TMC8670 Datasheet

Hardware Version V1.0 | Document Revision V1.00 • 2021-Feb-04

The TMC8670 is a CANopen-over EtherCAT (CoE) field oriented control (FOC) servo controller for torque, velocity, and position control. It comes with a fully integrated EtherCAT Slave Controller (ESC), a flexible sensor engine for different position feedback and current sensing options, as well as a complete CANopen-over-EtherCAT firmware stack for the CiA DS402 device profile. TMC8670 is a building block that enables a servo controller with only a couple of components.

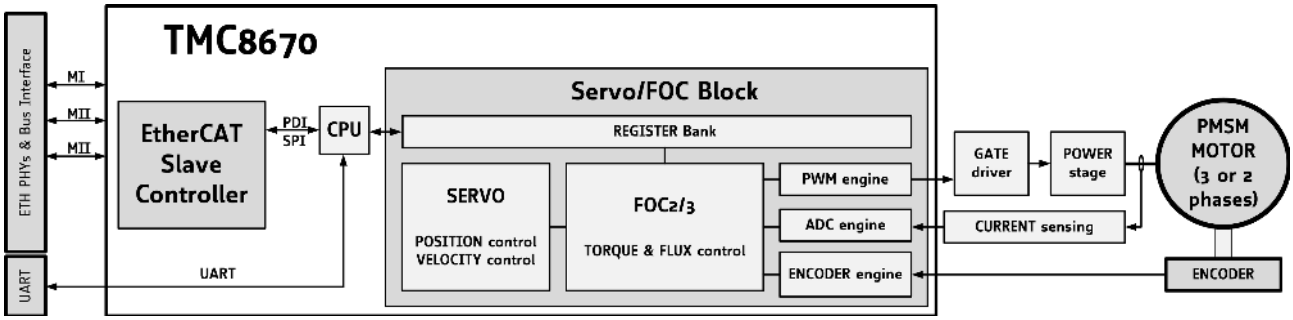
Features

- Field Oriented Control (FOC) Servo Controller
- Torque Control (FOC), Velocity Control, Position Control
- Sensor Engine (Hall analog/digital, Encoder analog/digital)
- Support for 3-Phase PMSM and 2-Phase Stepper Motors
- PWM Engine including SVPWM
- Integrated EtherCAT Slave Controller, CoE protocol CiA 402 drive profile
- UART interface

Applications

- Robotics
- Semiconductor Handling
- Factory Automation
- Laboratory Automation
- Manufacturing
- IIoT Applications

Simplified Block Diagram



Read entire documentation.

TRINAMIC
MOTION CONTROL
Now part of Maxim Integrated

Contents

1 Product Features

TMC8670 is a highly integrated SoC providing the interface between an EtherCAT real-time field bus and the local drive application. It includes the real-time MAC layer for EtherCAT, the application software stack for the CiA DS402 CANopen device profile, and the complete servo control block in dedicated hardware with interfaces to ADCs and position feedback.
TMC8670 offers an extremely high function density in a small scale package.

Advantages:

- Fully standard compliant and proven EtherCAT Slave Controller and State Machine
- Highly integrated Servo Controller with rich feature set vs. package size
- Robust silicon technology
- Saves board space & reduces BOM
- Long-term availability

Major Features:

- Integrated EtherCAT Slave Controller with 2 MII ports for Ethernet bus interfacing
- Complete firmware stack with EtherCAT State Machine and CANopen over EtherCAT stack based on CiA DS402 device profile
- Firmware update via EtherCAT or via UART
- Fully integrated hardware servo controller with field-oriented control and rich interface support
- Two digital incremental encoder interfaces
- Analog SinCos encoder interface
- Digital hall sensor interface
- Analog hall sensor interface
- Flexible ADC interface to connect to external SPI ADCs or delta sigma modulators
- Industrial temperature range -40°C to +125°C
- Package: 325-pin BGA chip scale package with 0.5mm pitch, 11mm x 11mm

2 Order Codes

Order Code	Description	Size
TMC8670-BI	TMC8670 Advanced EtherCAT® Servo Controller in 325-pin BGA chip scale package with 0.5mm pitch	11mm x 11mm
TMC8670-EVAL	Evaluation Board for TMC8670-BI, compatible with the modular Landungsbruecke system, RJ45 twisted pair copper interface	79mm x 85mm

Table 1: TMC8670 order codes

Trademark and Patents

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

3 Principles of Operation / Key Concepts

3.1 General Device Architecture

Figure 1 shows the general device architecture and major connections of TMC8670.
The EtherCAT Slave Controller (ESC) is realized in dedicated logic and provides two MII interfaces to external Ethernet PHYs suitable for EtherCAT.
The ESC connects to the integrated microcontroller, which executes the EtherCAT State Machine (ESM) and the CiA DS402 CANopen protocol stack. A debug UART interface connects to the MCU for debugging and firmware updates.
The firmware in the MCU controls the servo and field-oriented control (FOC) block, which is completely realized in dedicated logic. All PI-loops for position, velocity, and torque are fully configurable.
The FOC block drives external gate driver, which in turn are switching a power stage for 3-phase brushless motors or 2-phase stepper motors.
The FOC block provides a set of interfaces for different types of current sensing and position feedback. Current sensing and encoders are external components to the TMC8670.

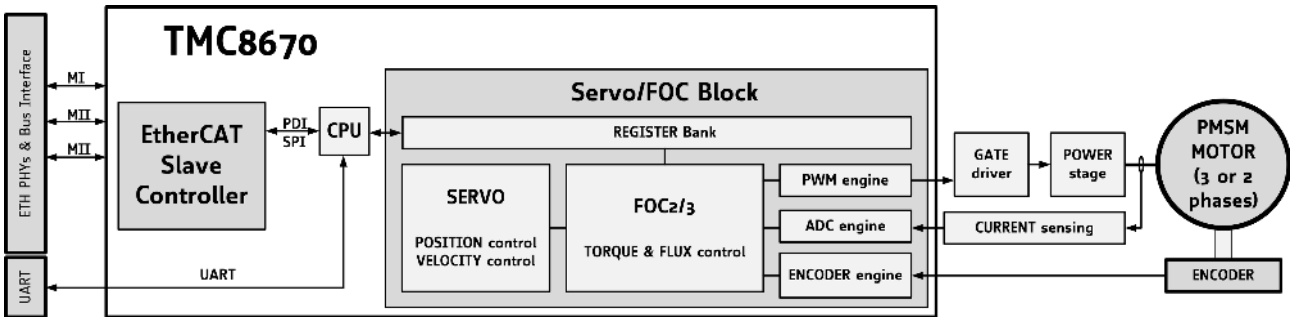


Figure 1: General device architecture

3.2 EtherCAT Slave Controller

TMC8670 contains a standard-conform and proven ESC engine providing real-time EtherCAT MAC layer functionality to EtherCAT slaves. It connects via MII interface to standard Ethernet PHYs and provides a digital control interface to a local application controller

The ESC part of TMC8670 provides the following EtherCAT-related features. More information is available in Section 7.

- Two MII interfaces to external Ethernet PHYs plus management interface
- Four Fieldbus Memory Management Units (FMMU)
- Four Sync Managers (SM)
- 4 KByte of Process Data RAM (PDRAM)
- 64 bit Distributed Clocks support
- IIC interface for an external SII-EEPROM for ESC configuration

3.3 Microcontroller and Firmware Stack

The integrated microcontroller system contains and controls the application layer of TMC8670. Thereby, the firmware is split up into a bootloader section and the application layer section. The bootloader allows

for future firmware updates. The application layer comprises the ESM to communicate with the ESC and the CANopen-over-EtherCAT (CoE) protocol stack. The CoE stack is based on the CiA DS402 device profile for drives. It controls the hardware servo/FOC controller block. The application layer also supports File-Transfer-over-EtherCAT (FoE), which is used for remote firmware updates via the EtherCAT master.

3.4 Servo/FOC Controller

The integrated servo/FOC controller is completely realized in dedicated logic. Its control registers are directly mapped into the microcontrollers address space. It offloads the microcontroller from the repetitive and time-consuming computation tasks of control loop processing, FOC Park and Clark transformations, PWM generation, and interfacing to ADCs and position feedback. The servo/FOC controller supports PWM frequencies and current loop frequencies of up to 100kHZ. It not only supports 3-phase brushless motors but also 2-phase stepper motors and single phase motors, for example DC motors.
More information is given the FOC Basics Section.

3.5 Flexible Sensor Engine

A versatile and flexible sensor engine is part of the servo/FOC controller block of TMC8670. The sensor engine handles digital hall sensors, digital incremental encoders, analog hall sensors, and analog sin-cos-sensors. Together with the relevant sensor parameters, it maps the measured sensor position to 16 bit signed values (s16) for the FOC engine.

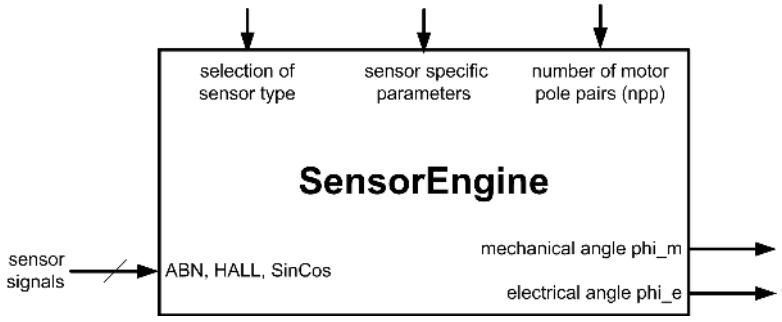


Figure 2: TMC8670 Sensor Engine maps position sensor signals to mechanical angels and electrical angels as direct input for the FOC engine.

ADC Interfaces The TMC8670 is a pure digital IC with interfaces for external ADCs. As ADC one can either select LTC2351 from Linear Technology or Delta Sigma Modulators (AD7401). As an alternative to Delta Sigma Modulators, the TMC8670 supports low cost comparators (e.g. LM339) together with some passive components to form delta sigma modulators.

Digital Encoder Interfaces The digital encoder interface support a wide range of encoders with different resolutions, signal polarities and zero pulses.

Analog Encoder Interfaces The analog encoder interface is for analog hall signals - two phase SinCos or three phase - and for analog (incremental) encoders. An interpollator for SinCos encoders is integrated.

Digital Hall Sensor Interface The digital hall signal interface enables digital hall signals for initialization of incremental encoders. The digital hall signal interface can be used directly for the FOC. For torque ripple reduction an interpolator for the digital hall signals is integrated.

Analog Hall Sensor Interface The interface for analog hall signals is the same interface as available for SinCos analog encoders.

3.6 Communication Interfaces

Field Bus Interface TMC8670 provides two MII ports to connect to 100-Mbit Ethernet PHYs that connect to the field bus. One port is the dedicated EtherCAT IN port. The second port is the dedicated EtherCAT OUT port. Depending on the physical medium (twisted pair copper or passive optical fiber) an external transformer circuit connects to the RX and TX lines.

IIC SII EEPROM Interface The IIC EEPROM interface is intended to be a point-to-point interface between TMC8670 and the SII EEPROM with TMC8670 being the master. Depending on the EEPROM's capacity the addressing mode must be properly set using the PROM_SIZE configuration pin.

Configuration of the EtherCAT Slave Controller is done during boot time with configuration information read from the SII EEPROM after reset or power cycling. This information must be (pre)programmed into the SII EEPROM. This can be done via the EtherCAT master using a so-called EtherCAT Slave Information (ESI) file in standardized XML format.

Debug UART Interfaces TMC8670 has two UART interfaces that allow for basic local debugging. The MCU UART directly connects to the microcontroller and can also be used for local firmware updates. The HW UART directly connects to the servo/FOC controller block and allows for direct control via register read/write of this function block alone. This is usable for local tuning, monitoring of the registers, and debugging.
More details on the two debug UART interfaces are given in the Debug UARTs' section

3.7 Software- and Tool-Support

Evaluation Board An evaluation board is available for the TMC8670 with standard RJ45 connectors and transformers for interfacing twisted pair copper media.

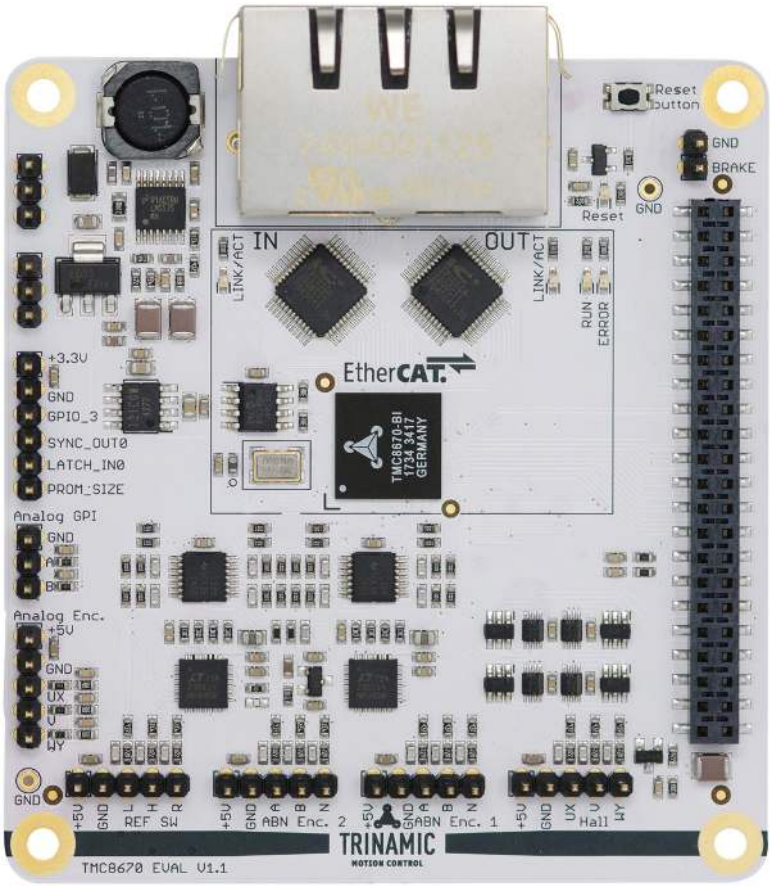


Figure 3: TMC8670 Evaluation Board

The complete board design files are available for download and can be used as reference. All information is available for download on the specific product page on TRINAMIC's website at https://www.trinamic.com/support/eval-kits/.

TMCL-IDE The TMCL-IDE is TRINAMIC's primary tool (for Windows PCs) to control TRINAMIC modules and evaluation boards. Besides, it provides feature like remote firmware updates, module monitoring options, and specific Wizard support. The TMCL-IDE can be used along with TRINAMIC's modular evaluation board system.

Info	The TMLC-IDE is not an EtherCAT master system!

The TMC8670-EVAL can be accessed via the UART interface of the evaluation board to try out the servo functions without using an EtherCAT master in the first place.

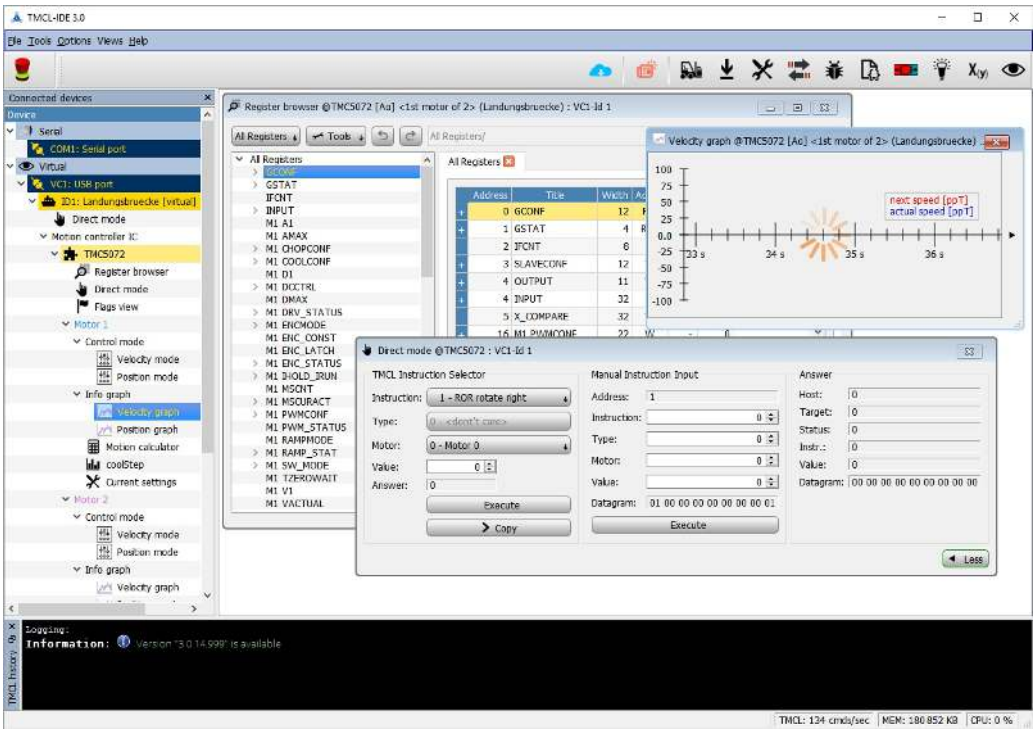


Figure 4: TMCL-IDE

The latest version and additional information is available for download from TRINAMIC's website at https://www.trinamic.com/support/software/tmcl-ide/.

4 Device Pin Definitions

4.1 Pinout and Pin Coordinates of TMC8670-BA

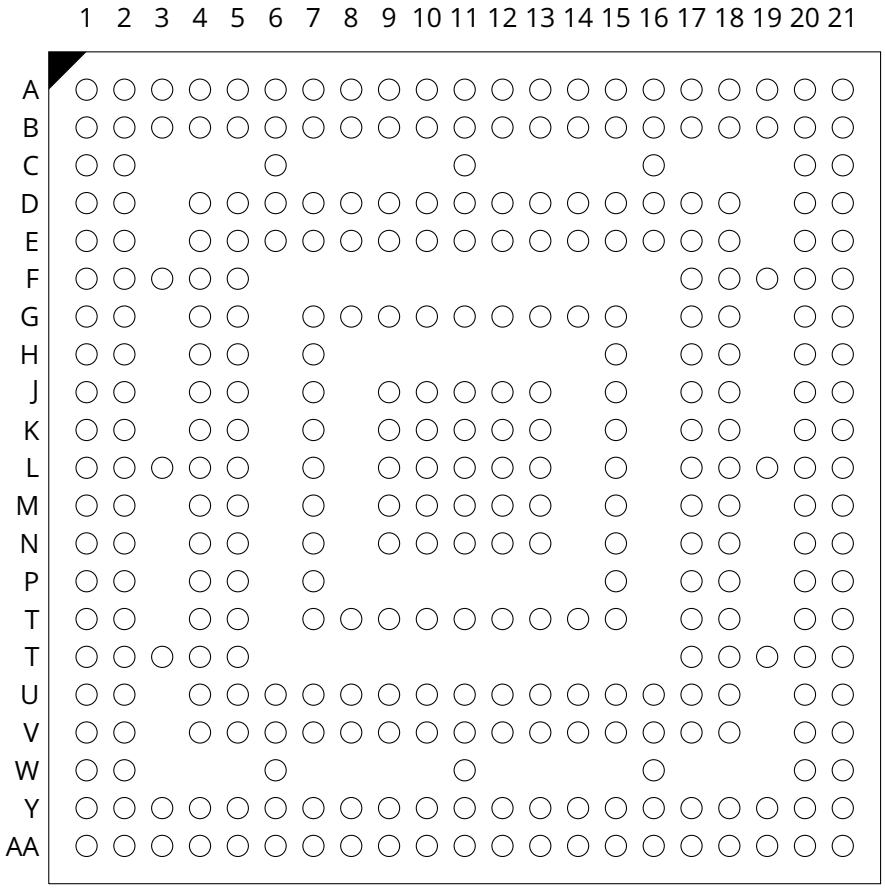


Figure 5: TMC8670-BI Pinout top view

4.2 Pin Numbers and Signal Descriptions

Pins not listed in the following table are N.C. (not connected).
Pin types are I = input, O = output, PU = has pull-up, PD = has pull-down.

Name	Pin	Type	Function
General Signals			
NRESET	M9	I	Low active system reset, pull up to VDD_3V3 with 10K
CLK_25MHZ	P1	I	25MHz Reference Clock Input, connect to clock source with <25ppm or better, typically same clock source as used for the ETH PHYs.
CLKOUT_25MHZ	H17	O	

Name	Pin	Type	Function
EtherCAT SII EEPROM IOs			
PROM_CLK	F1	O	External IIC SII EEPROM clock signal, **use 1K pull up resistor to VDD_3V3**
PROM_DATA	F2	I/O	External IIC SII EEPROM data signal, **use 1k pull up resistor to VDD_3V3**
PROM_SIZE	K1	I, PU	Selects between two different EEPROM sizes since the communication protocol for SII EEPROM access changes if a size > 16kBit is used (an additional address byte is required then). 0 = up to 16kBit EEPROM, 1 = 32 kBit-4Mbit EEPROM, has weak internal pull-up

EtherCAT Status LEDs			
LED_RUN	D1	O	ESM Run Status LED, connect to green LED (Anode) 0 = LED off, 1 = LED on
LED_ERR	D2	O	ESM Error Status LED, connect to red LED (Anode) 0 = LED off, 1 = LED on
LED_LINK_IN	E1	O	ETH Link In Port Status and Activity, connect to green LED (Anode) 0 = LED off, 1 = LED on
LED_LINK_OUT	F3	O	ETH Link Out Port Status and Activity, connect to green LED (Anode) 0 = LED off, 1 = LED on

Distributed Clocks Synchronization			
LATCH_IN0	K2	I, PD	Distributed Clocks Latch Input, has weak internal pull-down
SYNC_OUT0	K4	O	Distributed Clocks Synchronization Output

Name	Pin	Type	Function
MII Interface to external ETH PHY (EtherCAT IN Port)			
MII1_LINK	F18	I	Link indication input
MII1_RXCLK	G18	I	Receive clock
MII1_RXD[0]	F19	I	Receive data bit 0
MII1_RXD[1]	F20	I	Receive data bit 1
MII1_RXD[2]	E21	I	Receive data bit 2
MII1_RXD[3]	E20	I	Receive data bit 3
MII1_RXDV	G21	I	Receive data valid signal
MII1_RXER	G20	I	Receive error signal
MII1_TXCLK	E18	I	Transmit clock
MII1_TXD[0]	D21	O	Transmit data bit 0
MII1_TXD[1]	C21	O	Transmit data bit 1
MII1_TXD[2]	C20	O	Transmit data bit 2
MII1_TXD[3]	B21	O	Transmit data bit 3
MII1_TX_EN	E17	O	Transmit enable
MII Interface to external ETH PHY (EtherCAT OUT Port)			
MII2_LINK	K18	I	Link indication input
MII2_RXCLK	L18	I	Receive clock
MII2_RXD[0]	N21	I	Receive data bit 0
MII2_RXD[1]	M20	I	Receive data bit 1
MII2_RXD[2]	L20	I	Receive data bit 2
MII2_RXD[3]	L21	I	Receive data bit 3
MII2_RXDV	N20	I	Receive data valid signal
MII2_RXER	M18	I	Receive error signal
MII2_TXCLK	J17	I	Transmit clock
MII2_TXD[0]	L19	O	Transmit data bit 0
MII2_TXD[1]	K21	O	Transmit data bit 1
MII2_TXD[2]	J20	O	Transmit data bit 2
MII2_TXD[3]	J21	O	Transmit data bit 3
MII2_TX_EN	J18	O	Transmit enable

Name	Pin	Type	Function
ETH PHY Interface Configuration Pins and Management Interface			
LINK_POLARITY	R9	I, PD	selects polarity of the ETH PHYs link signal: 0 = low active, 1 = high active
MII1_TX_SHIFT[0]	K15	I	Used for clock shift compensation on TX port
MII1_TX_SHIFT[1]	K17	I	Used for clock shift compensation on TX port
MII2_TX_SHIFT[0]	L15	I	Used for clock shift compensation on TX port
MII2_TX_SHIFT[1]	L17	I	Used for clock shift compensation on TX port
MCLK	H20	O	PHY management clock, connect all ETH PHYs to this bus
MDIO	H21	I/O	PHY management data, connect all ETH PHYs to this bus if required, **use 4K7 pull up resistor to VDD_3V3**
Motor Position Feedback Signals			
ENC_A	N1	I, PU	incremental encoder signal A
ENC_B	N2	I, PU	incremental encoder signal B
ENC_N	P2	I, PU	incremental encoder null pulse N
ENC_2_A	V6	I, PU	2nd incremental encoder signal A
ENC_2_B	V7	I, PU	2nd incremental encoder signal B
ENC_2_N	W6	I, PU	2nd incremental encoder null pulse N
HALL_UX	M4	I, PU	digital Hall signal associated to U (H1)
HALL_V	N4	I, PU	digital Hall signal associated to V (H2)
HALL_WY	P4	I, PU	digital Hall signal associated to W (H1)
ENC_ADC_CSN	L3	O	analog encoder SPI ADC LTC2351 CONV
ENC_ADC_MISO	L1	I	analog encoder SPI ADC LTC2351 SDO
ENC_ADC_SCK	L2	O	analog encoder SPI ADC LTC2351 SCK

