TR ELECTRONIC
. . . Passionate about Sensors.


# TR 58 Series Rotary Encoder

# Module and AOI Configuration Guide

email: customercare@trelectronic.com | web: www.trelectronic.com

US: Head Office: P.O. Box 4448, Troy, Michigan, 48099 | Tel: (248)-244-2280 | Fax: (248)-244-2283 | Toll Free: 1-800-709-3300
Canada: Head Office: P.O. Box 2543, Station B, London, Ontario, N6A 4G9 | Tel: (519)-452-1999 | Fax: (519)-452-1177 | Toll Free: 1-800-265-9483

*Contents*

## Introduction

This guide is intended to help install and use a TR 58 Series Ethernet IP encoder quickly and successfully. It goes through all of the critical implementation steps in detail including: EDS (Electronic Data Sheets) installation, RSLogix module configuration, and AOI (Add-On-Instruction) implementation.

It is important to note that this guide does not need to be followed to have success with a TR 58 Series encoder. There are plenty of ways to successfully integrate the device into an automation environment. However, this guide is recommended as an efficient way to ensure the encoder is configured as expected, and position values are properly understood.

### Prerequisites

Before beginning this guide, it is recommended you have the following files downloaded and stored in a known location:

- Required:
  - Device user manual: “Absolute Encoder C__-58”.
    - http://www.tr-electronic.com/service/downloads/operating-manuals/encoder-and-linear-transducer.html#c17692
  - Device EDS file: “04710022_TR_C_SERIES_1_2.eds” and icon “04710022_TR_C_SERIES.ico”.
    - http://www.tr-electronic.com/service/downloads/file-download.html
- Required if you plan on using the AOI:
  - AOI file: “TR_58_Series_AOI.L5X”

## Add Module and Import AOI

The first step to successfully using the AOI is adding the TR_58_Series module and corresponding AOI to your project RSLogix project.

### Add module

Ethernet IP communications require that a module for the encoder is added to the IO tree in your RSLogix project. There are two possible modules that could be used: TR C-Series Encoder or a Generic Ethernet Module. If you plan on using the AOI provided, the TR C-Series Encoder must be used. It is still recommended to use the TR C-Series Encoder even if you do not plan to use the AOI, but if you'd like to use the Generic Ethernet Module then this document will not be helpful. Please refer to "Absolute Encoder C__-58" for the IO and Configuration Assembly Object information.

The AOI was designed to work with EDS version 1.2 of the TR C-Series Encoder module. It is possible you've previously installed version 1.1. If you install 1.2 with 1.1 already installed, the module properties for any modules previously added will not be available. Fortunately, the encoders will still continue to work as expected. After 1.2 is installed, any new TR C-Series Encoders added will be version 1.2.

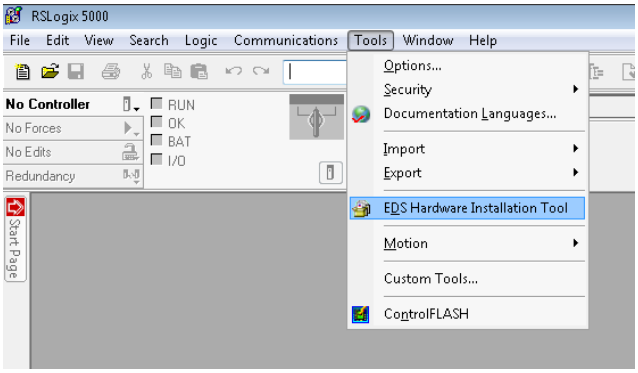
If you've already installed version 1.2 of the TR C-Series Encoders, this step can be skipped.

If version 1.2 must be installed, perform the following step.

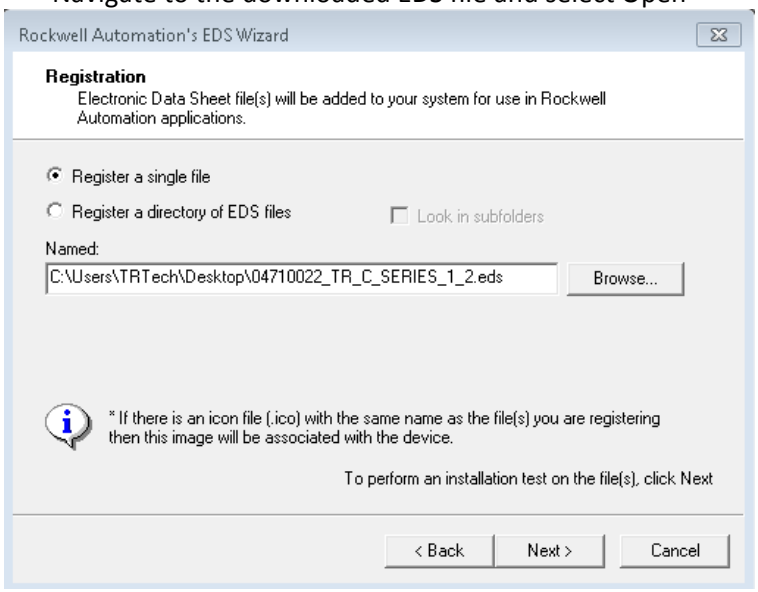
## *Install EDS*

To install the EDS file, open RSLogix and select:

- Tools
  - EDS Hardware Installation Tool




- Within the Rockwell Automation's EDS Wizard select:
  - Next
  - Next – with Register and EDS file(s) selected
  - Browse:
    - Navigate to the downloaded EDS file and select Open

