this object are listed in Table 30.

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| PDO Mapping | No |
| Value Range | 3 |
| Default Value | 3 |

| | |
|---|---|
| Sub-Index | 1h to 3h (X = 1 to 3) |
| Description | Math Y Function X Operator |
| Access | RW |
| PDO Mapping | No |
| Value Range | See Table 15 |
| Default Value | 9 (Plus) |

### 3.4.48.Object 5010h: Constant Field Value

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

***Object Description***

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

***Entry Description***

| | |
|---|---|
| Sub-Index | 0 |
| Description | Largest sub-index supported |
| Access | RO |
| PDO Mapping | No |
| Value Range | 12 |
| Default Value | 12 |

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

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

| | |
|---|---|
| Sub-Index | 3 |
| Description | Constant FV 3 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 10.00 |

| | |
|---|---|
| Sub-Index | 4 |
| Description | Constant FV 4 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 20.00 |

| | |
|---|---|
| Sub-Index | 5 |
| Description | Constant FV 5 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 30.00 |

| | |
|---|---|
| Sub-Index | 6 |
| Description | Constant FV 6 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 40.00 |

| | |
|---|---|
| Sub-Index | 7 |
| Description | Constant FV 7 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 50.00 |

| | |
|---|---|
| Sub-Index | 8 |
| Description | Constant FV 8 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 60.00 |

| | |
|---|---|
| Sub-Index | 9 |
| Description | Constant FV 9 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 70.00 |

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

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

| | |
|---|---|
| Default Value | 90.00 |

| | |
|---|---|
| Sub-Index | 12 |
| Description | Constant FV 12 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Float32 |
| Default Value | 100.00 |

### 3.4.49.Object 5040h: Fault Detection Field Value

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

***Object Description***

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

***Entry Description***

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

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

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

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

### 3.4.50. Object 5041h: FD Set Threshold

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

***Object Description***

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

***Entry Description***

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

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

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

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

### 3.4.51. Object 5042h: FD Clear Threshold

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Subindex | 1h |
| Description | Over Temperature Clear Threshold |
| Access | RW |
| PDO Mapping | No |
| Value Range | 500 to 4001h at subindex 1 [ºC x 10] |
| Default Value | 850 (85.0ºC) |

| | |
|---|---|
| Subindex | 2h |
| Description | Over Voltage Clear Threshold |
| Access | RW |
| PDO Mapping | No |
| Value Range | 4002h at subindex 3 to<br>4001h at subindex 2 [V x 10] |
| Default Value | 480 (48.0V) |

| | |
|---|---|
| Subindex | 3h |
| Description | Under Voltage Clear Threshold |
| Access | RW |
| PDO Mapping | No |
| Value Range | 4001h at subindex 3 to<br>4002h at subindex 2 [V x 10] |
| Default Value | 120 (12.0V) |

### 3.4.52.Object 5050h: FD Enable Error Check 3 Faults

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

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

***Object Description***

| | |
|---|---|
| Index | 5050h |
| Name | FD Enable Error Checking 3 Faults |
| Object Type | ARRAY |
| Data Type | UNSIGNED8 |

***Entry Description***

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

| | |
|---|---|
| Sub-index | 1h |
| Description | Error Check for FD 1 to FD 3 |
| Access | RW |
| PDO Mapping | No |
| Value Range | Bit Value 0 = Error Check Disabled<br>Bit Value 1 = Error Check Enabled |
| Default Value | 00h (all error check disabled) |

### 3.4.53.Object 5051h: FD Error Response Delay

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

***Object Description***

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

***Entry Description***

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

| | |
|---|---|
| Subindex | 1h to 3h |
| Description | Error Delay, FD 1 to FD 3 |
| Access | RW |
| PDO Mapping | No |
| Value Range | 0 to 10000 [ms] |
| Default Value | 5000 [ms] |

### 3.4.54.Object 5550h: Enable Automatic Updates

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

***Object Description***

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

***Entry Description***

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

### 3.4.55.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 MAG-IN-CO 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] |

# APPENDIX A - TECHNICAL SPECIFICATION

*Specifications are indicative and subject to change. Actual performance will vary depending on the application and operating conditions. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on [https://www.axiomatic.com/service/](https://www.axiomatic.com/service/).*

## Power

| | |
|---|---|
| Power Supply Input - Nominal | 12 V, 24 Vdc or 48 Vdc nominal; 8…60 Vdc<br>The minimum allowable supply voltage for the power pin is 7 Vdc. |
| Surge Protection | Meets the surge requirements of DIN EN13309 |
| Reverse Polarity Protection | Provided |

## Input

| | |
|---|---|
| Input | 1 Magnetic Pick-up type input<br>Frequency Range: 0.5 Hz to 20 kHz<br>Amplitude: 100 mVAC to 100VAC (RMS) |

## General Specifications

<table>
<tr><td>Memory</td><td>STM32F103CBT7<br>32-bit, 128 Kbytes Flash Program Memory</td></tr>
<tr><td>CAN Port</td><td>1 CAN 2.0B port, protocol CiA CANopen®</td></tr>
<tr><td>User Interface</td><td>EDS file</td></tr>
<tr><td>Control Logic</td><td>The Converter comes pre-programmed with standard logic.</td></tr>
<tr><td>Sampling Time</td><td>The input is tied to a 16-bit timer.</td></tr>
<tr><td>Quiescent Current Draw</td><td>9.1 mA @ 24Vdc; 16 mA @ 12Vdc Typical</td></tr>
<tr><td>Operating Conditions</td><td>-40 to 85°C (-40 to 185°F)</td></tr>
<tr><td>Weight</td><td>0.10 lb. (0.045 kg)</td></tr>
<tr><td>Protection Rating</td><td>IP67</td></tr>
<tr><td>EMC Compliance</td><td>CE marking</td></tr>
<tr><td>Vibration</td><td>MIL-STD-202G, Test 204D and 214A (Sine and Random)<br>10 g peak (Sine); 7.86 Grms peak (Random)</td></tr>
<tr><td>Shock</td><td>MIL-STD-202G, Test 213B, 50 g</td></tr>
<tr><td>Enclosure and Dimensions</td><td>Plastic Enclosure, Nylon 6-6 with 30% glass fill<br>Integral connector (equivalent TE Deutsch P/N: DT04-6P)<br>Refer to the dimensional drawing.</td></tr>
<tr><td>Electrical Connections</td><td>6-pin connector (equivalent TE Deutsch P/N: DT04-6P)<br>A mating plug kit is available as Axiomatic P/N: AX070119.
<table>
<tr><th colspan="2">CAN and I/O Connector</th></tr>
<tr><th>Pin #</th><th>Description</th></tr>
<tr><td>1</td><td>BATT+</td></tr>
<tr><td>2</td><td>Input +</td></tr>
<tr><td>3</td><td>CAN_H</td></tr>
<tr><td>4</td><td>CAN_L</td></tr>
<tr><td>5</td><td>Input GND</td></tr>
<tr><td>6</td><td>BATT-</td></tr>
</table></td></tr>
<tr><td>S/W Reflashing</td><td>Via the Axiomatic Electronic Assistant KIT, P/Ns: AX070502 or AX070506K</td></tr>
</table>

## OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

## OUR COMPANY

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

## QUALITY DESIGN AND MANUFACTURING

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

## WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

## COMPLIANCE

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

## SAFE USE

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

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

## SERVICE

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

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

## DISPOSAL

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

## CONTACTS

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

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