# 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	1
2	The Hardware Setup – Components	2
3	The Hardware Setup – Jumper Settings	3
4	The Hardware Setup – Wiring	3
5	TMC8462 – Configuration of the MFC IO Functions	5
6	TMC8462 EtherCAT Slave Configuration Using the Configuration Wizard	6
7	TMC8462 EtherCAT Slave Configuration Using TwinCAT	13
8	Summary and Conclusion	25
9	Links and Additional Information	25
10	Revision History	25

#### **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.

©2018 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany Download newest version at: www.trinamic.com





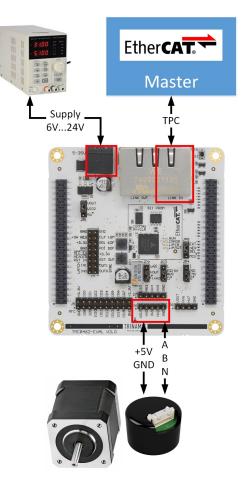
AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 2 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

#### 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.





AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 3 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

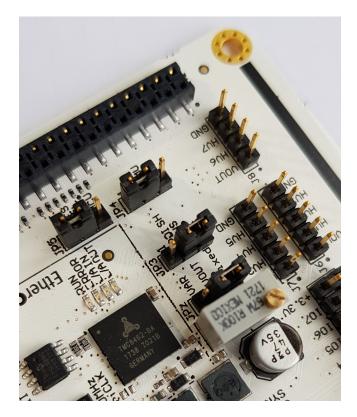
## 3 The Hardware Setup – Jumper Settings

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.



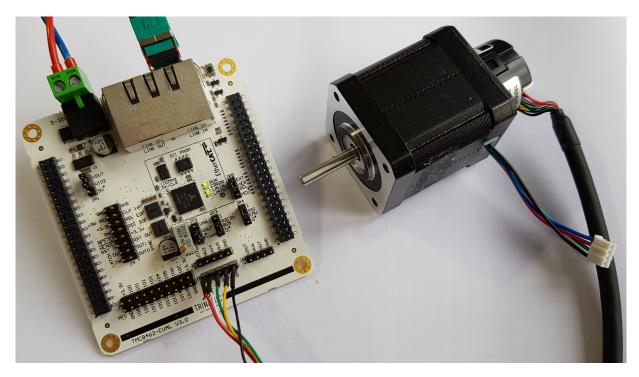
The encoder itself requires 5V DC for operation. This 5V can be provided directly from the adjustable DC/DC buck regulator inside the TMC8462. No additional DC/DC regulator component is required. Also, the encoder's quadrature and index signals, which are typically at 5V level, can directly be connected 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.





AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 5 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

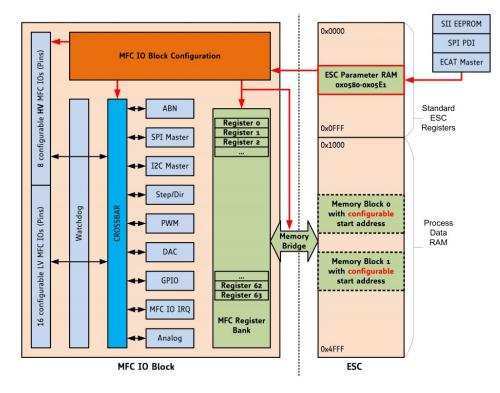
## **5** TMC8462 – Configuration of the MFC IO Functions

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 subblocks – including the ABN block, the crossbar, and the HV IOs we are going to use in this application note.





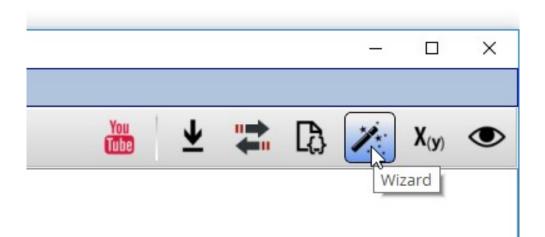
#### 6 TMC8462 EtherCAT Slave Configuration Using the Configuration Wizard

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.)

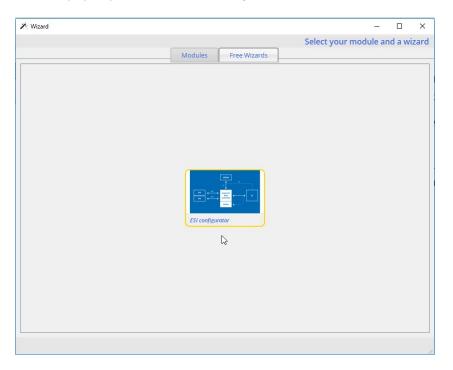
TRINAMICs's TMCL IDE, used to control all standard modules and evaluation boards, provides an easy to use offline wizard that allows generating the TMC8462 configuration vector in an easy "click & copy" style. The following steps show how to use that wizard.

- 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.

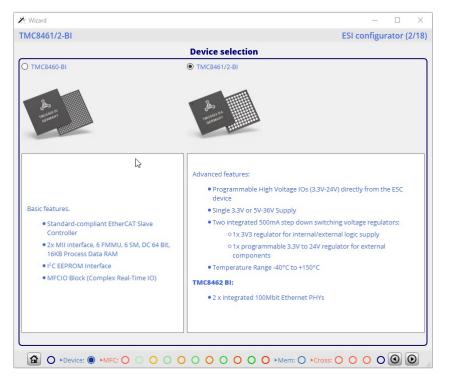


4. The wizard starts with an overview. Simply click on the lower right navigation arrow to go on to the device selection.