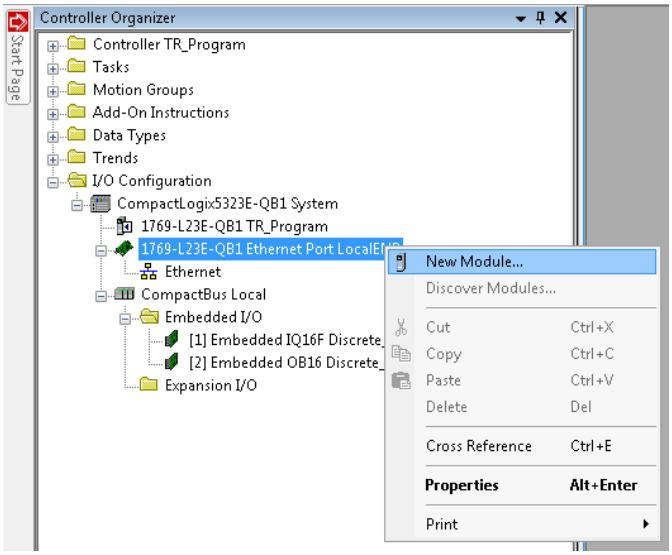



  - Next, Next, Next, Next, Finish

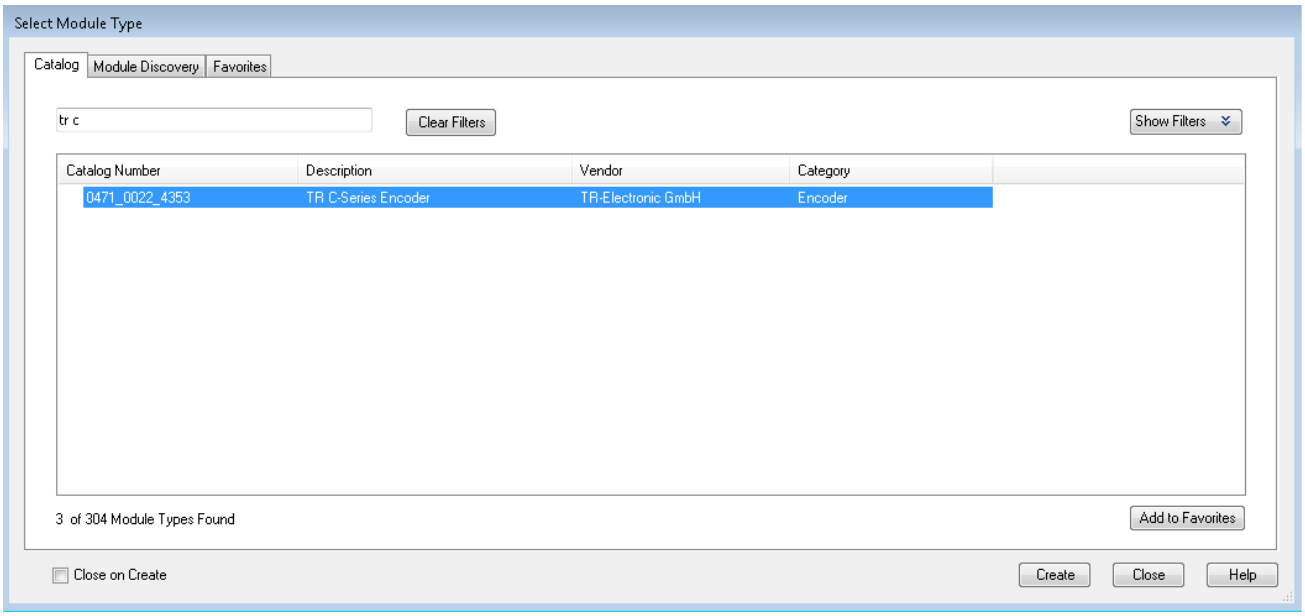
## *Add Module*

Once the correct EDS file has been installed, a module needs to be added to the project. Start by opening your working project.

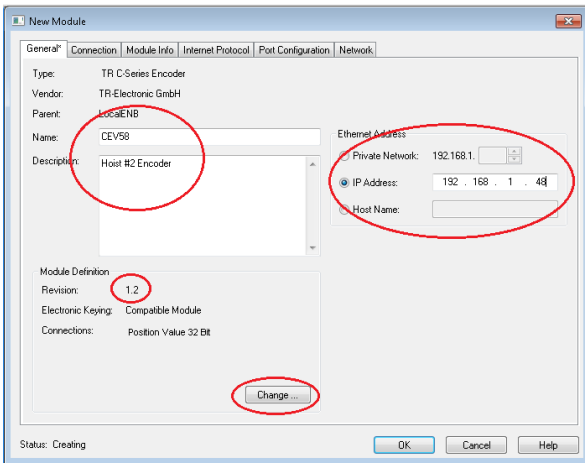
- Right click on your Ethernet module in the IO tree and select new module



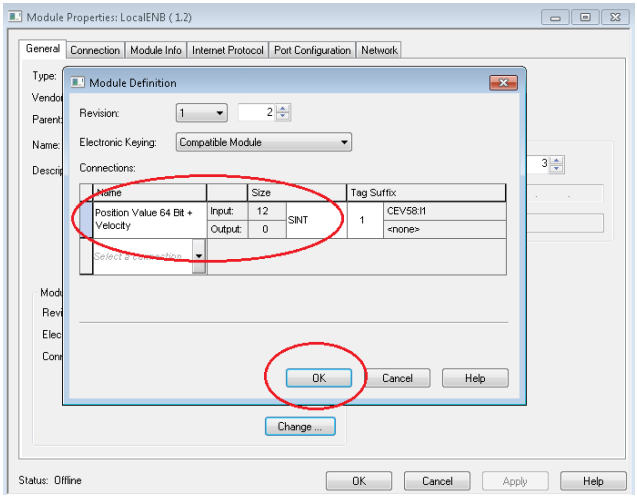

- Search for “tr c”, select TR C-Series Encoder and then select Create.



- Give the module a name, description (optional) and IP address.
    - This guide will continue using the name CEV58. However you may choose a name of your preference. Whenever CEV58 is referenced in this document, replace it with your module name.
- Verify that the revision is 1.2. If not, ensure the latest EDS file was downloaded and try re-installing.
- Select Change




- Within the Module Definition ensure the Connections Name and Size match below. This is a requirement for the AOI to function.



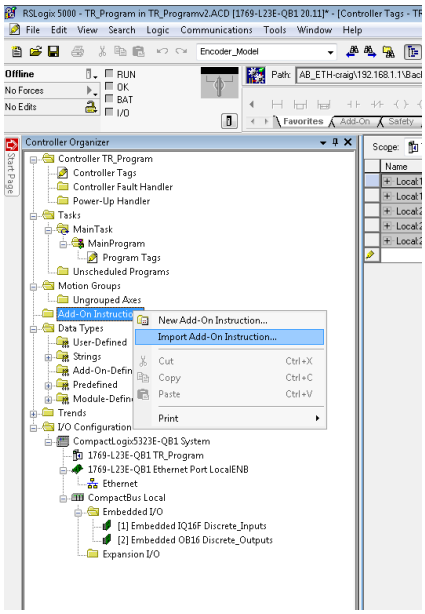

If a popup appears, click Yes. Click OK again to exit Module Properties.

If you do not plan on using the AOI, then please skip to the "Configure Encoder" section. It is the final section of the manual that will be helpful for module configuration only.

