# AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs

Document Revision V1.00 • 2018-Aug-07

**This application note demonstrates how we can easily turn a standard incremental encoder into an EtherCAT slave with TRINAMIC's EtherCAT Slave Controller Chips. No software stack is required for that. Only configuration of the controller chip.**

## Contents



## 1 Introduction

Encoders of different types (optical, magnetic, and capacitive) with incremental/quadrature outputs are widely used in motor control, robotic applications, and automation. So far, quadrature encoders are very often used directly at a motors back-bell and near to the motor drive/controller unit. With the success and ongoing adoption of distributed control systems and industrial strength (real-time) communication buses demands for networked encoder systems are increasing.

EtherCAT is the industrial real-time Ethernet bus with the best performance parameters, biggest market growth, and fastest growing community. It relies on the standard Ethernet physical layer and requires dedicated interface ICs – so-called EtherCAT Slave Controllers (ESC) – to handle the datalink layer EtherCAT protocol and its mechanisms.

As one of just a hand full of companies TRINAMIC offers suchlike ESC solutions. Besides their general applicability for all kinds of EtherCAT slave applications TRINAMIC's ESCs are specialized for motor and motion control applications.

This application note is a step by step guide on how to transform a standard quadrature encoder into a simple EtherCAT slave using TRINAMIC's TMC8461/TMC8462 ESC without any software coding and minimal hardware requirements.



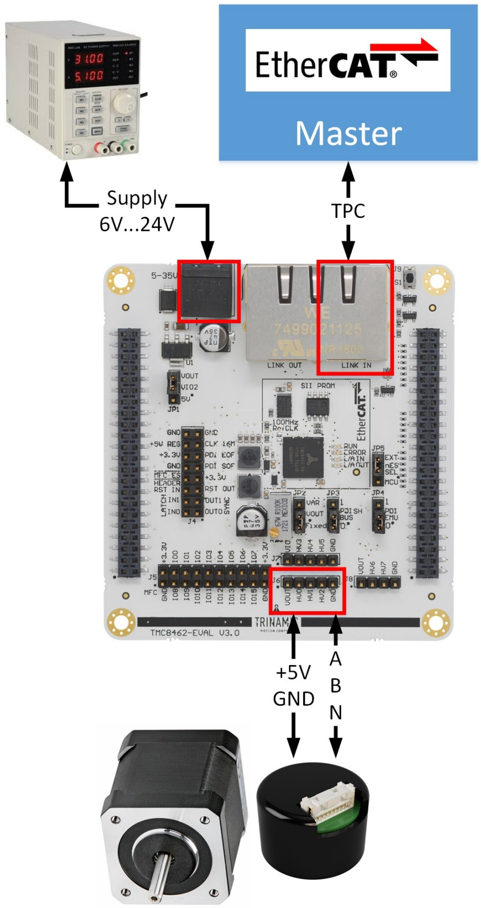
TRINAMIC
MOTION CONTROL

# 2 The Hardware Setup – Components

The hardware components used to demonstrate this are listed below. The URLs point to additional information for each component.

- TMC8462 EtherCAT Slave Controller chip.
  https://www.trinamic.com/products/integrated-circuits/details/tmc8462-ba/
- TMC8462 evaluation board
  https://www.trinamic.com/support/eval-kits/details/tmc8462-eval/
- TMCS-28 optical encoder kit
  https://www.trinamic.com/products/drives/encoder-details/tmcs-28/
- NEMA17 stepper motor equipped with the TMCS-28
- Desktop power supply with up to 24V DC
- From software side TRINAMIC's TMCL-IDE is used for offline configuration of the TMC8462 as well as a Beckhoff TwinCAT EtherCAT master system.
  https://www.trinamic.com/support/software/tmcl-ide/
  http://www.beckhoff.de/twincat/

The following diagram shows the general components and how to connect them in this setup.



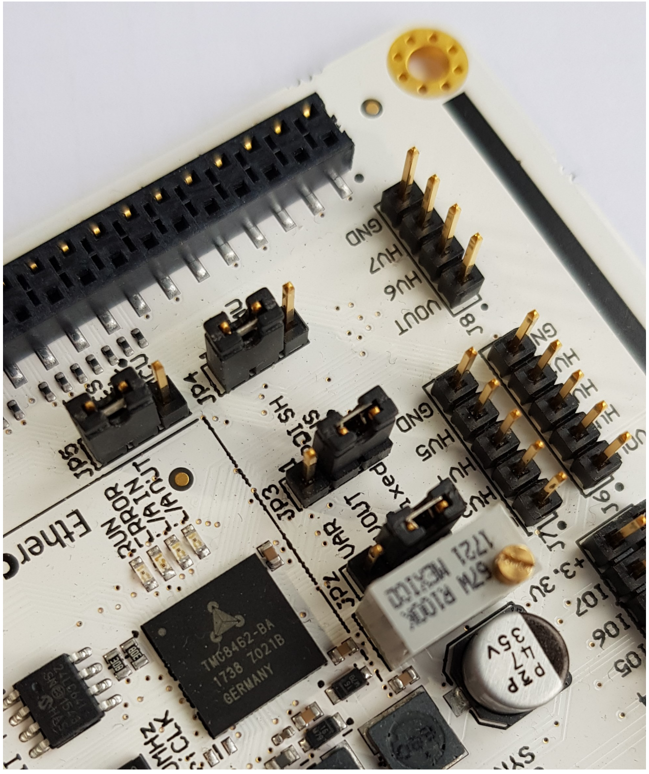
# 3 The Hardware Setup – Jumper Settings

The TMC8462 must be configured on the evaluation board to operate without a dedicated software stack and EtherCAT State Machine in device emulation mode by setting jumper JP4 onto pins 2 and 3 (DEVICE_EMULATION = 1). This way, no MCU and software stack is needed.

The reference voltage for HV IO group 1 on the evaluation board is driven by VOUT, which is the output voltage of the internal adjustable DC/DC buck regulator of TMC8462. Jumper JP2 allows configuration of VOUT. It can be either adjustable with the onboard potentiometer or use a fixed voltage divider to have VOUT = 5V. The latter one is required here for the encoder supply and VIO1 (= IO voltage for HV IO group 1). That is, jumper JP2 must have a jumper bridge on pins 1 and 2.

The other jumpers on the evaluation board are not of interest for this setup.

The following photo shows the required jumper settings for JP2 and JP4 on the board.




# 4 The Hardware Setup – Wiring

The evaluation board for the TMC8462 offers many interface options. Nevertheless, for this example application only the internal MFC IO Encoder Unit, 3 out of the 8 available high voltage IOs, and the internal DC/DC regulators of TMC8462 are actually needed. For a customized board solution this results in very low external part count and keeps the whole circuit small.

The evaluation board is powered from an external supply with 6V...35V. Any voltage in between fits. The TMC8462 can handle that.

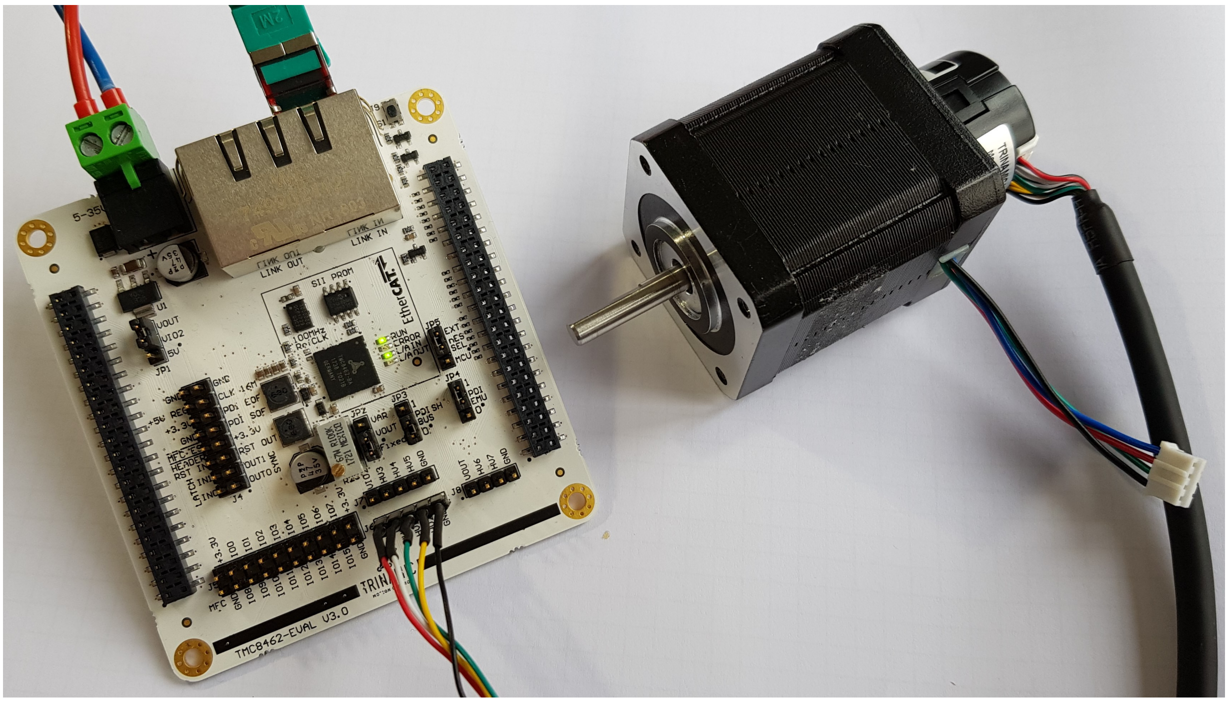
The encoder itself requires 5V DC for operation. This 5V can be provided directly from the adjustable DC/DC buck regulator inside the TMC8462. No additional DC/DC regulator component is required. Also, the encoder's quadrature and index signals, which are typically at 5V level, can directly be connected to the high voltage IO pins of the TMC8462. There is no need for additional level shifting circuits to scale down the external signals to a VCCIO like 3.3V.

The TMCS-28 encoder offers differential outputs for the quadrature signals but we only need the positive signals to be connected as single-ended inputs to the TMC8462 evaluation board.

| Pin # | Cable color | Signal name | Use in this demo | Pin on TMC8462-EVAL |
|---|---|---|---|---|
| 1 | Red | VCC (5V) | Yes | J6, Pin 1 (VOUT) |
| 2 | Black | GND | Yes | J6, Pin 5 (GND) |
| 3 | White | A+ | Yes | J6, Pin 2 (HV0) |
| 4 | White/Black | A- | No | |
| 5 | Green | B+ | Yes | J6, Pin 3 (HV1) |
| 6 | Green/Black | B- | No | |
| 7 | Yellow | Z+ | Yes | J6, Pin 4 (HV2) |
| 8 | Yellow/Black | Z- | No | |
| 9 | Blue or thick black | Shield | No | |

*Table 1: Used encoder signals*

The photo shows how things are connected for this setup. We use a voltage of 10V from our supply.



