Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

TMC8460-BI

Integrated EtherCAT Slave Controller with Enhanced Functionality

TRINAMIC® Motion Control GmbH & Co. KG
Hamburg, GERMANY
www.trinamic.com

The TMC8460 is an EtherCAT Slave Controller (ESC) used for EtherCAT communication. It provides the interface for data exchange between EtherCAT master and the slave's local application controller.

TMC8460 also provides a large set of complex real-time IO features in hardware with focus on motor and motion control applications and systems.

Focus

- Easiest to use ESC / Simplicity / Industrial
- License-free / requirements for customer
- Robust / Availability / Flexibility
- Automation / Drives / Robotics / Semiconductor / Multi Axis Systems / Motor & Motion Control

Features

- Standard compliant EtherCAT Slave Controller register set with 2 MII ports, 6 FMMU, 6 Sync Managers, Distributed clocks (64 bit), 16 Kbyte ESC RAM size
- External I²C EEPROM
- SPI Process Data Interface (PDI) with up to 30Mbit/s
- SPI interface for Trinamic Multi-Function and Control IO Block (MFCIO) with up to 30Mbit/s
- Optional Device Emulation mode
- Trinamic Multi-Function and Control IO Block (MFCIO)
    - 8 Digital general purpose IO, individually configurable
    - Incremental encoder input (ABN), single ended
    - Step & direction output with internal step rate generator
    - 3-ch PWM block with configurable frequency, duty cycle, dead times
    - Generic SPI master interface with up to 4 slaves, e.g., to connect Trinamic ICs
    - Configurable IRQ and event signal
    - Configurable watchdog for outputs and inputs
    - Simple configuration via EEPROM or from MCU
- 16MHz CLK output, e.g., for MCU or Trinamic ICs or other peripherals
- Operating voltages: 3V3 and 1V2
- Industrial temperature range: -40°C to +100°C
- Package: VFGG400, 17mmx17mm Very Fine Pitch Ball Grid Array, 0.8mm pitch

Table of Contents

List of Figures

List of Tables

1 Abbreviations

AL	
BOOT	Special boot state of the EtherCAT state machine
CS	Chip Select (SPI bus signal)
ECAT	EtherCAT (or sometimes used for EtherCAT Master Interface)
EEPROM	Electrically Erasable Programmable Read Only Memory. Non-volatile memory used to store EtherCAT Slave Information (ESI). Connected to the SII
ENI	EtherCAT Network Information file, holds information on the complete EtherCAT bus structure and its connected slaves
EoF	End of Frame
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information, stored in SII EEPROM, holds slave specific configuration information
ETG	EtherCAT Technology Group, www.ethercat.org
EtherCAT	Ethernet in Control and Automation Technology
FCS	Frame Check Sequence
FMMU	Fieldbus Memory Management Unit
GPI	General Purpose Input(s)
GPO	General Purpose Output(s)
I2C	Inter-Integrated Circuit, serial bus used for SII EEPROM connection
INIT	Initial state of the EtherCAT state machine
IRQ	Interrupt Request
MAC	Media Access Control layer
MCU	Microcontroller Unit
MFCIO	Multi Function and Control Input Output
MI	(PHY) Management Interface
MII	Media Independent Interface: Standardized interface between the Ethernet MAC and PHY
OP	Operational state of the EtherCAT state machine
PDI	Process Data Interface or Physical Device Interface: an interface that allows access to ESC from the process side
PDO	Process Data Object
PHY	Physical layer device that converts data from the Ethernet controller to electric or optical signals
PREOP	Pre-operational state of the EtherCAT state machine
PWM	Pulse Width Modulation
RAM	Random Access Memory. ESC have User RAM and Process Data RAM
RX	Receive path
SAFEOP	Safe operational state of the EtherCAT state machine
SII	EtherCAT Slave Information Interface
SM	SyncManager
SoF	Start of Frame
SPI	Serial Peripheral Interface
TX	Transmit path
µC	
XML	Extensible Markup Language: Standardized definition language that can be interpreted by nearly all parsers.
S/D	Step and Direction interface
PDRAM	Process Data RAM of the ESC
MBx	Memory Block x
IEC	International Electrotechnical Commission
ESM	EtherCAT State Machine

2 Principles of Operation

2.1 Key Concepts

2.1.1 General Information on EtherCAT

EtherCAT (Ethernet in Control and Automation Technology) has been developed and patented by Beckhoff. It is an Ethernet-based technology for data transmission and application control in real time. All process data for all connected nodes are transmitted in a single frame. All nodes connected to the bus interpret, process, and modify their data „on the fly". Ethernet frames are not buffered inside a node but are directly forwarded with minimum additional delay. Maintaining a strict master-slave communication, data exchange utilizes mailbox mechanisms and PDOs (Process Data Objects).

To ensure real-time behavior, frame processing and forwarding requires special hardware.
This special hardware is called ESC (EtherCAT Slave Controller) like the Trinamic TMC8460, TMC8461, or TMC8462.

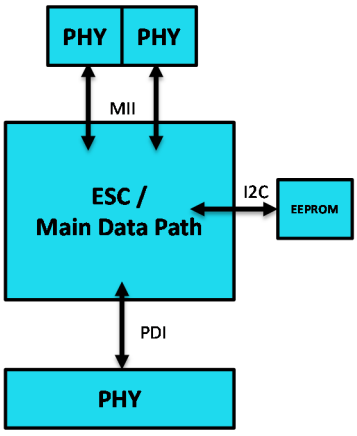
EtherCAT does not require software interaction for data transmission inside the slaves. EtherCAT only defines the MAC layer (similar to CAN). Higher layer protocols are implemented in software on microcontrollers connected to the ESC.