## Import AOI

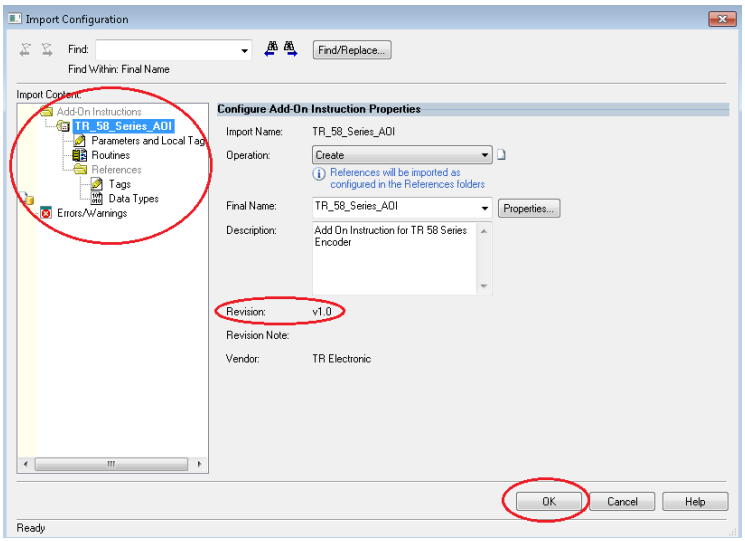
Now that a module has been added to the program, we need to add the logic that will interact with the module. This piece of logic is an AOI (Add-On Instruction) that must be imported into the project.

- Right click on Add-On Instructions in RSLogix under the Controller Organizer and select Import Add-On Instruction.

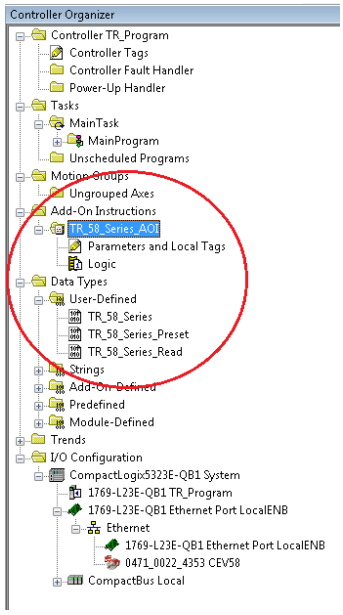



- Navigate to the AOI file that was downloaded and select Import:

- An Import Configuration dialog box will open with information regarding changes that come along with importing this AOI (Routines, Tags, Data Types, etc). Navigate around this dialog box for more information, then select OK.
    - Note the revision. If you've previously installed and are using an earlier revision, please review the consequences of overwriting. It might be best to change the "final name" of the AOI and imported Data Types to something else, so the new AOIs and UDT can be installed without impacting the old one.



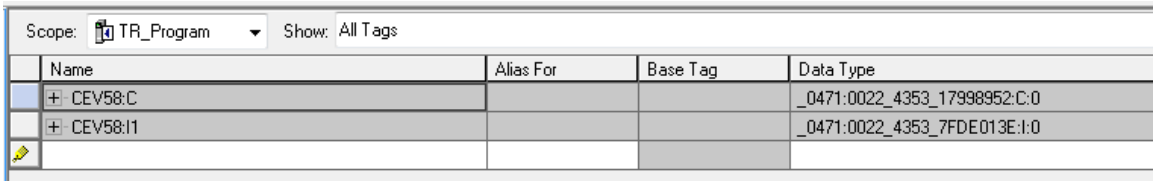

- When the AOI is imported, you'll see it under Add-On-Instruction in Controller Properties. In addition, under Data Types -> User Defined, you should see 3 new datatypes. These are the new datatypes introduced when the AOI was imported. They will be discussed in further detail in later sections.



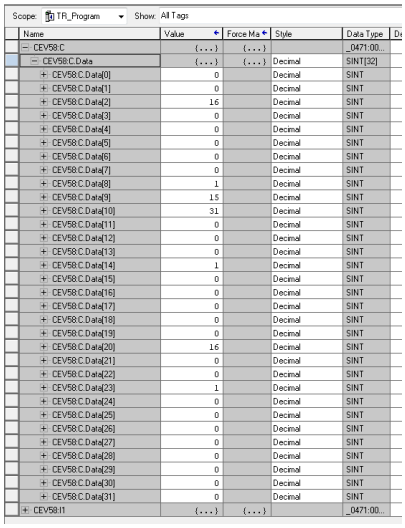
# Configuring Encoder and AOI

## Configure Encoder

After adding the module, you'll see two new tags in your tag database. The tag name will depend on what you named the new module, which was "CEV58" in this example.



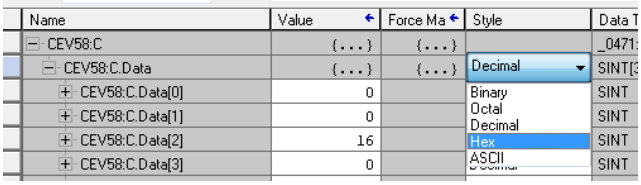

CEV58:I1 is the IO Assembly and contains the feedback/readings directly from the encoder. CEV58:C is the Configuration Assembly and is used to quickly configure the encoder upon successful connection with the PLC. Expanding the configuration assembly looks like below. The initial values have been populated as defined in the EDS file:




Refer to Appendix Configuration Assembly to edit these values as required.

It is important to note that the configuration assembly values must be populated byte by byte. Recommended procedure for updated these values is:

- Set CEV58:C.Data style to Hex.



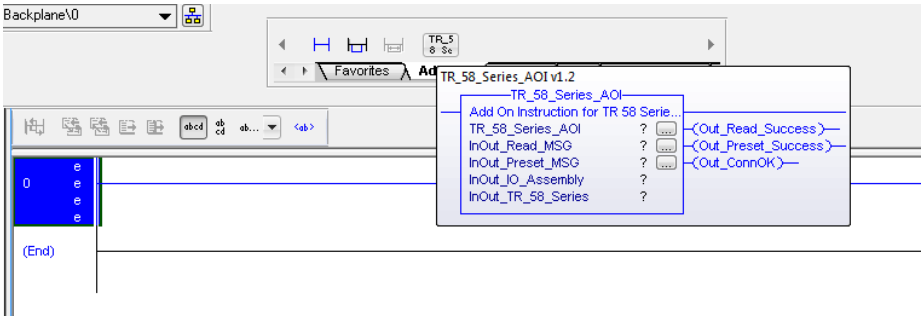

- Convert desired value into Hex
- Break converted Hex number into bytes (2 Hex digits)
- Enter value byte by byte into CEV58:C.Data array

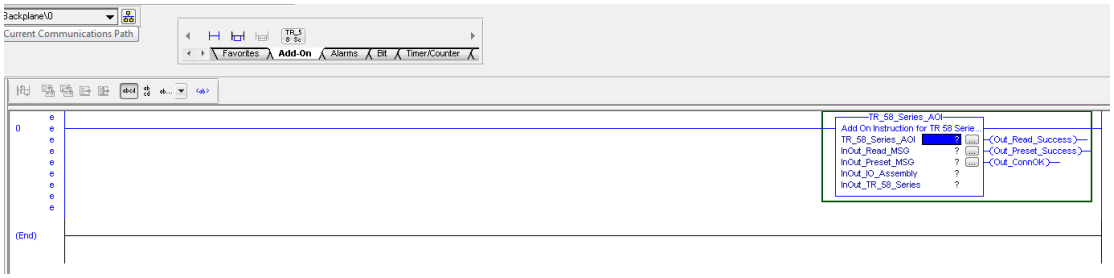
Refer to Appendix Configuration Assembly Mapping Example for a conversion example.