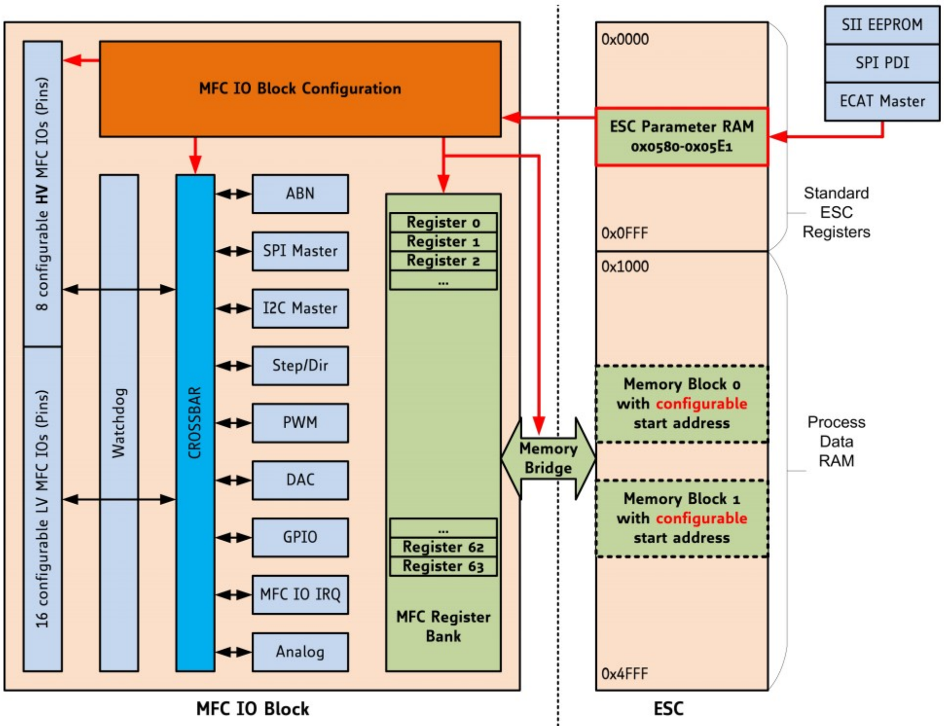
# 5 TMC8462 – Configuration of the MFC IO Functions

After reset and at power up, an EtherCAT Slave Controller requires at least basic configuration data from an external configuration EEPROM to properly configure the communication ports, operation mode, and feature availability – and so does the TMC8462. This EEPROM is called SII EEPROM (Slave Information Interface) and connects via I2C to the TMC8462. Besides the standard configuration parameters for basic operation, the SII EEPROM may also contain additional configuration information for SyncManagers, FMMUs, PDO/SDO data objects as well as additional configuration categories.

An XML file with a defined structure and XML schemata contains the slave configuration in a readable form and can easily be modified by hand. An EtherCAT master like TwinCAT by Beckhoff imports that slave configuration XML file and uploads or updates it into the SII-EEPROM of the slave in binary format.

The power-on configuration of the TMC8462's MFC IO functions is defined in the XML file as a "category 1" data block. It is a configuration vector of multiple bytes. The TMC8462 reads category 1 data from the SII EEPROM and automatically writes this data to its registers at addresses 0x0580-0x05FF (the so-called ESC Parameter RAM). This is the easiest way to configure the TMC8462 EtherCAT slave controller – to include all configuration data inside the SII EEPROM and have it automatically loaded at power-up or reset. Another way is to have these configuration bytes directly written to 0x0580- 0x05FF by the EtherCAT master or an MCU after power-up. The configuration can be changed at any time.

The image above shows the flow of the configuration information in the TMC8462. MFC IO configuration comes from the SII EEPROM at boot-time and is written into the ESC Parameter RAM. This register area directly maps into the MFC IO block where it is used as configuration data for all the functions and sub-blocks – including the ABN block, the crossbar, and the HV IOs we are going to use in this application note.

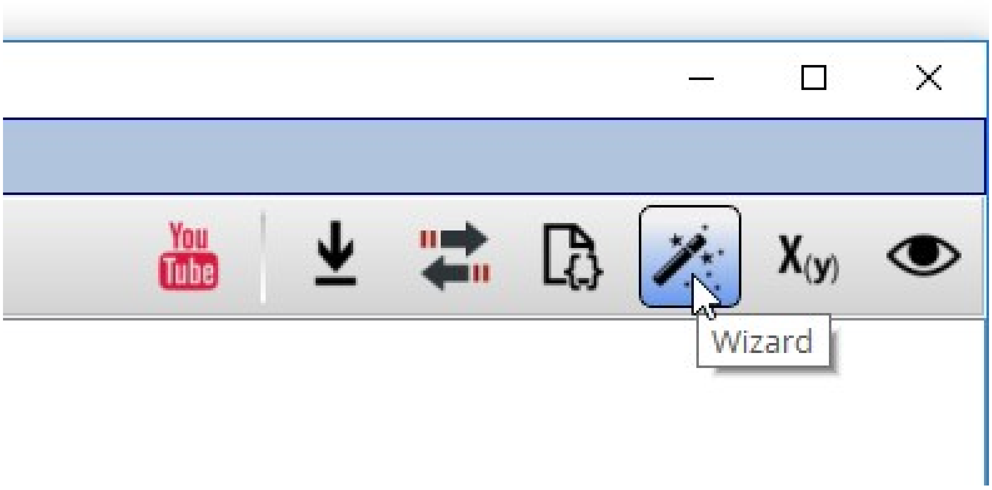


# 6 TMC8462 EtherCAT Slave Configuration Using the Configuration Wizard

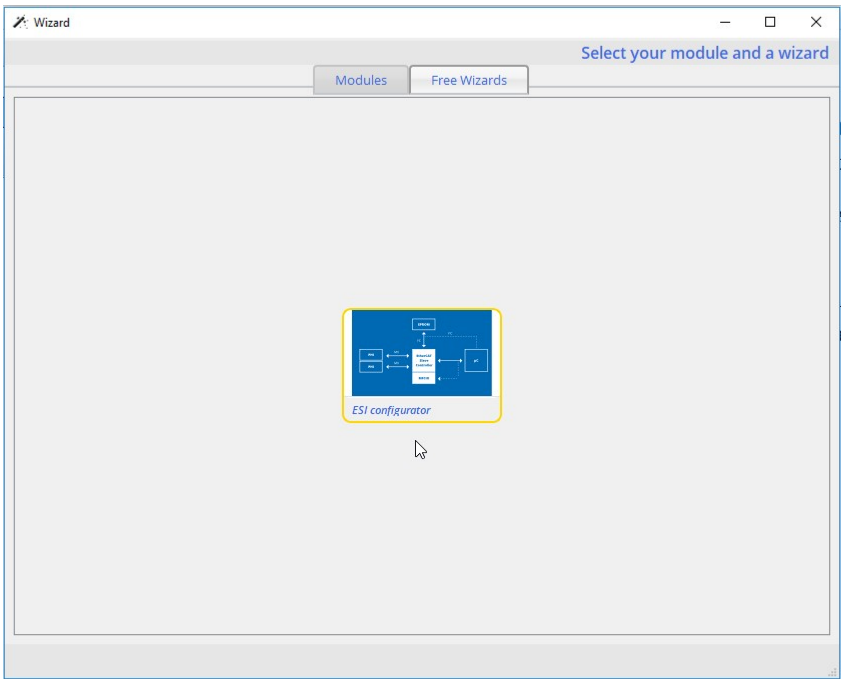
This section discusses how the configuration vector for the MFC IO block is actually defined and where it must be included for proper slave configuration.

(More information and details on the MFC IO block and configuration are available in the TMC8462 datasheet. Find links at the end of this application note.)
TRINAMICs's TMCL IDE, used to control all standard modules and evaluation boards, provides an easy to use offline wizard that allows generating the TMC8462 configuration vector in an easy "click & copy" style. The following steps show how to use that wizard.

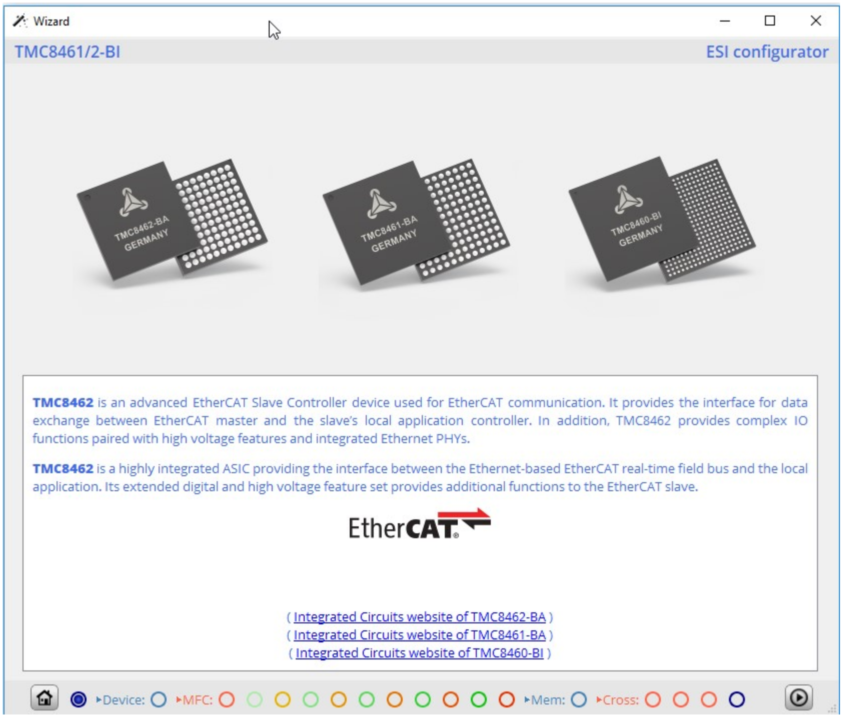
1. Download, install, and open the TMCL-IDE. Find links and URLs at the end of the document.
2. In the upper right button menu, select the wizard stick.