Reference Switch Signals			
REF_SW_H	H5	I, PU	Home Reference Switch
REF_SW_L	H4	I, PU	Left Reference Switch
REF_SW_R	J4	I, PU	Right Reference Switch

Name	Pin	Type	Function
Motor and Supply Current Measurement Signals			
SPI_ADC_CSN	U8	O	analog current measurement SPI ADC LTC2351 CONV
SPI_ADC_SCK	U9	O	analog current measurement SPI ADC LTC2351 SCK
SPI_ADC_MISO	U10	I, PU	analog current measurement SPI ADC LTC2351 SDO
ADC_PHASE_MISO_2ND	U11	I, PU	analog current measurement SPI ADC LTC2351 SDO
MCLK_AENC_UX	P5	IO	DS-Mod Clock analog encoder/analog Hall U or X
MCLK_AENC_VN	R4	IO	DS-Mod Clock analog encoder/analog Hall V or N
MCLK_AENC_WY	U4	IO	DS-Mod Clock analog encoder/analog Hall W or Y
MCLK_AGPI_A	T2	IO	DS-Mod Clock for Analog General Purpose Input AGPI_A
MCLK_AGPI_B	U2	IO	DS-Mod Clock for Analog General Purpose Input AGPI_B
MCLK_I_UX	V1	IO	DS-Mod Clock for Analog Current Sense Voltage of I_U or I_X
MCLK_I_WY	AA2	IO	DS-Mod Clock for Analog Current Sense Voltage of I_W or I_Y
MCLK_VM	T3	IO	DS-Mod Clock for (down-divided) motor supply voltage of V_M
MDAT_AENC_UX	R5	I	DS-Mod Data Stream for analog encoder/analog Hall U or X
MDAT_AENC_VN	T5	I	DS-Mod Data Stream for analog encoder/analog Hall V or N
MDAT_AENC_WY	U5	I	DS-Mod Data Stream for analog encoder/analog Hall W or Y
MDAT_AGPI_A	T1	I	DS-Mod Data Stream for Analog General Purpose Input AGPI_A
MDAT_AGPI_B	U1	I	DS-Mod Data Stream for Analog General Purpose Input AGPI_B
MDAT_I_UX	W1	I	DS-Mod Data Stream for Analog Current Sense Voltage of I_U or I_X
MDAT_I_WY	W2	I	DS-Mod Clock for Analog Current Sense Voltage of I_W or I_Y
MDAT_I_UX_2ND	Y1	I, PU	DS-Mod Data stream for Analog Current Sense Voltage of I_U or I_X
MDAT_I_WY_2ND	Y2	I, PU	DS-Mod Data stream for Analog Current Sense Voltage of I_W or I_Y
MDAT_VM	R2	I	DS-Mod Data stream for (down-divided) motor supply voltage of V_M

Name	Pin	Type	Function
PWM Signals			
PWM_UX1_H	Y7	O	Digital gate control signal for High Side of Phase U (FOC3) or X1 (FOC2)
PWM_UX1_L	AA7	O	Digital gate control signal for Low Side of Phase U (FOC3) or X1 (FOC2)
PWM_VX2_H	Y8	O	Digital gate control signal for High Side of Phase V (FOC3) or X2 (FOC2)
PWM_VX2_L	AA8	O	Digital gate control signal for Low Side of Phase V (FOC3) or X2 (FOC2)
PWM_WY1_H	Y10	O	Digital gate control signal for High Side of Phase W (FOC3) or Y1 (FOC2)
PWM_WY1_L	AA10	O	Digital gate control signal for Low Side of Phase W (FOC3) or Y1 (FOC2)
PWM_Y2_H	AA11	O	Digital gate control signal for High Side of Phase Y2 (FOC2)
PWM_Y2_L	Y11	O	Digital gate control signal for Low Side of Phase Y2 (FOC2)

Additional Control Signals			
ENABLE_OUT	W11	O	enable output
BRAKE_CHOPPER	Y9	O	brake chopper control signal

Debug UART Interfaces and Debug I/Os			
STATUS_OUT	M5	O	status signal output
RXD_HWI	G5	I, PU	HW debug UART, RxD input
TXD_HWO	G4	O	HW debug UART, TxD output
RXD_MCU	G2	I, PU	MCU debug UART, RxD input
TXD_MCU	G1	O	MCU debug UART, TxD output
MCU_GPO_15	V8	O	reserved, keep open
MCU_GPO_16	V10	O	reserved, keep open
MCU_GPO_17	V11	O	reserved, keep open
MCU_GPO_18	U12	O	reserved, keep open
PDI_IRQ	K5	O	reserved, keep open (GPIO_3 on TMC8670 EVAL V.1.1)

Name	Pin	Type	Function
Device Supply and Ground			
VDD_1V2	K10, K11, L10, L11, M12, M13, N12, N13, R12, R13, U14, V14, V16, W16		1.2V DC Core supply voltage, **use 100nF filter capacitors**
VDD_3V3	M7, U15, V12, K9, H7, G15, R14, E2, J5, M2, N5, V2, V5, AA9, R10, V9, D20, F17, J15, K20		3.3V supply voltage for I/Os, PLL, and NVM, **use 100nF filter capacitors**
GND	G7, H15, R15, A1, A11, A16, A21, A6, AA1, AA12, AA14, AA15, AA16, AA18, AA19, AA20, D12, D17, D7, F21, F4, G14, H1, H18, J10, J11, J7, K12, K13, L12, L13, L4, M10, M11, M17, M21, N10, N11, P15, P7, R1, T21, T4, U13, U16, U17, U6, U7, V13, V15, Y12, Y14, Y15, Y16, Y18, Y19, Y20, Y6, J9		Supply Ground

Explicitly Not Connected Pins		
N.C.	R11, Y4, V17, V18, Y13, Y17, Y5, D18, G12, G8, AA5, AA4, AA13, AA17, B13, B18, B3, B8, E14, E9, J12, J13, W20, N15, R18	not connected

Name	Pin	Type	Function
Test Pins only			
DUMMY_OUT	G17	O	reserved, keep open
JTAG_TCK	L9	I	JTAG test clock, pull up to VDD_3V3 with 1K
JTAG_TDI	N9	I	JTAG Test data, N.C.
JTAG_TDO	R7	O	JTAG Test data, N.C.
JTAG_TMS	AA3	I	JTAG Test mode select, N.C.
JTAG_TRSTB	Y3	I	JTAG Test reset, pull down to GND with 1K
JTAGSEL	V4	I	JTAG Select line, pull up to VDD_3V3 with 1K

Table 2: Pin and Signal description for TMC8670-BA

5 Device Usage and Handling

5.1 Reference Clock

TMC8670 and the external Ethernet PHYs must share the same clock source. For proper operation a stable and accurate 25MHz clock source is required. The recommended initial accuracy must be at least 25ppm or better.

TMC8670 has been successfully used with the following crystal oscillators so far (this list ist not limited to the mentioned parts):

- FOX Electronics FOX924B TCXO, 25.0MHz, 2.5ppm, 3.3V
- TXC 7M-25.000MAAJ-T XO 25.0MHz, 30ppm
- CTS 636L5C025M00000, 25MHz, 25ppm

5.2 Ethernet PHY Connection

For connection to the Ethernet physical medium and to the EtherCAT master, TMC8670 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.

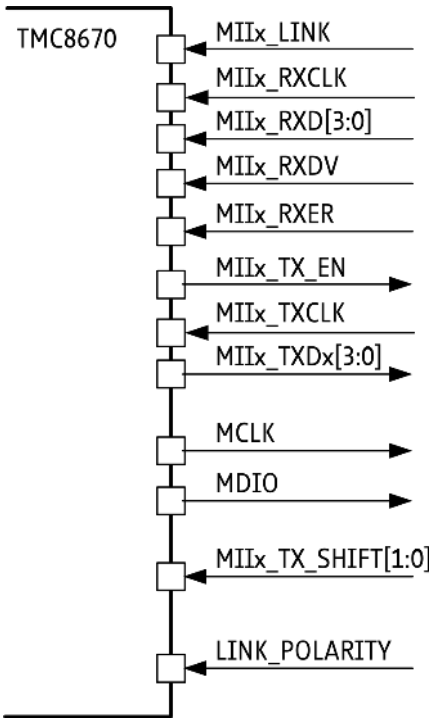


Figure 6: MII interface

TMC8670 pin	Description
MIIx_LINK	Active link input signal, active high/active low determined by LINK_POLARITY pin
MIIx_RXCLK	Receive clock input
MIIx_RXD[3:0]	Receive data inputs (4 bit wide)
MIIx_RXDV	Receive data valid input
MIIx_RXER	Receive error input
MIIx_TX_EN	Transmit enable output
MIIx_TXCLK	Transmit clock input, optional for automatic phase compensation
MIIx_TXD[3:0]	Transmit data output (4 bit wide)
MCLK	PHY MI configuration clock output
MDIO	PHY MI configuration data in-/output
MIIx_TX_SHIFT[1:0]	Phase compensation of MII TX signals, tie either to GND or VDD_3V3
LINK_POLARITY	Active level of MIIx_LINK signal, tie either to GND or VDD_3V3

Table 3: MII signal description

TMC8670 requires Ethernet PHYs with MII interface. The MII interface of TMC8670 is optimized for low additional delays by omitting a transmit FIFO. Additional requirements to Ethernet PHYs exist and not every Ethernet PHY is suited. Please see the Ethernet PHY Selection Guide provided by the ETG: http://download.beckhoff.com/download/Document/EtherCAT/Development_products/AN_PHY_Selection_GuideV2.6.pdf.

TMC8670 has been successfully tested in combination with the following Ethernet PHYs so far:

- IC+ IP101GA: http://www.icplus.com.tw
- Micrel KSZ8721BLI: http://www.micrel.com
- Micrel KSZ8081: http://www.micrel.com

The clock source of the Ethernet PHYs is the same as for the TMC8670.

LINK_POLARITY
This pin allows configuring the polarity of the link signal of the PHY. PHYs of different manufacturers may use different polarities at the PHY's pins.
In addition, some PHYs allow for bootstrap configuration with pull-up and pull-down resistors. This bootstrap information is used by the PHY at power-up/reset and also influences the polarity of the original pin function.

ETH PHY Addressing The TMC8670 addresses Ethernet PHYs using the logical port numbers 0 (LINK IN port) and 1 (LINK OUT port). Typically, the Ethernet PHY addresses should correspond with the logical port number, so PHY addresses have to be set to 0 and 1 accordingly using the ETH PHYs' bootstrap and configuration options.

MII_TX_SHIFT[1:0] TMC8670 and Ethernet PHYs share the same clock source. TX_CLK from the PHY has a fixed phase relation to the MII interface TX part of TMC8670Thus, TX_CLK must not be connected and the delay of a TX FIFO inside the IP Core is saved. In order to fulfill the setup/hold requirements of the PHY, the phase shift between TX_CLK and MIIx_TX_EN and MIIx_TXD[3:0] has to be controlled.

- Manual TX Shift compensation with additional delays for MIIx_TX_EN/MIIx_TXD[3:0] of 10, 20, or 30 ns. Such delays can be added using the TX Shift feature and applying MIIx_TX_SHIFT[1:0]. MIIx_TX_SHIFT[1:0] determine the delay in multiples of 10 ns for each port. Set MIIx_TXCLK to zero if manual TX Shift compensation is used.
- Automatic TX Shift compensation if the TX Shift feature is selected: connect MIIx_TXCLK and the automatic TX Shift compensation will determine correct shift settings. Set MIIx_TX_SHIFT[1:0] to 0 in this case.

5.3 External Circuitry and Applications Examples

5.3.1 Supply and Filtering

There should be one 100nF cap for each two VDD_1V2 pins. There should be one 100nF cap for circa each two VDD_3V32 pins. They should be placed as near as possible to the pins.

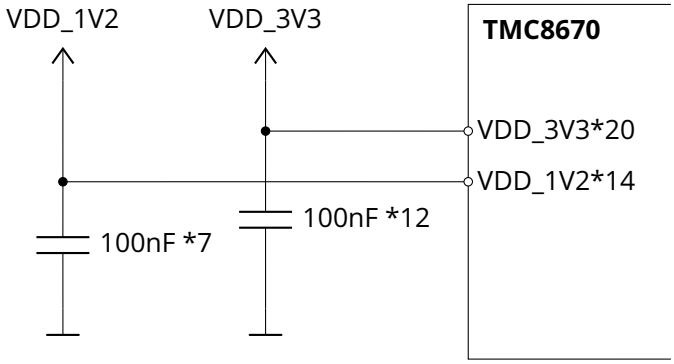


Figure 7: PLL supply filter

5.3.2 Status LED Circuit

The TMC8670 has 4 status LED outputs. All outputs are supplied from VDD_3V3, and drive a LED with current limiting resistor to GND. The use of low current LED is recommended to keep supply current low and to stay within the current limit of 10mA per pin. The appropriate resistor value must be chosen for the selected LED's forward voltage.
For a 2V forward voltage at 2mA, a value of ca. 680 Ohm is a reasonable value.
The LED colors are defined by **ETG.1300** (available on www.ethercat.org).

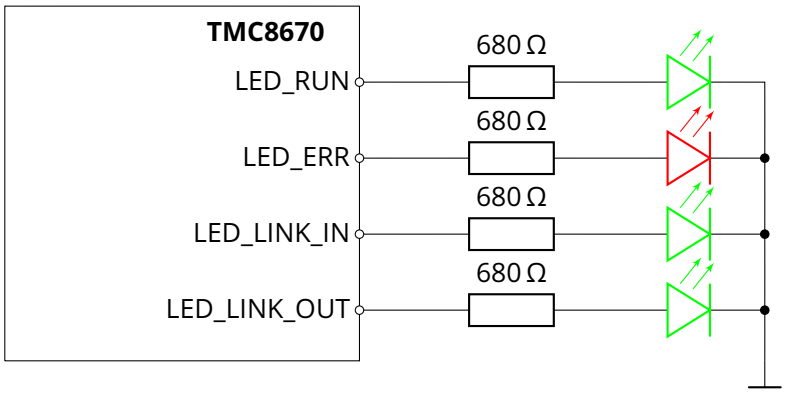


Figure 8: Status LED circuit

5.3.3 SII EEPROM Circuit

An IIC EEPROM is required for operation with the SII interface. Its size can be up to 4MBit. While the access protocol of the IIC EEPROMs is standardized, the addressing procedure changes from one address byte up to 16kBit to two address bytes from 32kBit.

Up to 16kBit the PROM_SIZE pin must be tied to GND, above that, it must be tied to VDD_3V3.

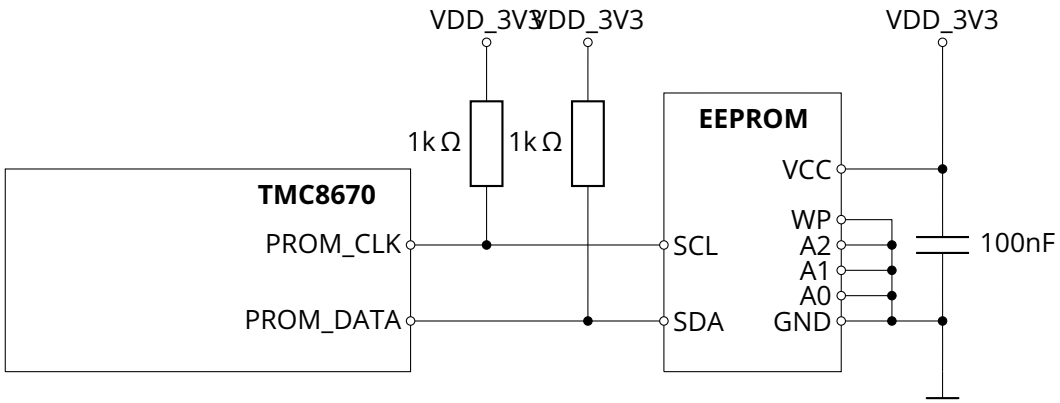


Figure 9: SII EEPROM circuit

5.4 Incremental Encoder Connection

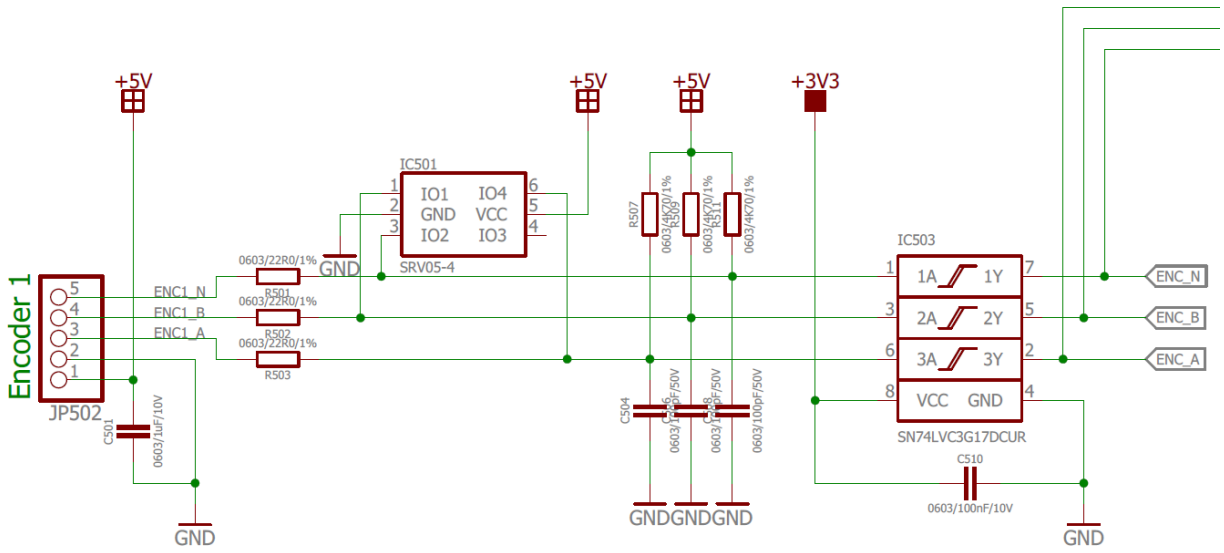


Figure 10: Example circuit for connecting an incremental encoder with level shifters from typically 5V to 3.3V

5.4.1 Incremental ABN Encoder

The incremental encoders give two phase shifted incremental pulse signals A and B. Some incremental encoders have an additional null position signal N or zero pulse signal Z. An incremental encoder (called ABN encoder or ABZ encoder) has an individual number of incremental pulses per revolution. The number of incremental pulses define the number of positions per revolution (PPR). The PPR might mean pulses per revolution or periods per revolution. Instead of positions per revolution some incremental encoder vendors call these CPR counts per revolution.

The PPR parameter is the most important parameter of the incremental encoder interface. With that, it forms a modulo (PPR) counter, counting from 0 to (PPR-1). Depending on the direction, it counts up or down. The modulo PPR counter is mapped into the register bank as a dual ported register. the user can over over write it with an initial position. The ABN encoder interface provides both, the electrical position and the multi-turn position are dual-ported read-write registers.

Note	The PPR parameter must be set exactly according to the used encoder.

The N pulse from an encoder triggers either sampling of the actual encoder count to fetch the position at the N pulse or it re-writes the fetched n position on an N pulse. The N pulse can either be uses as stand alone pulse or and-ed with NAB = N and A and B. It depends on the decoder what kind of N pulse has to be used, either N or NAB. For those encoder with precise N pulse within on AB quadrat, the N pulse must be used. For those encoders with N pulse over four AB quadrants one can enhance the precision of the N pulse position detection by using NAB instead of N.

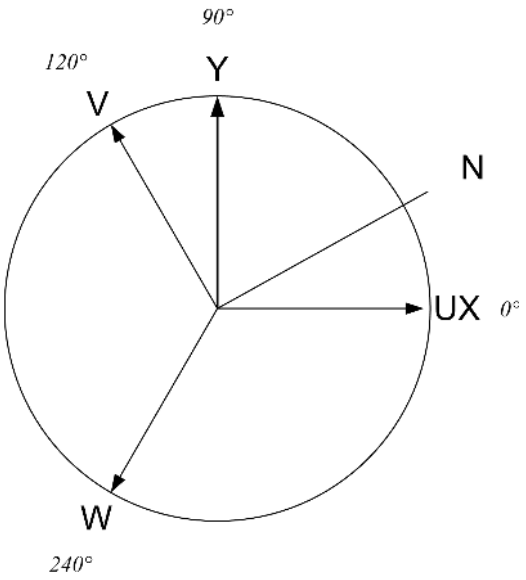


Figure 11: ABN Incremental Encoder N Pulse

The polarity of N pulse, A pulse and B pulse are programmable. The N pulse is for re-initialization with each turn of the motor. Once fetched, the ABN decoder can be configured to write back the fetched N pulse position with each N pulse.

Note	Incremental encoders are available with N pulse and without N pulse.

Note	The ABN encoder interface has a direction bit to set once the direction of motion for the application.

Logical ABN = A and B and N might be useful for incremental encoders with low resolution N pulse to enhance the resolution. On the other hand, for incremental encoders with high resolution n pulse a logical abn = a and b and n might totally suppress the resulting n pulse.

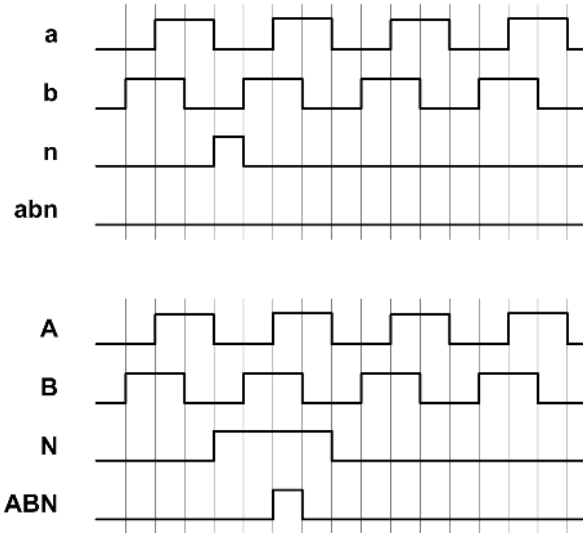


Figure 12: Encoder ABN Timing - high precise n pulse and less precise N pulse

5.4.2 Secondary Incremental ABN Encoder

For commutating a motor with FOC one selects a position sensor source (digital incremental encoder, digital hall, analog hall, analog incremental encoder, ...) that is mounted close to the motor. The inner FOC loop control torque and flux of the motor based on the measured phase currents and the electrical angle of the rotor.

The TMC8670 is equipped with a secondary incremental encoders interface. This secondary encoder interface is available as source for velocity control or position control. This is for applications where a motor turns an object with a gear to position the object. An example is a robot arm where a motor moves an angle with a the mechanical angle of the arm as the target.

Info	The secondary incremental encoder is not available for commutation (phi_e) for the inner FOC. In others words, there is no electrical angle phi_e selectable from the secondary encoder.

5.4.3 Open Loop Encoder

For initial system setup the encoder engine is equipped with an open loop position generator. With one can turn the motor open-loop by specifying speed in rpm and acceleration in rpm/s together with a voltage UD_EXT in D direction. So, the open-loop encoder it is not a real encoder, it just gives positions as an encoder does. The open-loop decoder has a direction bit to define once the direction of motion for the application.

Note	The open loop encoder is useful for initial ADC setup, encoder setup, hall signal validation, and for validation of the number of pole pairs of a motor. The open loop encoder turns a motor open with programmable velocity in unit [RPM] with programmable acceleration in unit [RPM/s].

So, with the open loop encoder one can turn a motor without any position sensor and without any current measurement as the first step of doing the system setup. With the turning motor one can adjust the ADC scales and offsets and set up positions sensors (hall, incremental encoder, ...) according to resolution, orientation, direction of rotation.