## Configure the AOI

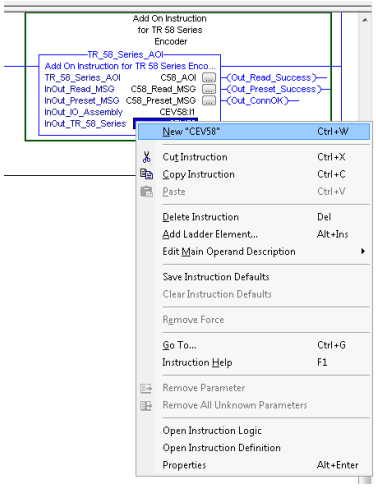
Now that the AOI has been imported, it has to be implemented in ladder logic.

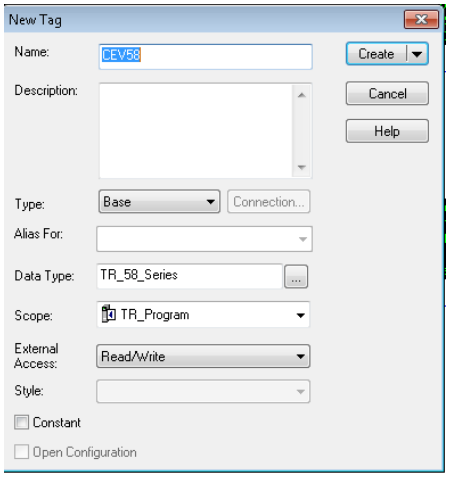
- The new instruction can be called by navigating to the desired routine and rung by selecting it on the standard instruction toolbar:



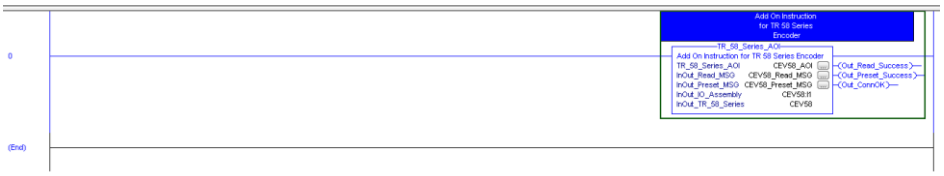

- Now we have "called" the AOI. However in it requires certain variables to be passed to it to function properly. Variable fields can be edited by double clicking.
    - InOut_TR_58_Series field:
        - Enter desired encoder name, you can re-use the module name as I have done here. You'll have to create a new variable (of type TR_58_Series) to use the AOI as discussed later. Note the scope (adjust if necessary) and add a description to the tag if you wish.



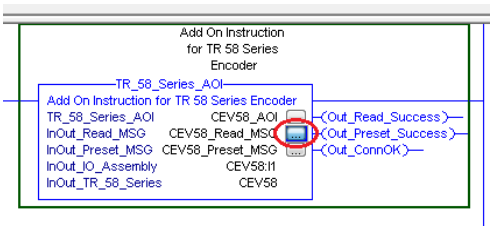



- InOut_IO_Assembly field:
  - Enter CEV58:I1
    - NOTE: The AOI requires this variable to be of a certain type as defined in the module properties (Position Value 64 Bit + Velocity, SINTs). If you'd like to change the module properties to something else, you'll have to change the AOI parameters accordingly.
- TR_58_Series_AOI field:
  - Enter CEV58_AOI and create a new tag.
- InOut_Read_MSG field:
  - Enter CEV58_Read_MSG and create a new tag.
- InOut_Preset_MSG field:
  - Enter CEV58_Preset_MSG and create a new tag.

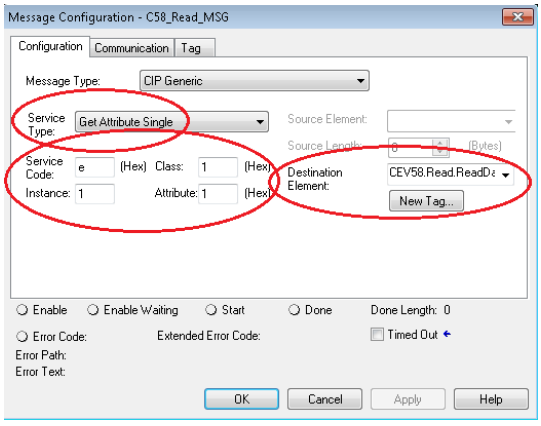
- The rung should now look something like this:



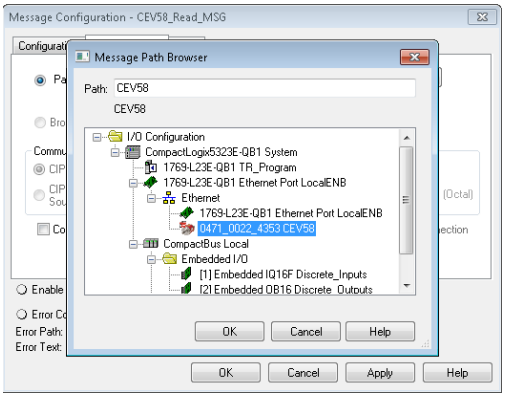
- Now the MSG instruction must be configured. Start with the CEV58_Read_MSG. Select the "…" button:




  - Enter the following information (Destination Element reads CEV58.Read.ReadData[0])
    - Destination element is critical. If a different value is entered the MSG instruction may still yield a successful result, which will set the AOI's success output even though data could be misallocated.

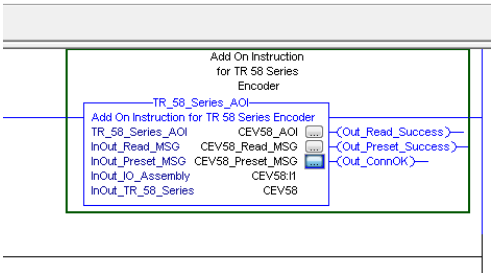


- Then select the Communication tab to configure the path. Click Browse and navigate to the new module:

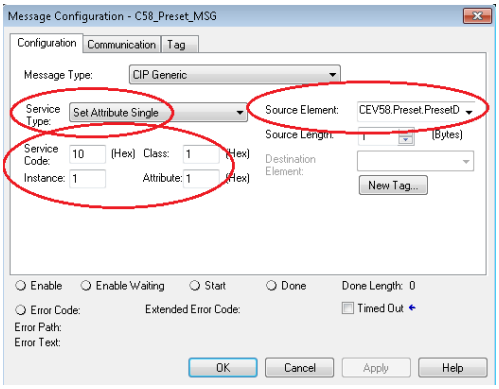



  - Select OK on the Message Path Browser to close the dialog box.
  - Select OK to close the Message Configuration dialog box.