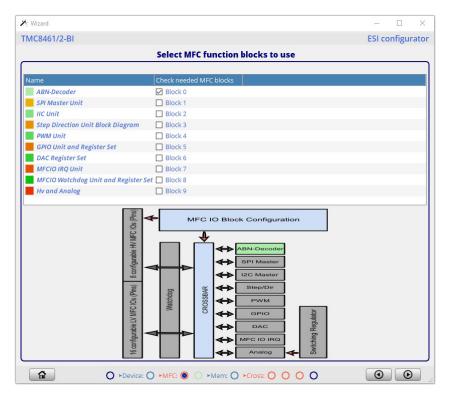




5. In the device selection choose the TMC8461/2 and use the navigation error on the lower right to go on to the next view – MFC IO block selection.



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.





AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 9 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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.

s (Pins)	N	IFC IO B	lock Configuration						
HV MFC IOS		•	ABN-Decoder			1			_
configurable					Name ENC_MODE	Check needed registers Register 0	Access	Trigger source Always triggered	
- United		-  ◆	SPI Master				read	Always triggered	•
8 C		€	I2C Master			Register 2	write	Always triggered	
(suic	Бõ	AR 🗲	Step/Dir			Register 3	read	Always triggered	<b>_</b>
MFC IOs (Pins)	Watchdog		PWM			Register 4	write	Always triggered	
MFC	Ň	CR(	GPIO	tor		Register 5	read	Always triggered	•
16 configurable LV		+ + +		Switching Regulator					

! 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.



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs 10 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

🖉 Wizard	$\searrow$		– 🗆 X
TMC8461/2-BI	- 0		ESI configurator (5/9
	Memor	y blocks	
Block 0 fo	or write registers	Block 1 for read registers	
Stant address	0x1000 🕞	Stant address	0x1100 🖨
Name Start address	End address	Name Start address End address	
✓ ABN-Decoder		✓ ABN-Decoder	
ENC_MODE 0x1000	0x1001	ENC_X(r) 0x1100 0x1103	
Padding 0x1002	0x1003	ENC_LATCH 0x1104 0x1107	
ENC_CONST 0x1004	0x1007	End address	0x1107
End Address	0x1007		0.1107
000000000000000000000000000000000000000	0000000000000000 <mark>000000000000000000000</mark>	0000101010 <mark>000000000000000000000000000</mark>	000000000000000000000000000000000000000
<b>a</b>	O Device: O MFC: O O Mem	: • Fcross: • • • • • • • • • • • • • • • • • •	00

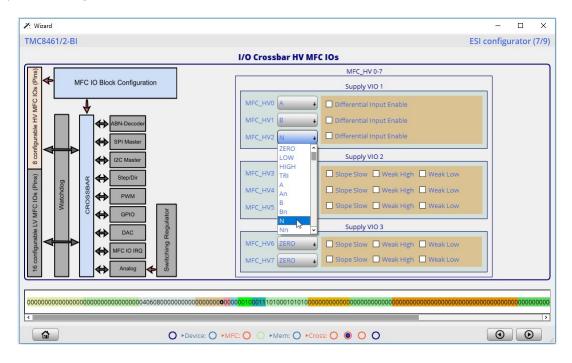
9. The next view configures the signal assignment for the low voltage IOs. We do not need the LV IOs in this example, so simply skip that view and go to the next view for the high voltage IOs.

K: Wizard			- 🗆 ×
TMC8461/2-BI			ESI configurator (6/9)
I/0 (	Crossbar LV MFC IOs		
MFC IO Block Configuration MFC IO Block Configuration ABN-Decoder ABN-Decoder SPI Master COULDING PWM PWM GPIO DAC Analog WICLING WICLING MFC IO IRQ Analog MFC IO IRQ MFC	MFCI000         ZERO         •           MFCI001         ZERO         •           MFCI002         ZERO         •           MFCI003         ZERO         •           MFCI004         ZERO         •           MFCI005         ZERO         •           MFCI006         ZERO         •           MFCI007         ZERO         •	3.3V GPIO pins of MFCIO 8-15 MFCIO08 ZERO • MFCIO10 ZERO • MFCIO11 ZERO • MFCIO12 ZERO • MFCIO12 ZERO • MFCIO13 ZERO • MFCIO14 ZERO • MFCIO15 ZERO •	000000000000000000000000000000000000000
Device:      NFC:	○ ▶Mem: ○ ▶Cross: ◎ ○ (	0 0	



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • Application Note • Document Revision V1.00 • 2018-Aug-07

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.

X: Wizard	-		×
ТМС8461/2-ВІ Е	SI confi	gurato	(8/9)
Switching regulator configuration			
MFC IO Block Configuration		00010000	000000
	0		



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs 12 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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.