The ETG takes care for standardization activities and conformance testing. EtherCAT is integrated into the following major standards: IEC 61158 (protocols and services), IEC 61784-2 (communication profiles for devices), IEC 61800-7 (drive profiles and communication), SEMI standard E54.20 (since 2007).

For detailed information on the EtherCAT technology, the EtherCAT core mechanisms, and major features we refer to the official standard documentations and guidelines available from ETG (www.ethercat.org, ETG.1000), IEC (http://www.iec.ch, see numbers above), and Beckhoff (http://www.beckhoff.de, technical specification).

2.1.2 EtherCAT Slave Controller (ESC)

The TMC8460 is a standard-conform dedicated EtherCAT Slave Controller providing EtherCAT MAC layer functionality to EtherCAT slaves.
It connects via standard of-the-shelf Ethernet PHYs to the physical medium and provides a digital control interface to a local application controller while also providing the option for standalone operation.

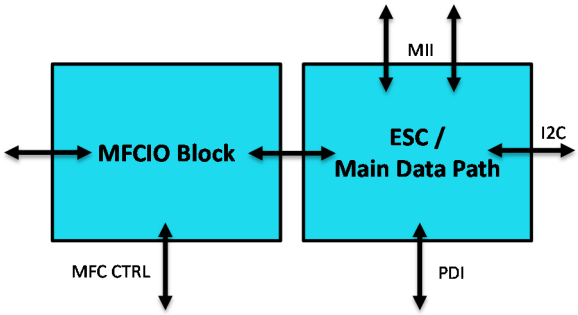


MAJOR ETHERCAT FEATURES

- Ethernet PHY interface: 2 x MII
- 6 FMMUs & 6 Sync Managers
- 16 Kbyte Process Data RAM
- 64 bit Distributed Clocks
- I²C interface for external EEPROM
- SPI Process Data Interface (PDI) with up to 30Mbit/s / optional Device Emulation mode

2.1.3 Trinamic Multi-Function and Control IO Block

Besides the proven EtherCAT functionality and the main EtherCAT data path, TMC8460 comes with a dedicated hardware block providing a configurable set of complex real-time IO functions to smart embedded systems. This IO functionality is called Multi-Function Control and IO block – MFCIO. Its special focus is on motor and motion control applications and systems while it is not limited to this application area.
The MFCIO block combines various functional sub-blocks that are helpful in an embedded design to reduce complexity, simplify the bill of materials (BOM), and to provide hardware acceleration to compute intensive tasks or time critical tasks. These functions can be used from the local application controller using a dedicated SPI interface or can directly be mapped into the Process Data RAM for direct access by the EtherCAT master.



General Purpose IOs

- There are up to 8 outputs or up to 8 inputs
- Each IO is individually configurable

Incremental Encoder Unit

- Incremental encoder inputs (ABN) with configurable counting constant, polarity, N-signal behavior and latch on N-signal
- 32 bit count register

Step & Direction Unit

- Simple internal step rate generator
- Configurable step pulse width and polarity
- Continuous mode or one-shot mode with configurable step number
- Counter for steps that have been done

3-ch PWM

- configurable frequency, duty cycle, polarity, dead times, polarity per channel

SPI Master Interface

- To directly connect to a TMC driver/controller or other SPI slaves
- Up to 4 slaves
- Configurable speed, mode, datagram width up to 64 bits (longer datagrams are possible)

IRQ / Event Output

- Common IRQ signal to indicate various events triggered by the MFCIO block
- Mask register to enable/disable certain event triggers

WATCHDOG

- Configurable for all inputs and outputs
- Outputs will be assigned with configurable level @ watchdog event
- Inputs will trigger a watchdog event only
- ECAT SoF and PDI SPI Chip Select can be monitored with watchdog as well

EMERGENCY SWITCH INPUT

- If used all functional outputs are set to a configurable safe state when the switch is not actively driven high
- Low active: must be pulled high for normal operation if used.

2.2 Configuration Options

The TMC8460 must be configured after power-up for proper operation. The EtherCAT part is automatically configured using configuration data from the connected I2C EEPROM.
The MFCIO block can also be configured using EEPROM configuration data. The EEPROM must therefore contain additional configuration data with category '1', which is automatically copied to ESC configuration RAM at addresses 0x0580:0x05FF.
Another way to configure the MFCIO block is to directly write the configuration bits to this RAM area using the ECAT or the PDI interface.
The MFCIO block can be used directly by a local application controller independent of the EtherCAT data path. Therefore, no upfront configuration is required since the MFCIO blocks comes with a dedicated SPI interface allowing complete access to its functions and local register set.

2.3 Control Interfaces

2.3.1 Ethernet Interface

For connection to the Ethernet physical medium and to the EtherCAT master, the TMC8460-BI offers two MII ports (media independent interface) and connects to standard 100Mbit/s Ethernet PHYs or 1Gbit/s Ethernet PHYs running in 100Mbit/s mode.

2.3.2 Process Data Interface

The Process Data Interface (PDI) is an SPI interface. The TMC8460-BI provides an SPI slave interface with ca. 30Mbit/s to connect to a local MCU or application controller. Typically, the local application controller contains the EtherCAT slave stack to control the EtherCAT state machine and to process state change requests by the EtherCAT master. Pulling the external configuration pin PDI_EMULATION high switches to Device Emulation mode. In Device Emulation mode, the TMC8460-BI can be used in standalone mode without MCU since state change requests by the master are directly forwarded to the state registers inside the TMC8460's ESC part. The PDI SPI interface remains active and can be used by an MCU in device emulation state.

2.3.3 Multi-Function and Control IO Block Interface

The MFCIO block of the TMC8460-BI comes with a dedicated SPI slave interface to allow direct access from a local application controller. It is called MFC CTRL SPI interface. This interface to the MFCIO block's functions is always available even if the EtherCAT state machine is currently not in operational state (OP). Protocol structure and timing are identical to the PDI SPI.