5.5 Hall Signal Connection

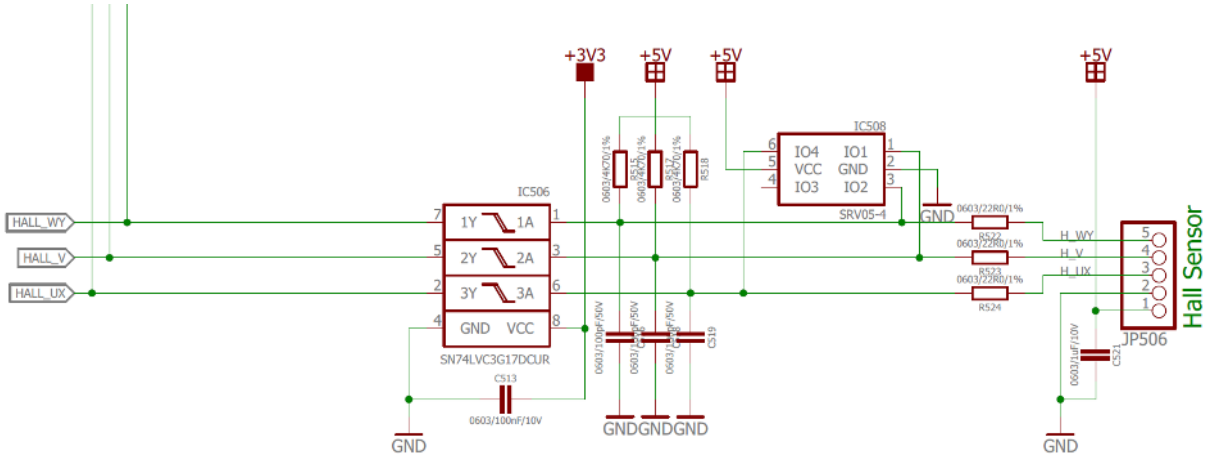


Figure 13: Example circuit for connecting Hall sensor signals with level shifters from typically 5V to 3.3V

5.5.1 Digital Hall Sensor Interface with optional Interim Position Interpolation

The digital hall interface is the position sensor interface for digital Hall signals. The digital Hall signal interface first maps the digital Hall signals to an electrical position PHI_E_RAW. An offset PHI_E_OFFSET can be used to rotate the orientation of the Hall signal angle. The electrical angle PHI_E is for commutation. Optionally, the default electrical positions of the Hall sensors can be adjusted by writes into the associated registers.

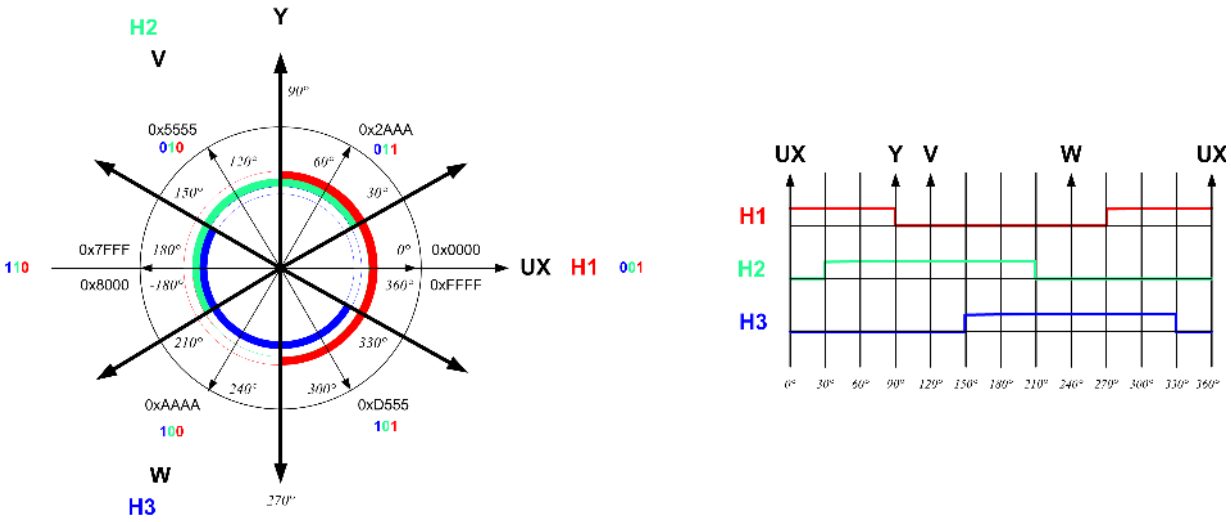


Figure 14: Hall Sensor Angles

Hall sensors give an absolute positions within an electrical period with a resolution of 60° as 16 bit positions (s16 resp. u16) PHI. With activated interim Hall position interpolation the user gets high resolution interim positions, when the motor is running at speed beyond 60 rpm.

5.5.2 Digital Hall Sensor - Interim Position Interpolation

For lower torque ripple the user can switch on the position interpolation of interim Hall positions. This function is useful for motors that are compatible with sine wave commutation, but equipped with digital hall sensors.
When the position interpolation is switched on, it becomes active on speed beyond 60 rpm. For lower speed it automatically disables. This is important especially, when the motor has to be at rest.
Hall Sensor position interpolation might fail, when Hall sensors signals are not properly placed in the motor. Please adjust hall sensor positions for this case.

5.5.3 Digital Hall Sensors - Masking and Filtering

Sometimes digital Hall sensor signals get disturbed by switching events in the power stage. The TMC8670 can automatically mask switching distortions by correct setting of the HALL_MASKING register. When a switching event occurs, the Hall sensor signals are held for HALL_MASKING value times 10 ns. In this way Hall sensor distortions are eliminated. Uncorrelated distortions can be filtered via a digital filter of parametrizable length. If the input signal to the filter does not change for HALL_DIG_FILTER times 5 us, the signal can pass the filter. This filter eliminates issues with bouncing Hall signals.

5.5.4 Digital Hall Sensors together with Incremental Encoder

If a motor is equipped with both Hall sensors and incremental encoder, the Hall sensors can be used for the initialization as a low resolution absolute position sensor and later the incremental encoder can be used as a high resolution sensor for commutation.

5.6 ADC Interfaces

The ADC interface is for measurement of sense voltages from sense resistor amplifiers for current measurement and for measurement of analog hall signals or analog encoder signals. There are two variants of external ADC interfaces supported: Delta Sigma ADC formed by linear comparator LM339 with two resistors and one caparitor per channel. External SPI ADC LTC2351 from Linear Technology. Both ADC groups (A and B) can be selected separately to process either dsADC or SPI ADC.

The TMC8670 evaluation board (TMC8670 EVAL V.1.1) is equipped with LMC339 delta sigma ADC frontends and LTC2351 SPI ADC frontends to enable evaluation of both alternatives.

5.6.1 ADC Interface - Delta Sigma Modulator

As external delta sigma modulator the linear quad comparator LM339 is recommended together with $R_{PU} = 1K\Omega(1\%)$, $R_C = 100K\Omega(1\%)$, and $R_I = 100K\Omega(1\%)$, and $C = 100pF(5\%)$.

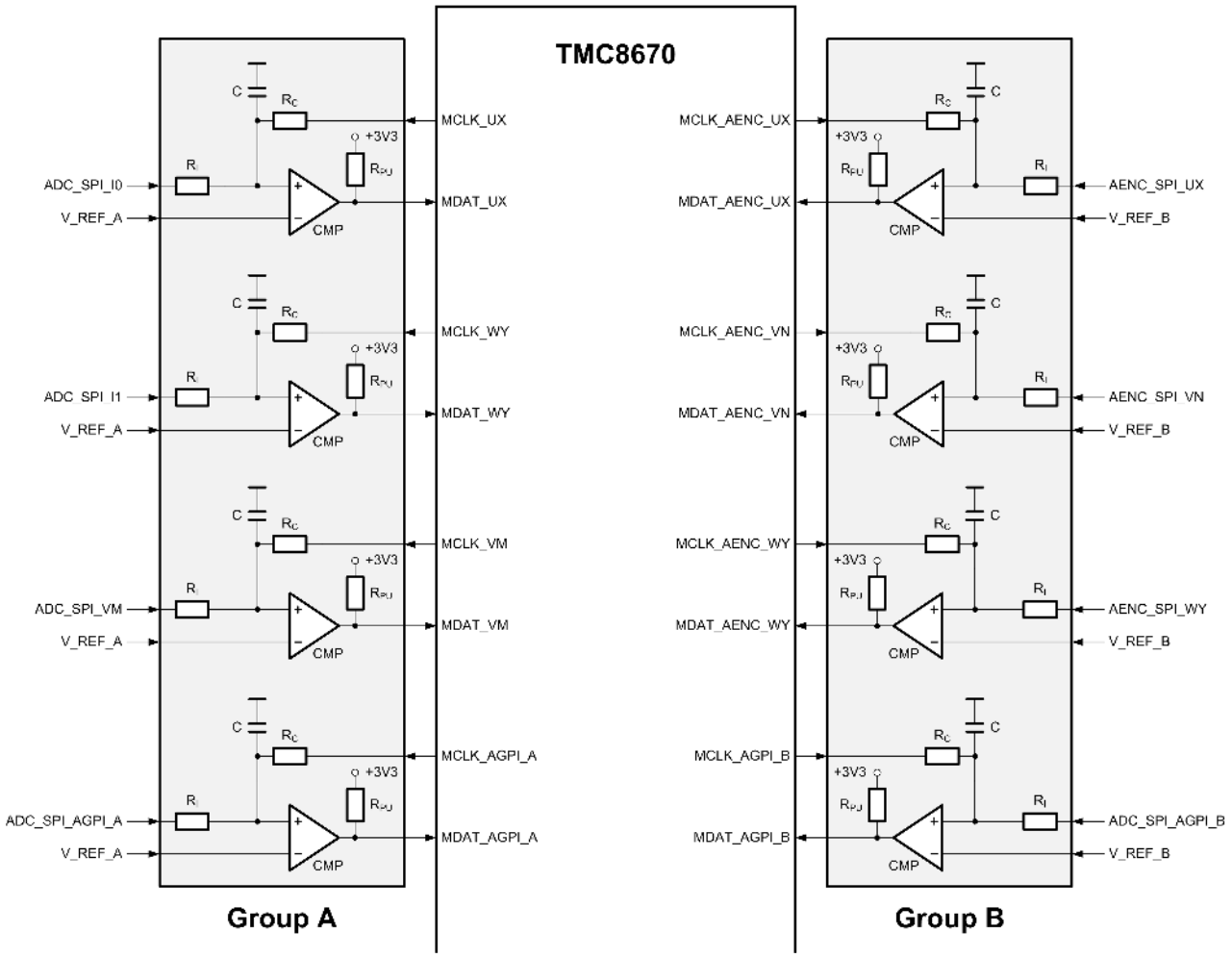


Figure 15: TMC8670 Delta Sigma ADC Configuration

5.6.2 ADC Interface - SPI ADC

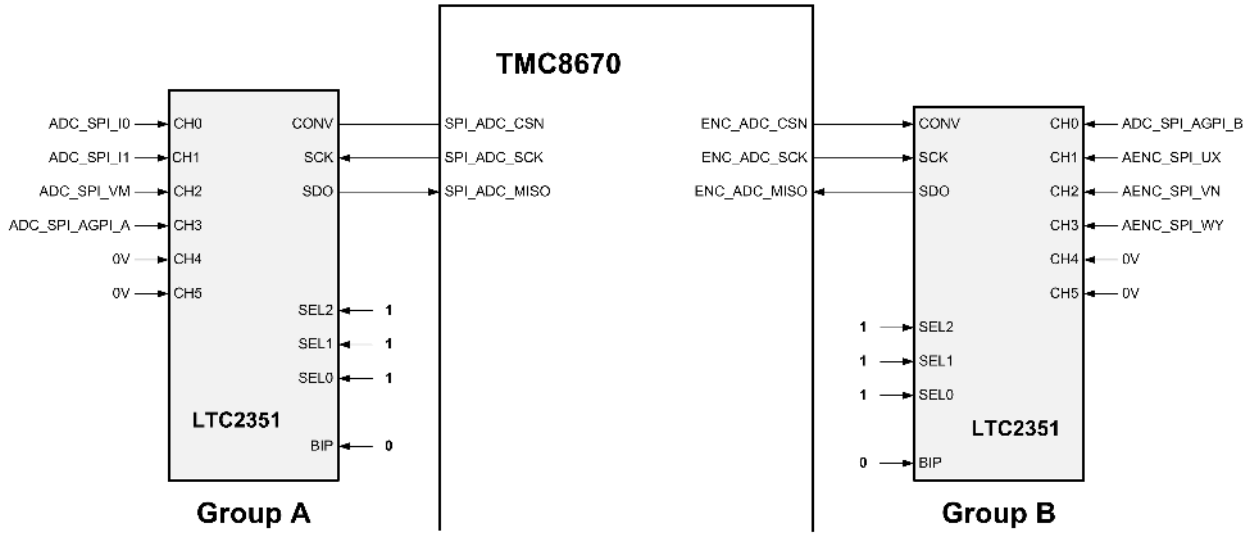


Figure 16: TMC8670 configuration for LTC2351 SPI ADCs

Whene using LTC2351 as ADC frontend for the TMC8670, one can add filter elements $R_{CHP} = 50\Omega(1\%)$ and $C_{PN} = 47pF(5\%)$ for spike suppressen on each ADC analog input channel of Group A (CH0, CH1, CH2, CH3, CH4, CH5) and of Group B (CH0, CH1, CGH2, CH3).

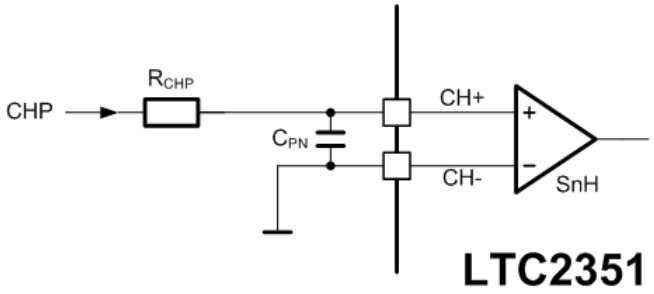


Figure 17: LTC2351 with input filter elements for noise reduction and spike reduction

5.6.3 Analog Hall and Analog Encoder Interface (SinCos of 0°90° or 0°120°240°)

An analog encoder interface is part of the decoder engine. It is able to handle analog position signals of 0° and 90° and 0° 120° 240°. The analog decoder engine adds offset and scales the raw analog encoder signals and calculates the electrical angle PHI_E from these analog position signals.

An individual signed offset is added each associated raw ADC channel and scaled by its associated scaling factors according to

$$\text{AENC\_VALUE} = (\text{AENC\_RAW} + \text{AENC\_OFFSET}) \cdot \text{AENC\_SCALE} \quad (1)$$

In addition, the AENC_OFFSET is for conversion of unsigned ADC values into signed ADC values as required for the FOC.

Info For details on the individual registers and how to access them please check the TMC8670 firmware manual.

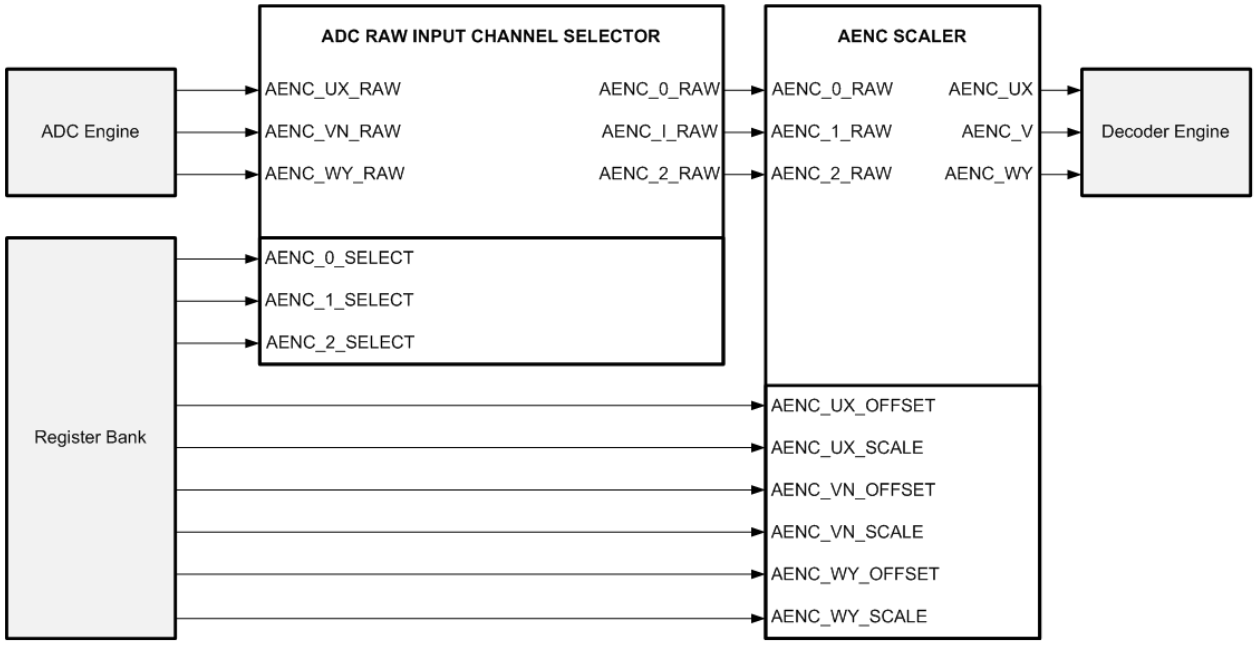


Figure 18: Analog Encoder (AENC) Selector & Scaler w/ Offset Correction

Info The analog N pulse is just a raw ADC value. Scaling, offset correction, hand handling of analog N pulse similar to N pulse handling of digital encoder N pulse is not implemented for analog encoder.

5.6.4 Analog Position Decoder (SinCos of 0°90° or 0°120°240°)

The extracted positions from the analog decoder are available for read out from registers.

5.7 Brake Chopper Connection

The brake chopper signal from the TMC8670 is just a digital 3V3 logic level signal. It can be used as switching / trigger signal for an external brake chopper circuit.

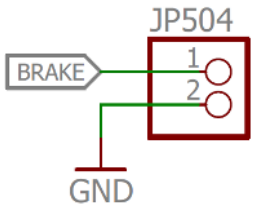


Figure 19: TMC8670 Brake Chopper Connection

5.8 UART Interfaces

The TMC8670 is equipped with two independant UART interfaces for initial evaluation and visualization purposes: The first UART software interface for the TMCL-IDE with TMCL protocol and the second UART hardware register interface with a five byte procotol.

The software UART (TXD_MCU, RXD_MCU) is for optional firmware updates with TMCL-IDE without EtherCAT master and for debugging purposes during EtherCAT stack development. With an EtherCAT master one can update via FoE.

The hardware UART (TXD_HWO, RXD_HWI) allows direct access to the TMC8670 registers handled by an arbiter. It is for debugging purposes and allows transparent access to internal registers of the TMC8670. It is intended to support EtherCAT slave controller hardware development.

Note	Both interfaces are intended for initial evaluation, debugging, development, and monitoring purposes. It is recommended not to over-write data from outside the EtherCAT into registers of the TMC8670 via these interfaces during regular operation. Read of data might be used for monitoring purposes during debugging or validation of own developments.

5.8.1 UART Software Interface for TMCL-IDE

The software UART interface is a simple three Pin (GND, TXD_MCU, RXD_MCU) 3.3V UART Interface with 115200 bps communication speed, one start bit, eight data bits, one stop bit, and no parity bits (1N8). With an 3.3V-UART-to-USB adapter cable (e.g. FTDI TTL-232R-RPi) the user can directly access registers of the TMC8670 via the TMCL-IDE.

5.8.2 UART Hardware Register Interface

The UART hardware register interface is a simple three Pin (GND, TXD_HWO, RXD_HWI) 3.3V UART Interface with up to 3 Mbit/s transfer speed with one start bit, eight data bits, one stop bit, and no parity bits (1N8). The default speed is 9600 bps. Other supported speeds are 115200 bps, 921600 bps, and 3000000 bps. This UART port enables In-System-Setup-Support by multiple-ported register access of the TMC8670.

This UART datagram consists of five bytes. This UART interface has a time out feature: Five bytes of a UART datagram need to be send within one second. A pause of sending more than one second causes a time out and sets the UART protocol handler back into idle state. In other words, waiting for more than one second in sending via UART ensures that the UART protocol handler is in idle state. The UART is inactive with the RXD input pulled to high.

A simple UART hardware register access example:

```
0x81 0x00 0x00 0x00 0x00 // 1st write 0x00000000 into address 0x01 (CHIPINFO_ADDR)
0x00 0x00 0x00 0x00 0x00 // 2nd read register 0x00 (CHIPINFO_DATA), returns 0x38363730
```

Why UART Interface? It might be useful during system setup phase by simple access to some internal registers without disturbing the application and without changing the actual user application software and without adding additional debugging code that might disturb the application software itself. It enables access for monitoring purposes with its simple and direct five byte protocol.

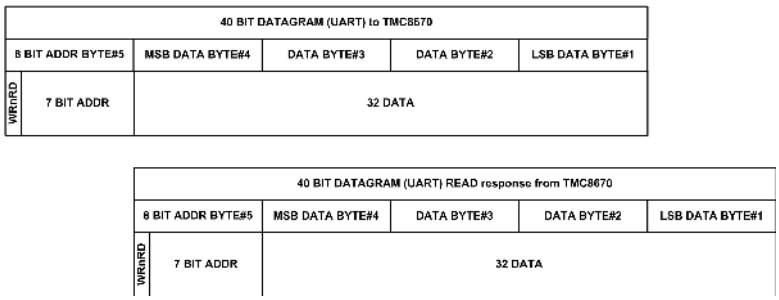


Figure 20: UART Read Datagram (TMC8670 register read via UART)

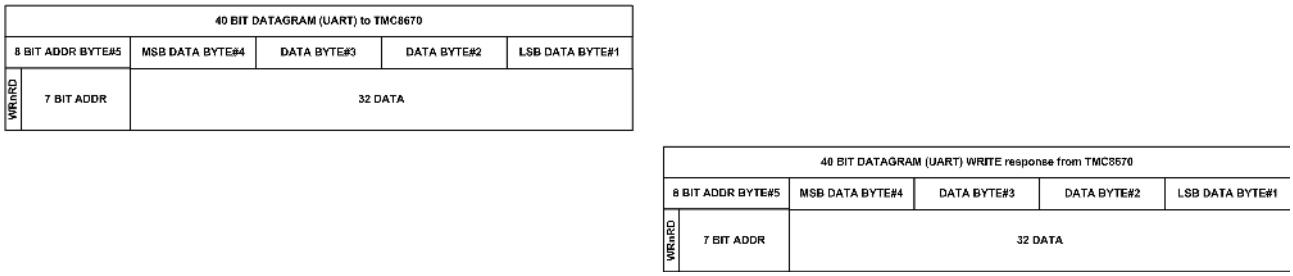


Figure 21: UART Write Datagram (TMC8670 register write via UART)

6 FOC Basics

This section gives a short introduction into some basics of Field Oriented Control (FOC) of electric motors.

6.1 Why FOC?

The Field Oriented Control (FOC) alternatively named Vector Control (VC) is a method for most energy efficient turning an electric motor.

6.2 What is FOC?

The Field Oriented Control was independently developed by K. Hasse, TU Darmstadt, 1968, and by Felix Blaschke, TU Braunschweig, 1973. The FOC is a current regulation scheme for electro motors that takes the orientation of the magnetic field and the position of the rotor of the motor into account regulating the strength in the way that the motor gives that amount of torque that is requested as target torque. The FOC maximizes active power and minimize idle power - that finally results in power dissipation - by intelligent closed-loop control illustrated by the cartoon figure 22.

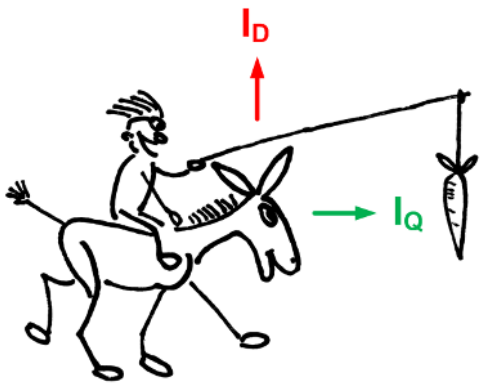


Figure 22: Illustration of the FOC basic principle by cartoon: Maximize active power and minimize idle power and minimize power dissipation by intelligent closed-loop control.

6.3 Why FOC as pure Hardware Solution?

The initial setup of the FOC is usually very time consuming and complex, although source code is freely available for various processors. This is because the FOC has many degrees of freedom that all need to fit together in a chain in order to work.