🗶 Wizard			_		×
TMC8461/2-BI			ESI configu	irator	(9/9)
	ESI generation				
Mark & Copy the <data> entry to your esi</data>	file <eeprom> section</eeprom>				
<pre><!-- Category 1 is required fo<br--><!-- Use category 0x0601 and c<br--><!-- Write addr 0x0507:0x0504<br--><!-- Write Addr 0x0507:0x0508<br--><!-- Write Addr 0x0509:0x0508<br--><!-- Write Addr 0x0503:0x0502<br--><category> <category> <catno>1</catno> &gt; <catno>2049</catno> <category> <category></category></category></category></category></pre>	ritten with TwinCAT> hange manually:> <= 0x000000000> <= 0x0001>	000000000000000000000000000000000000000	0000000000		~
					>
Definition for C code					
//Prototype of spi write funct void HW_MfcWrite(UINT16 Addres					Î
//Address mapping definitions					
<pre>#define TMC846x_ENC_MODE #define TMC846x_ENC_STATUS</pre>	0x0000 0x0010				
#define TMC846x ENC X W	0x0020	45			
#define TMC846x ENC X R	0x0030				
#define TMC846x ENC CONST	0x0040				
#define TMC846x ENC LATCH	0x0050				
#define TMC846x SPI RX DATA	0x0060				
#define TMC846x SPI TX DATA	0x0070				
#define TMC846x SPI CONF	0x0080				
#define TMC846x_SPI_STATUS	0x0090				
#define TMC846x_SPI_LENGTH	0x00A0				
#define TMC846x_SPI_TIME	0x00B0				
#define TMC846x_IIC_TIMEBASE	0x006?				
<pre>#define TMC846x_IIC_CONTROL</pre>	0x007?				
#define TMC846x_IIC_STATUS	0x008?				~
	● ►Device: ● ►MFC: ● ● ►Mem: ● ►Cross: ● ● ●		٩		

Now, we continue with the next section.



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 13 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

## 7 TMC8462 EtherCAT Slave Configuration Using TwinCAT

<u>OPTION A:</u> 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>**.

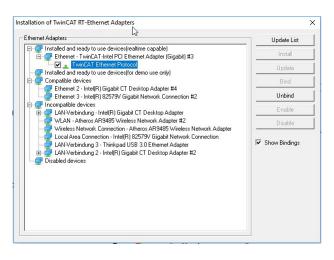
To make configuration in TwinCAT easier, it is suggested to also define the mapped encoder registers as RX and TX PDOs in the XML file. This way they are directly readable/writeable without too much handwork and mouse clicking.

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

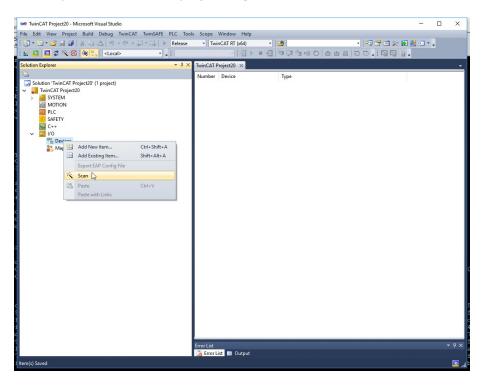
The following steps show how to configure and start the TMC8462-EVAL with our encoder example in the Beckhoff TwinCAT master system:

- 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.

👓 TwinCAT Project20 - Microsoft Visual Studio		-		×
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Tool				
🔚 🕶 📲 🖬 🖉 📓 👗 🐴 🖄 🖃 - (* - 💭 - 🖳 🕨 Release	🔹 TwinCAT RT (x64) 🔹 🔯	<u> </u>	-	
i 🔛 🚨 🧔 🔨 🧐 🙆 🐛 <local> 🔹 🚦</local>	- ] - ] - ] = -] = -] = -] = -] = -] =			
Solution Explorer - # ×	TwinCAT Project20 ×		ľ	•
Solution TwinCAT Project20' (1 project)	General Adapter         EhercAT         Online         Cole - Online           No         Addr         Name         State         CRC           Mo         Addr         Name         ERR PREOP         0			<
<ul> <li></li></ul>	Number Box Name Online Address Type Ir I Box 2 (TMCS462-EVAL D 1001 TMCS462-EVAL D ■ InfoData State 18 Comparison of the state of the stat	1 Size		ф ×
Item(s) Saved				<b>.</b>
nem(s) serve				<u> </u>

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...".



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs 16 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

👓 TwinCAT Project20 - Microsoft Visual Studio		- 0	x
File Edit View Project Build Debug TwinCAT TwinSAFE PLC To	ls Srone Window Heln		~
in • · · · · · · · · · · · · · · · · · ·	- TwinCAT RT (x64) - 20 - 20 - 20 - 20 - 20 - 20 - 20 - 2		
1 1 1 2 × 6 0 1 <local></local>	- · · · · · · · · · · · · · · · · · · ·		
Solution Explorer V 2 4 >	TwinCAT Project20 ×       General Adapter EhreCAT Online       No       Addr. Name       State       CRC       Intervent Boortstate       Request 'PREOP' state       Request 'SAFEOP' state       Disce 'Occ' Core Creater Stating       Boon Tim/CSAFE-EVAL De       Insize Out Size E-Bus (m Linked to       State     18		
	Error List		• ¶ ×
Item(s) Saved	👻 Emerida 🖬 Ordnud		ia. 🗖

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.