2.3.4 SPI Bus Sharing

Both SPI interfaces – PDI SPI and MFC CTRL SPI – can share the same SPI bus signals using two chip select signals. This reduces overall number of signals on the PCB and requires only one SPI interface on the local MCU. The external configuration pin SHARED_SPI_BUS needs to be pulled high for bus sharing. In this case, the PDI SPI bus is used as shared bus interface together with the chip select line of the MFC CTRL SPI interface.

2.3.5 Configuration Inputs

External package pins allow for selection of configuration options that typically do not change during operation by directly connecting them to 3V3 or ground. These package pins can also be controlled by GPIOs of the local application controller.
These options are for example external EEPROM's size, PHY addressing and MII TX clock shift configuration, polarities of the PHYs' link indicator, device emulation mode and enabling of the 16MHz clock output.

2.3.6 EEPROM Interface

An EEPROM containing boot-up configuration data is required for ESC operation. The EEPROM must come with a standard I2C interface and connects to the PROM interface of the TMC8460. EEPROMs of different size can be used. There is a difference in the I2C protocol when EEPROM parts with >16kbits memory size are used.

2.4 Software View

As seen from an EtherCAT master system, the TMC8460-BI is part of an EtherCAT slave using the register set and functionality according to the EtherCAT standard. It works together with other EtherCAT slaves on the same bus and is accessible via the Ethernet physical medium. According to the slave configuration (ESI) and network configuration (ENI)

As seen from the local application controller, the TMC8460-BI is a peripheral SPI slave device with a number of control and status registers that are accessible using the PDI SPI interface for the main EtherCAT data path or the MFC CTRL SPI interface for the MFCIO block.
For proper EtherCAT functionality, the local application controller runs the EtherCAT slave stack to process master requests regarding state changes in the state machine for example using the PDI SPI interface. In device emulation mode the PDI SPI interface is not used since master requests are handled inside the ESC.

The MFCIO block functions can directly be used with the MFC CTRL SPI interface. This can be done even without using the EtherCAT slave controller part.

Trinamic's TMCL IDE (http://www.trinamic.com/software-tools/tmcl-ide) can be used to access the device with the TMC8460-BI evaluation board. All registers are accessible via the two SPI interfaces. The configuration memory area for the MFCIO block can be read and modified.

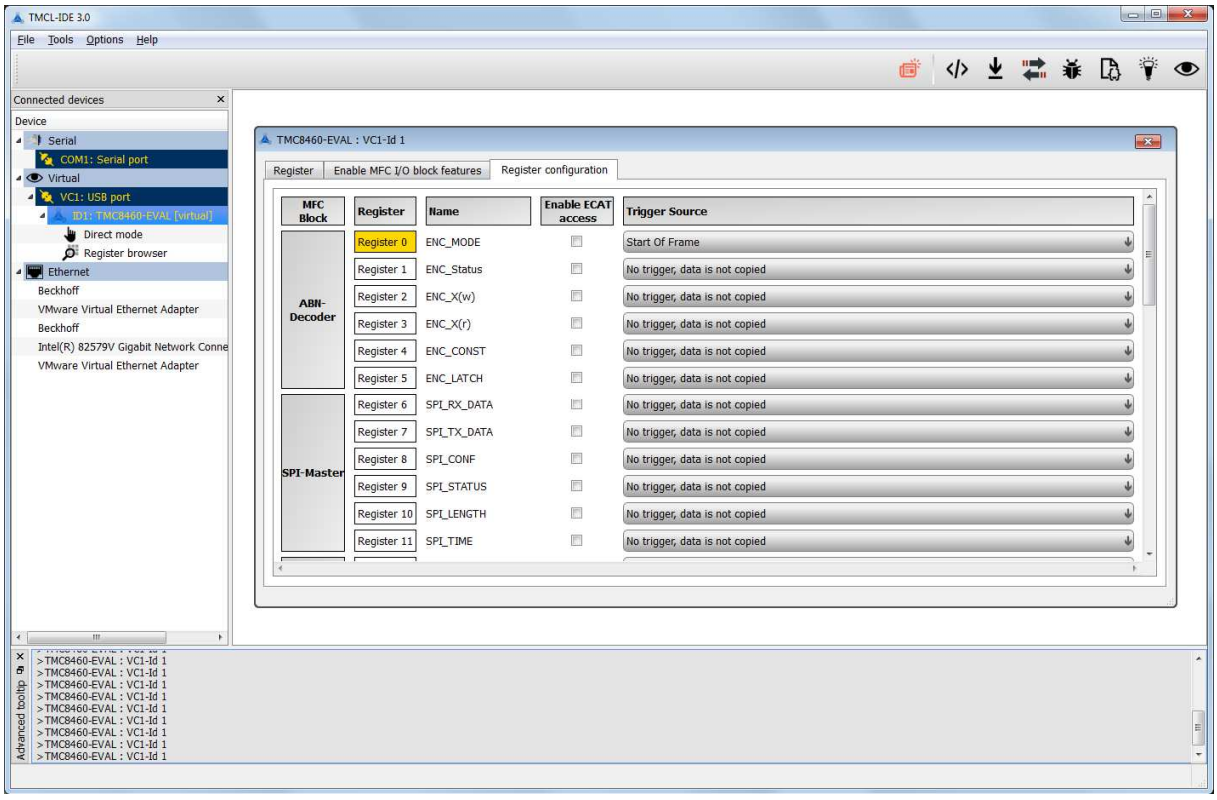


Figure 1 - TMCL-IDE with direct register access to the TMC8460-BI on its evaluation board

A wizard helps and simplifies the configuration and setup of the TMC8460-BI to your specific needs and provides code examples for your configuration to be used inside you microcontroller firmware and the EEPROM for startup configuration.