- Then the CEV58_Preset_MSG message must be configured. Click on the "…" button:




  - Enter the following information (Source Element reads CEV58.Preset.PresetData[0]).
    - Source element is critical. See notes regarding destination element for the read message above.




  - Configure the communication path as shown for the read message.
  - Click OK to close the Message Configuration dialog box.

The module and AOI are now configured and ready to use. The program can be downloaded to the controller and set to run mode if appropriate.

## Using the AOI

Importing the AOI also introduced a new UDT (User-Defined Data Type) called TR_58_Series. Refer to Appendix TR_58_Series UDT for detailed information about the data types. The UDT and AOI work together to perform 3 main tasks:

1. Parse encoder feedback into an understandable and usable format. This parsing takes place every scan.
    a. Position feedback: CEV58.Position, _LSW and _MSW.
    b. Velocity feedback: CEV58. Velocity
2. Read important encoder parameters: CEV58.Read.XXXXXXXX. **This is done so that a user can validate the settings read from the encoder against their intended configuration.**
    a. Automatic:
        i. Two seconds after a successful re-connection between PLC and encoder.
    b. On Demand: CEV58.Read.Command
        i. Read begins upon positive transition of trigger bit. This bit will be automatically cleared when the read is successful or a read failure is detected.
    c. Outputs: CEV58_AOI.Out_Read_Success
        i. Cleared when read initiated or connection between encoder and PLC is lost.
        ii. Set upon successful read.
        iii. Once the AOI has been setup and a successful read has occurred, there should be no read failures in the future (assuming the encoder is powered and connected). However if you want to monitor explicitly for read failures you may use CEV58_Read_MSG.ER.
3. Perform a preset: CEV58.Preset.Preset, _LSW and _MSW
    a. On Demand: CEV58.Preset.Command
        i. Preset operation begins upon positive transition of trigger bit, so it is recommended that this bit is set using a one-shot (ONS instruction). This bit will be automatically cleared when the preset operation is successful or an error is detected.
        ii. Once the AOI has been setup and a successful preset has occurred, there should be no preset failures in the future (assuming the encoder is powered and connected). However if you want to monitor explicitly for preset failures you may use CEV58_Preset_MSG.ER.
    b. Outputs: CEV58_AOI.Out_Preset_Success
        i. Cleared on first scan or when preset operation is initiated.
        ii. Set upon successful preset operation.

The AOI has an additional output called CEV58_AOI.ConnOK. This bit is set to true when the PLC and encoder have been connected successfully for two seconds. It is cleared when a communication fault is detected.

AOI output usage is at the programmer discretion. They can be used for in logic, but do not need to be.

# Appendix

## TR_58_Series UDT

| Member Variable | | Data Type | Usage | Description |
|---|---|---|---|---|
| Position_LSW | | DINT (32 bits) | Read Only (CEV58.Position_LSW) | IO Assembly from Device - Position value least significant word |
| Position_MSW | | DINT (32 bits) | Read Only (CEV58.Position_MSW) | IO Assembly from Device - Position value most significant word |
| Position | | LINT (64 bits)* | Read Only (CEV58.Position) | IO Assembly from Device - Position value |
| Velocity | | DINT (32 bits) | Read Only (CEV58.Velocity) | IO Assembly from Device - Velocity value |
| Read | PositionSensorType | SINT (8 bits) | Read Only (CEV58.Read.PositionSensorType) | Read From Device - Attribute: 11 |
| | Direction | BOOL (1 bit) | Read Only (CEV58.Read.Direction) | Read From Device - Attribute: 12 |
| | MeasuringUnitsPerSpan | DINT (32 bits) | Read Only (CEV58.Read.MeasuringUnitsPerSpan) | Read From Device - Attribute: 16 |
| | VelocityFormat | DINT (32 bits) | Read Only (CEV58.Read.VelocityFormat) | Read From Device - Attribute: 25 |
| | TotalMeasuringRange_LSW | DINT (32 bits) | Read Only (CEV58.Read.TotalMeasuringRange_LSW) | Read From Device - Attribute: 101 -LSW |
| | TotalMeasuringRange_MSW | DINT (32 bits) | Read Only (CEV58.Read.TotalMeasuringRange_MSW) | Read From Device - Attribute: 101 -MSW |
| | TotalMeasuringRange | LINT (64 bits)* | Read Only (CEV58.Read.TotalMeasuringRange) | Read From Device - Attribute: 101 |
| | NumberOfRevolutionsNumerator | DINT (32 bits) | Read Only (CEV58.Read.NumberOfRevolutionsNumerator) | Read From Device - Attribute: 102 |
| | NumberOfRevolutionsDivsisor | DINT (32 bits) | Read Only (CEV58.Read.NumberOfRevolutionsDivsisor) | Read From Device - Attribute: 103 |
| | TRParameter | BOOL (1 bit) | Read Only (CEV58.Read.TRParameter) | Read From Device - Attribute: 105 |
| | Alarms | INT (16 bits) | Read Only (CEV58.Read.Alarms) | Read From Device - Attribute: 44 |
| | Warnings | INT (16 bits) | Read Only (CEV58.Read.Warnings) | Read From Device - Attribute: 47 |
| | EncoderFirmwareNumber | STRING | Read Only (CEV58.Read.EncoderFirmwareNumber) | Read From Device - Attribute: 113 |
| | EncoderFirmwareVersion | STRING | Read Only (CEV58.Read.EncoderFirmwareVersion) | Read From Device - Attribute: 114 |
| | ReadData | SINT[8] (8x8 bits) | N/A | Internal working variables - do not use. |
| | Command | BOOL (1 bit) | Read/Write (CEV58.Read.Command) | Read Params Trigger |
| Preset | Preset_LSW | DINT (32 bits) | Write Only (CEV58.Preset.Preset_LSW) | Write To Device - Attribute: 104 - LSW |
| | Preset_MSW | DINT (32 bits) | Write Only (CEV58.Preset.Preset_MSW) | Write To Device - Attribute: 104 - MSW |
| | Preset | LINT (64 bits)* | Read Only (CEV58.Preset.Preset) | Write To Device - Attribute: 104 |
| | PresetData | SINT[8] (8x8 bits) | N/A | Internal working variables - do not use. |
| | Command | BOOL (1 bit) | Read/Write (CEV58.Preset.Command) | Preset Operation Trigger |

**Note:** If TotalMeasuringRange value is less than 32 bits, then only the _LSW variables need to be considered in logic.
**Note:** Attribute numbers refer to Object 0x23 Position Sensor Object. Definitions can be found in the "Absolute Encoder C__-58".
***Note:** All of the LINT datatypes above can be found broken into two (LSW and MSW) DINTs. This is because the built in RSLogix instructions compatible with LINTs are extremely limited. The LINTs have been provided for visual aid, but all mathematic operations and comparisons must be performed on the DINTs without a special library installed. The LINTS are all READ ONLY and will be overwritten by corresponding LINTS.