Available EEPROM Descriptions:	Show Hidden Devices	OK
Beckhoff Automation GmbH & Co. KG		
- A Trinamic Motion Control GmbH & Co. KG		Cancel
MFCIO_Test		
EVAL boards		
	with all MFCIO registers (642265170 / 65815)	
A TMC8461-EVAL Default (6422651		
- A TMC8462-BOB Default (6422651)		
TMC8462-BOB Device Emulation S		
- A TMC8462-EVAL Default (6422651		
TMC8452-BOB Device Emulation A	EN Appnote (642265088 / 1)	
CoE Drives	42	
CoE Drives		
⊞- <b>▲</b> S		
		Browse
		blowse



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".

👓 TwinCAT Project20 - Microsoft Visual Studio		- 0	×
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too	s Scope Window Help		
- 🔚 🕶 📲 🗃 😹 🔏 🖄 🖄 🔊 - (* - 💭 - 🖏 ) 🕨 Release	🔹 TwinCAT RT (x64) 🔹 🙋 🔹 🔹 🖓 🛣 💥 😥 🛃 🗉 🖕		
i 🔐 🚨 🥔 🛠 🌾 🎯 🍖 - 🐛 - Local> 🔹 🚦	- · · · · · · · · · · · · · · · · · · ·		
Solution Explorer 👻 म 🗙	TwinCAT Project20 ×		-
	General Adapter EtherCAT Online CoE - Online		^
Solution 'TwinCAT Project20' (1 project)			
SYSTEM	No Addr Name State CRC		
MOTION	1 1001 Box 1 (TMC8462-EVAL Default) ERR PREOP 0		
DI PLC			
SAFETY & C++			
✓ ☑ I/O			
✓ <sup>0</sup> <sup>™</sup> <sub>□</sub> Devices			
	Shift+A		
📲 Image-Ir 🔛 Add Existing Item Shift	+Alt+A		
SyncUnit X Remove Del			
✓ Frm0	State: OP Counter Cyclic Queued		1
Frm0 Save Device 2 (EtherCAT) As	Pre-Op Safe-Op Op Send Frames 148156 + 20453		
Frm0 Append EtherCAT Cmd     Slave Amound Duramic Containers	Promos / sec 500 + 34		
DevS	Lost Frames 0 + 0		×
V Dutputs Online Reset	Box Name Online Address Type In Size Out Size E-Bus (m Linked to		
Frm0 Online Reload     Frm0 Online Delete	Box 1 (TMC8462-EVAL D 1001 TMC8462-EVAL De		
Dev(	Data State 18		
> 🛄 InfoData			
> A Box 1 (TI Change Id Mappings Change To			
Copy Ctrl- Cut Ctrl-			
Paste Ctri-			
Paste with Links			
Independent Project File			
e Disable			
- Distric			
	Error List		- 4 ×
	🖹 Enzelist. 🗖 Ordenst		
Item(s) Saved			🔤 .d

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

Check Configuration			×
Found Items:	Disable >	Configured Items:	
Box 2 (TMC8462-80B Device Emulation ABN Appnote)	Ignore > Delete >	Box 1 (TMC8462-EVAL Default)	
	Copy Before >     Copy After >     Change to >		
	>> Copy All >>		
	OK Cancel		
Extended Information		L	



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 18 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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.

🗢 TwinCAT Project20 - Microsoft Visual Studio		- 🗆 X
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too	ols Scope Window Help	
1 日 · 四 · 四 · 日 · 日 · 日 · 日 · 日 · ○ · ○ · ○ · ○ · ○	<ul> <li>TwinCAT RT (x64)</li> <li>図 (2010)</li> <li>(2010)</li> <li>(</li></ul>	
🗄 🎦 💆 🛠 🌾 🚱 🍖 <local> 🔹 🕫</local>	-   型 ▶ = 包  確保強性の   占 当 当   ひ ♡ - : 問 型 և -	
Solution Explorer - 4 ×	TwinCAT Project20 ×	<u> </u>
Solution Experts 4 4 X Solution Experts 2 Solution TwinCAT Project20 (1 project) Solution TwinCAT Project20 Solution TwinCAT Project20 Project 2 Solution TwinCAT Project20 Project 2 Solution TwinCAT Project20 Project 2 Solution Solution Profile Characteristic Solution All Approves (1) Solution All Approve	Non-Add     Name       Actual State:     OP       Actual State:     OP       Actual State:     OP       Oper CRC     Oper Frames       Dear Frames     Courter       Lot Frames     O       Number     Box 2 (TMC8462-BOB Device Emulation ABN Appnde)       OP     O	E-Bus (m Linked
]	📽 InputToggle 0 🖬 InfoData	
a	👻 State 8	, , , , , , , , , , , , , , , , , , ,
	Error List	<del>-</del> 4 ×
	Error List	• <del>4</del> ×
Item(s) Saved		i. 🖬

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></eeprom>	
<bytesize>2048</bytesize>	
ConfigData 00030000000000000000000000000000000/Conf	igData>
<configdata>050F0366FF0000000000</configdata>	
(configured, cool cool cool cool cool cool cool coo	
Category 1 is required for parameter loading</td <td>&gt;</td>	>
but can not directly be written with TwinCAT</td <td></td>	
(other master systems might allow)</td <td>&gt;</td>	>
<pre><!-- Use category 0x0801 and change manually:</pre--></pre>	>
</math 1.) Write addr 0x0507:0x0504 <= 0x00000040	>
	5
2.) Write Addr 0x0509:0x0508 <= 0x0001</td <td>&gt;</td>	>
3.) Write Addr 0x0503:0x0502 <= 0x0281</td <td>&gt;</td>	>
<category></category>	
</td <td></td>	
<catno>1</catno>	
>	
<catno>2049</catno>	
<data>00000000000000000000000000000000000</data>	000000000000000000000000000000000000000
00000000000000000000000000000000000000	
(Category)	