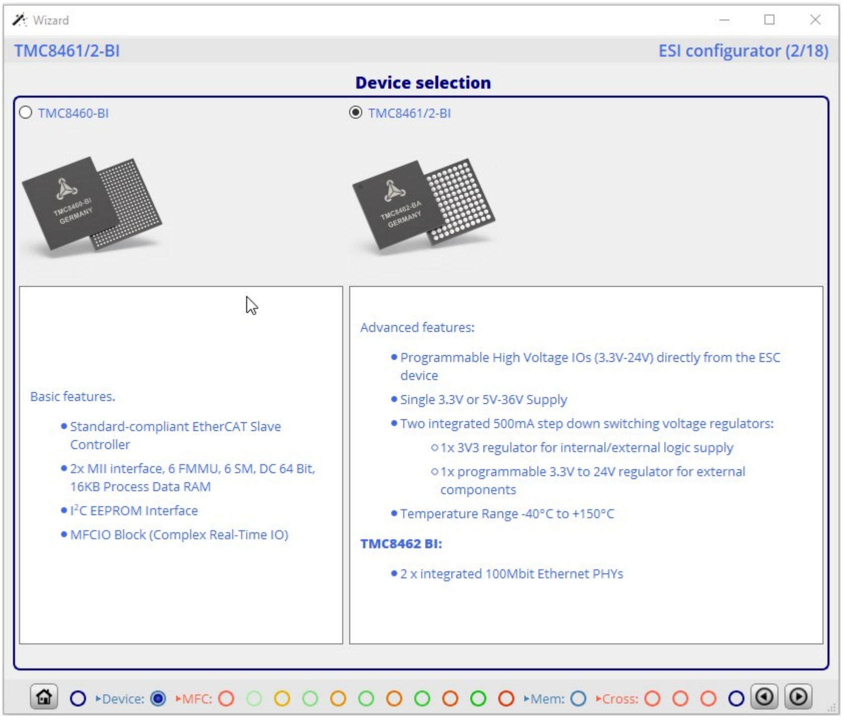
3. The Wizard windows pops up. Select the ESI configurator wizard on the Free Wizards tab.



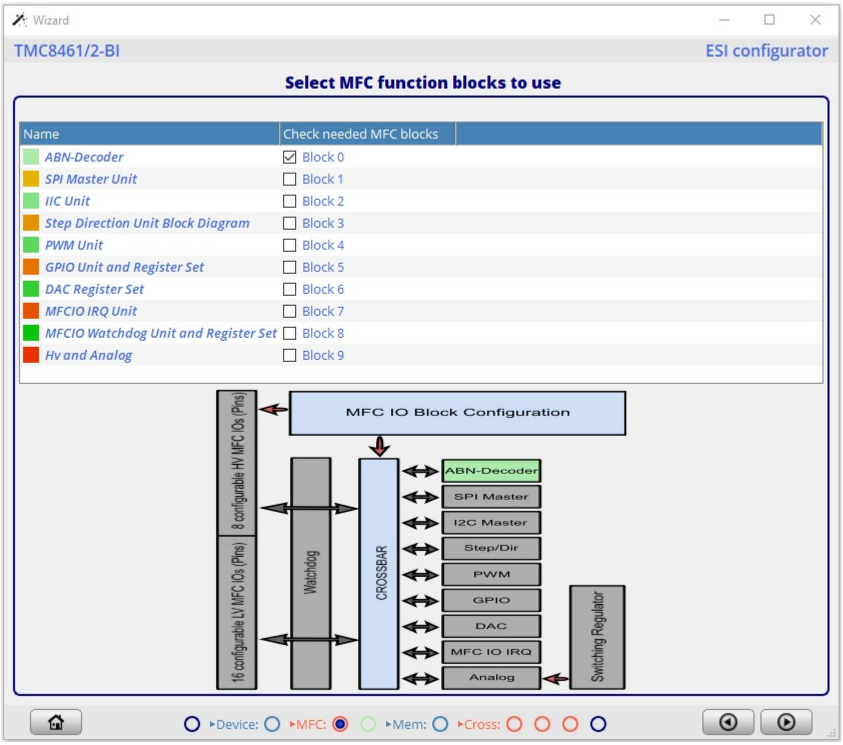

4. The wizard starts with an overview. Simply click on the lower right navigation arrow to go on to the device selection.



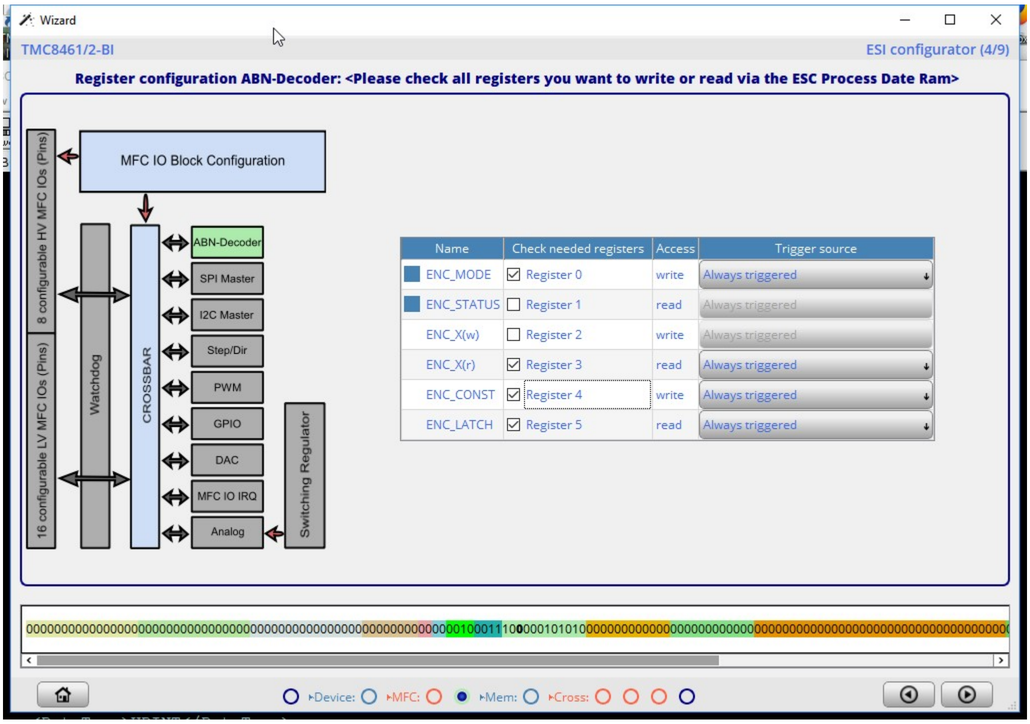
5. In the device selection choose the TMC8461/2 and use the navigation error on the lower right to go on to the next view – MFC IO block selection.




6. By default, all MFC IO blocks are checked in the MFC IO block selection view. Since we only require the ABN decoder block uncheck all others and go on to the next view.



7. This view lets one configure the mapping of the registers of the selected MFC IO blocks into the Process Data RAM (PDRAM) of the EtherCAT Slave Controller TMC8462. The master can only read and write from/to the PDRAM (not from/to the MFC IO registers directly). Select the 4 registers ENC_MODE, ENC_X(r), ENC_CONST, and ENC_LATCH. Keep all trigger sources at "always triggered". Then go on to the next view.

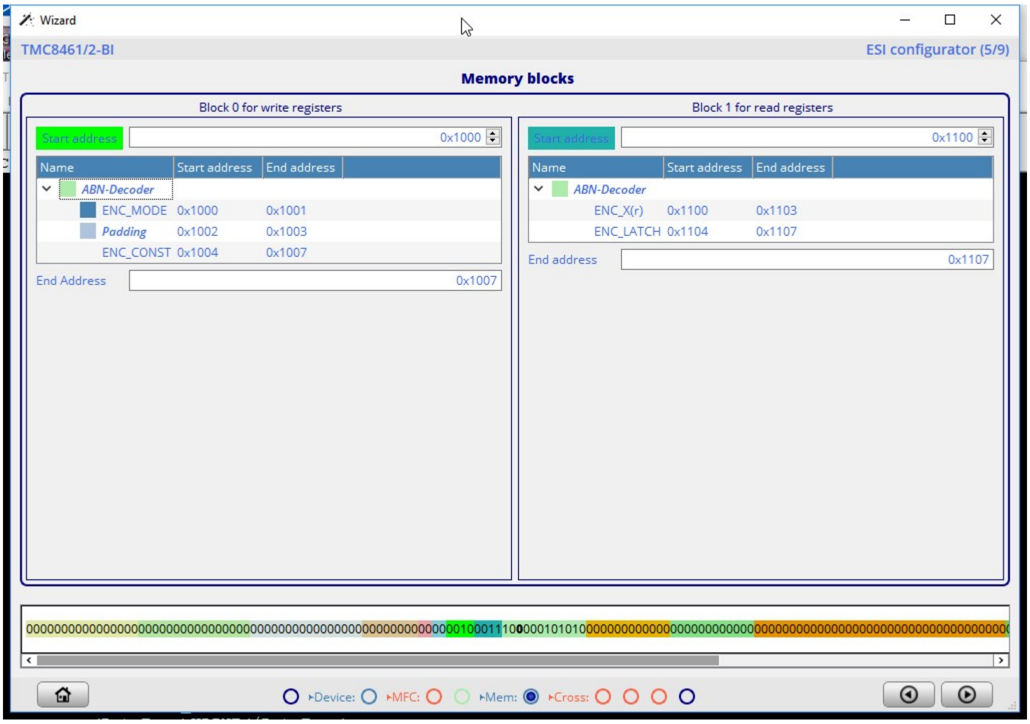



! As a small side note – on the bottom of each view you already see the colored configuration vector, which will be used in the XML file to configure the TMC8462 accordingly.

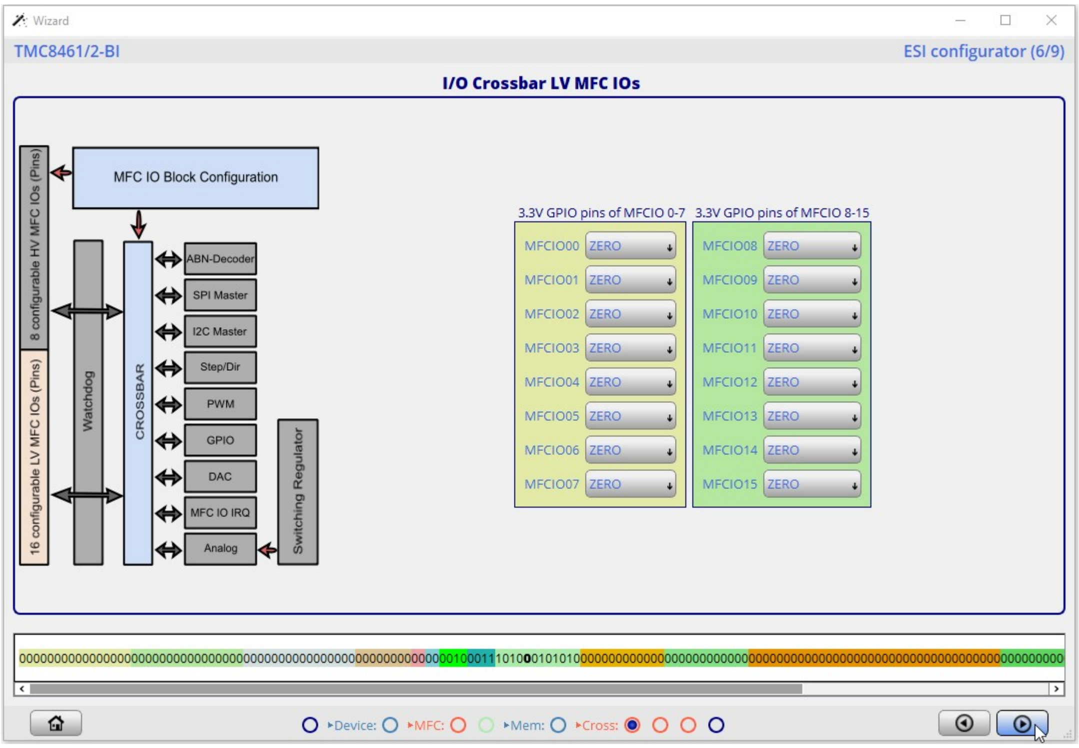
8. The MFC IO registers we selected for PDRAM mapping in the previous step must be mapped to a certain address inside the PDRAM. TMC8462 assigns all selected registers into two (2) defined blocks – one memory block for all read registers and one memory block for all write registers. This view is used to configure the actual start addresses of these two memory blocks, which are called memory block 0 (write registers) and memory block 1 (read registers). Order, lengths, and individual start addresses of each mapped register inside the memory blocks are defined and follow a fixed scheme. The tables in this view of the wizard show this information.