The hardware FOC as an existing standard building block drastically reduces the effort in system setup. With that of the shelf building block, the starting point of FOC is the setup of the parameters for the FOC and no longer the setup and implementation of the FOC itself and building and programming of required interface blocks. The real parallel processing of hardware blocks de-couples the higher lever application software from high speed real time tasks and simplifies the development of application software. With the TMC8670, the user is free to use its qualified CPU together with its qualified tool chain and it frees the user from fighting with processer specific challenges concerning interrupt handling and direct memory access. There is no need for a dedicated tool chain to access TMC8670 registers and to operate it - just SPI (or UART) communication needs to be enabled for a given CPU.

The integration of the FOC as a SoC (System-on-Chip) drastically reduces the number of required components and reduces the required PCB space. This is in contrast to classical FOC servos formed by motor

block and separate controller box wired with motor cable and encoder cable. The high integration of FOC, together with velocity controller and position controller as a SoC, enables the FOC as a standard peripheral component that transforms digital information into physical motion. Compact size together with high performance and energy efficiency especially for battery powered mobile systems are enabling factors when embedded goes autonomous.

6.4 How does FOC work?

Two force components act on the rotor of an electric motor. One component is just pulling in radial direction (ID) where the other component tangentially pulling (IQ) is applying torque. The ideal FOC performs a closed loop current regulation that results in a pure torque generating current IQ without direct current ID.

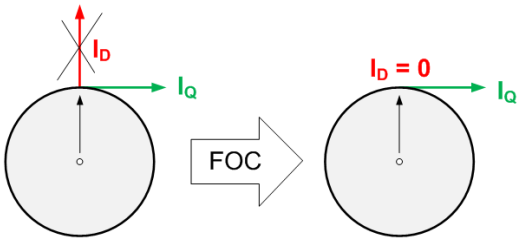


Figure 23: FOC optimizes torque by closed loop control while maximizing IQ and minimizing ID to 0

From top point of view, the FOC for three phase motors uses three phase currents of the stator interpreted as a current vector (Iu; Iv; Iw) and calculates three voltages interpreted as a voltage vector (Uu; Uv; Uw) taking the orientation of the rotor into account in a way that only a torque generating current IQ results.

From top point of view, the FOC for two phase motors uses two phase currents of the stator interpreted as a current vector (Ix; Iy) and calculates two voltages interpreted as a voltage vector (Ux; Uy) taking the orientation of the rotor into account in a way that only a torque generating current IQ results.

To do so, the knowledge of some static parameters (number of pole pairs of the motor, number of pulses per revolution of a used encoder, orientation of encoder relative to magnetic axis of the rotor, count direction of the encoder) is required together with some dynamic parameters (phase currents, orientation of the rotor).

The adjustment of P parameter and I parameters of two PI controllers for closed loop control of the phase currents depends on electrical parameters (resistance, inductance, back EMF constant of the motor that is also the torque constant of the motor, supply voltage) of the motor.

6.5 What is required for FOC?

The FOC needs to know the direction of the magnetic axis of the stator of the motor together with the magnetic axis of the rotor of the motor. The magnetic direction of the magnetic axis of the stator is calculated from the currents thought the phases of the motor. The magnetic direction of the rotor is determined by an encoder device.

For the FOC one needs to measure the currents through the coils of the stator and the angle of the rotor. The measured angle of the rotor needs to be adjusted to the magnetic axes.

The challenge of the FOC is the high number of degrees of freedom of all parameters together.

6.5.1 Coordinate Transformations - Clarke, Park, iClarke iPark

The FOC requires different coordinate transformations formulated as a set of matrix multiplications. These are the Clarke Transformation (Clarke), the Park Transformation (Park), the inverse Park Transformation (iPark) and the inverse Clarke Transformation (iClarke). Some put Park and Clarke together as DQ transformation and Park and Clarke as inverse DQ transformation.

The TMC8670 takes care of the required transformations and get the user rid from fighting with details of implementation of theses transformations.

6.5.2 Measurement of Stator Coil Currents

The measurement of the stator coil currents is required for the FOC to calculate a magnetic axis ot of the stator field caused by the currents flowing through the stator coils.

Coil current stands for motor torque in context of FOC. This is because motor torque is proportional to motor current, defined by the torque constant of a motor. In addition, the torque depends on the orientation of the rotor of the motor relative to the magnetic field produced by the current through the coils of the stator of the motor.

6.5.3 Stator Coil Currents I_U, I_V, I_W and Association to Terminal Voltages U_U, U_V, U_W

The correct association between stator terminal voltages U_U, U_V, U_W and stator coil currents I_U, I_V, I_W is essential for the FOC. In addition to the association, the signs of each current channel needs to fit. Signs of the current can be adapted numerically by the ADC scaler. The mapping of ADC channles is programmable via configurations registers for the ADC selector. Initial setup is supported by the integrated open loop encoder block that can turn a motor open loop.

6.5.3.1 Chain of Gains for ADC Raw Values

An ADC raw value is a result of a chain of gains that determine it. A coil current I_SENSE flowing through a sense resistor causes a voltage difference according to Ohm's law. The resulting ADC raw value is result of the analog signal path according to

$$\text{ADC\_RAW} = (\text{I\_SENSE} * \text{ADC\_GAIN}) + \text{ADC\_OFFSET}. \tag{2}$$

The ADC_GAIN is a result of a chain of gains with individual signs. The sign of the ADC_GAIN is positive or negative, depending on the association of connections between sense amplifier inputs and the sense resistor terminals. The ADC_OFFSET is the result of electrical offsets of the phase current measurement signal path. For the TMC8670 the maximum ADC_RAW value ADC_RAW_MAX = $(2^{16} - 1)$ and the minimum ADC raw value is ADC_RAW_MIN = 0.

$$\begin{aligned} \text{ADC\_GAIN} = & \quad (\quad \text{I\_SENSE\_MAX} * \text{R\_SENSE} \quad) \\ & * \quad \text{SENSE\_AMPLIFIER\_GAIN} \\ & * \quad (\quad \text{ADC\_RAW\_MAX}/\text{ADC\_U\_MAX} \quad) \end{aligned} \tag{3}$$

For the FOC, the ADC_RAW is scaled by the ADC scaler of the TMC8670 together with subtraction of offset to compensate it. Internally, the TMC8670 FOC engine calculates with s16 values. So, the ADC scaling needs to be chosen that the measures currents fit into the s16 range. With the ADC scaler, one can choose a scaling with physical units like $[mA]$. A scaling to $[mA]$ covers a current range of $-32A \ldots +32A$ with $m[A]$ resolution. For higher currents con can go to un-usual units like centi Ampere $[cA]$ covering $-327A \ldots +327A$ or deci Ampere $-3276A \ldots +3276A$.

ADC scaler and offset compensators are for mapping of raw ADC values to s16 scaled an offset cleaned current measurement values that are adequate for the FOC. ADC scaling factor and ADC offset removal value needs to be programmed into associated registers. Finally, a current is mapped to an ADC raw value that is numerically mapped to signed ADC value with removed offset by the ADC scaler.

6.5.4 Measurement of Rotor Angle

Determination of the rotor angle is either by done by sensors (digital encoder, analog encoder, digital hall sensors, analog hall sensors) or sensorless by reconstruction of the rotor angle from measurements of electrical parameters with or without a mathematical model of the motor. Currently, there is no sensorless methods available for FOC that work in a general purpose way as a sensor down to velocity zero.

The TMC8670 does not support sensorless FOC.

6.5.5 Measured Rotor Angle vs. Magnetic Axis of Rotor vs. Magnetic Axis ot Stator

The rotor angle, measured by an encoder, needs to be adjusted to the magnetic axis ot the rotor. This is because an incremental encoder has an arbitrary orientation relative to the magnetic axis of the rotor and the rotor has an arbitrary orientation to magnetic axis of the stator.

The direction of counting depends on the encoder, its mounting, and wiring and polarities of encoder signals and motor type. So, the direction of encoder counting is programmable for comfortable definition for a given combination of motor and encoder.

6.5.5.1 Direction of Motion - Magnetic Field vs. Position Sensor

For FOC it is essential, that the direction of revolution of the magnetic field is compatible with the direction of motion of the rotor position reconstructed from encoder signals: For revolution of magnetic field with positive direction the decoder position need to turn into same positive direction. For revolution of magnetic field with negative direction the decoder position need to turn into same negative direction.

With an absolute encoder, once adjusted to the relative orientation of the rotor and to the relative orientation of the stator, one could start the FOC without initialization of the relative orientations.

6.5.5.2 Bang-Bang Encoder Initialization

For Bang-Bang initialization one sets a current into direction D that is strong enough the move the rotor into the desired direction.

6.5.5.3 Encoder Initialization using Hall Sensors

The encoder can initialized using digital Hall sensor signals. Digital Hall sensor signal give absolute positions within each electrical period with a resolution of sixty degree. If the hall sensor signals are used to initialize the encoder position on the first change of a Hall sensor signal, one gets an absolute reference within the electrical period for commutation.

6.5.5.4 Encoder Minimum Movement Initialization

For encoder minimal movement initialization, one slowly increases a current into direction D and adjusts an offset of measured angel in a way the rotor of the motor does not move during initialization while the offset of measured angel is determined.

6.5.6 Knowledge of Relevant Motor Parameters and Position Sensor (Encoder) Parameters

6.5.6.1 Number of Pole Pairs of a Motor

The number of pole pairs is an essential motor parameter. It defines the ratio between electrical revolutions and mechanical revolutions. For a motor with one pole pair one mechanical revolution is equivalent to one electrical revolution. For a motor with npp pole pairs, one mechanical revolution is equivalent to npp electrical revolutions, with n = 1, 2, 3, 4, ...

Some define the number of poles NP instead of number of pole pairs NPP for a motor, which results in a factor of two that might cause confusion. For the TMC8670 we use NPP number of pole pairs.

6.5.6.2 Number of Encoder Positions per Revolution

For the encoder, the number of positions per revolution (PPR) is an essential parameter. The number of positions per revolution is essential for the FOC.

Some encoder vendors give the number of lines per revolution (LPR) or just named line count (LC) as encoder parameter. Line count and positions per revolution might differ by a factor of four. This is because of the quadrature encoding - A signal and B signal with phase shift - that give four positions per line and enables the determination of direction of revolution. Some encoder vendors associate counts per revolution (CPR) or pulses per revolution associated to PPR acronym.

The TMC8670 uses PPR as Positions Per Revolution as encoder parameter.

6.5.7 Proportional Integral (PI) Controllers for Closed Loop Current Control

Last but not least two PI controllers are required for the FOC. The TMC8670 is equipped with two PI controllers. One for control of torque generating current I_Q and one to control current I_D to zero.

6.5.8 Pulse Width Modulation (PWM) and Space Vector Pulse Width Modulation (SVPWM)

The PWM power stage is must have for energy efficient motor control. The PWM engine of the TMC8670 just needs a couple of parameters to set PWM frequency fPWM and switching pauses for high side switches tBBM_H and for low side switches tBBM_L. Some control bis are for programming of power switch polarities for maximum flexibility in selection in gate drivers for the power MOS-FETs. An additional control bit selects SVPWM on or off. The TMC8670 allows change of PWM frequency by a single parameter during operation.

Whit this, the TMC8670 is advanced compared to software solutions where PWM and SVPM configuration of CPU internal peripherals normally needs settings of many parameters.

6.5.9 Orientations, Models of Motors, and Coordinate Transformations

The orientation of magnetic axes (U, V, W for FOC3 resp. X, Y for FOC2) is essential for the FOC together with the relative orientation of the rotor. Here the rotor is modelled by a bar magnet with one pole pair (n_pole_pairs = 1) with magnetic axis in north-south-direction.
The actual magnetic axis of the stator - formed by the motor coils - is determined by measurement of the coil currents.
The actual magnetic axis of the rotor is determined by incremental encoder or by hall sensors. Incremental encoders need an initialization of orientation, where hall sensors give an absolute orientation but with low resolution. A combination of hall sensor and incremental encoder is useful for start-up initialization.

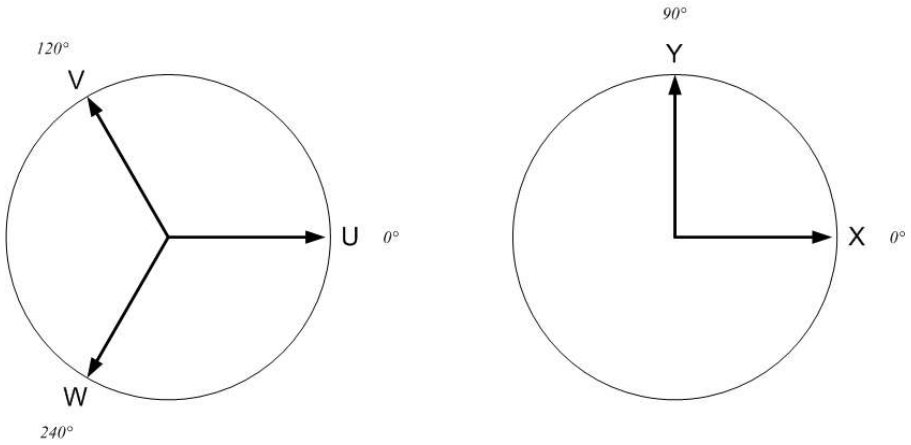


Figure 24: Orientations UVW (FOC3) and XY (FOC2)

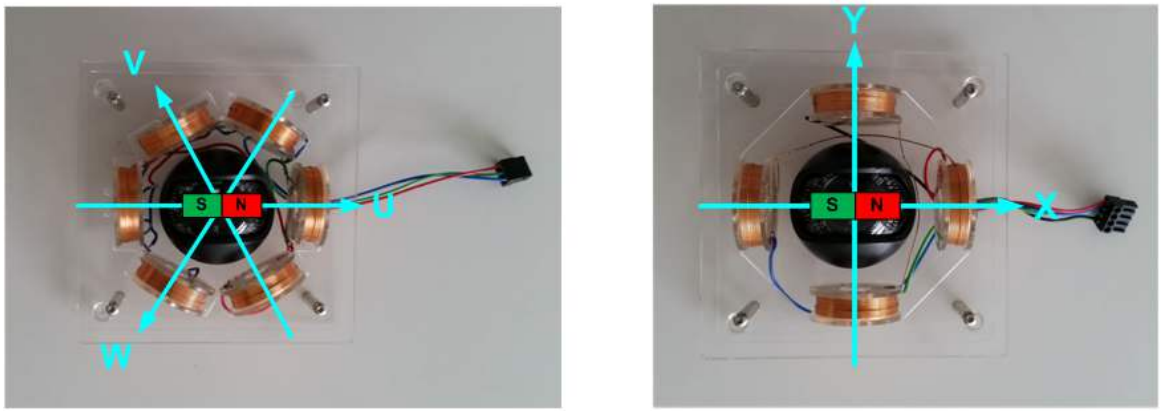


Figure 25: Compass Motor Model w/ 3 Phases UVW (FOC3) and Compass Motor Model w/ 2 Phases (FOC2)

6.6 FOC23 Engine

Info Support for the TMC8670 is integrated into the TMCL-IDE including wizards for set up and configuration. With the TMCL-IDE configuration and operation can be done in a few steps and the user gets direct access to all registers of the TMC8670.

The FOC23 engine performs the inner current control loop for the torque current I_Q and the flux current I_D including the required transformations. Programmable limiters take care of clipping of interim results. Per default, the programmable circular limiter clips U_D and U_Q to U_D_R = $\sqrt{(2)}\cdot$ U_Q and U_R_R = $\sqrt{(2)}\cdot$ U_D. PI controllers perform the regulation tasks.

6.6.1 PI Controllers

PI controllers are used for current control and velocity control. A P controller is used for position control. The D part is not yet supported. The user can choose between two PI controller structures. Classic PI controller structure which is also used in the TMC4670 and the Advanced PI Controller Structure. The Advanced PI Controller Structure shows better performance in dynamics and is recommended for high performance applications.

6.6.2 PI Controller Calculations - Classic Structure

The PI controllers in the classic Structure perform the following calculation

$$\mathrm{dXdT} = \mathrm{P}\cdot\mathrm{e} + \mathrm{I}\cdot\int_0^t \mathrm{e(t)}\,\mathrm{d}t \tag{4}$$

with

$$\mathrm{e} = \mathrm{X\_TARGET} - \mathrm{X} \tag{5}$$

where X_TARGET stands for target flux, target torque, target velocity, or target position with error e, that is the difference between target value and actual values. The time constant $\mathrm{d}t$ is $1\mu s$ with the integral part is divided by 256.

Info Changing the I-parameter of the Classic PI Controller during operation causes the controller output to jump, as the control error is first integrated and then gained by the I parameter. Be careful during controller tuning or use the advanced PI Controller Structure.

6.6.3 PI Controller Calculations - Advanced Structure

The PI controllers in the Advanced Controller Structure perform the calculation

$$\mathrm{dXdT} = \mathrm{P}\cdot\mathrm{e} + \int_0^t \mathrm{P}\cdot\mathrm{I}\cdot\mathrm{e(t)}\,\mathrm{d}t \tag{6}$$

with

$$\mathrm{e} = \mathrm{X\_TARGET} - \mathrm{X} \tag{7}$$

where X_TARGET represents target flux, target torque, target velocity, or target position with control error e that is the difference between target value and actual values. The time constant $\mathrm{d}t$ is set according to

the PWM period. Velocity and Position controller evaluation can be down sampled by a constant factor when needed.

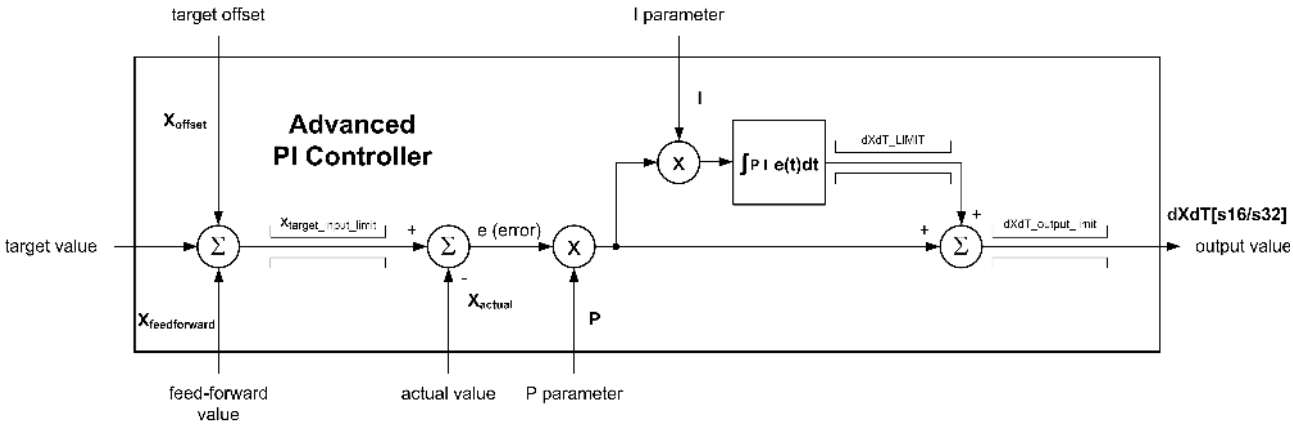


Figure 26: Advanced PI Controller

6.6.4 PI Controller - Clipping

The limiting of target values for PI controllers and output values of PI controllers is programmable. Per power on default these limits are set to maximum values. During initialization these limits should be properly set for correct operation and clipping. The target input is clipped to X_TARGET_LIMIT. The output of a PI controller is named dXdT, because it gives the desired derivative d/dt as a target value to the following stage: The position (x) controller gives velocity (dx/dt). The output of the PI Controller is clipped to dXdT_LIMIT. The error integral of (4) is clipped to dXdT_LIMIT / I in the classic controller structure and the integrator output is clipped to dXdT_LIMIT in the advanced controller structure.

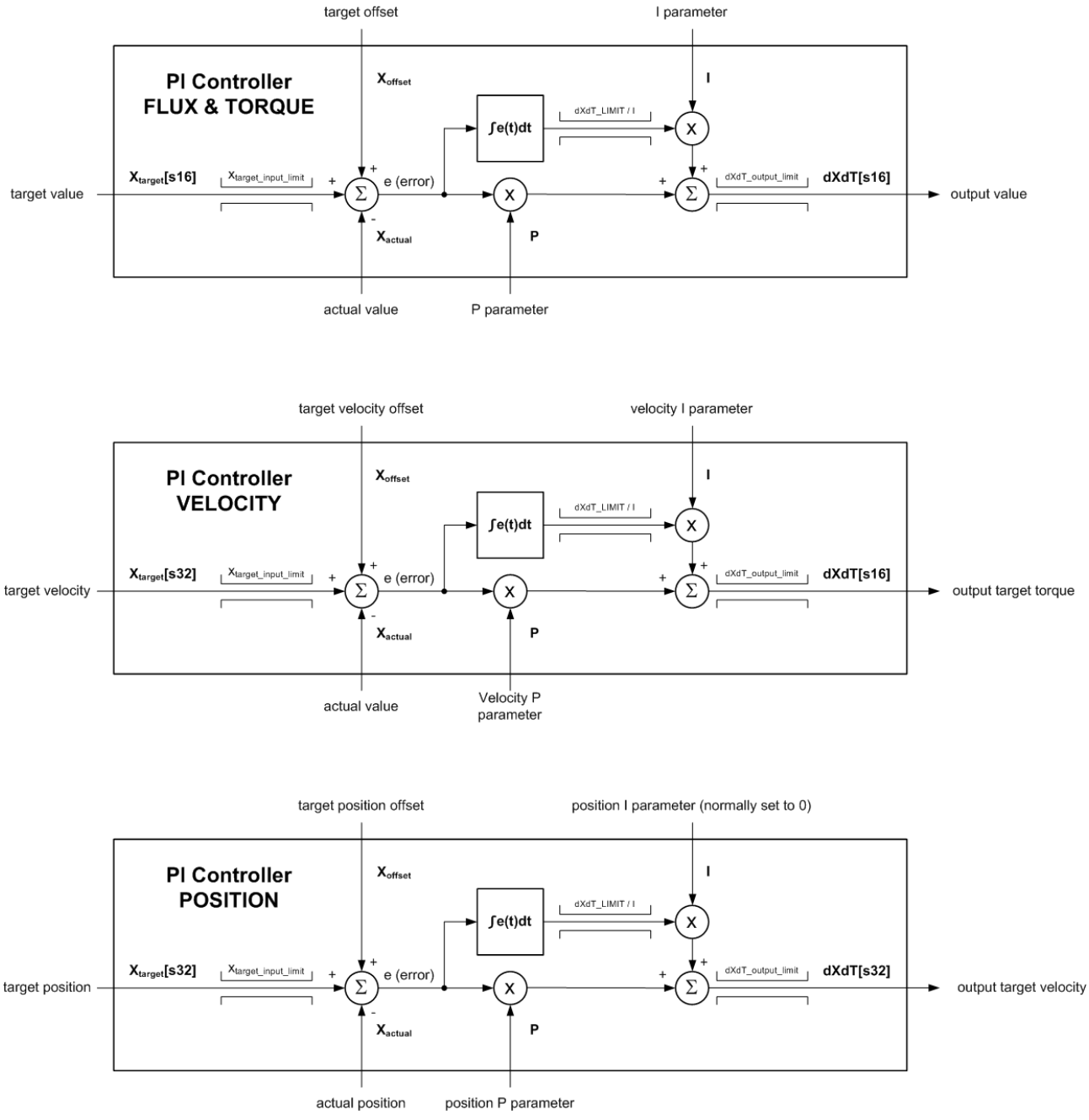


Figure 27: PI Architectures

6.6.5 PI Flux & PI Torque Controller

The P part is represented as q8.8 and I is the I part represented as q0.15.

6.6.6 PI Velocity Controller

The P part is represented as q8.8 and I is the I part represented as q0.15.

6.6.7 P Position Controller

For the position regulator, the P part is represented as q4.12 to be compatible with the high resolution positions - one single rotation is handled as an s16.

This is because e = x - x_target might result in larger e[s32] for x[s32] and x_target[s32] represented as s32 for e = x - x_target for x[s16] and x_target[s16] represented as s16.

6.6.8 Inner FOC Control Loop - Flux & Torque

The inner FOC loop (figure 28) controls the flux current to the flux target value and the torque current to the desired torque target. The inner FOC loop performs the desired transformations according to figure 29 for 3-phase motors (FOC3). For 2-phase motors (FOC2) both Clark (CLARK) transformation and inverse Clark (iCLARK) a by-passed. For control of DC motors transformations are bypassed and only the first full bridge (X1 and X2) is used.

The inner FOC control loop gets a target torque value (I_Q_TARGET), which represents acceleration, the rotor position, and the measured currents as input data. Together with the programmed P and I parameters, the inner FOC loop calculates the target voltage values as input for the PWM engine.