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs 19 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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

😎 TwinCAT Project20 - Microsoft Visual Studio				- c	×
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too	ls Scope Window Help				
🖥 🔚 • 🔛 • 💕 🚚 🝠 👗 🐁 🖄 🕫 - 🗠 - 💭 - 🖏 🔺 Release	<ul> <li>TwinCAT RT (x64)</li> <li>W</li> </ul>	- 🔍 🕾 🖄	k: 🛃 🛃 🗉 📲		
🔛 🛄 🥩 🌂 🎯 🐾 - <local> 🔹 🚽</local>	·  관 ▶ = 관 액 (21%	= = O   古古首   U U <mark>-</mark> : 日日	1 (a -		
Solution Explorer - 4 ×	TwinCAT Project20 ×				-
Solution 'TwinCAT Project20' (1 project)	General Adapter EtherCAT Online CoE - Online	ie .			<u>^</u>
V TwinCAT Project20	No Addr Name	State	CRC		
> C SYSTEM	1 1001 Box 1 (TMC8462-EVAL Devi	ce Emulation ABN Appnote) OP	0		
01 PLC		Request 'INIT' state			
x SAFETY		Request 'PREOP' state			
🚱 C++		Request 'SAFEOP' state			
✓ □ I/O ✓ ■ Devices		Request 'OP' state			
Device 2 (EtherCAT)		Request 'BOOTSTRAP' state	_		
Image Image-Info		Clear 'ERROR' state			
> 2 SyncUnits		EEPROM Update			
- V 🛄 Inputs		Firmware Update			
😕 Frm0State	Actual State: OP Cou	Inter Advanced Settings			
Frm0WcState     Frm0InputToggle	Init Pre-Op Safe-Op Op Ser	nd Fra Properties			
Prmoinput loggie     SlaveCount		mes /			
≠ DevState		t Fran Export			¥
🗸 🛄 Outputs	Number Box Name	Online	Address Type	In Size Out Size E-Bus (m	h Linked
Frm0Ctrl	Box 1 (TMC8462-EVAL Device     Readable	Emulation ABN Appnote)	1001 TMC8462-EVAL De	8.0 8.0	
Printoweeth     DevCtrl	ENC X	>0<			
e > 🛄 InfoData	ENC LATCH	0			
> A Box 1 (TMC8462-EVAL Device Emulation ABN Appnote)	Writable				
Mappings	ENC_MODE	0			
	Padding_2Byte	0			
v .	ENC_CONST	0			
	WcState	0			
	InputToggle	1			
	InfoData				
J	😴 State	8			
ш.	¢				>
	Error List				- 4 ×
	😤 Error Lirt 📼 Output				
Item(s) Saved					ii. 🗖

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.

General	Memory							
Behavior Timeout Settings	Start Offset:	0500	Offs		Dec	Hex	Char	^
Identification	Length:	0400	0500	EEPROM Assign	0	0000		
FMMU / SM Init Commands	Working Counter:	1	0502	EEPROM Ctrl/Status	128	0080		
Distributed Clock			0504	EEPROM Address Lo	10	000a		
ESC Access	Auto Reload	Reload	0506	EEPROM Address Hi	0	0000		
E <sup>2</sup> PROM	Compact View	Write	0508	EEPROM Data 0	12288	3000	.0	
FPGA	Use Fixed Addr		050a	EEPROM Data 1	9800	2648	H&	
Memory			050c	EEPROM Data 2	0	0000		
0	EtherCAT Slave C	Controller Type	050e	EEPROM Data 3	0	0000		
	Unspecified		0510	Phy MIO Ctrl/Status	6	0006		
	O ESC 10/20		0512	Phy MIO Address	0	0000		
	O IP core O ET1100		0514	Phy MIO Data	0	0000		
	O ET1200	0516	MIO access	0	0000			
			0518	MIO port status A/B	8227	2023	#	~
	PDI Type O Unspecified	0 16 μC (8, a)	Bits	Name	Value	Enum		
	() SPI (5)	○ 8 μC (9, a) ○ 16 μC (10, s) ○ 8 μC (11, s)						



- 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.