   ! Remember the addresses of these registers as this structure is required when configuring each register as RX and TX PDO for easier read out via the EtherCAT master.

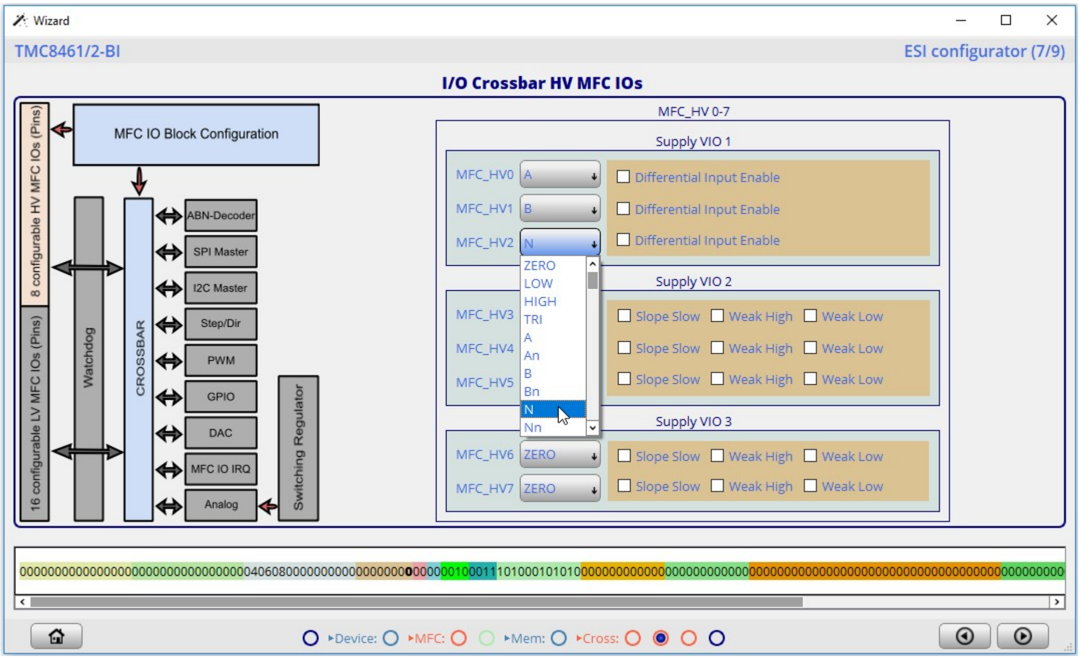
   For now, just keep the standard start addresses of both memory blocks (0x1000 and 0x1100) and go to the next view of the wizard.



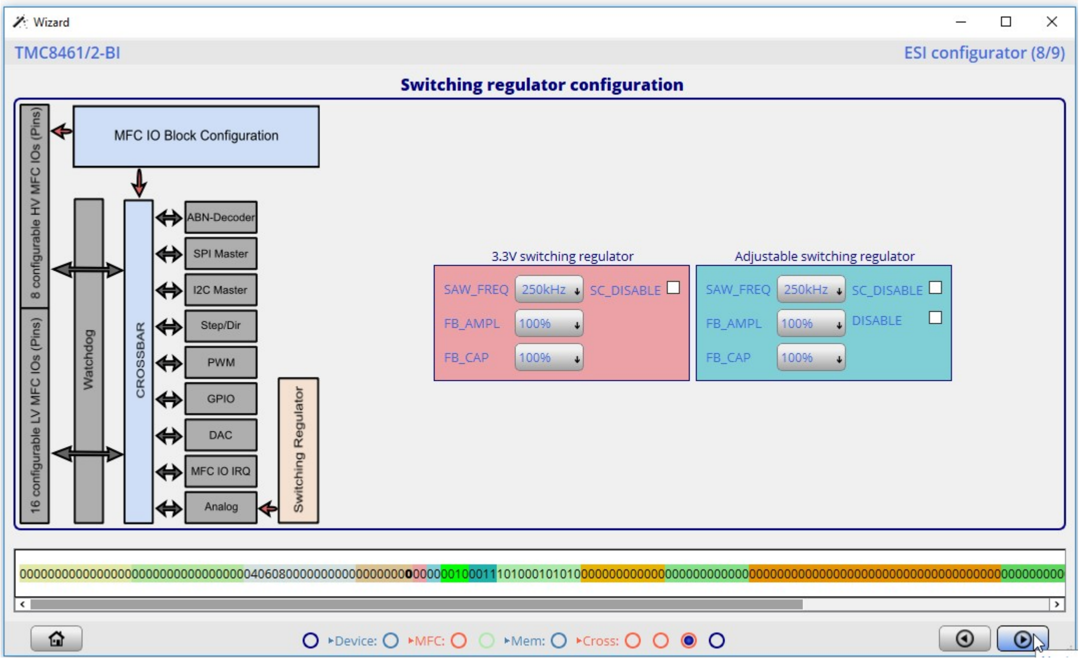

9. The next view configures the signal assignment for the low voltage IOs. We do not need the LV IOs in this example, so simply skip that view and go to the next view for the high voltage IOs.



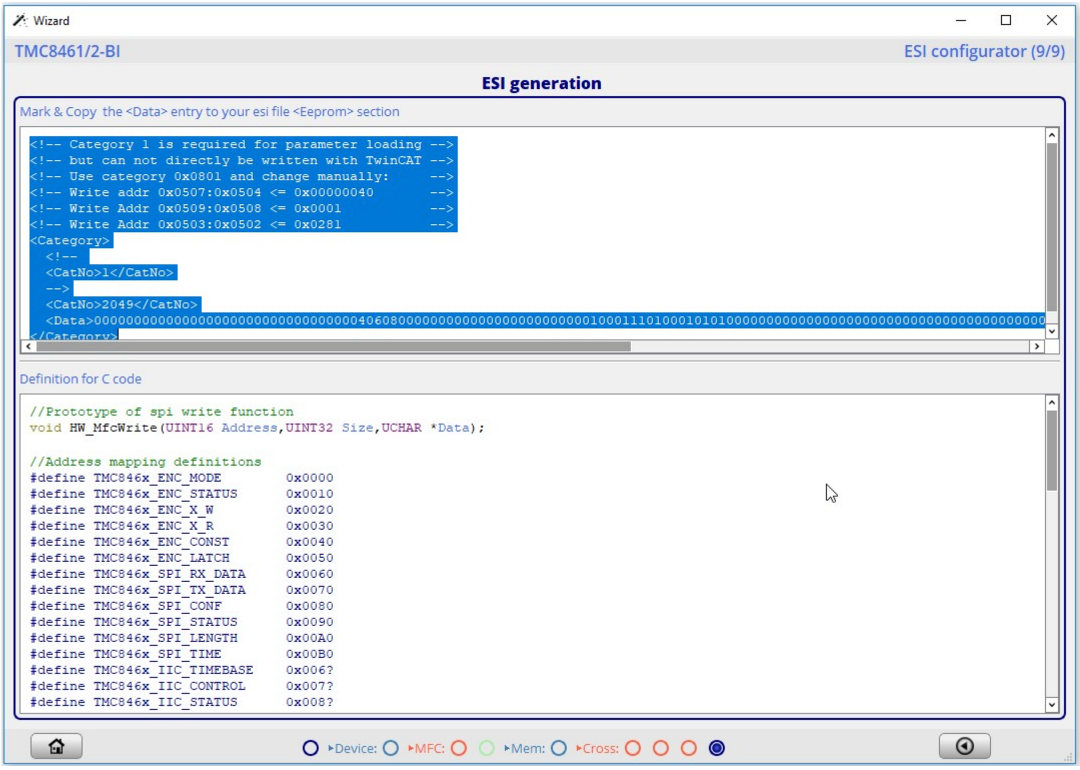
10. Based on the table in the beginning of this application note we assign the quadrature encoder signals to the desired HV IOs. We want to use pins HV0, HV1, and HV2. They all belong to the same pin group and are supplied with VIO 1 (which we configured with 5V already). Simply choose the signals A, B, and N from the drop down select buttons for these 3 high voltage IOs. Do not check the differential option. Then go to the next view of the wizard.




11. The next view can be skipped. Nothing to do here. Go to the final view of the Wizard.



12. The final view of the wizard gives you a textual output and summary of the configuration you have just made. We need the upper part for the XML configuration file. Simply copy that whole block to include it in your XML file.



Now, we continue with the next section.

# 7 TMC8462 EtherCAT Slave Configuration Using TwinCAT

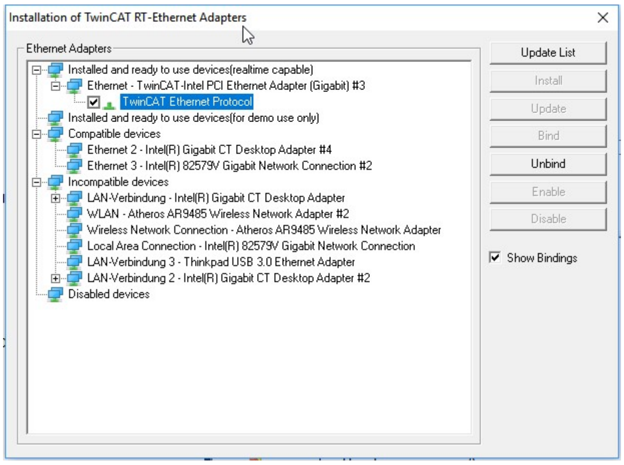
OPTION A: If you are familiar with generating your own XML file simply copy & add the configuration block from the previous section in your XML file into the <Eeprom> part. Add the block into your file as is (with **<CatNo>2049</CatNo>**.