## Configuration Assembly

| | Default (Hex) | Default (Decimal) | Default (Decimal) | Possibilities | Description | When TR Parameter = 0 | When TR Parameter = 1 |
|---|---|---|---|---|---|---|---|
| CEV58:C.Data[0] | 00 | 0 | 0 | 0 or 1 | Direction Toggle – Attribute: 12 | | |
| CEV58:C.Data[1] | 00 | 0 | 4096 | CEX-58 (1-32 768)<br>COX-58 (1-262 144) | Measuring Units Per Span<br>Attribute: 16 | Used For Configuration | Configuration Value Ignored - Leave As Default<br>Attribute Value is Calculated |
| CEV58:C.Data[2] | 10 | 16 | | | | | |
| CEV58:C.Data[3] | 00 | 0 | | | | | |
| CEV58:C.Data[4] | 00 | 0 | | | | | |
| CEV58:C.Data[5] | 00 | 0 | 16777216 | 16 - 4294967295 | Total Measuring Range (<32 Bit)<br>Attribute: 17 | Used For Configuration | Configuration Value Ignored - Leave As Default<br>Attribute Value is Calculated |
| CEV58:C.Data[6] | 00 | 0 | | | | | |
| CEV58:C.Data[7] | 00 | 0 | | | | | |
| CEV58:C.Data[8] | 01 | 1 | | | | | |
| CEV58:C.Data[9] | 0F | 15 | 7951 | 7940 (steps/s), 7941 (steps/ms),<br>7950 (rev/s), 7951 (rev/min) | Velocity Format<br>Attribute: 25 | | |
| CEV58:C.Data[10] | 1F | 31 | | | | | |
| CEV58:C.Data[11] | 00 | 0 | 16777216 | CEX-58 (16-8 388 608 000)<br>COX-58 (16-67 108 864 000) | Total Measuring Range (<64 Bit)<br>Attribute: 101 | Configuration Value Ignored - Leave As Default<br>Attribute Value is Calculated | Used For Configuration |
| CEV58:C.Data[12] | 00 | 0 | | | | | |
| CEV58:C.Data[13] | 00 | 0 | | | | | |
| CEV58:C.Data[14] | 01 | 1 | | | | | |
| CEV58:C.Data[15] | 00 | 0 | | | | | |
| CEV58:C.Data[16] | 00 | 0 | | | | | |
| CEV58:C.Data[17] | 00 | 0 | | | | | |
| CEV58:C.Data[18] | 00 | 0 | | | | | |
| CEV58:C.Data[19] | 00 | 0 | 4096 | 1-256000 | Number Of Revolutions – Numerator<br>Attribute: 102 | Configuration Value Ignored - Leave As Default<br>Attribute Value is Calculated | Used For Configuration |
| CEV58:C.Data[20] | 10 | 16 | | | | | |
| CEV58:C.Data[21] | 00 | 0 | | | | | |
| CEV58:C.Data[22] | 00 | 0 | | | | | |
| CEV58:C.Data[23] | 01 | 1 | 1 | 1-16384 | Number Of Revolutions – Divisor<br>Attribute: 103 | Configuration Value Ignored - Leave As Default<br>Attribute Value is Calculated | Used For Configuration - Recommend to leave at 1 |
| CEV58:C.Data[24] | 00 | 0 | | | | | |
| CEV58:C.Data[25] | 00 | 0 | | | | | |
| CEV58:C.Data[26] | 00 | 0 | | | | | |
| CEV58:C.Data[27] | 00 | 0 | 0 | 0 or 1 - **Recommended to set to 1** | TR – Parameter - Attribute: 105 | | |
| CEV58:C.Data[28] | 00 | 0 | 0 | 0 | Reserved | | |
| CEV58:C.Data[29] | 00 | 0 | | | | | |
| CEV58:C.Data[30] | 00 | 0 | | | | | |
| CEV58:C.Data[31] | 00 | 0 | | | | | |

**NOTE:** The offline PLC file should be updated with the new values to ensure old values aren't re-loaded upon re-download. This can be done by updating the offline file (recommended) or uploading tag values upon saving the program.

**NOTE:** Detailed information regarding TR Parameter can be found in Appendix TR Parameter

## Configuration Assembly Mapping Example

| | Desired Value (Decimal) | Desired Value (Hex) | Desired Value (Hex) | Desired Value (Singed Decimal)* |
|---|---|---|---|---|
| CEV58:C.Data[0] | 0 | 0 | 00 | 0 |
| CEV58:C.Data[1] | Left as default - 4096 | Left as default – 10 00 | 00 | 0 |
| CEV58:C.Data[2] | | | 10 | 16 |
| CEV58:C.Data[3] | | | 00 | 0 |
| CEV58:C.Data[4] | | | 00 | 0 |
| CEV58:C.Data[5] | Left as default - 16777216 | Left as default – 1 00 00 00 | 00 | 0 |
| CEV58:C.Data[6] | | | 00 | 0 |
| CEV58:C.Data[7] | | | 01 | 1 |
| CEV58:C.Data[8] | | | 00 | 0 |
| CEV58:C.Data[9] | 7940 | 1F04 | 04 | 4 |
| CEV58:C.Data[10] | | | 1F | 31 |
| CEV58:C.Data[11] | 18 000 000 000 | 4 30E2 3400 | 00 | 0 |
| CEV58:C.Data[12] | | | 34 | 52 |
| CEV58:C.Data[13] | | | E2 | -30 (2's complement of 226)** |
| CEV58:C.Data[14] | | | 30 | 48 |
| CEV58:C.Data[15] | | | 04 | 4 |
| CEV58:C.Data[16] | | | 00 | 0 |
| CEV58:C.Data[17] | | | 00 | 0 |
| CEV58:C.Data[18] | | | 00 | 0 |
| CEV58:C.Data[19] | 8192 | 20 00 | 00 | 0 |
| CEV58:C.Data[20] | | | 20 | 32 |
| CEV58:C.Data[21] | | | 00 | 0 |
| CEV58:C.Data[22] | | | 00 | 0 |
| CEV58:C.Data[23] | 1 | 01 | 01 | 1 |
| CEV58:C.Data[24] | | | 00 | 0 |
| CEV58:C.Data[25] | | | 00 | 0 |
| CEV58:C.Data[26] | | | 00 | 0 |
| CEV58:C.Data[27] | 1 | 01 | 01 | 1 |
| CEV58:C.Data[28] | 0 | 00 | 00 | 0 |
| CEV58:C.Data[29] | | | 00 | 0 |
| CEV58:C.Data[30] | | | 00 | 0 |
| CEV58:C.Data[31] | | | 00 | 0 |

***Note:** Conversion from Hex bytes to decimal bytes is not required if the style was changed to Hex, as recommended.

****Note:** The 2's complement must be taken in this case. Please review Appendix Unsigned Vs Signed.

## Unsigned Vs Signed

RSLogix uses signed integers (SINTs, DINTs) which can catch some users off guard. This is general RSLogix behaviour and can't be influenced by an AOI, so this appendix has been included to help the user's understanding.