Identification       Length:       0400       0500 EEPROM Assign       0         FMMU / SM       Working Counter:       1       0500 EEPROM Assign       0         Distributed Clock       ESC Access       Auto Reload       Reload       0506 EEPROM Address Lo       0         E EPROM       □ Compact View       Writs       0508 EEPROM Data 0       0       0506 EEPROM Data 1       0         FPGA       □ Use Fixed Addr       0506 EEPROM Data 2       0       0       0506 EEPROM Data 3       0         EtherCAT Slave Controller Type       0500 EEPROM Data 3       0       0       0510 Phy MIO Ctrl/Status       0         0 ET100       ○ ET100       ○ 516 MIO access       0       0       0       0	ec Hex 0000 28 0080 4 0040 0000 2288 3000 2648 0000 2648 0000 0000	Char   @.  .0 H&	<b>^</b>
FMMU / SM       Length:       0400       0500       EPROM Assign       1         Distributed Clock       ESC Access       Auto Reload       0504       EEPROM Address Lo       0         ESC Access       Auto Reload       Reload       0506       EEPROM Address Lo       0         EFGA       Ouse Fixed Addr       0506       EEPROM Data 0       0       0506       EEPROM Data 1       0         BherCAT Slave Controller Type       Ispecified       0510       Phy MIO Ctrl/Status       0       0510       Phy MIO Address       0         OFID Phy MIO Address       OFID Phy MIO Ctrl/Status       0       0510       Phy MIO Address       0         OFID Phy MIO Ctrl/Status       OFID Phy MIO Data       0       0514       Phy MIO Data       0         OFIT Phy MIO Ctrl/Status       OFID Phy MIO Data       0       0       0       0       0	28         0080           4         0040           0000         0000           2288         3000           300         2648           0000         0000	 @.  .0 H& 	
Init Commands       Working Counter:       1       0502       EEPROM Ctrl/Status         Distributed Clock       ESC Access       Auto Reload       0504       EEPROM Address Lo         EYPROM       Compact View       Wing       0508       EEPROM Data 0       0502       EEPROM Data 1       0504       EEPROM Data 2       0506       EEPROM Data 2       0506       EEPROM Data 2       0506       EEPROM Data 1       0502       EEPROM Data 2       0506       EEPROM Data 1       0506       EEPROM Data 2       0506       EEPROM Data 2       0506       EEPROM Data 3       0510       Phy MIO Address       0512       Phy MIO Address       0514       Phy MIO Data       0516       MIO access       0518       MIO port status A/B       0518       MIO port status A/B       0518       MIO port status A/B       0518	4 0040 0000 2288 3000 300 2648 0000	©.  .0 H&	
Distributed Clock ESC Access Auto Reload Reload 0506 EEPROM Address Lo EPROM Compact View With 0508 EEPROM Data 0 FPGA Use Fixed Addr 0508 EEPROM Data 1 BherCAT Slave Controller Type 050e EEPROM Data 2 BherCAT Slave Controller Type 050e EEPROM Data 3 OS0e EEPROM Data 3 Distributed Clock 0506 EEPROM Data 1 Soc EEPROM Data 2 Dispecfied 0510 Phy MIO Ctrl/Status 0 Dispecfied 0512 Phy MIO Address 0 Dispecfied 0512 Phy MIO Address 0 Dispecfied 0516 MIO access 0 Distributed Clock 0518 MIO port status A/B	0000 2288 3000 300 2648 0000	 .0 H&	
Be <sup>1</sup> PROM FPGA Memory     Use Fixed Addr     Use Fixed Addr     OS08 EEPROM Data 0     OS08 EEPROM Data 1     OS06 EEPROM Data 1     OS06 EEPROM Data 2     OS06 EEPROM Data 2     OS06 EEPROM Data 3     OS06 EEPROM Data 3     OS06 EEPROM Data 3     OS06     EEPROM Data     OS16     MIO Data     OS16     MIO port status A/B     S	2288 3000 300 2648 0000	.0 H&	
FPGA       Use Fixed Addr       050a       EEPROM Data 0         Memory       Use Fixed Addr       050a       EEPROM Data 1       9         EtherCAT Slave Controller Type       050a       EEPROM Data 2       0         © Unspecified       0510       Phy MIO Ctrl/Status       0         O EET 100       0514       Phy MIO Data       0         O ET1200       0516       MIO port status A/B       0	300 2648 0000	H& 	
Memory     Use Fixed Addr     050a EEPROM Data 1     050c EEPROM Data 2       EtherCAT Slave Controller Type     050e EEPROM Data 3     050e EEPROM Data 3       Image: I	0000		
EtherCAT Slave Controller Type       050c       EEPROM Data 2         Image: Controller Type       050c       EEPROM Data 3         Image: Controller Type       050c       EEPROM Data 3         Image: Controller Type       0510       Phy MIO Ctrl/Status         Image: Controller Type       0512       Phy MIO Address         Image: Controller Type       0514       Phy MIO Address         Image: Controller Type       0516       MIO Data         Image: Controller Type       0516       MIO access         Image: Controller Type       0518       MIO port status A/B			
Obsective Ctricking of the second secon	0000	· · · · · · · · · · · · · · · · · · ·	
ESC 10/20         0510         Fity Mile Cally Status         0           IP core         0512         Phy MIle Address         0           ET1100         0516         MIO access         0           0516         MIO port status A/B         0         0			
O IP core         0512         Phy MIO Address         0           O ET1100         0516         MIO Data         0           O ET1200         0518         MIO port status A/B         4	0006		
O ET1100         0514         Phy MIO Data         0           O ET1200         0516         MIO access         0           0518         MIO port status A/B         4	0000		
O ET1200 0516 MIO access 0518 MIO port status A/B	0000		
0518 MIO port status A/B	0000		
	227 2023	#	~
PDI Type ◯ Unspecified ◯ 16 µC (8, a) Bits Hex	Dec		
O bightal (4) 08 µC (9, a) ● Digital (4) 08 µC (9, a) ● SPI (5) 016 µC (10, s)	10		
⊖ Bridge (7) ⊖ 8 μC (11, s)			