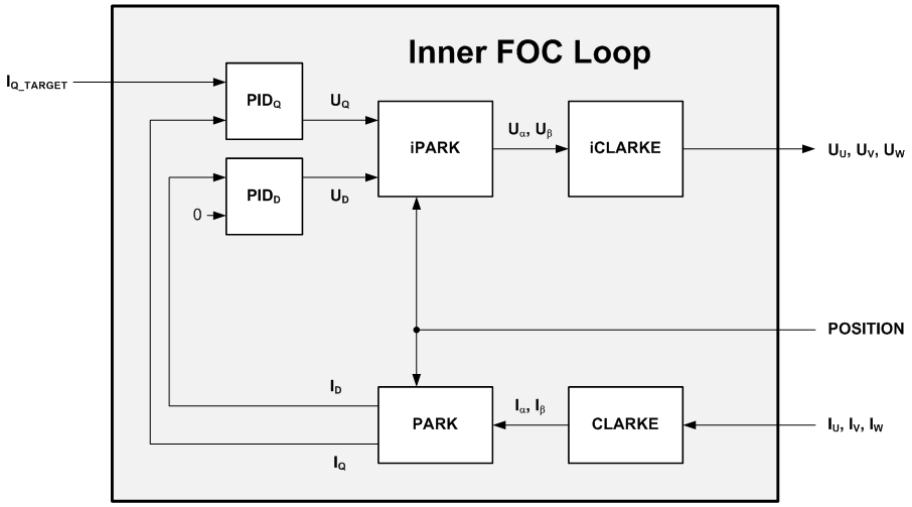


Figure 28: Inner FOC Control Loop

6.6.9 FOC Transformations and PI(D) for control of Flux & Torque

The Clarke transformation (CLARKE) maps three motor phase currents (I_U, I_V, I_W) to a two dimensional coordinate system with two currents (I_α, I_β). Based on the actual rotor angle determined by an encoder or via sensorless techniques, the Park transformation (PARK) maps these two currents to a quasi-static coordinate system with two currents (I_D, I_Q). The current I_D represents flux and the current I_Q represents torque. The flux just pulls on the rotor but does not effect torque. The torque is effected by I_Q. Two PI controllers determine two voltages (U_D, U_Q) to drive desired currents for a target torque and a target flux. The determined voltages (U_D, U_Q) are re-transformed into the stator system by the inverse Parke transformation (iPARK). The inverse Clarke Transformation (iCLARKE) transforms these two currents into three voltages (U_U, U_V, U_W). Theses three voltage are the input of the PWM engine to drive the power stage.

In case of the FOC2, Clarke transformation CLARKE and inverse Clarke Transformation iCLARKE are skipped.

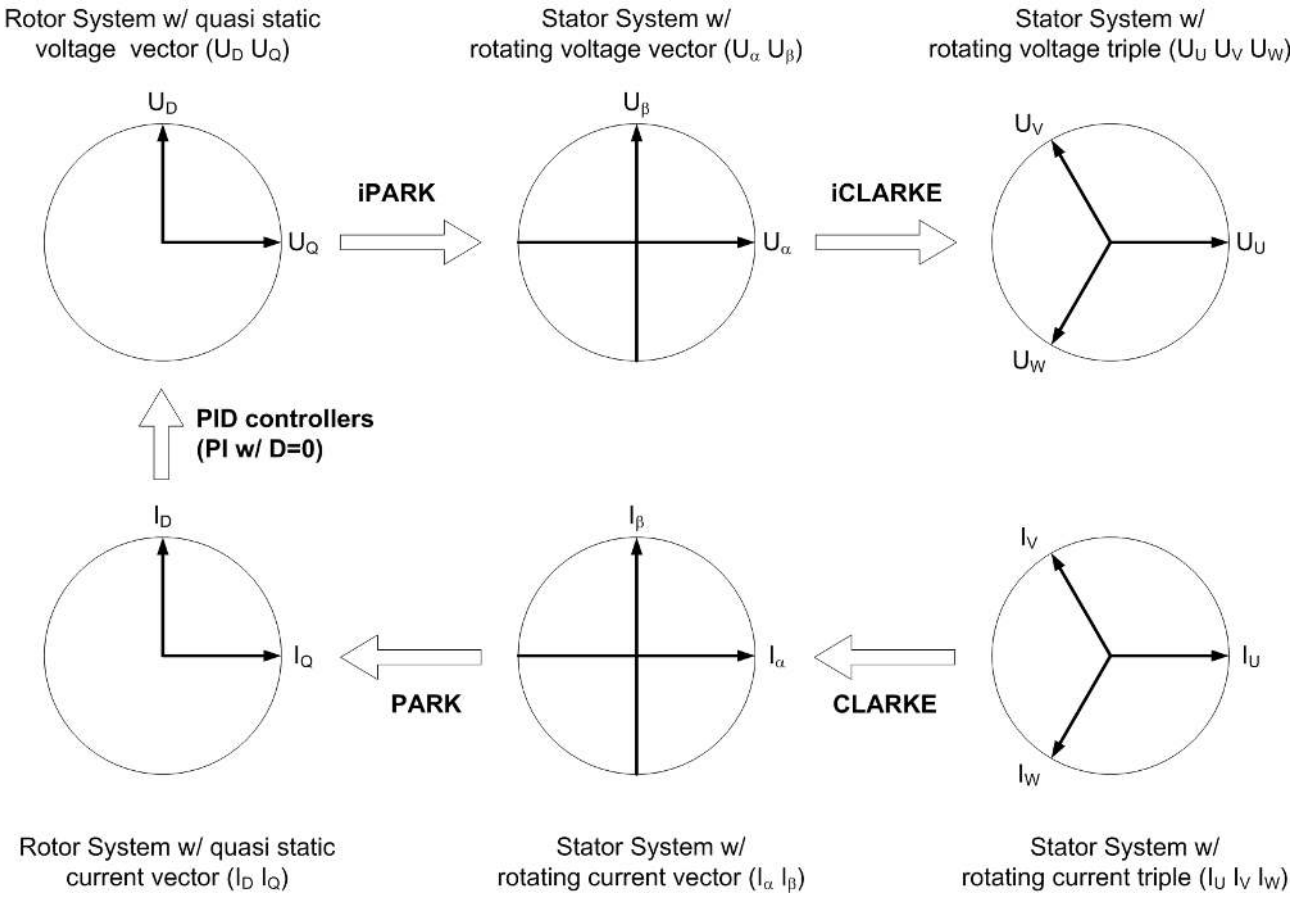


Figure 29: FOC3 Transformations (FOC2 just skips CLARKE and iCLARKE)

6.6.10 Motion Modes

The user can operate the TMC8670 in several motion modes. Standard Motion Modes are position control, velocity control and torque control, where target values are fed into the controllers via register access. The motion mode UD_UQ_EXTERN allows the user to set voltages for open loop operation and for tests during setup.

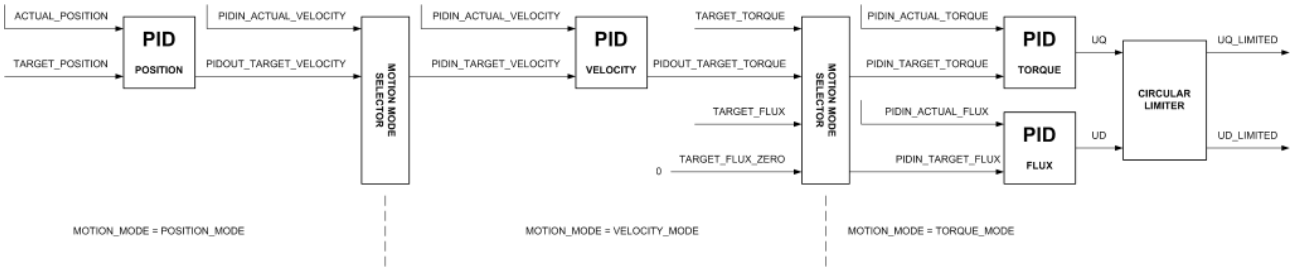


Figure 30: Standard Motion Modes

In position control mode the user can feed the step and direction interface to generate a position target value for the controller cascade. Additional motion modes are the motion mode for Encoder Initialisation (ENCODER_INIT_MINI_MOVE) and motion modes where target values are fed into the TMC8670 via

PWM interface (Pin: PWM_IN) or analog input via pin AGPI_A. These motion modes are recommended for applications, where reference values have to be easily distributed.

There are additional Motion Modes, which are using input from the PWM_I input and the AGPI_A input. Input signals can be scaled via a standard scaler providing offset and gain correction. The interface can be configured via the Registers SINGLE_PIN_IF_OFFSET_SCALE and SINGLE_PIN_IF_STATUS_CFG, where also the status of the interface can be monitored. PWM input signals, which are out of frequency range can be neglected. In case of wrong input data, last correct position is used or velocity and torque are set to zero.

7 EtherCAT Slave Controller Description

7.1 General EtherCAT Information

TMC8670 contains a proven and standard-conform EtherCAT Slave Controller (ESC) providing real-time EtherCAT MAC layer functionality to EtherCAT slave devices. The ESC part of TMC8670 provides the following EtherCAT-related features:

- 4 KByte of Process Data RAM (PDRAM): The PDRAM is a dual ported RAM, which allows exchange of data from the EtherCAT master to the local application.
- Four Sync Managers (SM): Sync Managers are used to control and secure the data exchange via the PDRAM in terms of data consistency, data security, and synchronized read/write operations on the data objects. Two modes –buffered mode and mailbox mode – are available.
- Four Fieldbus Memory Management Units (FMMU): FMMUs are used for mapping of logical addresses to physical addresses. The EtherCAT master uses logical addressing for data than spans multiple slaves. An FMMU can map such a logical address range to a continuous local physical address range.
- 64 bit Distributed Clock support (DC) as core function for EtherCAT's hard real-time capabilities.
- IIC interface for external SII EEPROM for ESC configuration: After reset and at power up, the ESC requires reading basic (and advanced) configuration data from an external SII EEPROM to properly configure interfaces, operation modes, and and feature availability. The SII EEPROM may be read and written by the master or the local application controller as well.
- SPI Process Data Interface (PDI): The PDI is the interface between the local application controller and the ESC. Application-specific process data and EtherCAT control and status information for the EtherCAT State Machine (ESM) is exchanged via this interface. This interface is internal to the TMC8670.

To manufacture own slaves devices, a registration with the EtherCAT Technology Group (ETG) is required. More information and resources on the EtherCAT technology and the EtherCAT standard are available here:

- EtherCAT Technology Group (ETG) (http://www.ethercat.org/http://www.ethercat.org/)
- EtherCAT is standardized by the IEC (http://www.iec.ch/http://www.iec.ch/) and filed as IEC-Standard 61158.

7.2 EtherCAT Register Overview

TMC8670 has an address space of 8 KByte.
The first block of 4KByte (0x0000:0x0FFF) is reserved for the standard ESC- and EtherCAT-relevant configuration and status registers. The Process Data RAM (PDRAM) starts at address 0x1000 and has a size of 4 KByte.

Address	Length (Byte)	Description
		ESC Information
0x0000	1	Type
0x0001	1	Revision
0x0002:0x0003	2	Build
0x0004	1	FMMUs supported
0x0005	1	SyncManagers supported
0x0006	1	RAM Size
0x0007	1	Port Descriptor
0x0008:0x0009	2	ESC Features supported

		Station Address
0x0010:0x0011	2	Configured Station Address
0x0012:0x0013	2	Configured Station Alias

		Write Protection
0x0020	1	Write Register Enable
0x0021	1	Write Register Protection
0x0030	1	ESC Write Enable
0x0031	1	ESC Write Protection

		Data Link Layer
0x0100:0x0103	4	ESC DL Control
0x0108:0x0109	2	Physical Read/Write Offset
0x0110:0x0111	2	ESC DL Status

Address	Length (Byte)	Description
		Application Layer
0x0120:0x0121	2	AL Control
0x0130:0x0131	2	AL Status
0x0134:0x0135	2	AL Status Code
0x0138	1	RUN LED Override
0x0139	1	ERR LED Override

		PDI
0x0140	1	PDI Control
0x0141	1	ESC Configuration
0x014E:0x014F	1	PDI Information
0x0150	4	PDI SPI Slave Configuration
0x0151	4	SYNC/LATCH PDI Configuration
0x0152:0x0153	4	Extended PDI SPI Slave Configuration

		Interrupts
0x0200:0x0201	2	ECAT Event Mask
0x0204:0x0207	4	AL Event Mask
0x0210:0x0211	2	ECAT Event Request
0x0220:0x0223	4	AL Event Request

		Error Counters
0x0300:0x0307	4x2	RX Error Counter[3:0]
0x0308:0x030B	4x1	Forward RX Error Counter[3:0]
0x030C	1	ECAT Processing Unit Error Counter
0x030D	1	PDI Error Counter
0x030E	1	PDI Error Code
0x0310:0x0313	4x1	Lost Link Counter[3:0]

Address	Length (Byte)	Description
		Watchdogs
0x0400:0x0401	2	Watchdog Divider
0x0410:0x0411	2	Watchdog Time PDI
0x0420:0x0421	2	Watchdog Time Process Data
0x0440:0x0441	2	Watchdog Status Process Data
0x0442	1	Watchdog Counter Process Data
0x0443	1	Watchdog Counter PDI

		SII EEPROM Interface
0x0500	1	EEPROM Configuration
0x0501	1	EEPROM PDI Access State
0x0502:0x0503	2	EEPROM Control/Status
0x0504:0x0507	4	EEPROM Address
0x0508:0x050F	4/8	EEPROM Data

		MII Management Interface
0x0510:0x0511	2	MII Management Control/Status
0x0512	1	PHY Address
0x0513	1	PHY Register Address
0x0514:0x0515	2	PHY Data
0x0516	1	MII Management ECAT Access State
0x0517	1	MII Management PDI Access State
0x0518:0x051B	4	PHY Port Status

Address	Length (Byte)	Description
0x0600:0x06FF	**16x16**	**FMMU[15:0]**
+0x0:0x3	4	Logical Start Address
+0x4:0x5	2	Length
+0x6	1	Logical Start bit
+0x7	1	Logical Stop bit
+0x8:0x9	2	Physical Start Address
+0xA	1	Physical Start bit
+0xB	1	Type
+0xC	1	Activate
+0xD:0xF	3	Reserved

0x0800:0x087F	16x8	SyncManager[15:0]
+0x0:0x1	2	Physical Start Address
+0x2:0x3	2	Length
+0x4	1	Control Register
+0x5	1	Status Register
+0x6	1	Activate
+0x7	1	PDI Control

0x0900:0x09FF		Distributed Clocks (DC)
		DC Receive Times
0x0900:0x0903	4	Receive Time Port 0
0x0904:0x0907	4	Receive Time Port 1
0x0908:0x090B	4	Receive Time Port 2
0x090C:0x090F	4	Receive Time Port 3

Address	Length (Byte)	Description
		DC Time Loop Control Unit
0x0910:0x0917	4/8	System Time
0x0918:0x091F	4/8	Receive Time ECAT Processing Unit
0x0920:0x0927	4/8	System Time Offset
0x0928:0x092B	4	System Time Delay
0x092C:0x092F	4	System Time Difference
0x0930:0x0931	2	Speed Counter Start
0x0932:0x0933	2	Speed Counter Diff
0x0934	1	System Time Difference Filter Depth
0x0935	1	Speed Counter Filter Depth

		DC Cyclic Unit Control
0x0980	1	Cyclic Unit Control

		DC SYNC Out Unit
0x0981	1	Activation
0x0982:0x0983	2	Pulse Length of SYNC signals
0x0984	1	Activation Status
0x098E	1	SYNC0 Status
0x098F	1	SYNC1 Status
0x0990:0x0997	4/8	Start Time Cyclic Operation / Next SYNC0 Pulse
0x0998:0x099F	4/8	Next SYNC1 Pulse
0x09A0:0x09A3	4	SYNC0 Cycle Time
0x09A4:0x09A7	4	SYNC1 Cycle Time

Address	Length (Byte)	Description
		DC LATCH In Unit
0x09A8	1	Latch0 Control
0x09A9	1	Latch1 Control
0x09AE	1	Latch0 Status
0x09AF	1	Latch1 Status
0x09B0:0x09B7	4/8	Latch0 Time Positive Edge
0x09B8:0x09BF	4/8	Latch0 Time Negative Edge
0x09C0:0x09C7	4/8	Latch1 Time Positive Edge
0x09C8:0x09CF	4/8	Latch1 Time Negative Edge

		DC SyncManager Event Times
0x09F0:0x09F3	4	EtherCAT Buffer Change Event Time
0x09F8:0x09FB	4	PDI Buffer Start Event Time
0x09FC:0x09FF	4	PDI Buffer Change Event Time

0x0E00:0x0EFF	**256**	**ESC Specific**
0x0E00:0x0E07	8	Product ID
0x0E08:0x0E0F	8	Vendor ID

0x0F80:0x0FFF	128	**User RAM**
0x0F80:0x0FFF	20	reserved

		Process Data RAM
0x1000:0x1FFF	4KB	TMC8670

Table 4: TMC8670 EtherCAT Registers

For Registers longer than one byte, the LSB has the lowest and MSB the highest address.

7.3 EtherCAT Register Set

7.3.1 ESC Information

7.3.1.1 Type (0x0000)

Bit	Description	ECAT	PDI	Reset Value
7:0	Type of EtherCAT controller	r/-	r/-	TMC8460: 0xD0 TMC8461: 0xD0 TMC8462: 0xD0 TMC8670: 0xD0

Table 5: Register 0x0000 (Type)

7.3.1.2 Revision (0x0001)

Bit	Description	ECAT	PDI	Reset Value
7:0	Revision of EtherCAT controller	r/-	r/-	TMC8460: 0x60 TMC8461: 0x61 TMC8462: 0x61 TMC8670: 0x70

Table 6: Register 0x0001 (Revision)

7.3.1.3 Build (0x0002:0x0003)

Bit	Description	ECAT	PDI	Reset Value
15:0	Actual build of EtherCAT controller, minor version, maintenance version	r/-	r/-	TMC8460: 0x10 TMC8461: 0x11 TMC8462: 0x11 TMC8670: 0x10

Table 7: Register 0x0002 (Build)

7.3.1.4 FMMUs supported (0x0004)

Bit	Description	ECAT	PDI	Reset Value
7:0	Number of supported FMMU channels (or entities) of the EtherCAT slave controlller.	r/-	r/-	TMC8460: 6 TMC8461: 8 TMC8462: 8 TMC8670: 4

Table 8: Register 0x0004 (FMMUs)

7.3.1.5 SyncManagers supported (0x0005)

Bit	Description	ECAT	PDI	Reset Value
7:0	Number of supported SyncManager channels (or entities) of the EtherCAT Slave Controller	r/-	r/-	TMC8460: 6 TMC8461: 8 TMC8462: 8 TMC8670: 4

Table 9: Register 0x0005 (SMs)

7.3.1.6 RAM Size (0x0006)

Bit	Description	ECAT	PDI	Reset Value
7:0	Process Data RAM size supported by the EtherCAT Slave Controller in KByte	r/-	r/-	TMC8460: 16 TMC8461: 16 TMC8462: 16 TMC8670: 4

Table 10: Register 0x0006 (RAM Size)

7.3.1.7 Port Descriptor (0x0007)

Bit	Description	ECAT	PDI	Reset Value
	Port configuration: 00: Not implemented 01: Not configured (SII EEPROM) 10: EBUS 11: MII RMII RGMII			
1:0	Port 0	r/-	r/-	TMC8460: 11 TMC8461: 11 TMC8462: 11 TMC8670: 11
3:2	Port 1	r/-	r/-	TMC8460: 11 TMC8461: 11 TMC8462: 11 TMC8670: 11
7:4	not supported	r/-	r/-	0

Table 11: Register 0x0007 (Port Descriptor)

7.3.1.8 ESC Features supported (0x0008:0x0009)

Bit	Description	ECAT	PDI	Reset Value
0	FMMU Operation: 0: Bit oriented 1: Byte oriented	r/-	r/-	
1	Reserved	r/-	r/-	
2	Distributed Clocks: 0: Not available 1: Available	r/-	r/-	
3	Distributed Clocks (width): 0: 32 bit 1: 64 bit	r/-	r/-	
4	Low Jitter EBUS: 0: Not available, standard jitter 1: Available, jitter minimized	r/-	r/-	0
5	Enhanced Link Detection EBUS: 0: Not available 1: Available	r/-	r/-	0
6	Enhanced Link Detection MII 0: Not available 1: Available	r/-	r/-	

Bit	Description	ECAT	PDI	Reset Value
7	Separate Handling of FCS Errors: 0: Not supported 1: Supported, frames with wrong FCS and additional nibble will be counted separately in Forwarded RX Error Counter	r/-	r/-	
8	Enhanced DCSYNC Activation 0: Not available 1: Available NOTE: This feature refers to registers `0x981.(7:3)`, `0x0984`	r/-	r/-	
9	EtherCAT LRW command support: 0: Supported 1: Not Supported	r/-	r/-	
10	EtherCAT read/write command support 0: Supported 1: Not Supported	r/-	r/-	
11	Fixed FMMU/SyncManager configuration 0: Variable configuration 1: Fixed configuration (refer to documentation of supporting ESCs)	r/-	r/-	
15:12	Reserved	r/-	r/-	

Table 12: Register 0x0008:0x0009 (ESC Features)

7.3.2 Station Address

7.3.2.1 Configured Station Address (0x0010:0x0011)

Bit	Description	ECAT	PDI	Reset Value
15:0	Address used for node addressing (FPxx commands)	r/w	r/-	

Table 13: Register 0x0010:0x0011 (Station Addr)

7.3.2.2 Configured Station Alias (0x0012:0x0013)

Bit	Description	ECAT	PDI	Reset Value
15:0	Alias Address used for node addressing (FPxx commands) The use of this alias is activated by Register DL Control Bit 24 (0x0100.24/0x0103.0) NOTE: EEPROM value is only taken over at first EEPROM load after power-on reset.	r/-	r/w	

Table 14: Register 0x0012:0x0013 (Station Alias)

7.3.3 Write Protection

7.3.3.1 Write Register Enable (0x0020)

Bit	Description	ECAT	PDI	Reset Value
0	If write register protection is enabled, this register has to be written in the same Ethernet frame (value does not care) before other writes to this station are allowed. Write protection is still active after this frame (if Write Register Protection register is not changed).	r/w	r/-	
7:1	Reserved, wirte 0	r/-	r/-	

Table 15: Register 0x0020 (Write Register Enable)

7.3.3.2 Write Register Protection (0x0021)

Bit	Description	ECAT	PDI	Reset Value
0	Write register protection: 0: Protection disabled 1: Protection enabled Registers 0x0000-0x0137, 0x013A-0x0F0F are write protected, except for 0x0030	r/w	r/-	
7:1	Reserved, write 0	r/-	r/-	

Table 16: Register 0x0021 (Write Register Prot.)

7.3.3.3 ESC Write Enable (0x0030)

Bit	Description	ECAT	PDI	Reset Value
0	If ESC write protection is enabled, this register has to be written in the same Ethernet frame (value does not care) before other writes to this station are allowed. ESC write protection is still active after this frame (if ESC Write Protection register is not changed).	r/w	r/-	
7:1	Reserved, write 0	r/-	r/-	

Table 17: Register 0x0030 (ESC Write Enable)

7.3.3.4 ESC Write Protection (0x0031)

Bit	Description	ECAT	PDI	Reset Value
15:0	Write protect: 0: Protection disabled 1: Protection enabled All areas are write protected, except for 0x0030.	r/w	r/-	
7:1	Reserved, write 0	r/-	r/-	

Table 18: Register 0x0031 (ESC Write Prot.)