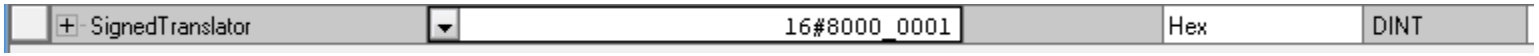
When the most significant bit of the integer is high, RSLogix will interpret and display this number as a negative. Using SINT as an example:

| Binary | RSLogix Value |
|---|---|
| 00000001 | 1 |
| 00000011 | 3 |
| 01111111 | 127 |
| 10000000 | -128 (2's complement of 128) |
| 10000001 | -127 (2's complement of 129) |
| 10000011 | -125 (2's complement of 131) |
| 11111111 | -1 (2's complement of 255) |

In the example in Appendix Configuration Assembly Mapping Example, the intended value is Hex E2 which translates using a calculator to 226. RSLogix will not actually allow a user to enter 226 into that field, because it is too large to fit in an 8 bit **signed** field. 226 in binary is 11100010. The most significant bit is high. Therefore one would have to use the 2's complement of 226 which is -30 (2's compliment help can be found online, and in the example below).

This also applies to DINTs which can be important to understand. For example, if your total measuring range is such that the most significant bit of the Position_LSW could go high (equal to or greater then 2, 147, 483, 648) then eventually your reading could go negative. At reading 2, 147, 483, 647 you'd expect the next reading (if increasing) to be 2, 147, 483, 648, but instead it would be -2, 147, 483, 648 (2's complement of 2, 147, 483, 648). The next reading you'd expect to be 2, 147, 483, 649 which would actually be -2 147 483 647. This is important if you have any logic which depends on readings over 2, 147, 483, 647.

An easy was to calculate the 2's complement of a number in RSLogix using an available tag:
Translate the number (2, 147, 483, 649) to Hex using a calculator.

- Change the Style of a DINT tag in RSLogix to Hex.
- Enter the number into the value column of that tag, in Hex (80 00 00 01).
  - 

- Then change the Style back to Decimal to reveal the 2's complement.
  - 

## TR Parameter

If TR = 0:

- Limited to 32 bits
- Configuration Parameter Inputs:
  - Total Measuring Range (<32 Bit)
  - Measuring Units Per Span
- Equation:
  - Number of Revolutions – Numerator (Attr.-ID 102) and Number of Revolutions – Divisor (Attr.-ID 103) will be overwritten in the encoder with a ratio that most closely satisfies:

$$\left(\frac{Number\ Of\ Revolutions -\ Numerator}{Number\ Of\ Revolutions -\ Divisor}\right) = Measuring\ Units\ Per\ Span * Total\ Measuring\ Range\ (< 32\ Bit)$$

- Configuration Parameters Ignored:
  - Total Measuring Range (<64 Bit)
  - Number Of Revolutions – Numerator
  - Number Of Revolutions – Divisor

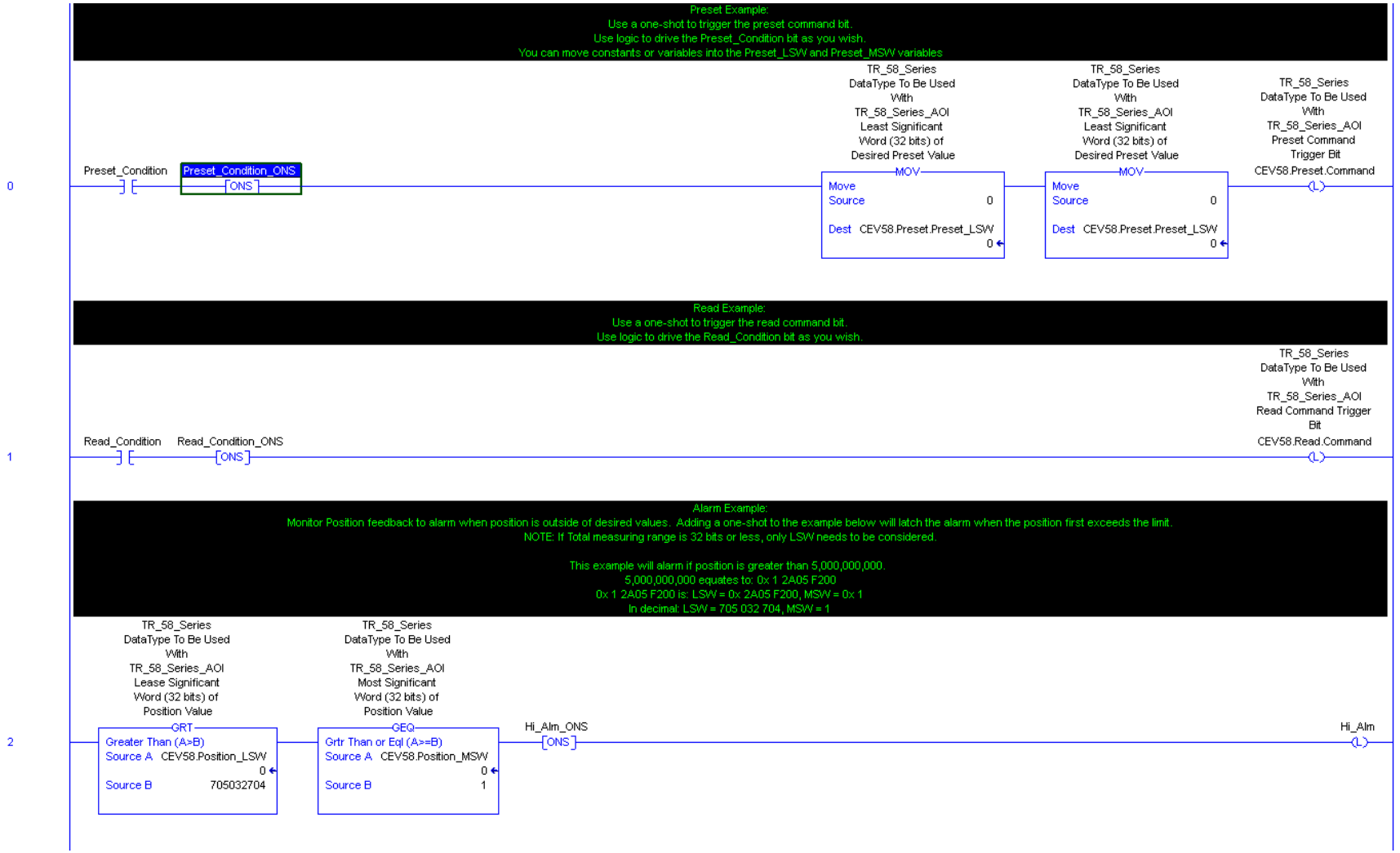
If TR = 1:

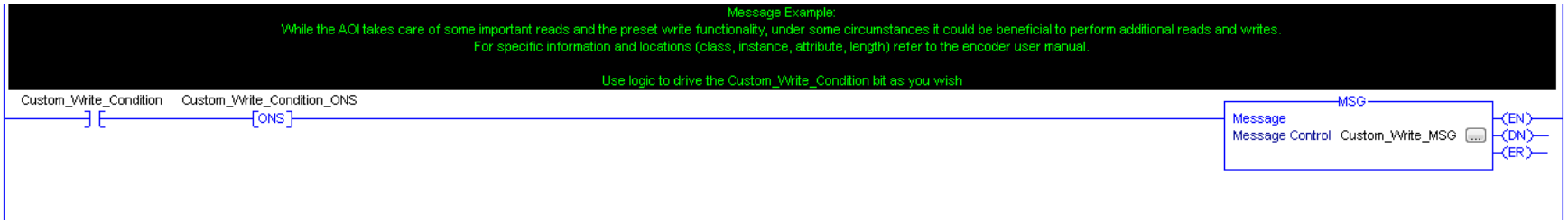
- Limited to 64 bits
- Configuration Parameter Inputs:
  - Total Measuring Range (<64 Bit)
  - Number Of Revolutions – Numerator
  - Number Of Revolutions - Divisor
- Equation:
  - Measuring Units Per Span attribute (Attr.-ID 16) will be overwritten in the encoder:

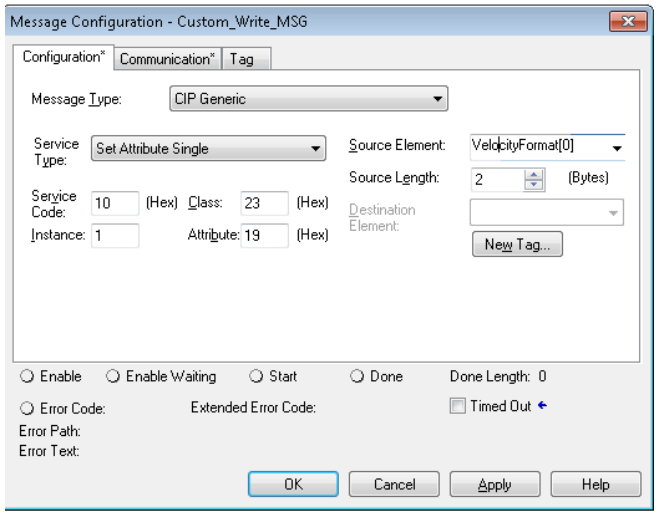
$$Measuring\ Units\ Per\ Span = \left(\frac{Number\ Of\ Revolutions -\ Numerator}{Number\ Of\ Revolutions -\ Divisor}\right) * Total\ Measuring\ Range\ (< 64\ Bit)$$

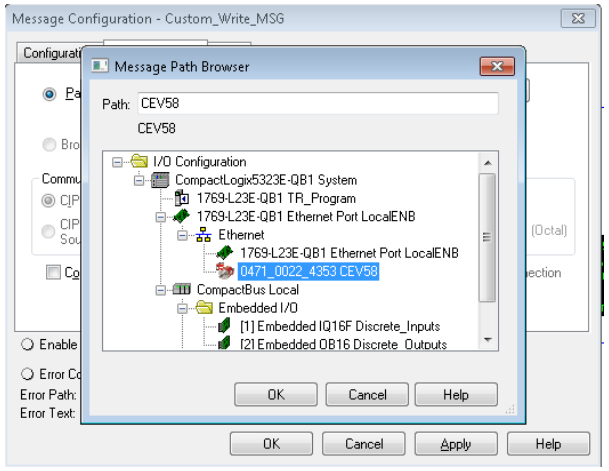
- Configuration Parameters Ignored:
  - Total Measuring Range (<32 Bit)
  - Measuring Units Per Span

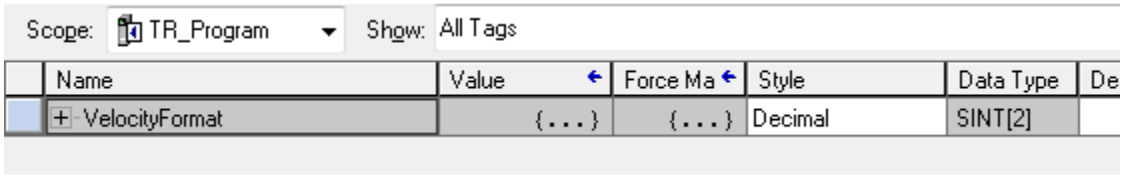
## Logic Examples




Message Example:
While the AOI takes care of some important reads and the preset write functionality, under some circumstances it could be beneficial to perform additional reads and writes.
For specific information and locations (class, instance, attribute, length) refer to the encoder user manual.
Use logic to drive the Custom_Write_Condition bit as you wish
3
Custom_Write_Condition
Custom_Write_Condition_ONS
ONS
MSG
Message
Message Control Custom_Write_MSG
EN
DN
ER


Message Configuration - Custom_Write_MSG
Configuration
Communication
Tag
Message Type: CIP Generic
Service Type: Set Attribute Single
Source Element: VelocityFormat[0]
Source Length: 2 (Bytes)
Service Code: 10 (Hex) Class: 23 (Hex)
Instance: 1 Attribute: 19 (Hex)
Destination Element:
New Tag...
Enable
Enable Waiting
Start
Done
Done Length: 0
Error Code:
Extended Error Code:
Timed Out
Error Path:
Error Text:
OK
Cancel
Apply
Help


Message Configuration - Custom_Write_MSG
Message Path Browser
Path: CEV58
CEV58
I/O Configuration
CompactLogix5323E-QB1 System
1769-L23E-QB1 TR_Program
1769-L23E-QB1 Ethernet Port LocalENB
Ethernet
1769-L23E-QB1 Ethernet Port LocalENB
0471_0022_4353 CEV58
CompactBus Local
Embedded I/O
[1] Embedded IQ16F Discrete_Inputs
[2] Embedded OB16 Discrete_Outputs
OK
Cancel
Help
Enable
Error Path:
Error Text:
OK
Cancel
Apply
Help


Scope: TR_Program
Show: All Tags
Name
Value
Force Ma
Style
Data Type
VelocityFormat
{...}
{...}
Decimal
SINT[2]