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

General	Memory								
Behavior Timeout Settings	Start Offset:	0500	Offs		Dec	Hex	Char	^	
Identification	Length:	0400	0500 EE	PROM Assign	0	0000			
FMMU / SM Init Commands	Working Counter:	1	0502 EE	PROM Ctrl/Status	128	0080			
- Distributed Clock			0504 EE	PROM Address Lo	64	0040	<mark>@</mark> .		
- ESC Access	Auto Reload	Reload	0506 EE	PROM Address Hi	0	0000			
E <sup>2</sup> PROM	Compact View	₹ nW	0508 EE	PROM Data 0	1	0001			
FPGA	Use Fixed Addr		050a EE	PROM Data 1	9800	2648	H&		
- Memory	_		050c EE	PROM Data 2	0	0000			
	EtherCAT Slave C	Controller Type	050e EE	PROM Data 3	0	0000			
	Unspecified     ESC 10/20     IP core     ET1100		0510 PH	y MIO Ctrl/Status	6	0006			
			0512 PH	y MIO Address	0	0000			
			0514 PH	iy MIO Data	0	0000			
	O ET1200	0516 M	IO access	0	0000				
			0518 M	IO port status A/B	8227	2023	#	¥	
	PDI Type	○ 16 µC (8, a)	Bits	Name	Hex			^	
	O Digital (4) O 8 µC (9, a)		0-63	Hex	00000000	26483000			
		O 16 µC (10, s)	0-63	Dec	642265088	3			
	O Bridge (7)	O 8 µC (11, s)	0-63	Time (GMT)	January 0	1, 2000 01:00	0:00 265:088	.00 265:088 🗸	
			<					>	



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 1 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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

General	Memory							
Timeout Settings	Start Offset:	0500	Offs		Dec	Hex	Char	^
Identification	Length:	0400	0500	EEPROM Assign	0	0000		
FMMU / SM Init Commands	Working Counter:	1	0502	EEPROM Ctrl/Status	641	0281		
Distributed Clock			0504	EEPROM Address Lo	64	0040	@.	
ESC Access	Auto Reload	Reload	0506	EEPROM Address Hi	0	0000		
<ul> <li>→ F<sup>2</sup>PROM</li> <li>→ FPGA</li> <li>→ Memory</li> </ul>	Compact View	Write	0508	EEPROM Data 0	1	0001		
	Use Fixed Addr			EEPROM Data 1	9800	2648	H&	
			050c	EEPROM Data 2	0	0000		
	EtherCAT Slave C	Controller Type	050e	EEPROM Data 3	0	0000		
	Unspecified		0510	Phy MIO Ctrl/Status	6	0006		
	O ESC 10/20		0512	Phy MIO Address	0	0000		
	O IP core		0514	Phy MIO Data	0	0000		
	O ET1200	0516	MIO access	0	0000			
			0518	MIO port status A/B	8227	2023	#	~
	PDI Type	○ 16 µC (8 a)	Bits	Name	Value	Enum		^
		() 8 μC (9, a)	0	Write access	0	FALSE		
		O 16 µC (10, s)	5	EEPROM emulation	0	Normal op	eration	
	O Bridge (7)	O 8 µC (11, s)	6	8 byte access	0	FALSE		
			7	2 byte address	1	TRUF		×

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.



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 22 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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).

😎 TwinCAT Project20 - Microsoft Visual Studio		- 🗆 X
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too		
🐂 🔂 = 🛅 = 😂 🔄 🥔 👗 🐜 🖄 🕫 - 🖓 - 🖓 - 🖏 🕨 Release	<ul> <li>TwinCAT RT (x64)</li> <li>Image: Image: I</li></ul>	-   🖏 🕾 🖄 🎾 🛃 💷 - 🖕
i 🔝 🛄 🥩 🌂 🎯 🐾 - <local> 🔹 🚽</local>	-  刃▶ =  액焊強相の 占占省 3	
Solution Explorer 🚽 🗸	TwinCAT Project20 ×	-
Solution 'TwinCAT Project20' (1 project)	General Adapter EtherCAT Online CoE - Online	^
	No Add Name Alt 1001 Box 1 (TMC3462:EVAL Device Emulation ABN Appindle)	State CRC ) OP 0
The Device 2 (EtherCAT)     Device 2 (EtherCAT)	Actual State: OP Courter Cyclic Int Pre-Op Safe-Op Op Farmer J 10104 Glear CRC Clear Frames J Los Frames J 0 Los Frames J 0	
	Number         Box Name           I         Box 1 (TMC8462-EVAL Device Emulation ABN Appnote)           Readable         Image: Second	Online         Address         Type         In Size         Out Size         E-Bus (m         Linker           i)         1001         TMC8462-EVAL De         8.0         8.0         8.0           4294967295         0         5.0         5.0         5.0         5.0         5.0
A Box 1 (TMC8462-EVAL Device Emulation ABN Appnote)     Mappings	ENC_CONST	0 0
	WcState         P3         Online Write           WcState         *3         Ohline Force           InputToggle         Kelesse Force           InfoData         Display Mode	0 0
ŭ	State     Add to Watch     Remove from Watch	8
1	Error List	
ltem(s) Saved	Change Link	
A second se	Clear Link(s)	

Set Value Dia	alog	×
Dec:	65536	OK
Hex:	0x0001 0000	Cancel
Float:	9.1835496e-041	
Bool:	0 1	Hex Edit
Binary:	00 00 01 00	4
Bit Size:	○1 ○8 ○16 ●32 (	○64 ○?



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 23 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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.