7.3.4 Data Link Layer

7.3.4.1 ESC DL Control (0x0100:0x0103)

Bit	Description	ECAT	PDI	Reset Value
0	Forwarding rule: 0: EtherCAT frames are processed, Non-EtherCAT frames are forwarded without processing 1: EtherCAT frames are processed, Non-EtherCAT frames are destroyed The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 - locally administered address) regardless of the forwarding rule.	r/-	r/-	
1	Temporary use of settings in Register 0x101: 0: permanent use 1: use for about 1 second, then revert to previous settings	r/-	r/-	
7:2	Reserved, write 0	r/-	r/-	
9:8	Loop Port 0: 00: Auto 01: Auto Close 10: Open 11: Closed Note Loop open means sending/receiving over this port is enabled, loop closed means sending/receiving is disabled and frames are forwarded to the next open port internally. Auto: loop closed at link down, opened at link up Auto Close: loop closed at link down, opened with writing 01 again after link up (or receiving a valid Ethernet frame at the closed port) Open: loop open regardless of link state Closed: loop closed regardless of link state	r/w*	r/-	
11:10	Loop Port 1: 00: Auto 01: Auto Close 10: Open 11: Closed	r/w*	r/-	
13:12	Loop Port 2: 00: Auto 01: Auto Close 10: Open 11: Closed	r/w*	r/-	
15:14	Loop Port 3: 00: Auto 01: Auto Close 10: Open 11: Closed	r/w*	r/-	

Bit	Description	ECAT	PDI	Reset Value
18:16	RX FIFO Size (ESC delays start of forwarding until FIFO is at least half full). RX FIFO Size/RX delay reduction** : Value (for MII): 0: -40 ns 1: -40 ns 2: -40 ns 3: -40 ns 4: no change 5: no change 6: no change 7: default default NOTE: EEPROM value is only taken over at first EEPROM load after power-on or reset	r/w	r/-	
19	EBUS Low Jitter: 0: Normal jitter / 1: Reduced jitter	r/w	r/-	0
21:20	Reserved, write 0	r/w	r/-	
22	EBUS remote link down signaling time: 0: Default (≈660 ms) 1: Reduced (≈80 μs)	r/w	r/-	0
23	Reserved, write 0	r/w	r/-	
24	Station alias: 0: Ignore Station Alias 1: Alias can be used for all configured address command types (FPRD, FPWR, ...)	r/w	r/-	
31:25	Reserved, write 0	r/-	r/-	

Table 19: Register 0x0100:0x0103 (DL Control)

* Loop configuration changes are delayed until end of currently received or transmitted frame at the port.
** The possibility of RX FIFO Size reduction depends on the clock source accuracy of the ESC and of every connected EtherCAT/Ethernet devices (master, slave, etc.). RX FIFO Size of 7 is sufficient for 100ppm accuracy, FIFO Size 0 is possible with 25ppm accuracy (frame size of 1518/1522 Byte).

7.3.4.2 Physical Read/Write Offset (0x0108:0x0109)

Bit	Description	ECAT	PDI	Reset Value
15:0	Offset of R/W Commands (FPRW, APRW) between Read address and Write address. RD_ADR = ADR and WR_ADR = ADR + R/W-Offset 0	r/w	r/-	

Table 20: Register 0x0108:0x0109 (R/W Offset)

7.3.4.3 ESC DL Status (0x0110:0x0111)

Bit	Description	ECAT	PDI	Reset Value
0	PDI operational/EEPROM loaded correctly: 0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)	r*/-	r/-	
1	PDI Watchdog Status: 0: Watchdog expired 1: Watchdog reloaded	r*/-	r/-	
2	Enhanced Link detection: 0: Deactivated for all ports 1: Activated for at least one port NOTE: EEPROM value is only taken over at first EEPROM load after power-on or reset	r*/-	r/-	
3	Reserved	r*/-	r/-	
4	Physical link on Port 0: 0: No link 1: Link detected	r*/-	r/-	
5	Physical link on Port 1: 0: No link 1: Link detected	r*/-	r/-	
6	Physical link on Port 2: 0: No link 1: Link detected	r*/-	r/-	
7	Physical link on Port 3: 0: No link 1: Link detected	r*/-	r/-	
8	Loop Port 0: 0: Open 1: Closed	r*/-	r/-	
9	Communication on Port 0: 0: No stable communication 1: Communication established	r*/-	r/-	
10	Loop Port 1: 0: Open 1: Closed	r*/-	r/-	
11	Communication on Port 1: 0: No stable communication 1: Communication established	r*/-	r/-	
12	Loop Port 2: 0: Open 1: Closed	r*/-	r/-	
13	Communication on Port 2: 0: No stable communication 1: Communication established	r*/-	r/-	

Bit	Description	ECAT	PDI	Reset Value
14	Loop Port 3: 0: Open 1: Closed	r*/-	r/-	
15	Communication on Port 3: 0: No stable communication 1: Communication established	r*/-	r/-	

Table 21: Register 0x0110:0x0111 (DL Status)

* Reading DL Status register from ECAT clears ECAT Event Request 0x0210.2.

Register 0x0111	Port 3	Port2	Port1	Port 0
0x55	No link, closed	No link, closed	No link, closed	No link, closed
0x56	No link, closed	No link, closed	No link, closed	Link, open
0x59	No link, closed	No link, closed	Link, open	No link, closed
0x5A	No link, closed	No link, closed	Link, open	Link, open
0x65	No link, closed	Link, open	No link, closed	No link, closed
0x66	No link, closed	Link, open	No link, closed	Link, open
0x69	No link, closed	Link, open	Link, open	No link, closed
0x6A	No link, closed	Link, open	Link, open	Link, open
0x95	Link, open	No link, closed	No link, closed	No link, closed
0x96	Link, open	No link, closed	No link, closed	Link, open
0x99	Link, open	No link, closed	Link, open	No link, closed
0x9A	Link, open	No link, closed	Link, open	Link, open
0xA5	Link, open	Link, open	No link, closed	No link, closed
0xA6	Link, open	Link, open	No link, closed	Link, open
0xA9	Link, open	Link, open	Link, open	No link, closed
0xAA	Link, open	Link, open	Link, open	Link, open
0xD5	Link, closed	No link, closed	No link, closed	No link, closed
0xD6	Link, closed	No link, closed	No link, closed	Link, open
0xD9	Link, closed	No link, closed	Link, open	No link, closed
0xDA	Link, closed	No link, closed	Link, open	Link, open

Table 22: Decoding port state in ESC DL Status register 0x0111 (typical modes only)

7.3.5 Application Layer

7.3.5.1 AL Control (0x0120:0x0121)

Bit	Description	ECAT	PDI	Reset Value
3:0	Initiate State Transition of the Device State Machine: 1: Request Init State 3: Request Bootstrap State 2: Request Pre-Operational State 4: Request Safe-Operational State 8: Request Operational State	r/(w)	r/ (wack)*	
4	Error Ind Ack: 0: No Ack of Error Ind in AL status register 1: Ack of Error Ind in AL status register	r/(w)	r/ (wack)*	
4	Device Identification: 0: No request 1: Device Identification request	r/(w)	r/ (wack)*	
15:6	Reserved, write 0	r/(w)	r/ (wack)*	

Table 23: Register 0x0120:0x0121 (AL Cntrl)

Note

AL Control register behaves like a mailbox if Device Emulation is off (0x0140.8=0): The PDI has to read/write* the AL Control register after ECAT has written it. Otherwise ECAT cannot write again to the AL Control register. After Reset, AL Control register can be written by ECAT. (Regarding mailbox functionality, both registers 0x0120 and 0x0121 are equivalent, e.g. reading 0x0121 is sufficient to make this register writeable again.)

If Device Emulation is on, the AL Control register can always be written, its content is copied to the AL Status register.
* PDI register function acknowledge by Write command is disabled: Reading AL Control from PDI clears AL Event Request 0x0220.0. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing AL Control from PDI clears AL Event Request 0x0220.0. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.5.2 AL Status (0x0130:0x0131)

Bit	Description	ECAT	PDI	Reset Value
3:0	Actual State of the Device State Machine: 1: Init State 3: Request Bootstrap State 2: Pre-Operational State 4: Safe-Operational State 8: Operational State	r*/-	r/(w)	
4	Error Ind: 0: Device is in State as requested or Flag cleared by command 1: Device has not entered requested State or changed State as result of a local action	r*/-	r/(w)	
5	Device Identification: 0: Device Identification not valid 1: Device Identification loaded	r*/-	r/(w)	
15:6	Reserved, write 0	r*/-	r/(w)	

Table 24: Register 0x0130:0x0131 (AL Status)

Note AL Status register is only writable from PDI if Device Emulation is off (0x0140.8=0), otherwise AL Status register will reflect AL Control register values.

* Reading AL Status from ECAT clears ECAT Event Request 0x0210.3.

7.3.5.3 AL Status Code (0x0134:0x0135)

Bit	Description	ECAT	PDI	Reset Value
15:0	AL Status Code	r/-	r/w	

Table 25: Register 0x0134:0x0135 (AL Status Code)

7.3.5.4 RUN LED Override (0x0138)

Bit	Description	ECAT	PDI	Reset Value
3:0	LED code: (FSM State) 0x0: Off (1-Init) 0x1-0xC: Flash 1x - 12x (4-SafeOp 1x) 0xD: Blinking (2-PreOp) 0xE: Flickering (3-Bootrap) 0xF: On	r/w	r/w	
4	Enable Override: 0: Override disabled 1: Override enabled	r/w	r/w	
7:5	Reserved, write 0	r/w	r/w	

Table 26: Register 0x0138 (RUN LED Override)

Note Changes to AL Status register (0x0130) with valid values will disable RUN LED Override (0x0138.4=0). The value read in this register always reflects current LED output.

7.3.5.5 ERR LED Override (0x0139)

Bit	Description	ECAT	PDI	Reset Value
3:0	LED code: 0x0: Off 0x1-0xC: Flash 1x - 12x 0xD: Blinking 0xE: Flickering 0xF: On	r/w	r/w	
4	Enable Override: 0: Override disabled 1: Override enabled	r/w	r/w	
7:5	Reserved, write 0	r/w	r/w	

Table 27: Register 0x0139 (ERR LED Override)

Note New error conditions will disable ERR LED Override (0x0139.4=0). The value read in this register always reflects current LED output.

7.3.6 PDI

7.3.6.1 PDI Control (0x0140)

Bit	Description	ECAT	PDI	Reset Value
7:0	Process data interface: 0x00: Interface deactivated (no PDI) ... 0x05: **SPI Slave** ... 0x80: On-chip bus Others: Reserved	r/-	r/-	TMC8460, TMC8461, TMC8462, TMC8670: 0x00 later EEPROM ADR 0x0000 only SPI Slave (0x05) is supported in the hardware

Table 28: Register 0x0140 (PDI Control)

7.3.6.2 ESC Configuration (0x0141)

Bit	Description	ECAT	PDI	Reset Value
0	Device emulation (control of AL status): 0: AL status register has to be set by PDI 1: AL status register will be set to value written to AL control register	r/w	r/-	
1	Enhanced Link detection all ports: 0: disabled (if bits [7:4]=0) 1: enabled at all ports (overrides bits [7:4])	r/-	r/-	
2	Distributed Clocks SYNC Out Unit: 0: disabled (power saving) / 1: enabled	r/-	r/-	
3	Distributed Clocks Latch In Unit: 0: disabled (power saving) / 1: enabled	r/-	r/-	
4	Enhanced Link port 0: 0: disabled (if bit 1=0) / 1: enabled	r/-	r/-	
5	Enhanced Link port 1: 0: disabled (if bit 1=0) / 1: enabled	r/-	r/-	
6	Enhanced Link port 2: 0: disabled (if bit 1=0) / 1: enabled	r/-	r/-	
7	Enhanced Link port 3: 0: disabled (if bit 1=0) / 1: enabled	r/-	r/-	

Table 29: Register 0x0141 (ESC Config)

7.3.6.3 PDI Information (0x014E:0x014F)

Bit	Description	ECAT	PDI	Reset Value
0	PDI register function acknowledge by write: 0: Disabled 1: Enabled	r/w	r/-	Depends on configuration
1	PDI configured: 0: PDI not configured 1: PDI configured (EEPROM loaded)	r/w	r/-	0
2	PDI active: 0: PDI not active 1: PDI active	r/w	r/-	0
3	PDI configuration invalid: 0: PDI configuration ok 1: PDI configuration invalid	r/w	r/-	0
7:4	Reserved	r/w	r/-	0

Table 30: Register 0x014E (PDI Information))

7.3.6.4 PDI SPI Slave Configuration (0x0150)

The PDI configuration register 0x0150 and the extended PDI configuration registers 0x0152:0x0153 depend on the selected PDI. The Sync/Latch[1:0] PDI configuration register 0x0151 is independent of the selected PDI. The TMC8460, TMC8461, TMC8462, and TMC8670 devices support SPI Slave PDI only.

Bit	Description	ECAT	PDI	Reset Value
1:0	SPI mode: 00: SPI mode 0 01: SPI mode 1 10: SPI mode 2 11: SPI mode 3 NOTE: SPI mode 3 is recommended for Slave Sample Code NOTE: SPI status flag is not available in SPI modes 0 and 2 with normal data out sample.	r/-	r/-	
3:2	SPI_IRQ output driver/polarity: 00: Push-Pull active low 01: Open Drain (active low) 10: Push-Pull active high 11: Open Source (active high)	r/-	r/-	
4	SPI_CSNL polarity: 0: Active low 1: Active high	r/-	r/-	
5	Data Out sample mode: 0: Normal sample (SPI_MISO and SPI_MOSI are sampled at the same SPI_CLK edge) 1: Late sample (SPI_MISO and SPI_MOSI are sampled at different SPI_CLK edges)	r/-	r/-	
7:6	Reserved, set EEPROM value 0	r/-	r/-	

Table 31: Register 0x0150 (PDI SPI CFG)

7.3.6.5 SYNC/LATCH Configuration (0x0151)

Bit	Description	ECAT	PDI	Reset Value
1:0	SYNC0 output driver/polarity: 00: Push-Pull active low 01: Open Drain (active low) 10: Push-Pull active high 11: Open Source (active high)	r/-	r/-	TMC8461: 10 TMC8462: 10
2	SYNC0/LATCH0 configuration: 0: LATCH0 Input 1: SYNC0 Output	r/-	r/-	TMC8461: 1 TMC8462: 1
3	SYNC0 mapped to AL Event Request register 0x0220.2: 0: Disabled 1: Enabled	r/-	r/-	TMC8461, TMC8462: depends on configuration

Bit	Description	ECAT	PDI	Reset Value
5:4	SYNC1 output driver/polarity: 00: Push-Pull active low 01: Open Drain (active low) 10: Push-Pull active high 11: Open Source (active high)	r/-	r/-	TMC8461: 10 TMC8462: 10
6	SYNC1/LATCH1 configuration*: 0: LATCH1 input 1: SYNC1 output	r/-	r/-	TMC8461: 1 TMC8462: 1
7	SYNC1 mapped to AL Event Request register 0x0220.3: 0: Disabled 1: Enabled	r/-	r/-	TMC8461, TMC8462: depends on configuration

Table 32: Register 0x0151 (SYNC/LATCH CFG)

7.3.6.6 PDI SPI Slave Extended Configuration (0x0152:0x0153)

Bit	Description	ECAT	PDI	Reset Value
15:0	Reserved, set EEPROM value 0	r/-	r/-	TMC8461: 0 TMC8462: 0

Table 33: Register 0x0152:0x0153 (PDI SPI extCFG)

7.3.7 Interrupts

7.3.7.1 ECAT Event Mask (0x0200:0x0201)

Bit	Description	ECAT	PDI	Reset Value
15:0	ECAT Event masking of the ECAT Event Request Events for mapping into ECAT event field of EtherCAT frames: 0: Corresponding ECAT Event Request register bit is not mapped 1: Corresponding ECAT Event Request register bit is mapped	r/w	r/-	

Table 34: Register 0x0200:0x0201 (ECAT Event M.)

7.3.7.2 AL Event Mask (0x0204:0x0207)

Bit	Description	ECAT	PDI	Reset Value
31:0	AL Event masking of the AL Event Request register Events for mapping to PDI IRQ signal: 0: Corresponding AL Event Request register bit is not mapped 1: Corresponding AL Event Request register bit is mapped	r/-	r/w	

Table 35: Register 0x0204:0x0207 (AL Event Mask)

7.3.7.3 ECAT Event Request (0x0210:0x0211)

Bit	Description	ECAT	PDI	Reset Value
0	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from ECAT for ECAT controlled Latch Units, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event)	r/-	r/-	
1	Reserved	r/-	r/-	
2	DL Status event: 0: No change in DL Status 1: DL Status change (Bit is cleared by reading out DL Status 0x0110:0x0111 from ECAT)	r/-	r/-	

Bit	Description	ECAT	PDI	Reset Value
3	AL Status event: 0: No change in AL Status 1: AL Status change (Bit is cleared by reading out AL Status 0x0130:0x0131 from ECAT)	r/-	r/-	
4 5 ...	Mirrors values of each SyncManager Status: 0: No Sync Channel 0 event 1: Sync Channel 0 event pending 0: No Sync Channel 1 event 1: Sync Channel 1 event pending ... 0: No Sync Channel 7 event 1: Sync Channel 7 event pending	r/w	r/-	
15:12	Reserved	r/-	r/-	

Table 36: Register 0x0210:0x0211 (ECAT Event R.)

7.3.7.4 AL Event Request (0x0220:0x0223)

Bit	Description	ECAT	PDI	Reset Value
0	AL Control event: 0: No AL Control Register change 1: AL Control Register has been written[1] (Bit is cleared by reading AL Control register 0x0120:0x0121 from PDI)	r/-	r/-	
1	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from PDI, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event. Available if Latch Unit is PDI controlled)	r/-	r/-	
2	State of DC SYNC0 (if register 0x0151.3=1): (Bit is cleared by reading SYNC0 status 0x098E from PDI, use only in Acknowledge mode)	r/-	r/-	
3	State of DC SYNC1 (if register 0x0151.7=1): (Bit is cleared by reading of SYNC1 status 0x098F from PDI, use only in Acknowledge mode)	r/-	r/-	

[1] AL control event is only generated if PDI emulation is turned off (PDI Control register 0x0140.8=0)

Bit	Description	ECAT	PDI	Reset Value
4	SyncManager activation register (SyncManager register offset 0x6) changed: 0: No change in any SyncManager 1: At least one SyncManager changed (Bit is cleared by reading SyncManager Activation registers 0x0806 etc. from PDI)	r/-	r/-	
5	EEPROM Emulation: 0: No command pending 1: EEPROM command pending (Bit is cleared by acknowledging the command in EEPROM command register 0x0502 from PDI)	r/-	r/-	
6	Watchdog Process Data: 0: Has not expired 1: Has expired (Bit is cleared by reading Watchdog Status Process Data 0x0440 from PDI)	r/-	r/-	
7	Reserved	r/-	r/-	
8 9 ... 23	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending ... 0: No SyncManager 15 interrupt 1: SyncManager 15 interrupt pending	r/-	r/-	
31:24	Reserved	r/-	r/-	

Table 37: Register 0x0220:0x0223 (AL Event R.)

7.3.8 Error Counters

Errors are only counted if the corresponding port is enabled.

7.3.8.1 RX Error Counter[3:0] (0x0300:0x0307)

Bit	Description	ECAT	PDI	Reset Value
7:0	Invalid frame counter of Port y (counting is stopped when 0xFF is reached).	r/ w(clr)	r/-	
15:8	RX Error counter of Port y (counting is stopped when 0xFF is reached). This is coupled directly to RX ERR of MII interface.	r/ w(clr)	r/-	

Table 38: Register 0x0300:0x0307 (RX Err Cnt)

7.3.8.2 Forward RX Error Counter[3:0] (0x0308:0x030B)

Bit	Description	ECAT	PDI	Reset Value
7:0	Forwarded error counter of Port y (counting is stopped when 0xFF is reached).	r/ w(clr)	r/-	

Table 39: Register 0x0308:0x030B (FW RX Err Cnt)

Note Error Counters 0x0300-0x030B are cleared if one of the RX Error counters 0x0300-0x030B is written. Write value is ignored (write 0).

7.3.8.3 ECAT Processing Unit Error Counter (0x030C)

Bit	Description	ECAT	PDI	Reset Value
7:0	ECAT Processing Unit error counter (counting is stopped when 0xFF is reached). Counts errors of frames passing the Processing Unit (e.g., FCS is wrong or datagram structure is wrong).	r/ w(clr)	r/-	

Table 40: Register 0x030C (Proc. Unit Err Cnt)

Note Error Counter 0x030C is cleared if error counter 0x030C is written. Write value is ignored (write 0).

7.3.8.4 PDI Error Counter (0x030D)

Bit	Description	ECAT	PDI	Reset Value
7:0	PDI Error counter (counting is stopped when 0xFF is reached). Counts if a PDI access has an interface error.	r/ w(clr)	r/-	

Table 41: Register 0x030D (PDI Err Cnt)

Note Error Counter 0x030D and Error Code 0x030E are cleared if error counter 0x030D is written. Write value is ignored (write 0).

7.3.8.5 PDI Error Code (0x030E)

Bit	Description	ECAT	PDI	Reset Value
	SPI access which caused last PDI Error. Cleared if register 0x030D is written.	r/-	r/-	
2:0	Number of SPI clock cycles of whole access (modulo 8)	r/-	r/-	
3	Busy violation during read access	r/-	r/-	
4	Read termination missing	r/-	r/-	
5	Access continued after read termination byte	r/-	r/-	
7:6	SPI command CMD[2:1]	r/-	r/-	

Table 42: Register 0x030E (PDI Err Code)

Note Error Counter 0x030D and Error Code 0x030E are cleared if error counter 0x030D is written. Write value is ignored (write 0).

7.3.8.6 Lost Link Counter[3:0] (0x0310:0x0313)

Bit	Description	ECAT	PDI	Reset Value
7:0	Lost Link counter of Port y (counting is stopped when 0xff is reached). Counts only if port loop is Auto.	r/w(clr)	r/-	

Table 43: Register 0x0310:0x0313 (LL Counter)

Note Only lost links at open ports are counted. Lost Link Counters 0x0310-0x0313 are cleared if one of the Lost Link Counters 0x0310-0x0313 is written. Write value is ignored (write 0).

7.3.9 Watchdogs

7.3.9.1 Watchdog Divider (0x0400:0x0401)

Bit	Description	ECAT	PDI	Reset Value
15:0	Watchdog Time PDI: number or basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)	r/w	r/-	

Table 44: Register 0x0400:0x0401 (WD Divider)

7.3.9.2 Watchdog Time PDI (0x0410:0x0411)

Bit	Description	ECAT	PDI	Reset Value
15:0	Watchdog Time PDI: number or basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)	r/w	r/-	

Table 45: Register 0x0410:0x0411 (WD Time PDI)

Note Watchdog is disabled if Watchdog time is set to 0x0000. Watchdog is restarted with every PDI access.

7.3.9.3 Watchdog Time Process Data (0x0420:0x0421)

Bit	Description	ECAT	PDI	Reset Value
15:0	Watchdog Time Process Data: number of basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)	r/w	r/-	

Table 46: Register 0x0420:0x0421 (WD Time PD)

Note There is one Watchdog for all SyncManagers. Watchdog is disabled if Watchdog time is set to 0x0000. Watchdog is restarted with every write access to SyncManagers with Watchdog Trigger Enable Bit set.

7.3.9.4 Watchdog Status Process Data (0x0440:0x0441)

Bit	Description	ECAT	PDI	Reset Value
15:0	Watchdog Status of Process Data (triggered by SyncManagers) 0: Watchdog Process Data expired 1: Watchdog Process Data is active or disabled	r/-	r/ (w ack)*	
0	Reserved	r/-	r/ (w ack)*	

Table 47: Register 0x0440:0x0441 (WD Status PD)

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220.6. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220.6. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.9.5 Watchdog Counter Process Data (0x0442)

Bit	Description	ECAT	PDI	Reset Value
7:0	Watchdog Counter Process Data (counting is stopped when 0xFF is reached). Counts if Process Data Watchdog expires.	r/ w(clr)	r/-	

Table 48: Register 0x0442 (WD Counter PD)

Note Watchdog Counters 0x0442-0x0443 are cleared if one of the Watchdog Counters 0x0442-0x0443 is written. Write value is ignored (write 0).

7.3.9.6 Watchdog Counter PDI (0x0443)

Bit	Description	ECAT	PDI	Reset Value
7:0	Watchdog PDI counter (counting is stopped when 0xFF is reached). Counts if PDI Watchdog expires.	r/ w(clr)	r/-	

Table 49: Register 0x0443 (WD Counter PDI)

Note Watchdog Counters 0x0442 & 0x0443 are cleared if one of the Watchdog Counters 0x0442 & 0x0443 is written. Write value is ignored (write 0).

7.3.10 SII EEPROM Interface

Address	Length (Byte)	Description
		SII EEPROM Interface
0x0500	1	EEPROM Configuration
0x0501	1	EEPROM PDI Access State
0x0502:0x0503	2	EEPROM Control/Status
0x0504:0x0507	4	EEPROM Address
0x0508:0x050F	4/8	EEPROM Data

Table 50: SII EEPROM Interface Register Overview

7.3.10.1 EEPROM Configuration (0x0500)

Bit	Description	ECAT	PDI	Reset Value
0	EEPROM control is offered to PDI: 0: no 1: yes (PDI has EEPROM control)	r/w	r/-	
1	Force ECAT access: 0: Do not change Bit 0x0501.0 1: Reset Bit 0x0501.0 to 0	r/w	r/-	
7:2	Reserved, write 0	r/w	r/-	

Table 51: Register 0x0500 (PROM Config)

7.3.10.2 EEPROM PDI Access State (0x0501)

Bit	Description	ECAT	PDI	Reset Value
0	Access to EEPROM: 0: PDI releases EEPROM access 1: PDI takes EEPROM access (PDI has EEPROM control)	r/-	r/(w)	
7:1	Reserved, write 0	r/-	r/-	

Table 52: Register 0x0501 (PROM PDI Access)

Note r/(w): write access is only possible if 0x0500.0=1 and 0x0500.1=0.