To make configuration in TwinCAT easier, it is suggested to also define the mapped encoder registers as RX and TX PDOs in the XML file. This way they are directly readable/writeable without too much handwork and mouse clicking.

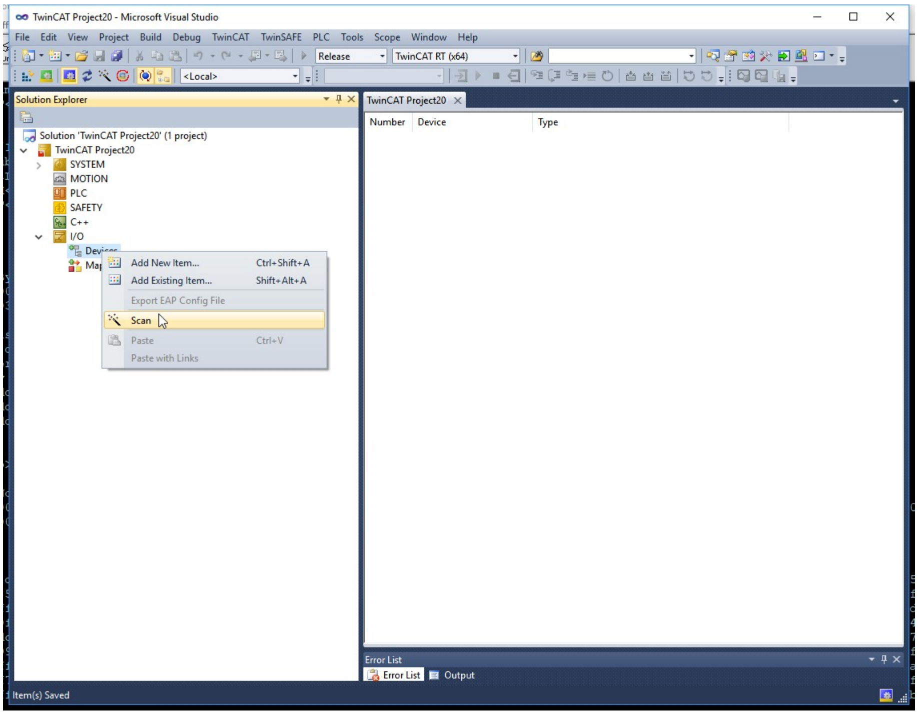
OPTION B: An easier way is to simply download the pre-configured example XML file from TRINAMIC's website:
https://www.trinamic.com/support/eval-kits/details/tmc8462-eval/
The name of the XML file is: **TMC8462-EVAL_DeviceEmulation_ABN_Appnote.xml**

The following steps show how to configure and start the TMC8462-EVAL with our encoder example in the Beckhoff TwinCAT master system:

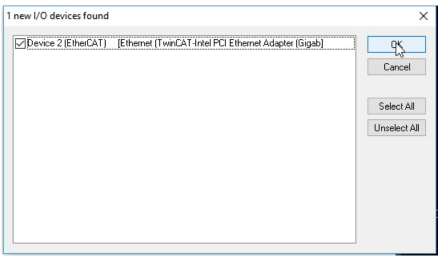
1. Copy the XML file **TMC8462-EVAL_DeviceEmulation_ABN_Appnote.xml** into the device description folder of your TwinCAT installation. This should be located here:
   `<installation drive>:/.../TwinCAT/3.1/Config/Io/EtherCAT/`
2. Prerequisite is that you already have your network adapter properly configured for use with TwinCAT. Check the proper bindings as well.



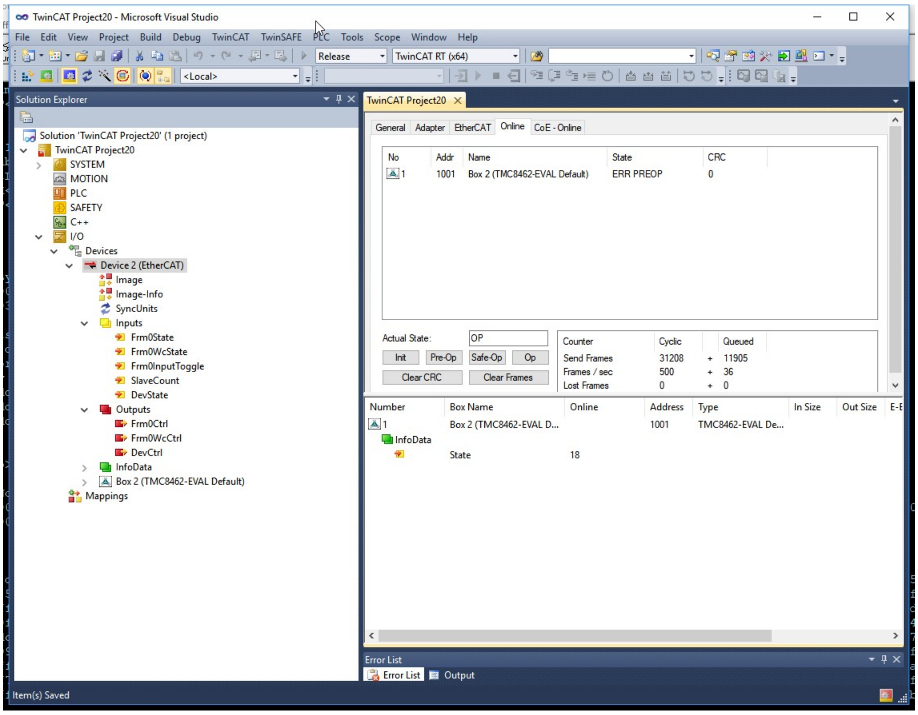
3. Open and start a plain new TwinCAT project. Right-click on "Devices" and select "Scan".



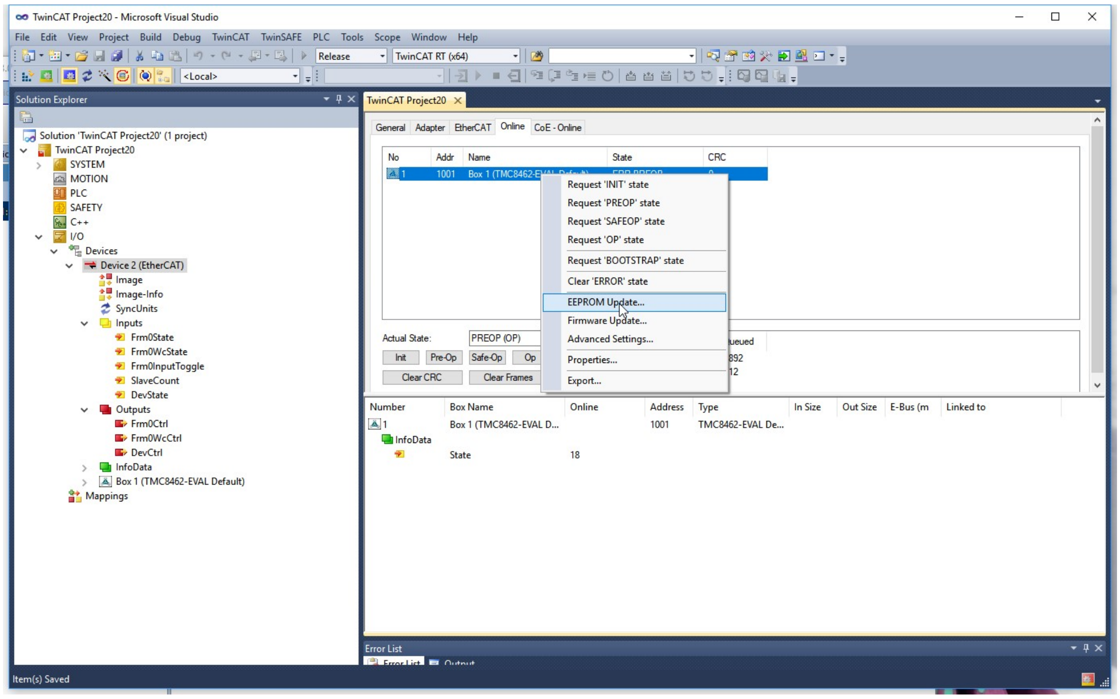

4. The scan should finish and show your EtherCAT capable interfaces. Select your interface and click "OK". Also scan for boxes (= slaves). If you are asked to activate free run mode clock "Yes".



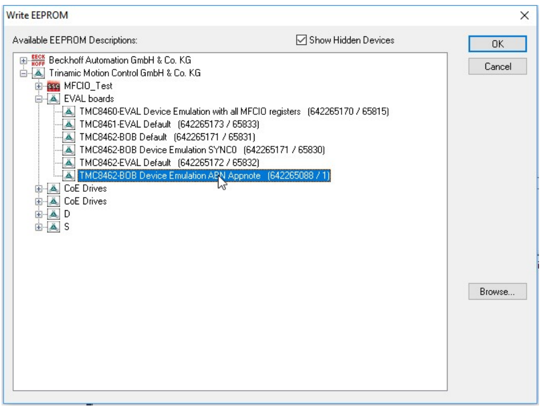
5. Now you should see the TMC8462-EVAL as Box 1 connected to your EtherCAT adapter. If it is a fresh TMC8462-EVAL the name of Box 1 might still be "TMC8462-EVAL Default", which is the default XML file and naming when the board is shipped. If you already programmed a different configuration into the SII EEPROM it might have a different name.



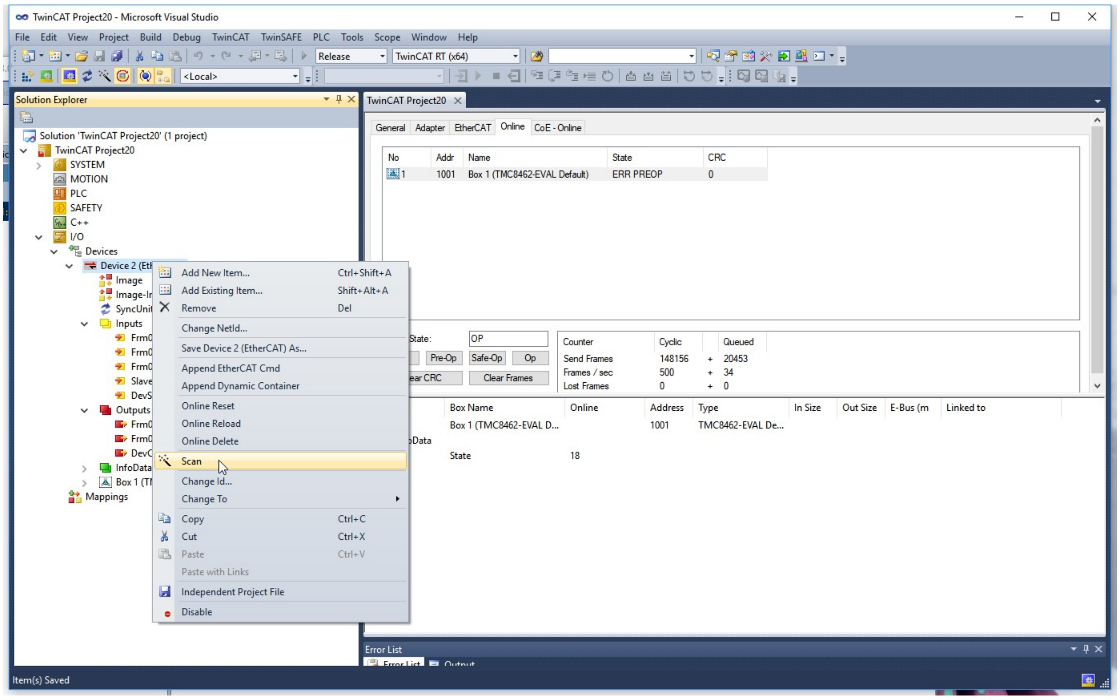

6. Now we want to upload our new XML file to the TMC8462-EVAL. Select the EtherCAT adapter and go to the "Online"-tab. Right-click on the TMC8462-EVAL box and select "EEPROM Update...".