Figure 2 - Wizard Start Screen

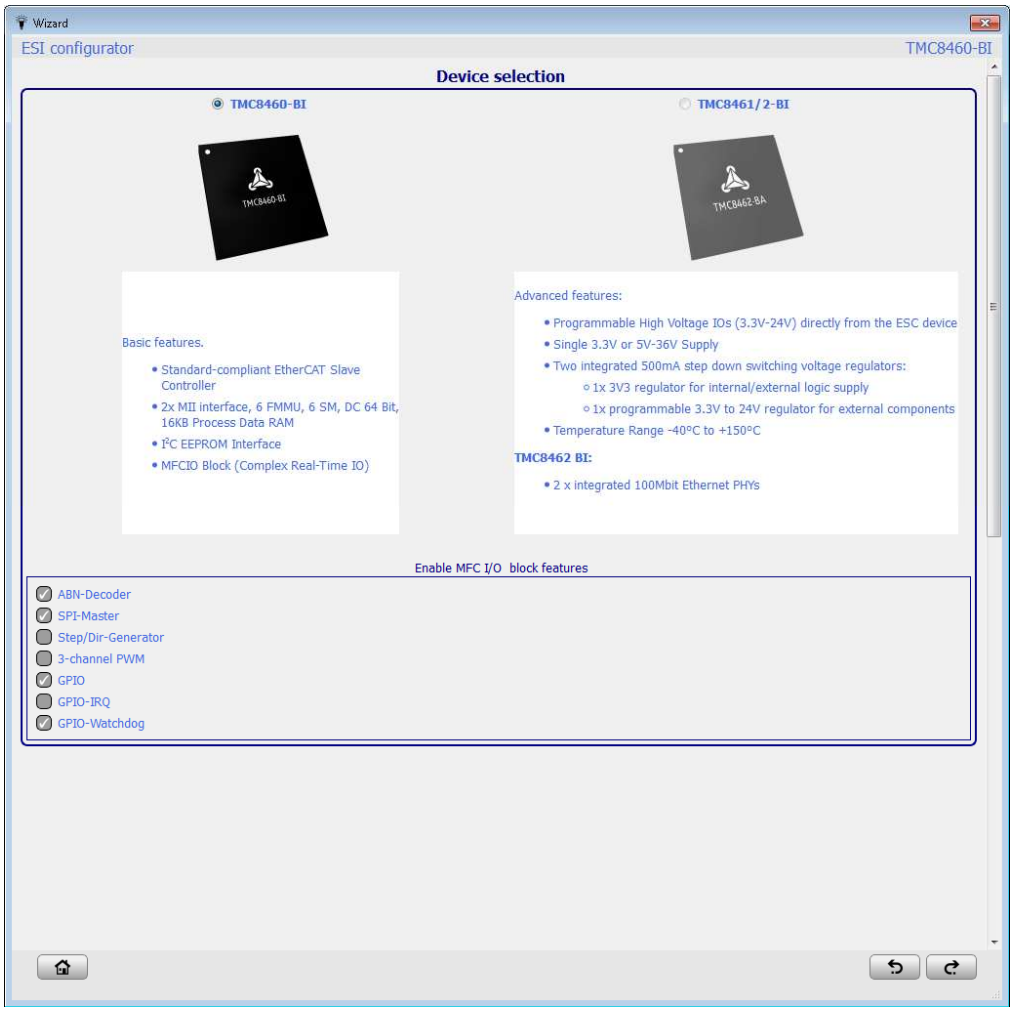


Figure 3 - Wizard Device Selection and Feature Selection

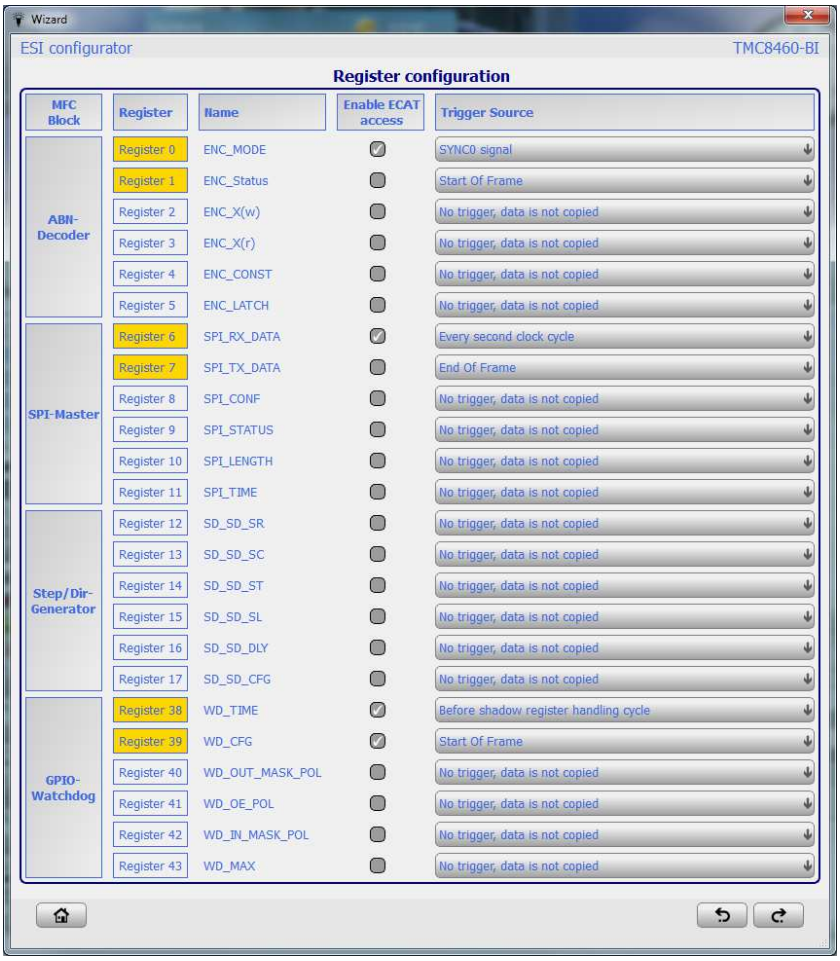


Figure 4 - Wizard Register Selection and Configuration View

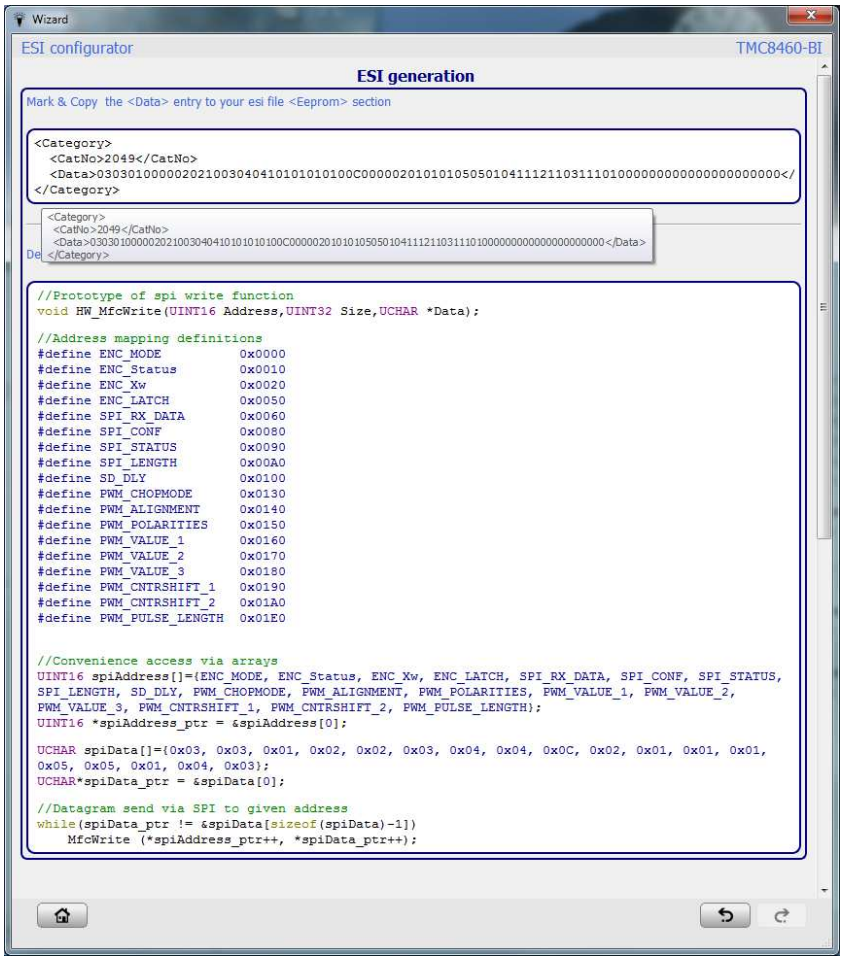


Figure 5 - Wizard output view with EEPROM configuration string and firmware C-code snippets

3 Device Usage and Handling

3.1 Sample Block Diagrams

The TMC8460 allows for flexible system architectures using a microcontroller running the Slave Stack Code (SSC) or using Device Emulation mode without a microcontroller.
The following examples show typical system architectures using the TMC8460.

3.1.1 Typical EtherCAT Slave architecture