7.3.10.3 EEPROM Control/Status (0x0502:0x0503)

Bit	Description	ECAT	PDI	Reset Value
0	ECAT write enable*2: 0: Write requests are disabled 1: Write requests are enabled This bit is always 1 if PDI has EEPROM control.	r/(w)	r/-	
4:1	Reserved, write 0	r/-	r/-	
5	EEPROM emulation: 0: Normal operation (I2C interface used) 1: PDI emulates EEPROM (I2C not used)	r/-	r/-	
6	Supported number of EEPROM read bytes: 0: 4 Bytes 1: 8 Bytes	r/-	r/-	
7	Selected EEPROM Algorithm: 0: 1 address byte (1KBit ...16KBit EEPROMs) 1: 2 address bytes (32KBit ...4 MBit EEPROMs)	r/-	r/- r/[w]	
10:8	Command register*1: Write: Initiate command. Read: Currently executed command Commands: 000: No command/EEPROM idle (clear error bits) 001: Read 010: Write 100: Reload Others: Reserved/invalid commands (do not issue) EEPROM emulation only: after execution, PDI writes command value to indicate operation is ready.	r/(w)	r/(w) r/[w]	
11	Checksum Error at in ESC Configuration Area: 0: Checksum ok 1: Checksum error	r/-	r/-	
12	EEPROM loading status: 0: EEPROM loaded, device information ok 1: EEPROM not loaded, device information not available (EEPROM loading in progress or finished with a failure)	r/-	r/-	
13	Error Acknowledge/Command*2: 0: No error 1: Missing EEPROM acknowledge or invalid command EEPROM emulation only: PDI writes 1 if a temporary failure has occurred.	r/-	r/- r/[w]	

Bit	Description	ECAT	PDI	Reset Value
14	Error Write Enable*2: 0: No error 1: Write Command without Write enable	r/-	r/-	
15	Busy: 0: EEPROM Interface is idle 1: EEPROM Interface is busy	r/-	r/-	

Table 53: Register 0x0502:0x0503 (PROM Cntrl)

Note r/(w): write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if EEPROM interface is busy (0x0502.15=1).

Note r/[w]: EEPROM emulation only: write access is possible if EEPROM interface is busy (0x0502.15=1). PDI acknowledges pending commands by writing a 1 into the corresponding command register bits (0x0502.10:8).
Errors can be indicated by writing a 1 into the error bit 0x0502.13. Acknowledging clears AL Event Request 0x0220.5.

*1 Write Enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (EEPROM Busy ends). Writing "'000"' to the command register will also clear the error bits [14:13]. Command bits [10:8] are ignored if Error Acknowledge/Command is pending (bit 13).
*2 Error bits are cleared by writing "'000"' (or any valid command) to Command Register Bits [10:8].

7.3.10.4 EEPROM Address (0x0504:0x0507)

Bit	Description	ECAT	PDI	Reset Value
31:0	EEPROM Address 0: First word (= 16 bit) 1: Second word ... Actually used EEPROM Address bits: [9:0]: EEPROM size up to 16 kBit [17:0]: EEPROM size 32 kBit ...4 Mbit [32:0]: EEPROM Emulation	r/(w)	r/(w)	

Table 54: Register 0x0504:0x0507 (PROM Address)

Note r/(w): write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if EEPROM interface is busy (0x0502.15=1).

7.3.10.5 EEPROM Data (0x0508:0x050F)

Bit	Description	ECAT	PDI	Reset Value
15:0	EEPROM Write data (data to be written to EEPROM) or EEPROM Read data (data read from EEPROM,. lower bytes)	r/(w)	r/[w]	
63:16	EEPROM Read data (data read from EEPROM, higher bytes)	r/-	r/- r[w]	

Table 55: Register 0x0508:0x050F (PROM Data)

Note r/(w): write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if EEPROM interface is busy (0x0502.15=1).

Note r/[w]: write access for EEPROM emulation if read or reload command is pending.

7.3.11 MII Management Interface

Address	Length (Byte)	Description
		MII Management Interface
0x0510:0x0511	2	MII Management Control/Status
0x0512	1	PHY Address
0x0513	1	PHY Register Address
0x0514:0x0515	2	PHY Data
0x0516	1	MII Management ECAT Access State
0x0517	1	MII Management PDI Access State
0x0518:0x051B	4	PHY Port Status

Table 56: MII Management Interface Register Overview

7.3.11.1 MII Management Control/Status (0x0510:0x0511)

Bit	Description	ECAT	PDI	Reset Value
0	Write enable*: 0: Write disabled 1: Write enabled This bit is always 1 if PDI has MI control.	r/(w)	r/-	
1	Management Interface can be controlled by PDI (registers 0x0516:0x0517): 0: Only ECAT control 1: PDI control possible	r/-	r/-	
2	MI link detection (link configuration, link detection, registers 0x0518:0x051B): 0: Not available 1: MI link detection active	r/-	r/-	
7:3	PHY address of port 0	r/-	r/-	
9:8	Command register*: Write: Initiate command. Read: Currently executed command Commands: 00: No command/MI idle (clear error bits) 01: Read 10: Write Others: Reserved/invalid commands (do not issue)	r/(w)	r/(w)	
12:10	Reserved, write 0	r/-	r/-	

Bit	Description	ECAT	PDI	Reset Value
13	Read error: 0: No read error 1: Read error occurred (PHY or register not available) Cleared by writing to this register.	r/(w)	r/(w)	
14	Command error: 0: Last Command was successful 1: Invalid command or write command without Write Enable Cleared with a valid command or by writing "'00"' to Command register bits [9:8].	r/-	r/-	
15	Busy: 0: MI control state machine is idle 1: MI control state machine is active	r/-	r/-	

Table 57: Register 0x0510:0x0511 (MI Cntrl/State)

Note r/ (w): write access depends on assignment of MI (ECAT/PDI). Write access is generally blocked if Management interface is busy (0x0510.15=1).

* Write enable bit 0 is self-clearing at the SOF of the next frame (or at the end of the PDI access), Command bits [9:8] are self-clearing after the command is executed (Busy ends). Writing "'00"' to the command register will also clear the error bits [14:13]. The Command bits are cleared after the command is executed.

7.3.11.2 PHY Address (0x0512)

Bit	Description	ECAT	PDI	Reset Value
0:4	PHY Address	r/(w)	r/(w)	
6:5	Reserved, write 0	r/-	r/-	
7	Show configured PHY address of port 0-3 in register 0x0510.7:3. Select port x with bits [4:0] of this register (valid values are 0-3): 0: Show address of port 0 (offset) 1: Show individual address of port x	r/(w)	r/(w)	

Table 58: Register 0x0512 (PHY Address)

Note r/ (w): write access depends on assignment of MI (ECAT/PDI). Write access is generally blocked if Management interface is busy (0x0510.15=1).

7.3.11.3 PHY Register Address (0x0513)

Bit	Description	ECAT	PDI	Reset Value
4:0	Address of PHY Register that shall be read/written	r/(w)	r/(w)	
7:5	Reserved, write 0	r/(w)	r/(w)	

Table 59: Register 0x0513 (PHY Register Address)

Note r/ (w): write access depends on assignment of MI (ECAT/PDI). Write access is generally blocked if Management interface is busy (0x0510.15=1).

7.3.11.4 PHY Data (0x0514:0x0515)

Bit	Description	ECAT	PDI	Reset Value
15:0	PHY Read/Write Data	r/(w)	r/(w)	

Table 60: Register 0x0514:0x0515 (PHY Data)

Note r/ (w): write access depends on assignment of MI (ECAT/PDI). Access is generally blocked if Management interface is busy (0x0510.15=1).

7.3.11.5 MII Management ECAT Access State (0x0516)

Bit	Description	ECAT	PDI	Reset Value
31:0	Access to MII management: 0: ECAT enables PDI takeover of MII management control 1: ECAT claims exclusive access to MII management	r/(w)	r/-	
31:0	Reserved, write 0	r/-	r/-	

Table 61: Register 0x0516 (MI ECAT State)

Note r/ (w): write access is only possible if 0x0517.0=0.

7.3.11.6 MII Management PDI Access State (0x0517)

Bit	Description	ECAT	PDI	Reset Value
0	Access to MII management: 0: ECAT has access to MII management 1: PDI has access to MII management	r/-	r/(w)	
1	Force PDI Access State: 0: Do not change Bit 0x0517.0 1: Reset Bit 0x0517.0 to 0	r/w	r/-	
7:2	Reserved, write 0	r/-	r/-	

Table 62: Register 0x0517 (MI PDI State)

7.3.11.7 PHY Port Status (0x0518:0x051B)

Bit	Description	ECAT	PDI	Reset Value
0	Physical link status (PHY status register 1.2): 0: No physical link / 1: Physical link detected	r/-	r/-	
1	Link status (100 Mbit/s, Full Duplex, Autonegotiation): 0: No link / 1: Link detected	r/-	r/-	
2	Link status error: 0: No error 1: Link error, link inhibited	r/-	r/-	
3	Read error: 0: No read error occurred 1: A read error has occurred Cleared by writing any value to at least one of the PHY Status Port registers.	r/(w/clr)	r/(w/clr)	
4	Link partner error: 0: No error detected / 1: Link partner error	r/-	r/-	
5	PHY configuration updated: 0: No update 1: PHY configuration was updated Cleared by writing any value to at least one of the PHY Status Port registers.	r/(w/clr)	r/(w/clr)	
31:0	Reserved	r/-	r/-	

Table 63: Register 0x0518+y (PHY Port Status)

Note r/(w): write access depends on assignment of MI (ECAT/PDI).

7.3.12 FMMUs

Address	Length (Byte)	Description
0x0600:0x06FF	**16x16**	**FMMU[15:0]**
+0x0:0x3	4	Logical Start Address
+0x4:0x5	2	Length
+0x6	1	Logical Start bit
+0x7	1	Logical Stop bit
+0x8:0x9	2	Physical Start Address
+0xA	1	Physical Start bit
+0xB	1	Type
+0xC	1	Activate
+0xD:0xF	3	Reserved

Table 64: FMMU Register Overview

For the following registers use y as FMMU number.
See the device features on how many FMMUs are supported in a specific ESC device.

7.3.12.1 Logical Start Address (+0x0:0x3)

Bit	Description	ECAT	PDI	Reset Value
31:0	Logical start address within the EtherCAT Address Space.	r/w	r/-	

Table 65: Register 0x06y0:0x06y3 (Log Start Addr)

7.3.12.2 Length (+0x4:0x5)

Bit	Description	ECAT	PDI	Reset Value
15:0	Offset from the first logical FMMU Byte to the last FMMU Byte + 1 (e.g., if two bytes are used then this parameter shall contain 2)	r/w	r/-	

Table 66: Register 0x06y4:0x06y5 (FMMU Length)

7.3.12.3 Logical Start bit (+0x6)

Bit	Description	ECAT	PDI	Reset Value
2:0	Logical starting bit that shall be mapped (bits are counted from least significant bit (=0) to most significant bit(=7)	r/w	r/-	
7:3	Reserved, write 0	r/-	r/-	

Table 67: Register 0x06y6 (Log. Start Bit)

7.3.12.4 Logical Stop bit (+0x7)

Bit	Description	ECAT	PDI	Reset Value
2:0	Last logical bit that shall be mapped (bits are counted from least significant bit (=0) to most significant bit(=7)	r/w	r/-	
7:3	Reserved, write 0	r/-	r/-	

Table 68: Register 0x06y7 (Log. Stop Bit))

7.3.12.5 Physical Start Address (+0x8:0x9)

Bit	Description	ECAT	PDI	Reset Value
	Physical Start Address (mapped to logical Start address)	r/w	r/-	

Table 69: Register 0x06y8:0x06y9 (Phy. Start Addr

7.3.12.6 Physical Start bit (+0xA)

Bit	Description	ECAT	PDI	Reset Value
2:0	Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit (=0) to most significant bit(=7)	r/w	r/-	
7:3	Reserved, write 0	r/-	r/-	

Table 70: Register 0x06yA (Phy. Start Bit)

7.3.12.7 Type (+0xB)

Bit	Description	ECAT	PDI	Reset Value
0	0: Ignore mapping for read accesses 1: Use mapping for read accesses	r/w	r/-	
1	0: Ignore mapping for write accesses 1: Use mapping for write accesses	r/w	r/-	
7:2	Reserved, write 0	r/-	r/-	

Table 71: Register 0x06yB (FMMU Type)

7.3.12.8 Activate (+0xC)

Bit	Description	ECAT	PDI	Reset Value
0	0: FMMU deactivated 1: FMMU activated. FMMU checks logical addressed blocks to be mapped according to mapping configured	r/w	r/-	
7:1	Reserved, write 0	r/-	r/-	

Table 72: Register 0x06yC (FMMU Activate)

7.3.12.9 Reserved (+0xD:0xF)

Bit	Description	ECAT	PDI	Reset Value
23:0	Reserved, write 0	r/-	r/-	

Table 73: Register 0x06yD:0x06yF (Reserved)

7.3.13 SyncManagers

Address	Length (Byte)	Description
0x0800:0x087F	**16x8**	**SyncManager[15:0]**
+0x0:0x1	2	Physical Start Address
+0x2:0x3	2	Length
+0x4	1	Control Register
+0x5	1	Status Register
+0x6	1	Activate
+0x7	1	PDI Control

Table 74: SyncManager Register Overview

For the following registers use y as SM number.
See the device features on how many SMs are supported in a specific ESC device.

7.3.13.1 Physical Start Address (+0x0:0x1)

Bit	Description	ECAT	PDI	Reset Value
15:0	Specifies first byte that will be handled by SyncManager	r/(w)	r/-	

*Table 75: Register 0x0800+y*8:0x0801+y*8 (Phy. Start Addr)*

Note r/(w): Register can only be written if SyncManager is disabled (+0x6.0 = 0).

7.3.13.2 Length (+0x2:0x3)

Bit	Description	ECAT	PDI	Reset Value
15:0	Number of bytes assigned to SyncManager (shall be greater 1, otherwise SyncManager is not activated. If set to 1, only Watchdog Trigger is generated if configured)	r/(w)	r/-	

*Table 76: Register 0x0802+y*8:0x0803+y*8 (SM Length)*

Note r/(w): Register can only be written if SyncManager is disabled (+0x6.0 = 0).

7.3.13.3 Control Register (+0x4)

Bit	Description	ECAT	PDI	Reset Value
1:0	Operation Mode: 00: Buffered (3 buffer mode) 01: Reserved 10: Mailbox (Single buffer mode) 11: Reserved	r/(w)	r/-	
3:2	Direction: 00: Read: ECAT read access, PDI write access. 01: Write: ECAT write access, PDI read access. 10: Reserved 11: Reserved	r/(w)	r/-	
4	Interrupt in ECAT Event Request Register: 0: Disabled 1: Enabled	r/(w)	r/-	
5	Interrupt in PDI Event Request Register: 0: Disabled 1: Enabled	r/(w)	r/-	
6	Watchdog Trigger Enable: 0: Disabled 1: Enabled	r/w	r/-	
7	Reserved, write 0	r/-	r/-	

*Table 77: Register 0x0804+y*8 (SM Control)*

Note r/(w): Register can only be written if SyncManager is disabled (+0x6.0 = 0).

7.3.13.4 Status Register (+0x5)

Bit	Description	ECAT	PDI	Reset Value
0	Interrupt Write: 1: Interrupt after buffer was completely and successfully written 0: Interrupt cleared after first byte of buffer was read NOTE: This interrupt is signaled to the reading side if enabled in the SM Control register.	r/-	r/-	
1	Interrupt Read: 1: Interrupt after buffer was completely and successful read 0: Interrupt cleared after first byte of buffer was written NOTE: This interrupt is signaled to the writing side if enabled in the SM Control register.	r/-	r/-	

Bit	Description	ECAT	PDI	Reset Value
2	Reserved	r/-	r/-	
3	Mailbox mode: mailbox status: 0: Mailbox empty 1: Mailbox full Buffered mode: reserved	r/-	r/-	
5:4	Buffered mode: buffer status (last written buffer): 00: 1. buffer 01: 2. buffer 10: 3. buffer 11: (no buffer written) Mailbox mode: reserved	r/-	r/-	
6	Read buffer in use (opened)	r/-	r/-	
7	Write buffer in use (opened)	r/-	r/-	

*Table 78: Register 0x0805+y*8 (SM Status)*

7.3.13.5 Activate (+0x6)

Bit	Description	ECAT	PDI	Reset Value
0	SyncManager Enable/Disable: 0: Disable: Access to Memory without SyncManager control 1: Enable: SyncManager is active and controls Memory area set in configuration	r/w	r/(w ack)*	
1	Repeat Request: A toggle of Repeat Request means that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)	r/w	r/-	
5:2	Reserved, write 0	r/-	r/(w ack)*	
6	Latch Event ECAT: 0: No 1: Generate Latch event if EtherCAT master issues a buffer exchange	r/w	r/(w ack)*	
7	Latch Event PDI: 0: No 1: Generate Latch events if PDI issues a buffer exchange or if PDI accesses buffer start address	r/w	r/(w ack)*	

*Table 79: Register 0x0806+y*8 (SM Activate)*

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI in all SMs which have changed activation clears AL Event Request 0x0220.4. Writing to this register from PDI is

not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI in all SMs which have changed activation clears AL Event Request 0x0220.4. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.13.6 PDI Control (+0x7)

Bit	Description	ECAT	PDI	Reset Value
0	Deactivate SyncManager: Read: 0: Normal operation, SyncManager activated. 1: SyncManager deactivated and reset SyncManager locks access to Memory area. Write: 0: Activate SyncManager 1: Request SyncManager deactivation NOTE: Writing 1 is delayed until the end of a frame which is currently processed.	r/-	r/w	
1	Repeat Ack: If this is set to the same value as set by Repeat Request, the PDI acknowledges the execution of a previous set Repeat request.	r/-	r/w	
7:2	Reserved, write 0	r/-	r/-	

*Table 80: Register 0x0807+y*8 (SM PDI Control)*

7.3.14 Distributed Clocks Receive Times

Depending on the available width of the Distributed Clocks feature the time stamp registers are either 32 bit (4 bytes) or 64 bits (8 bytes) wide. Please check the feature summary of the respective TRINAMIC ESC device.

7.3.14.1 Receive Time Port 0 (0x0900:0x0903)

Bit	Description	ECAT	PDI	Reset Value
31:0	Write: A write access to register 0x0900 with BWR or FPWR latches the local time of the beginning of the receive frame (start first bit of preamble) at each port. Read: Local time of the beginning of the last receive frame containing a write access to this register.	r/w (special function)	r/-	

Table 81: Register 0x0900:0x0903 (Rcv Time P0)

Note The time stamps cannot be read in the same frame in which this register was written.

7.3.14.2 Receive Time Port 1 (0x0904:0x0907)

Bit	Description	ECAT	PDI	Reset Value
31:0	Local time of the beginning of a frame (start first bit of preamble) received at port 1 containing a BWR or FPWR to Register 0x0900.	r/-	r/-	

Table 82: Register 0x0904:0x0907 (Rcv Time P1)

7.3.15 Distributed Clocks Time Loop Control Unit

Time Loop Control unit is usually assigned to ECAT. Write access to Time Loop Control registers by PDI (and not ECAT) depends on explicit hardware configuration and on the used ESC type. Check the device features for availability.

7.3.15.1 System Time (0x0910:0x0917)

Bit	Description	ECAT	PDI	Reset Value
0:63	ECAT read access: Local copy of System Time when frame passed the reference clock (i.e., including System Time Delay). Time latched at beginning of the frame (Ethernet SOF delimiter)	r	-	
63:0	PDI read access: Local copy of the System Time. Time latched when reading first byte (0x0910)	-	r	
31:0	Write access: Written value will be compared with the local copy of the System time. The result is an input to the time control loop. NOTE: written value will be compared at the end of the frame with the latched (SOF) local copy of the System time if at least the first byte (0x0910) was written.	(w) (special function)	r/-	
31:0	Write access: Written value will be compared with Latch0 Time Positive Edge time. The result is an input to the time control loop. NOTE: written value will be compared at the end of the access with Latch0 Time Positive Edge (0x09B0:0x09B3) if at least the last byte (0x0913) was written.	-	(w) (special function)	

Table 83: Register 0x0910:0x0917 (System Time)

Note Write access to this register depends upon ESC configuration (typically ECAT, PDI only with explicit ESC configuration: System Time PDI controlled).

7.3.15.2 Receive Time ECAT Processing Unit (0x0918:0x091F)

Bit	Description	ECAT	PDI	Reset Value
63:0	Local time of the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to Register 0x0900 NOTE: E.g., if port 0 is open, this register reflects the Receive Time Port 0 as a 64 Bit value.	r/-	r/-	

Bit	Description	ECAT	PDI	Reset Value

Table 84: Register 0x0918:0x091F (Rcv Time EPU)

7.3.15.3 System Time Offset (0x0920:0x0927)

Bit	Description	ECAT	PDI	Reset Value
63:0	Difference between local time and System Time. Offset is added to the local time.	r/(w)	r/(w)	

Table 85: Register 0x0920:0x0927 (Sys Time Offset)

Note Write access to this register depends upon ESC configuration (typically ECAT, PDI only with explicit ESC configuration: System Time PDI controlled). Reset internal system time difference filter and speed counter filter by writing Speed Counter Start (0x0930:0x0931) after changing this value.

7.3.15.4 System Time Delay (0x0928:0x092B)

Bit	Description	ECAT	PDI	Reset Value
31:0	Delay between Reference Clock and the ESC	r/(w)	r/(w)	

Table 86: Register 0x0928:0x092B (Sys Time Delay)

Note Write access to this register depends upon ESC configuration (typically ECAT, PDI only with explicit ESC configuration: System Time PDI controlled). Reset internal system time difference filter and speed counter filter by writing Speed Counter Start (0x0930:0x0931) after changing this value.

7.3.15.5 System Time Difference (0x092C:0x092F)

Bit	Description	ECAT	PDI	Reset Value
30:0	Mean difference between local copy of System Time and received System Time values	r/-	r/-	
31	0: Local copy of System Time greater than or equal received System Time 1: Local copy of System Time smaller than received System Time	r/-	r/-	

Table 87: Register 0x092C:0x092F (Sys Time Diff)

7.3.15.6 Speed Counter Start (0x0930:0x0931)

Bit	Description	ECAT	PDI	Reset Value
14:0	Bandwidth for adjustment of local copy of System Time (larger values → smaller bandwidth and smoother adjustment) A write access resets System Time Difference (0x092C:0x092F) and Speed Counter Diff (0x0932:0x0933). Minimum value: 0x0080 to 0x3FFF	r/(w)	r/(w)	
15	Reserved, write 0	r/(w)	r/-	

Table 88: Register 0x0930:0x931 (Speed Cnt Start)

Note Write access to this register depends upon ESC configuration (typically ECAT, PDI only with explicit ESC configuration: System Time PDI controlled).

7.3.15.7 Speed Counter Diff (0x0932:0x0933)

Bit	Description	ECAT	PDI	Reset Value
15:0	Representation of the deviation between local clock period and reference clock's clock period (representation: two's complement) Range: ±(Speed Counter Start - 0x7F)	r/-	r/-	

Table 89: Register 0x0932:0x0933 (Speed Cnt Diff)

Note Calculate the clock deviation after System Time Difference has settled at a low value as follows:

$$Deviation = \frac{SpeedCntDiff}{5*(SpeedCntStart+SpeedCntDiff+2)*(SpeedCntStart-SpeedCntDiff+2)}$$

7.3.15.8 System Time Difference Filter Depth (0x0934)

Bit	Description	ECAT	PDI	Reset Value
3:0	Filter depth for averaging the received System Time deviation. A write access resets System Time Difference (0x092C:0x092F)	r/(w)	r/(w)	
7:4	Reserved, write 0	r/-	r/-	

Table 90: Register 0x0934 (Sys Time Diff Filter)

Note Write access to this register depends upon ESC configuration (typically ECAT, PDI only with explicit ESC configuration: System Time PDI controlled).