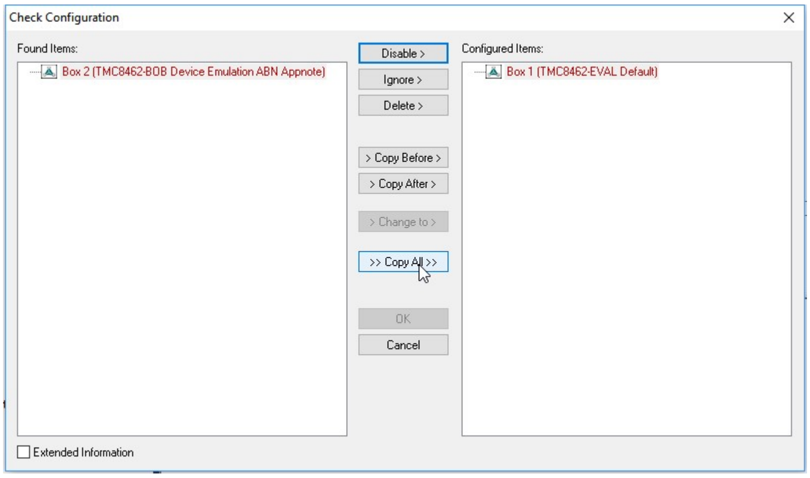
7. Select the "TMC8462-EVAL Device Emulation ABN Appnote" from the list of available EEPROM descriptions. Click "OK" and wait some seconds. TwinCAT is now updating the SII EEPROM with the selected XML description.



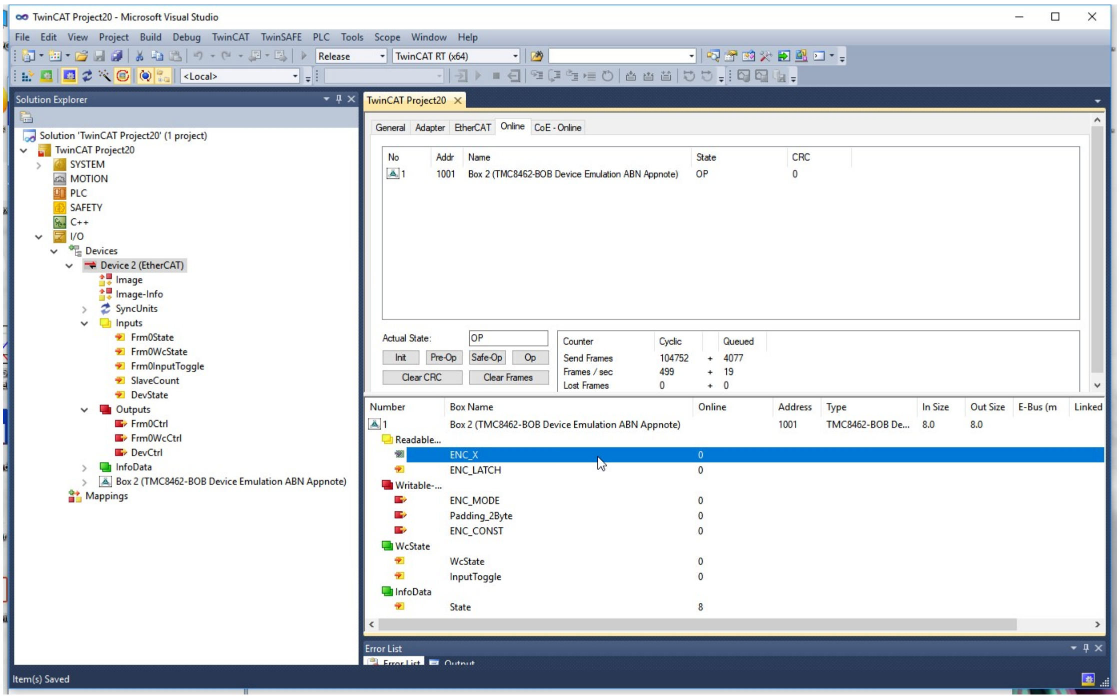
8. Since the configuration of our slave has changed, we need to re-scan the bus by right-clicking on our adapter and selecting "scan".




9. The window that appears shows you the changes on the bus. Click on "» Copy All »" and then "OK".



10. Now you already see the new name of your slave "TMC8462-EVAL Device Emulation ABN Appnote" as well as the RXPDOs and TXPDOs of the mapped MFC IO registers as they have been configured in the XML file.

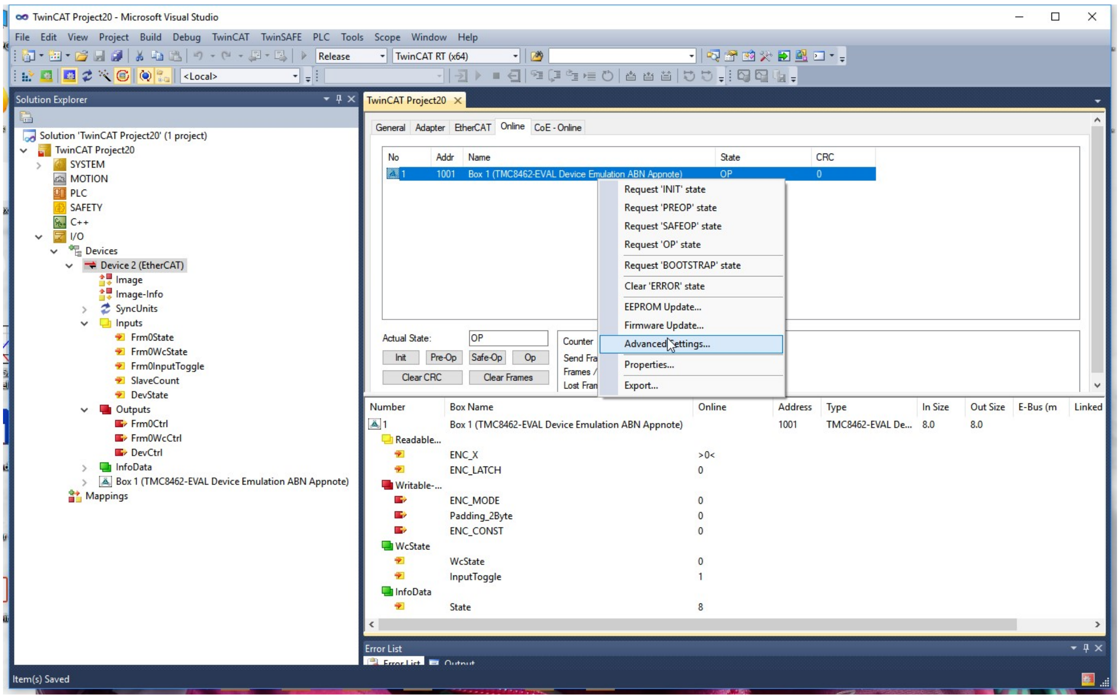



11. Nevertheless, the category number of the MFC IO configuration must still be changed from category 2049 to category 1 to be properly loaded at startup. (The reason is that TwinCAT prevents direct writing of category 1 configuration to the EEPROM.) We also pointed this out as comment in the provided XML file:

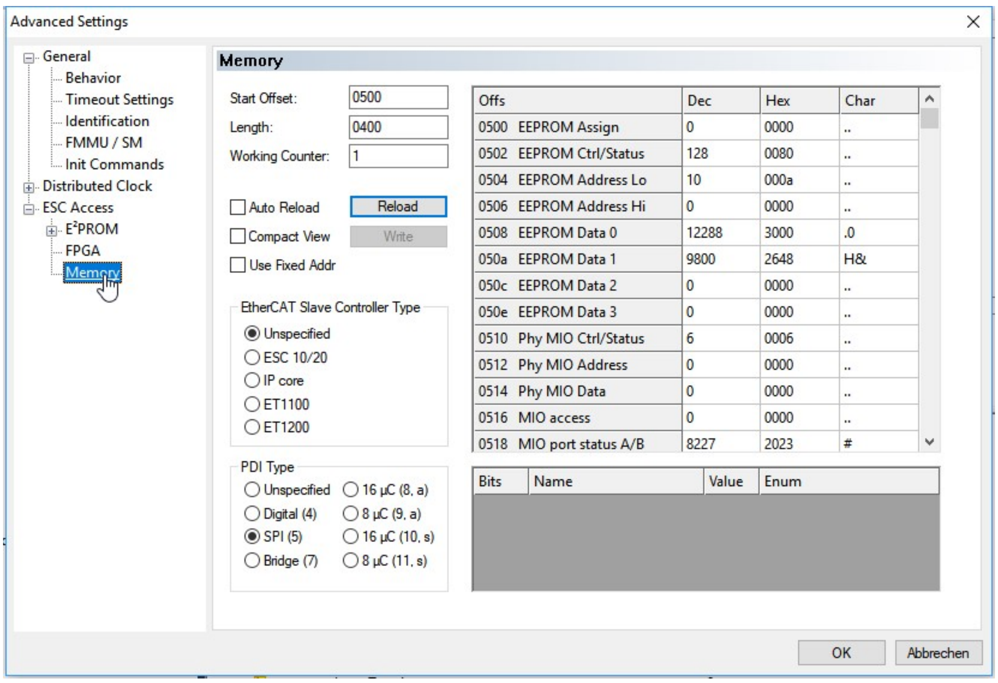
```
<Eeprom>
  <ByteSize>2048</ByteSize>
  <!-- ConfigData>000300000000000000000000000000</ConfigData -->
  <ConfigData>050F0366FF0000000000</ConfigData>

  <!-- Category 1 is required for parameter loading -->
  <!-- but can not directly be written with TwinCAT -->
  <!-- (other master systems might allow)           -->
  <!-- Use category 0x0801 and change manually:     -->
  <!-- 1.) Write addr 0x0507:0x0504 <= 0x00000040   -->
  <!-- 2.) Write Addr 0x0509:0x0508 <= 0x0001       -->
  <!-- 3.) Write Addr 0x0503:0x0502 <= 0x0281       -->
  <Category>
  <!--
    <CatNo>1</CatNo>
  -->
  <CatNo>2049</CatNo>
  <Data>000000000000000000000000000000000004060800000000000000000000
0000000000000000000000000000000078787878</Data>
  </Category>
```