The first application diagram shows the TMC8460 in a typical straightforward architecture. The PHYs connect to the TMC8460 using MII interface. Both PHYs and the TMC8460 have the same 25MHz clock source (see Section 3.9.2). The I2C EEPROM is connected to the TMC8460 and contains boot-up configuration required by the ESC after reset or power-cycling. The EEPROM is optionally connected to the µC to allow EEPROM updates via the MCU's firmware.
The µC connects to the TMC8460 using an SPI bus interface to the PDI SPI. The local application is connected to the µC and is controlled by the application layer inside the µC. The application interface depends on the application and is a generic placeholder in this diagram. In this example the MFCIO IO block is not used.

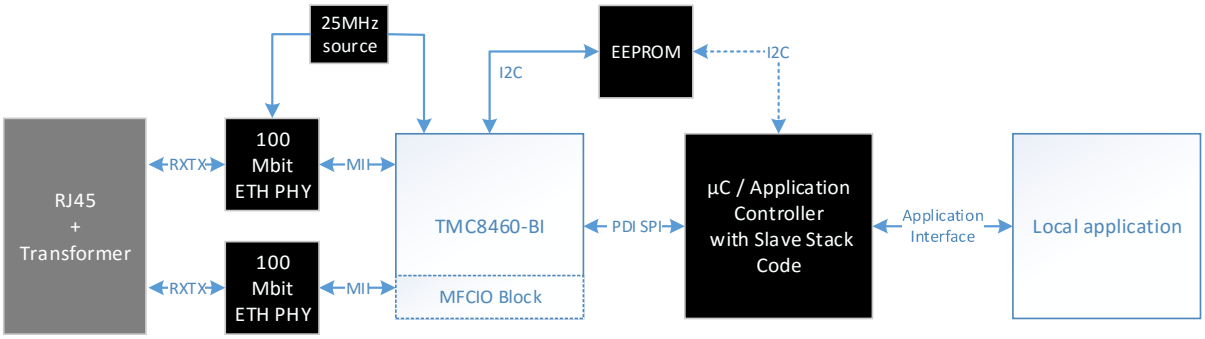


Figure 6 - Application diagram using only the local application controller to interface the application