🗢 TwinCAT Project20 - Microsoft Visual Studio			- 🗆 X
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too	s Scope Window Help		
N = 1 = 1 = 1 → 1 → 1 → 1 → 1 → 1 → 1 → 1	<ul> <li>TwinCAT RT (x64)</li> <li>Image: Image: I</li></ul>	• 🔍 🕾 🕺 🗶 🛃 🖬 • 🖕	
: 🔝 🛄 🤣 🌂 🌀 🍳 🐾 <local> 🔹 🚦</local>	-  刭 > = ┨ @ 焊強相 O   占占当	55.100 Q 4.	
Solution Explorer 🗾 👻 🕂 🗙	TwinCAT Project20 ×		•
Solution TwinCAT Project20 (1 project)  Solution TwinCAT Project20  Solution TwinCAT Project20  Solution  Place  Solution  So	General Adapter EhverCAT Online CoE - Online No Add Name Add Name Add Name Add Name Add Name Add Name Correct Counter Cyclic Int Pre-Op Safe-Op Op Former / social 2005 Former /	Queurd	
SlaveCount SlaveCount DevState Urput Borfm0Ctrl Borfm0Ctrl	Load Frames         0           Number         Box Name           Image: International Content of the state	+ 0 Online Address Type 1001 TMC8462	-EVAL De 8.0 8.0
DevCtrl	W ENC X	40018	
	#I     ENC_LATCH       Writeble       #P       ENC_MODE       #P       Padding_2Byte       #P       ENC_CONST       #WcState       #0       WcState       #0       WeState       #0       WeState       #0       WeState       #0       WeState       #0       WeState       #0       State	0 0 655336 0 1 8	
ũ.	<		>
	Error List		+ 4 ×
	🖹 Error Litt. 📼 Output		
Item(s) Saved			ii. 🗖

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.

👓 TwinCAT Project20 - Microsoft Visual Studio	-	σ×
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too		
🐂 🔂 • 🕮 • 📸 🛃 🍠 🕹 🛍 🖄 🔊 • (* • 💭 • 🖏 🔺 🖗 Release		
🗄 🔛 🚨 🤣 🌂 🌀 🎯 🐾 🛛 <local> 🔹 🚦</local>	·   2 ▶ ■ 2   程碑強信の  各名首  ひひ, : 13 12 14	
Solution Explorer 👻 म 🗙	× TwinCAT Project20 ×	-
	Variable Rags Online	
Solution 'TwinCAT Project20' (1 project)		
> G SYSTEM	Name: ENC_X	
MOTION	Type: UDINT	
SAFETY	Group: Readable-Registers Size: 4.0	
64 C++	Address: 26 (0x1A) User ID: 0	
✓ ☑ I/O ✓ <sup>47</sup> ⊟ Devices	Linked to	
Device 2 (EtherCAT)	Comment:	
tage timage timage time time time time time time time tim		
> 🕏 SyncUnits		
- ✓ □ Inputs ♥ Frm0State		
FrmUstate     FrmOWcState	~ · ·	
Frm0InputToggle	ADS Info: Port: 11, IGrp: 0x3040010, IOffs: 0x8000001A, Len: 4	
al 💎 SlaveCount		
V 🛄 Outputs	Full Name: TIID*Device 2 (EtherCAT)*Box 1 (TMC8462-EVAL Device Emulation ABN #	
Frm0Ctrl		
DevCtrl		
ié > 🔄 InfoData → 🔺 Box 1 (TMC8462-EVAL Device Emulation ABN Appnote)		
Box 1 (TMC8402-EVAL Device Emulation ABN Appnote)     Readable-Registers		
✓ ENC_X		
> 🛄 WcState		
> 🛄 InfoData		
a wappings		
a.		
	Error List	- 4 ×
Item(s) Saved		<u>.</u> .



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 24 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

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.

👓 TwinCAT Project20 - Microsoft Visual Studio	N - 0	×
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Too	ls Scope Window Help	
📔 🖥 • 🕮 • 💕 🚽 🥔 🐇 🔤 🖄 🔊 • (* • 💭 • 🖏 🕨 Release	- TwinCAT RT (x64) - 🙆 - 🥺 🐨 🖓 🕾 🕺 🖂 🗆 - 🚽	ľ
i 🔝 💶 🥩 🔨 🐻 💓 🐾 - «Local» 🔹 💡 i	-   - 1) ▶ ■ - 1) № 健健性()   Δ Δ Δ   5 5 <mark>,</mark> 1 🖓 № № .	
Solution Explorer 🔷 구 부 🗙	TwinCAT Project20 ×	-
Solution Explorer • 9 × Solution Explorer • 9 × Solution TwinCAT Project20 (1 project) Solution TwinCAT Project20 Solution TwinCate FindOveState FindOveState FindOveState SolveCount DeSolate SolveCount FindOveState FindOveState SolveCount Solve	TwincAT Project20 ×       Verable       Verable       Verable       Prove       Dermert:	
ENC_LATCH     Set State     With the Registers     With the Registers     With the Registers     With the Registers     InfoDat		
Appings		
	Error List	- 4 ×
Item(s) Saved		ii. 🖸



AN023: Transforming an Incremental Encoder into a Simple EtherCAT Slave using TRINAMIC's EtherCAT Slave Controller ICs • 25 / 25 Application Note • Document Revision V1.00 • 2018-Aug-07

#### 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