12. To do so, right click on the box/slave in the "Online" tab and select "Advanced Settings...".

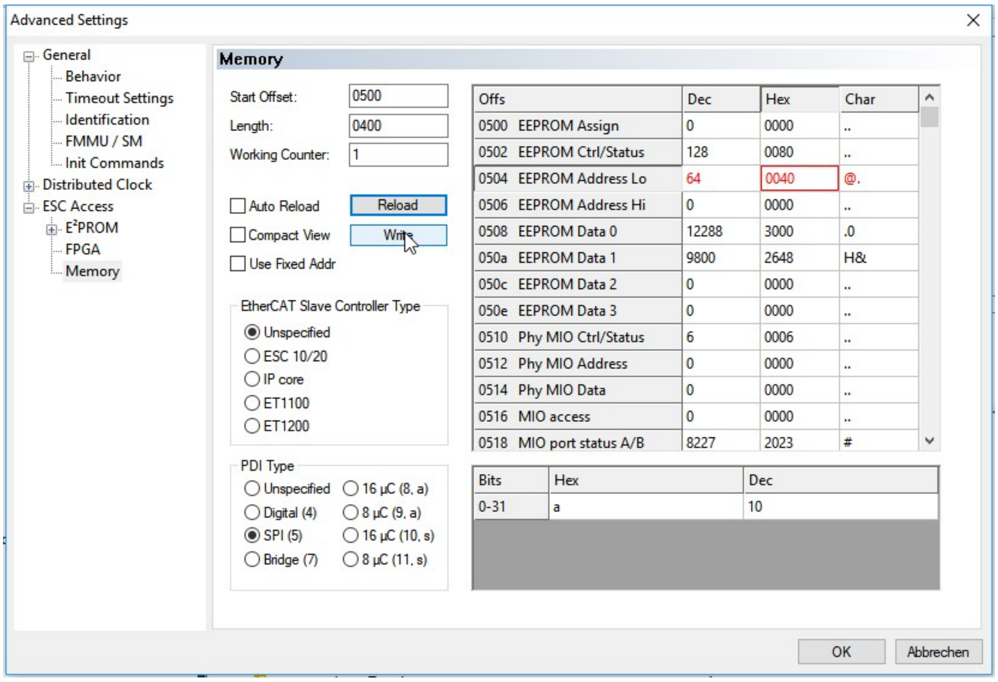



13. In the Advanced Settings window go to the "Memory" view in the "ESC Access" sub-tree. This view is a live view of the TMC8462 internal registers and PDRAM and allows for byte-wise read and write operations to it for debugging, testing, and modification. In the field for "Start Offset" enter 0500 to move to the required address range we need in this step, which is used to modify the EEPROM content via the TMC8462 directly.

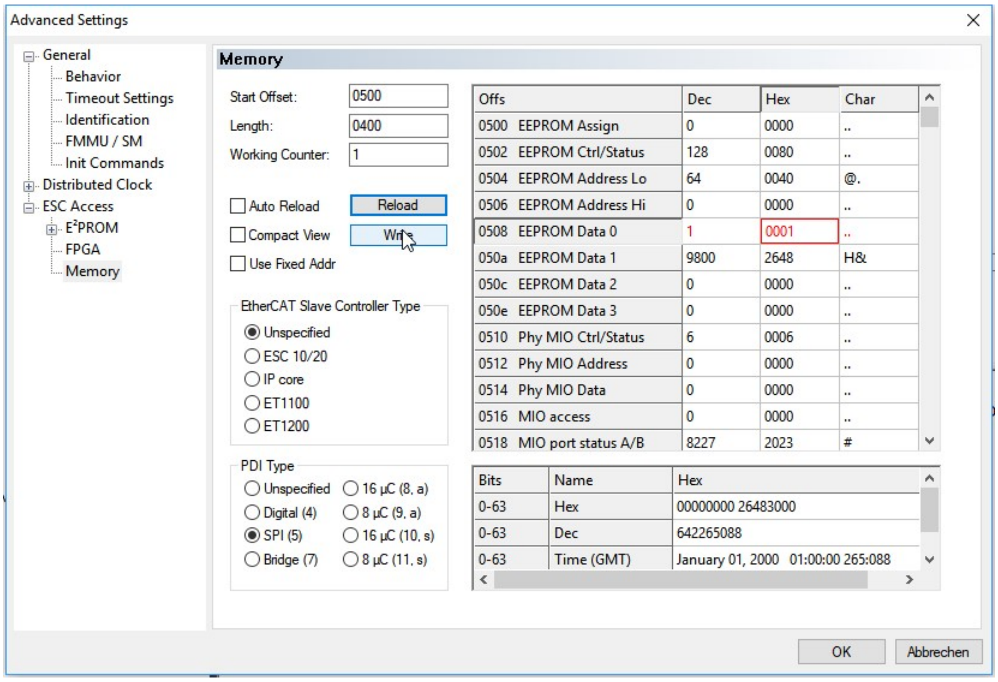


14. Now we perform the three steps as described in the textual description in step 11. They are all write operations to ESC register that lead to a write operation to the EEPROM to change the category number.

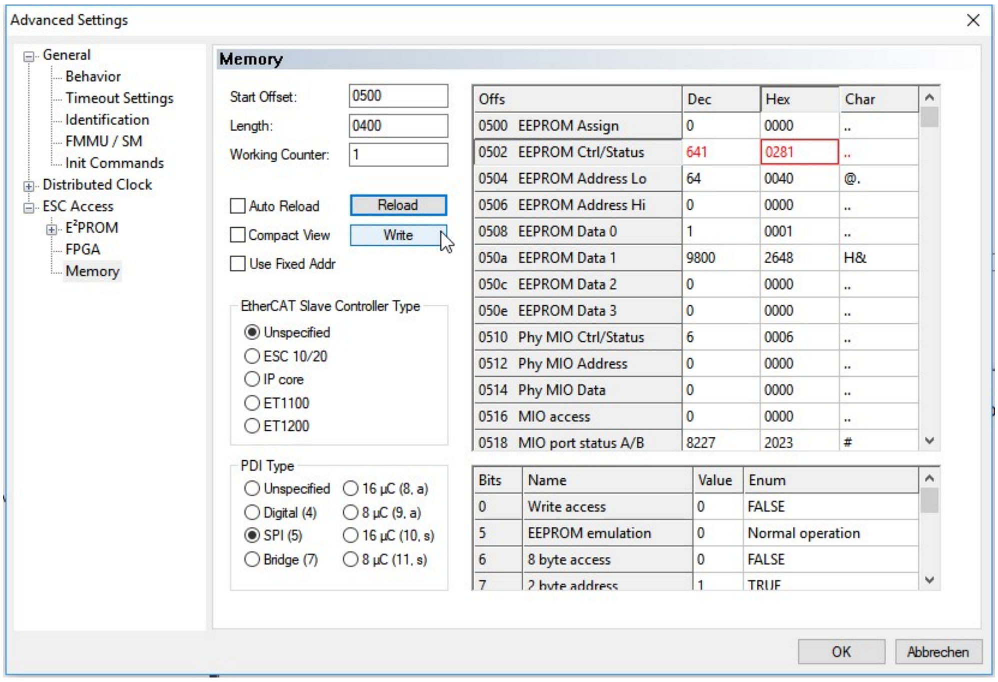
    (a) Double-click into the hex field of address 0x0504 (EEPROM Address Lo) and write a 0x0040 into it. Then click on write.




    (b) Double-click into the hex field of address 0x0508 (EEPROM Data 0) and write a 0x0001 into it. Then click on write.

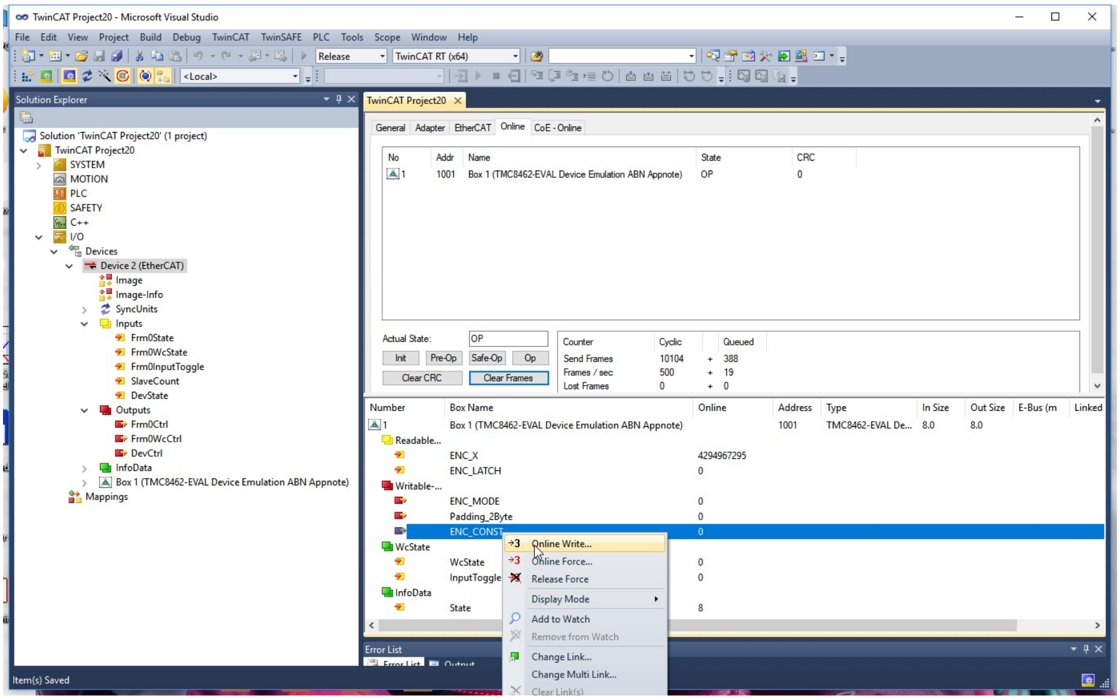


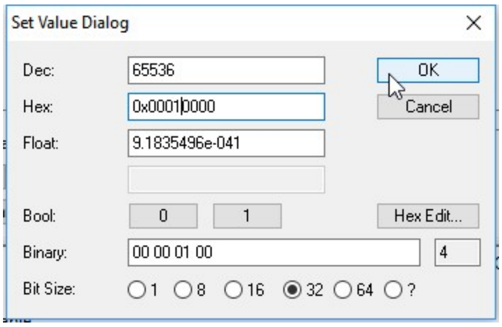
(c) Double-click into the hex field of address 0x0508 (EEPROM Data 0) and write a 0x0001 into it. Then click on write.