3.1.2 MFCIO block based Microcontroller Architecture

The second application diagram shows a similar architecture with µC but with extensive use of the MFCIO block features. The special functions of the MFCIO block allow relocating functionality from the µC to the TMC8460. That is, certain compute intense and time-critical functions are moved from software to hardware. The application layer in the µC can focus on interfacing to the real-time bus and for high-level control tasks of the application.
For example, an incremental encoder can directly be connected to the MFCIO block. The µC only reads back the actual position via the dedicated SPI interface MFC CTRL SPI. Additionally, SPI slave chips are directly connected to the MFCIO and not the µC, for example Trinamic's dedicated smart stepper motor drivers, dedicated hardware motion controllers, and simple S/D stepper motor drivers. The MFCIO block master SPI interface, the PWM functions, the S/D function, and the 16MHz clock output are used in this case. The application controller does not need to implement these firmware functions and interfaces but uses the available resources of the TMC8460 instead. Software development is simplified. Other application parts not covered by the MFCIO block still connect to the microcontroller.

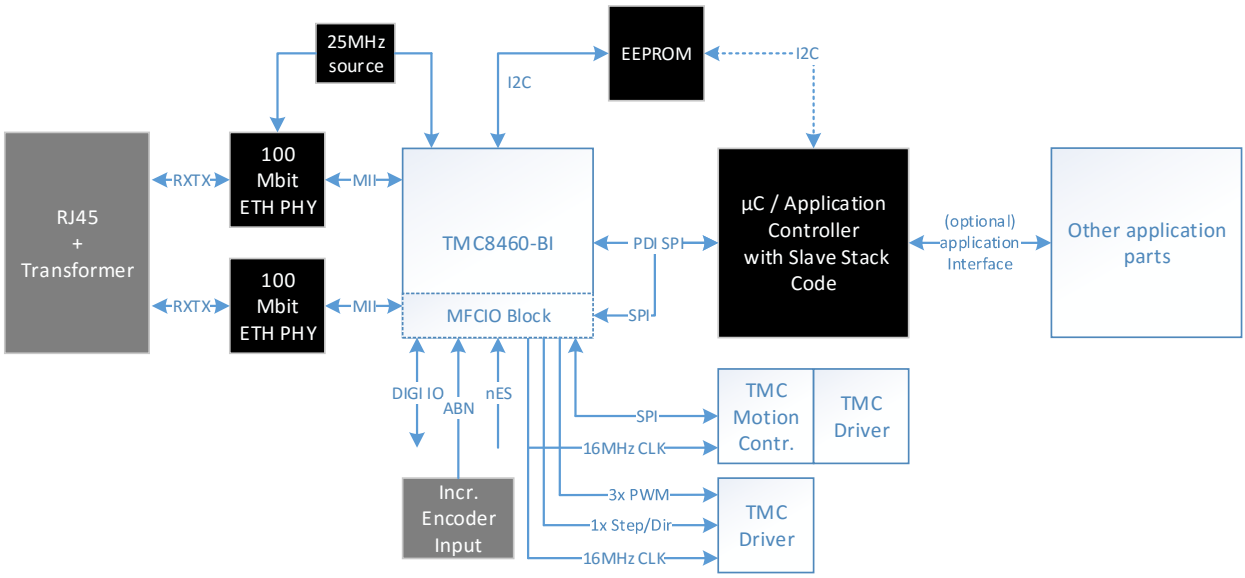


Figure 7 - Application diagram using the MFCIO block features to reduce software overhead and provide real-time hardware support to the MCU. Other application parts may still be connected to the MCU.

3.1.3 Device Emulation Example

The third application diagram shows a more compact architecture using device emulation mode. No µC is required. State machine change requests by the EtherCAT master are directly processed inside the ESC. The MFCIO block is the only application interface available in this architecture and provides the features mentioned under Section 2.1.3. For example, a simple stepper motor slave with hardware motion controller and encoder feedback can be set up without using a µC in the system by only using the features provided by the TMC8460. Since the registers and functions of the MFCIO block can directly be mapped into the PDRAM the EtherCAT master can control the slave.

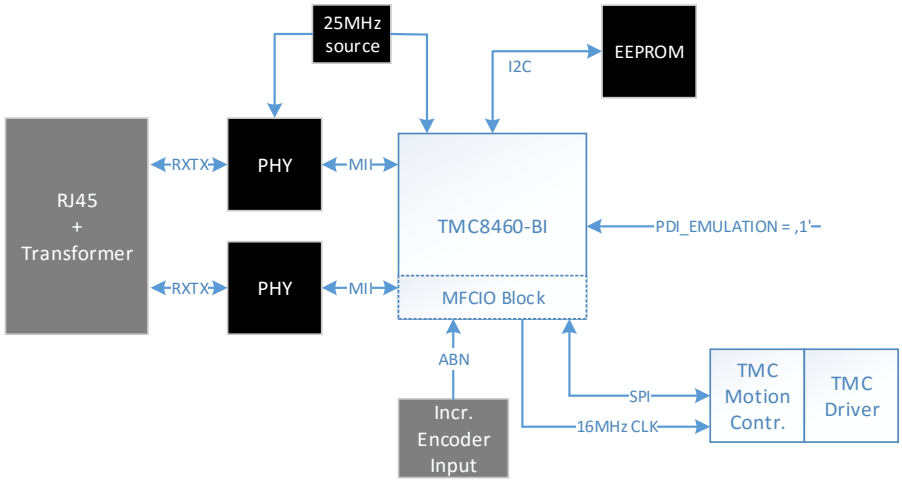
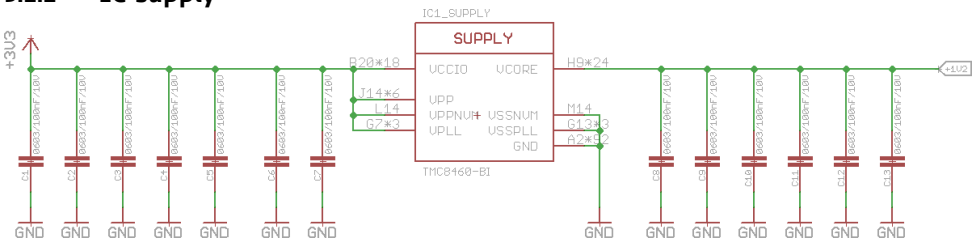


Figure 8 - Application diagram without MCU. The TMC8460 is used in device emulation mode. SPI slave chips and other application peripherals can be connected to the MFCIO block. The EtherCAT master can directly control all the application functions.

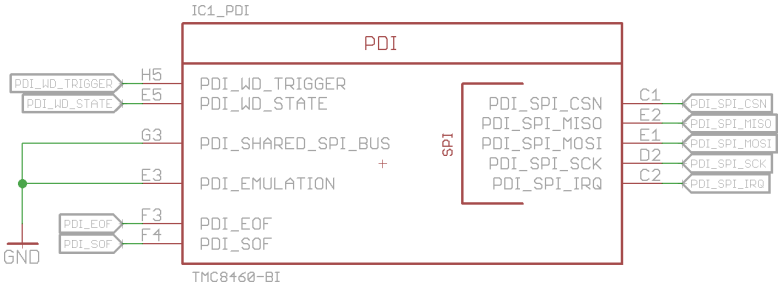
3.2 Samples Circuits

3.2.1 IC supply



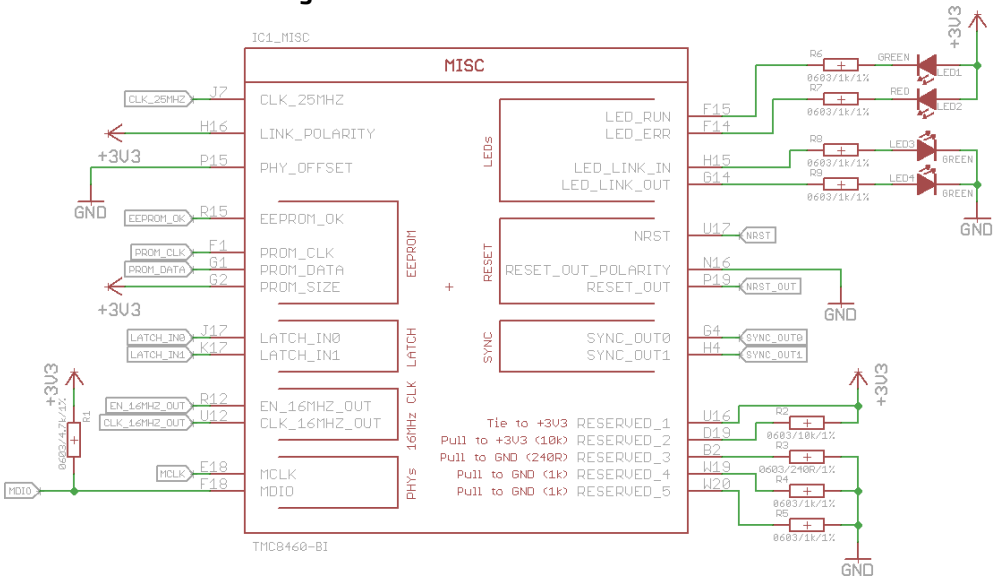
Only a minimal amount of decoupling capacitors is shown here.
If possible every supply pin (1.2V and 3.3V) should have a separate 100nF capacitor connected between it and GND as close to the pin as possible.
Larger capacitor values can be used on the 3.3V and 1.2V supply rails for increased stability.

3.2.2 PDI interface



This is the default configuration for the PDI SPI interface; both PDI_EMULATION and PDI_SHARED_SPI_BUS are tied to GND.
PDI__EMULATION = 0 means that the processor connected to the PDI SPI pins has full control over the ESC registers, the memory and the EtherCAT state machine.
PDI_SHARED_SPI_BUS = 0 means that the signals of the PDI SPI and MFC CTRL SPI buses are completely separate.
The processor can also use the extra signals for Start-/End-Of-Frame and the PDI Watchdog.

3.2.3 Miscellaneous signals



CLK_25MHZ is the clock input, which should be the same signal as the clock for both PHYs. The traces from the oscillator to the TMC8460 and the PHYs should have approximately the same length to avoid timing problems.

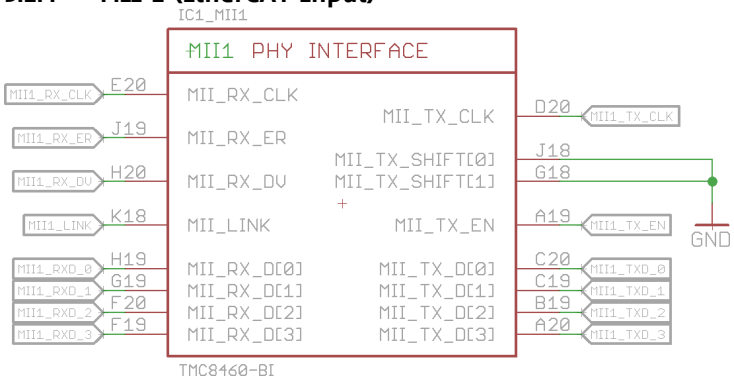
LATCH_IN0, LATCH_IN1, SYNC_OUT0, SYNC_OUT1, NRST_OUT, EN_16MHZ_OUT and CLK_16MHZ_OUT are optional signals that can be used depending on the specific use case.