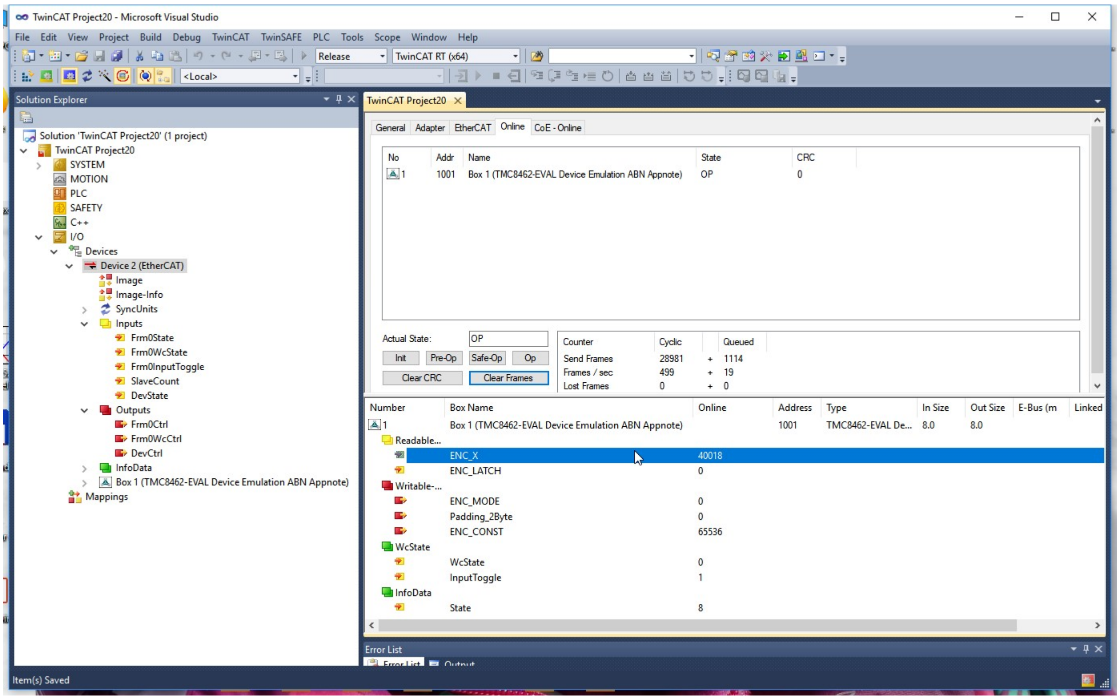

15. Now that we have successfully changed the category number of the MFC IO configuration vector, we need to reset the TMC8462-EVAL. Simply use the reset-button in the upper right corner next to the RJ45 connector of the board.
After some seconds, the TMC8462-EVAL reconnects and should go automatically into OP mode again.

16. Right now when moving the motor shaft by hand, the "ENC_X" RXPDO still does not change because the encoder counting constant is not yet set. We have to configure the encoder unit in the TMC8462 by setting an encoder counting constant to the "ENC_CONST" TXPDO.
Therefore, right-click on "ENC_CONST" and select "Online Write..." and write a `0x00010000` to it. This is `65536` in decimal representation and means that each quadrature encoder tick counts as one increment (1:1).

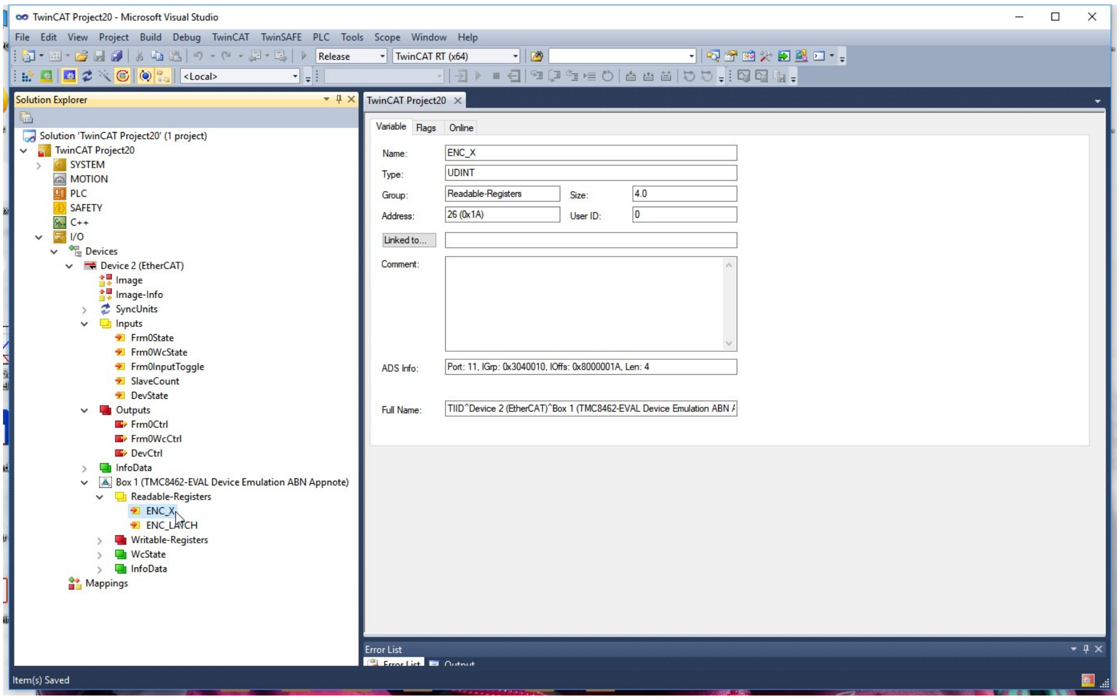


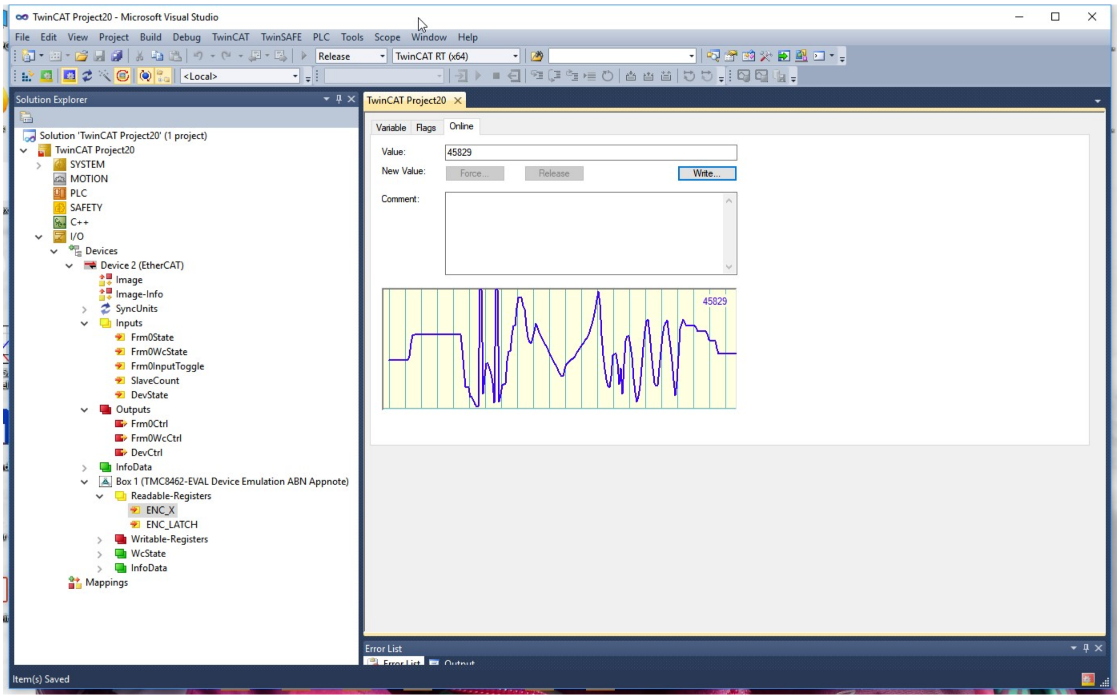
17. Now we are done and you can turn the motor and watch the RXPDO "ENC_X" change according to the movement. The encoder has 10K lines (= 40K increments). Simply turn the motor shaft by one full 360°approximately to verify this. The ENC_X value should now change to circa 40000.




18. To visualize the encoder counting up and down even better, we can also watch the ENC_X value in a diagram over time. To do so open the tree view of the TMC8462-EVAL box in the right tree diagram and left-click on ENC_X under "Readable-Registers". The right window changes.



19. Now click on the "Online" tab, which provides a representation of this RXPDO in a running time-scale diagram. Right-click into the diagram area to adapt the ranges since the default range is full 32bit and you would not see too much changes at full scale. In the following screen shot, a scale of 0 to 100K was chosen for better visualization.
When now turning the shaft of the motor by hand you see how the encoder position value ENC_X changes over time.



# 8 Summary and Conclusion

This application note showed that with very low hardware overhead and actually no additional software coding a simply quadrature encoder has been turned into an EtherCAT slave and connected to an EtherCAT bus. The EtherCAT master can now stream the actual position data from that encoder within the cyclic frames.

While this example sets up a quite simple EtherCAT slave device there are of course possible enhancements to this application note. An MCU could be used running the EtherCAT State Machine and an enhanced protocol stack with special device profiles for sensors for example. Additional SyncManagers mechanisms can be configured to take care of data integrity and synchronization between slave and master.

# 9 Links and Additional Information

- TRINAMIC Motion Control GmbH & Co.KG,
  `www.trinamic.com`
- TMC8462-BA,
  `https://www.trinamic.com/products/integrated-circuits/details/tmc8462-ba/`
- TMC8462 Evaluation Board (EVAL),
  `https://www.trinamic.com/support/eval-kits/details/tmc8462-eval/`
- TMC8462 Break Out Board (BOB),
  `https://www.trinamic.com/support/eval-kits/details/tmc8462-bob-eth/`
- TMCL-IDE,
  `https://www.trinamic.com/support/software/tmcl-ide/`
- General EtherCAT Information, EtherCAT Technology Group,
  `www.ethercat.org`
- TwinCAT EtherCAT Master, Beckhoff Automation GmbH & Co. KG,
  `www.beckhoff.com`

# 10 Revision History

| Version | Date | Author | Description |
|---|---|---|---|
| V1.00 | 07.08.2018 | SK | Initial release version |

*Table 2: Document Revision*