Configuration Pins with their setting in this example:

LINK_POLARITY	1 (+3.3V)	MII_LINK signal from PHYs is high active
PHY_OFFSET	0 (GND)	PHY address offset is 0
PROM_SIZE	1 (+3.3V)	EEPROM is 32kbit or larger (4Mbit max.)
RESET_OUT_POLARITY	0 (GND)	The RESET_OUT signal is low active

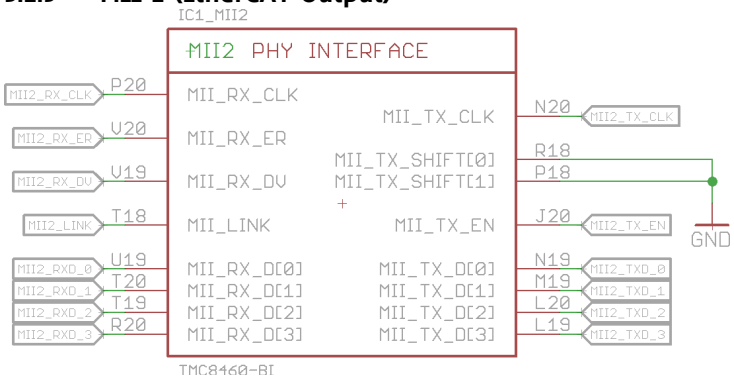
External resistors are required on the MDIO line and for four (4) of the reserved pads.

3.2.4 MII 1 (EtherCAT Input)



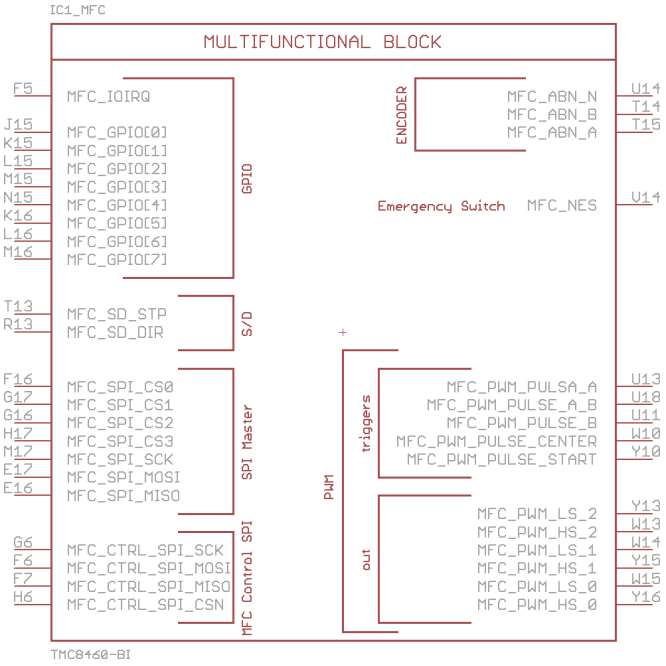
These are the connections to the first PHY, representing the input port of the EtherCAT device.

3.2.5 MII 2 (EtherCAT Output)



These are the connections to the second PHY, representing the output port of the EtherCAT device.

3.2.6 MFC I/Os



The I/Os of the MFC block are shown unconnected here as the actual connections depend on the intended use.

3.3 Pinout and Pin Description

The following table contains the pin description and required pin connections of the TMC8460-BI for the Very Fine Ball Pitch Grid Array package VF400.

Pins not listed in this table are not connected (n.c.) / leave open.

PKG. Pin	Pin Name / Function	DIR	Functional Description / Comments
U17	NRST	I	Low active system reset input
J7	CLK_25MHz	I	25MHz Reference Clock Input, connect to clock source with <25ppm or better, Typically same clock source as used for the PHYs
U12	CLK_16MHz_OUT	O	16MHz auxiliary clock output, enabled by EN_16MHz_OUT
R12	EN_16MHz_OUT	I	Enable signal for 16 MHz auxiliary clock output: 0 = off, 1 = on, 16 MHz available at CLK_16MHz_OUT
P19	RESET_OUT	O	TMC8460 auxiliary output / active level defined by RESET_OUT_POL, reset by ECAT (reset register 0x0040), RESET_OUT has to trigger nRESET, which clears RESET_OUT.
N16	RESET_OUT_POL	I	Configures active level of the RESET_OUT signal: 0 = low active, 1 = high active
R15	EEPROM_OK	O	Signal indicating that EEPROM has been loaded, 0 = not ready, 1 = EEPROM loaded
F1	PROM_CLK	O	External I2C EEPROM clock signal, Use 1K pull up resistor to 3.3V
G1	PROM_DATA	I/O	External I2C EEPROM data signal, Use 1k pull up resistor to 3V3
G2	PROM_SIZE	I	Selects between two different EEPROM sizes since the communication protocol for EEPROM access changes if a size > 16k is used (an additional address byte is required then). 0 = up to 16K EEPROM, 1 = 32 kbit-4Mbit EEPROM
G4	SYNC_OUT0	O	Distributed Clocks synchronization output 0, Typically connect to MCU
H4	SYNC_OUT1	O	Distributed Clocks synchronization output 1, typically connect to MCU
J17	LATCH_IN0	I	Latch input 0 for distributed clocks, connect to GND if not used.
K17	LATCH_IN1	I	Latch input 1 for distributed clocks, Connect to GND if not used.
F14	LED_ERR	O	Error Status LED, connect to red LED (Cathode) 0 = LED on, 1 = LED off
H15	LED_LINK_IN	O	Link In Port Activity, connect to green LED (Anode) 0 = LED off, 1 = LED on
G14	LED_LINK_OUT	O	Link Out Port Activity, connect to green LED (Anode) 0 = LED off, 1 = LED on