7.3.15.9 Speed Counter Filter Depth (0x0935)

Bit	Description	ECAT	PDI	Reset Value
3:0	Filter depth for averaging the clock period deviation. A write access resets the internal speed counter filter.	r/(w)	r/(w)	
7:4	Reserved, write 0	r/-	r/-	

Table 91: Register 0x0935 (Speed Cnt Filter Depth)

7.3.16 Distributed Clocks Cyclic Unit Control

7.3.16.1 Cyclic Unit Control (0x0980)

Bit	Description	ECAT	PDI	Reset Value
0	SYNC out unit control: 0: ECAT controlled 1: PDI controlled	r/w	r/-	
3:1	Reserved, write 0	r/-	r/-	
4	Latch In unit 0: 0: ECAT controlled 1: PDI controlled NOTE: Always 1 (PDI controlled) if System Time is PDI controlled. Latch interrupt is routed to ECAT/PDI depending on this setting	r/w	r/-	
5	Latch In unit 1: 0: ECAT controlled 1: PDI controlled NOTE: Latch interrupt is routed to ECAT/PDI depending on this setting	r/w	r/-	
7:6	Reserved, write 0	r/-	r/-	

Table 92: Register 0x0980 (Cyclic Unit Cntrl)

7.3.17 Distributed Clocks SYNC Out Unit

7.3.17.1 SYNC Out Activation (0x0981)

Bit	Description	ECAT	PDI	Reset Value
0	Sync Out Unit activation: 0: Deactivated 1: Activated	r/(w)	r/(w)	0
1	SYNC0 generation: 0: Deactivated 1: SYNC0 pulse is generated	r/(w)	r/(w)	0
2	SYNC1 generation: 0: Deactivated 1: SYNC1 pulse is generated	r/(w)	r/(w)	0
3	Auto-activation by writing Start Time Cyclic Operation (0x0990:0x0997): 0: Disabled 1: Auto-activation enabled. 0x0981.0 is set automatically after Start Time is written.	r/(w)	r/(w)	0
4	Extension of Start Time Cyclic Operation (0x0990:0x0993): 0: No extension 1: Extend 32 bit written Start Time to 64 bit	r/(w)	r/(w)	0
5	Start Time plausibility check: 0: Disabled. SyncSignal generation if Start Time is reached. 1: Immediate SyncSignal generation if Start Time is outside near future (see 0x0981.6)	r/(w)	r/(w)	0
6	Near future configuration (approx.): 0: $1/2$ DC width future (2^{31} ns or 2^{63} ns) 1: 2.1 sec. future (2^{31} ns)	r/(w)	r/(w)	0
7	SyncSignal debug pulse (Vasily bit): 0: Deactivated 1: Immediately generate one ping only on SYNC0-1 according to 0x0981.(2:1) for debugging This bit is self-clearing, always read 0.	r/(w)	r/(w)	0

Table 93: Register 0x0981 (SYNC Out Activation)

Note Write to this register depends upon setting of 0x0980.0.

7.3.17.2 Pulse Length of SYNC signals (0x0982:0x0983)

Bit	Description	ECAT	PDI	Reset Value
0	Pulse length of SyncSignals (in Units of 10ns) 0: Acknowledge mode: SyncSignal will be cleared by reading SYNC[1:0] Status register	r/-	r/-	0, later EEPROM ADR 0x0002

Table 94: Register 0x0982:0x0983 (SYNC Pulse Length)

7.3.17.3 Activation Status (0x0984)

Bit	Description	ECAT	PDI	Reset Value
0	SYNC0 activation state: 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending	r/-	r/-	0
1	SYNC1 activation state: 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending	r/-	r/-	0
2	Start Time Cyclic Operation (0x0990:0x0997) plausibility check result when Sync Out Unit was activated: 0: Start Time was within near future 1: Start Time was out of near future (0x0981.6)	r/-	r/-	0
7:3	Reserved	r/-	r/-	0

Table 95: Register 0x0984 (Activation Status)

7.3.17.4 SYNC0 Status (0x098E)

Bit	Description	ECAT	PDI	Reset Value
0	SYNC0 activation state: 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending	r/-	r/ (w ack)*	0
7:1	Reserved	r/-	r/ (w ack)*	0

Table 96: Register 0x098E (SYNC0 Status)

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220.2. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220.2. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.17.5 SYNC1 Status (0x098F)

Bit	Description	ECAT	PDI	Reset Value
0	SYNC1 activation state: 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending	r/-	r/ (w ack)*	0
7:1	Reserved	r/-	r/ (w ack)*	0

Table 97: Register 0x098F (SYNC1 Status)

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220.3. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220.3. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.17.6 Start Time Cyclic Operation / Next SYNC0 Pulse (0x0990:0x0997)

Bit	Description	ECAT	PDI	Reset Value
63:0	Write: Start time (System time) of cyclic operation in ns Read: System time of next SYNC0 pulse in ns	r/(w)	r/(w)	0

Table 98: Register 0x0990:0x0997 (Start Time Cyclic Operation)

Note Write to this register depends upon setting of 0x0980.0. Only writable if 0x0981.0=0. Auto-activation (0x0981.3=1): upper 32 bits are automatically extended if only lower 32 bits are written within one frame.

7.3.17.7 Next SYNC1 Pulse (0x0998:0x099F)

Bit	Description	ECAT	PDI	Reset Value
63:0	System time of next SYNC1 pulse in ns	r/-	r/-	0

Table 99: Register 0x0998:0x099F (Next SYNC1)

7.3.17.8 SYNC0 Cycle Time (0x09A0:0x09A3)

Bit	Description	ECAT	PDI	Reset Value
31:0	WTime between two consecutive SYNC0 pulses in ns. 0: Single shot mode, generate only one SYNC0 pulse.	r/(w)	r/(w)	0

Table 100: Register 0x09A0:0x09A3 (SYNC0 Cycle Time)

Note Write to this register depends upon setting of 0x0980.0.

7.3.17.9 SYNC1 Cycle Time (0x09A4:0x09A7)

Bit	Description	ECAT	PDI	Reset Value
31:0	Time between SYNC1 pulses and SYNC0 pulse in ns	r/(w)	r/(w)	0

Table 101: Register 0x09A4:0x09A7 (SYNC1 Cycle Time)

Note Write to this register depends upon setting of 0x0980.0.

7.3.18 Distributed Clocks LATCH In Unit

7.3.18.1 Latch0 Control (0x09A8)

Bit	Description	ECAT	PDI	Reset Value
0	Latch0 positive edge: 0: Continuous Latch active 1: Single event (only first event active)	r/(w)	r/(w)	0
1	Latch0 negative edge: 0: Continuous Latch active 1: Single event (only first event active)	r/(w)	r/(w)	0
7:2	Reserved, write 0	r/-	r/-	0

Table 102: Register 0x09A8 (Latch0 Control)

Note Write access depends upon setting of 0x0980.4.

7.3.18.2 Latch1 Control (0x09A9)

Bit	Description	ECAT	PDI	Reset Value
0	Latch1 positive edge: 0: Continuous Latch active 1: Single event (only first event active)	r/(w)	r/(w)	0
1	Latch01 negative edge: 0: Continuous Latch active 1: Single event (only first event active)	r/(w)	r/(w)	0
7:2	Reserved, write 0	r/-	r/-	0

Table 103: Register 0x09A9 (Latch1 Control)

Note Write access depends upon setting of 0x0980.5.

7.3.18.3 Latch0 Status (0x09AE)

Bit	Description	ECAT	PDI	Reset Value
0	Event Latch0 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch0 Time Positive Edge.	r/-	r/-	0
1	Event Latch0 negative edge. 0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch0 Time Negative Edge.	r/-	r/-	0
2	Latch0 pin state	r/-	r/-	0
7:3	Reserved	r/-	r/-	0

Table 104: Register 0x09AE (Latch0 Status)

7.3.18.4 Latch1 Status (0x09AF)

Bit	Description	ECAT	PDI	Reset Value
0	Event Latch1 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch1 Time Positive Edge.	r/-	r/-	0
1	Event Latch1 negative edge. 0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch1 Time Negative Edge.	r/-	r/-	0
2	Latch1 pin state	r/-	r/-	0
7:3	Reserved	r/-	r/-	0

Table 105: Register 0x09AF (Latch1 Status)

7.3.18.5 Latch0 Time Positive Edge (0x09B0:0x09B7)

Bit	Description	ECAT	PDI	Reset Value
63:0	Register captures System time at the positive edge of the Latch0 signal.	r(ack)/-	r/ (w ack)*	0

Table 106: Register 0x09B0:0x09B7 (Latch0 Time Pos Edge)

Note Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE.0 if 0x0980.4=0. Writing to this register from ECAT is not possible.

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980.4=1 clears Latch0 Status 0x09AE.0. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980.4=1 clears Latch0 Status 0x09AE.0. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.18.6 Latch0 Time Negative Edge (0x09B8:0x09BF)

Bit	Description	ECAT	PDI	Reset Value
63:0	Register captures System time at the negative edge of the Latch0 signal.	r(ack)/-	r/ (w ack)*	0

Table 107: Register 0x09B8:0x09BF (Latch0 Time Neg Edge)

Note Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE.1 if 0x0980.4=0. Writing to this register from ECAT is not possible.

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980.4=1 clears Latch0 Status 0x09AE.1. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980.4=1 clears Latch0 Status 0x09AE.1. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.18.7 Latch1 Time Positive Edge (0x09C0:0x09C7)

Bit	Description	ECAT	PDI	Reset Value
63:0	Register captures System time at the positive edge of the Latch1 signal.	r(ack)/-	r/ (w ack)*	0

Table 108: Register 0x09C0:0x09C7 (Latch1 Time Pos Edge)

Note Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch1 Status 0x09AF.0 if 0x0980.5=0. Writing to this register from ECAT is not possible.

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980.5=1 clears Latch1 Status 0x09AF.0. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980.5=1 clears Latch1 Status 0x09AF.0. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.18.8 Latch1 Time Negative Edge (0x09C8:0x09CF)

Bit	Description	ECAT	PDI	Reset Value
63:0	Register captures System time at the negative edge of the Latch1 signal.	r(ack)/-	r/ (w ack)*	0

Table 109: Register 0x09C8:0x09CF (Latch1 Time Neg Edge)

Note Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch1 Status 0x09AF.0 if 0x0980.5=0. Writing to this register from ECAT is not possible.

* PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980.5=1 clears Latch1 Status 0x09AF.1. Writing to this register from PDI is not possible.
PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980.5=1 clears Latch1 Status 0x09AF.1. Writing to this register from PDI is possible; write value is ignored (write 0).

7.3.19 Distributed Clocks SyncManager Event Times

7.3.19.1 EtherCAT Buffer Change Event Time (0x09F0:0x09F3)

Bit	Description	ECAT	PDI	Reset Value
31:0	Register captures local time of the beginning of the frame which causes at least one SM to assert an ECAT event	r/-	r/-	0

Table 110: Register 0x09F0:0x09F3 (ECAT Buffer Change Event Time)

Note Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

7.3.19.2 PDI Buffer Start Event Time (0x09F8:0x09FB)

Bit	Description	ECAT	PDI	Reset Value
31:0	Register captures local time when at least one SyncManager asserts an PDI buffer start event	r/-	r/-	0

Table 111: Register 0x09F8:0x09FB (PDI Buffer Start Event Time)

Note Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

7.3.19.3 PDI Buffer Change Event Time (0x09FC:0x09FF)

Bit	Description	ECAT	PDI	Reset Value
31:0	Register captures local time when at least one SyncManager asserts an PDI buffer start event	r/-	r/-	0

Table 112: Register 0x09FC:0x09FF (PDI Buffer Change Event Time)

Note Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

7.3.20 ESC Specific

7.3.20.1 Product ID (0x0E00:0x0E07)

Bit	Description	ECAT	PDI	Reset Value
63:0	Product ID	r/-	r/-	TMC8460: 0x0000000001008460 TMC8461: 0x0000000001108461 TMC8462: 0x0000000001108461 TMC8670: 0x0000000001008670

Table 113: Register 0x0E00:0x0E07 (Product ID)

7.3.20.2 Vendor ID (0x0E08:0x0E0F)

Bit	Description	ECAT	PDI	Reset Value
63:0	Vendor ID: [23:0] Company [31:24] Department NOTE: Test Vendor IDs [31:28]=0xE	r/-	r/-	TMC8460: 0x0000000100000286 TMC8461: 0x0000000100000286 TMC8462: 0x0000000100000286 TMC8670: 0x0000000100000286

Table 114: Register 0x0E08:0x0E0F (Vendor ID)

7.3.21 Process Data RAM

7.3.21.1 Process Data RAM (0x1000:0xFFFF)

The Process Data RAM starts at address 0x1000.
The size of the Process Data RAM depends on the device.

Bytes	Description	ECAT	PDI	Reset Value
- - -	Process Data RAM	(r/w)	(r/w)	Random/undefined

Table 115: Process Data RAM (0x1000:0xFFFF)

Note (r/w): Process Data RAM is only accessible if EEPROM was correctly loaded (register 0x0110.0 = 1).

Device	Process Data RAM Size	Upper RAM Address
TMC8460	16kBytes	0x4FFF
TMC8461	16kBytes	0x4FFF
TMC8462	16kBytes	0x4FFF
TMC8670	4kBytes	0x1FFF

Table 116: Process Data RAM Size

8 Electrical Ratings

8.1 Absolute Maximum Ratings

Note The maximum ratings may not be exceeded under any circumstances. Operating the circuit at or near more than one maximum rating at a time for extended periods shall be avoided by application design.

Parameter	Symbol	Min	Max	Unit
DC supply	VDD_3V3	-0.3	3.63	V
DC core supply voltage	VDD_1V2	-0.3	1.32	V
IO supply voltage	VDD_3V3	-0.3	3.63	V
PLL supply voltage	VDD_3V3	-0.3	3.63	V
Junction temperature	TJ	-55	125	°C
Storage temperature	$TSTG$	-65	150	°C

Table 117: Absolute Maximum Ratings for TMC8670-BI

8.2 Operational Ratings

Parameter	Symbol	Min	Typ	Max	Unit
DC supply	VDD_3V3	3.15	3.3	3.45	V
DC core supply voltage	VDD_1V2	1.14	1.2	1.26	V
IO supply voltage	VDD_3V3	3.15	3.3	3.45	V
PLL supply voltage	VDD_3V3	3.15	3.3	3.45	V
Operating Junction temperature	TJ	-40	25	100	°C

Table 118: Operational Ratings for TMC8670-BI

8.3 Digital I/Os

The following table contains information on the I/O characteristics. LVCMOS is a widely used switching standard and is defined by JEDEC (JESD 8-5). TMC8670supports LVCMOS standard LVCMOS33, which is a general standard for 3V3 applications.

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input voltage low level	VIL	VDD_3V3 = 3.3V	-0.3		0.8	V
Input voltage high level	VIH	VDD_3V3 = 3.3V	2.0		3.45	V
Weak pull-down	RPD	at VOL	9.9		14.5	kOhm
Weak pull-up	RPU	at VOH	9.98		14.9	kOhm
Input low current	IIL	V_{IN} = 0V	-		10	μA
Input high current	IIL	V_{IN} = VDD_3V3	-		10	μA
Output voltage low level	VOL	VDD_3V3 = 3.3V	-		0.4	V
Output voltage high level	VOH	VDD_3V3 = 3.3V	VDD_3V3-0.4		-	V
Output driver strength standard	IOUT_DRV			8		mA
Output capacitance	COUT			10		pF

Table 119: Digital I/Os DC Characteristics

8.4 Power Consumption

The values given here are typical values only. The real values depend on configuration, activity, and temperature.

Parameter	Symbol	Typical	Unit	%	Notes
Total power consumption	P_{TOTAL}	562	mW	100	$P_S + P_D$
Static power consumption	P_S	28	mW	5	
Dynamic power consumption	P_D	534	mW	95	

Table 120: TMC8670 power consumption

Parameter	Symbol	Power (mW)	Voltage (V)	Current (mA)	Notes
DC core supply voltage	VDD_1V2	470	1.2	392	
Supply voltage I/Os and PLL	VDD_3V3	92	3.3	28	

Table 121: TMC8670 power consumption by rail

8.5 Package Thermal Behavior

Dynamic and static power consumption cause the junction temperature of the TMC8670 to be higher than the ambient, case, or board temperature. The equations below show the relationships.

$Theta_{JA} = (TJ - TA)/P_{Total}$

$Theta_{JB} = (TJ - TB)/P_{Total}$

$Theta_{JC} = (TJ - TC)/P_{Total}$

Parameter	Symbol	Typ	Unit	Notes
Junction-to-ambient thermal resistance	$Theta_{JA}$	27.03	°C/W	at still air
		22.91	°C/W	at 1.0 m/s
		21.25	°C/W	at 2.5 m/s
Junction-to-board thermal resistance	$Theta_{JB}$	12.33	°C/W	
Junction-to-case thermal resistance	$Theta_{JC}$	1.54	°C/W	

Table 122: TMC8670 package thermal behavior

Symbols used:
TJ = Junction temperature
TA = Ambient temperature
TB = Board temperature measured 1.0mm away from the package
TC = Case temperature

$Theta_{JA}$ = Junction-to-ambient thermal resistance is determined under standard conditions specified by JEDEC (JESD-51), but it has little relevance in the actual performance of the product. It must be used with caution, but it is useful for comparing the thermal performance of one package with another. The maximum power dissipation allowed is calculated as follows:

Maximum power allowed = $(TJ_{MAX} - TA_{MAX})/Theta_{JA}$

$Theta_{JB}$ = Junction-to-board thermal resistance measures the ability of the package to dissipate heat from the surface of the chip to the PCB. As defined by the JEDEC (JESD-51) standard, the thermal resistance from the junction to the board uses an isothermal ring cold plate zone concept. The ring cold plate is simply a means to generate an isothermal boundary condition at the perimeter. The cold plate is mounted on a JEDEC standard board with a minimum distance of 5.0 mm away from the package edge.

$Theta_{JC}$ = Junction-to-case thermal resistance measures the ability of a device to dissipate heat from the surface of the chip to the top or bottom surface of the package. It is applicable to packages used with external heat sinks. Constant temperature is applied to the surface, which acts as a boundary condition. This only applies to situations where all or nearly all of the heat is dissipated through the surface in consideration.

9 Manufacturing Data

9.1 Package Dimensions

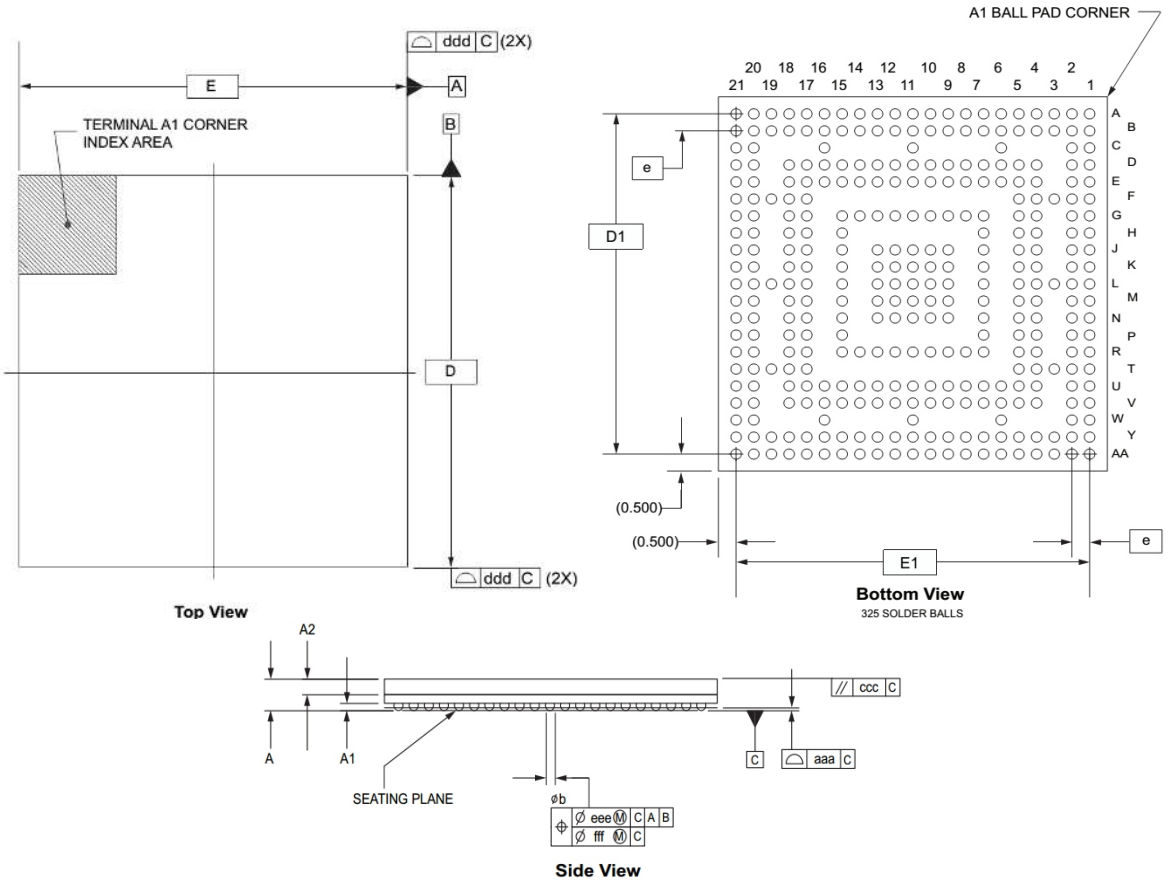


Figure 31: TMC8670-BI package outline drawing (dimensions are in millimeter)

Symbol	Min	Normal	Max
A			1.01
A1	0.15	0.21	
A2	0.40	0.45	0.50
aaa		0.08	
b	0.25	0.30	0.35
bbb		n.a.	
c	0.21	0.25	0.29
ccc		0.10	
D		11.00	
D1		10.00 BSC	

Symbol	Min	Normal	Max
E		11.00	
ddd		0.15	
E1		10.00 BSC	
e		0.5 TYP	
eee		0.15	

Table 123: Dimensions of TMC8670-BI (BSC = Basic Spacing between Centers)

9.2 Marking

The device marking is shown below.

Pin 1 location is highlighted with a dot.
yyww = date code.
LLLLL = lot number.

Figure 32: TMC8670-BI device marking

9.3 Board and Layout Considerations

- Example part libraries for different CAD tools are available as downloads on the respective IC product page on the TRINAMIC website at https://www.trinamic.com/products/integrated-circuits/.
- Package drawings, recommended land patterns, and soldering profiles for all TRINAMIC IC packages are available online at https://www.trinamic.com/support/help-center/ic-packages/
- TRINAMIC's evaluation boards are fully available as layout examples and recommendations and are free for download. Design data, Gerber data, and additional information is available at https://www.trinamic.com/support/eval-kits/.

10 Abbreviations

Abbreviation	Description
MCU	Microcontroller unit, application controller
AL	Application Layer
ASIC	Application Specific Integrated Circuit
CoE	CAN application protocol over EtherCAT
COMM	Common Anode or common cathode
CPU	Central Processing Unit
DC	Distributed Clocks
DPRAM	Dual Ported Random Access Memory
DS-Mod	Delta Sigma Modulator
ECAT	EtherCAT
ENI	EtherCAT Network Information (Information on Network configuration in XML format)
EOF	End of Frame
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information (device description/configuration data in XML format)
ESM	EtherCAT State Machine
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automation Technology
FMMU	Fieldbus Memory Management Unit
FoE	File Access over EtherCAT
GPIO	General Purpose I/O
GPI	General Purpose Input
GPO	General Purpose Output
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IRQ	Interrupt Request
LED	Light Emitting Diode
MI	(PHY) Management Interface
MII	Media Independent Interface
MISO	Master In - Slave Out
MOSI	Master Out - Slave In
PDI	Process Data Interface
PDO	Process Data Object

PDRAM	Process Data Random Access Memory
POF	Plastic/polymer Optical Fiber
RMS	Root Mean Square value
SII	Slave Information Interface
SM	SyncManager
SOF	Start of Frame
SPI	Serial Peripheral Interface
TMCL	TRINAMIC Motion Control Language
TTAP	TRINAMIC Technology Access Package
TTL	Transistor Transistor Logic
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
XML	Extended Mark-up Language

Table 124: Abbreviations used in this Manual

11 Figures Index

12 Tables Index

13 Revision History

13.1 IC Revision

Version	Date	Author	Description
V1.00	2017-OCT-30	LL/SL/ED	First IC version engineering samples.
V1.00	2018-MAY-31	LL	First IC version.

Table 125: IC Revision

13.2 Document Revision

Version	Date	Author	Description
V1.0	2017-OCT-31	SK, LL	Initial version
V1.0	2018-JUN-06	LL	pin names updated, ADC front-end part (LM339, LTC2351) updated
V1.0	2018-JUN-12	SK	typos fixed
V1.0	2019-MAR-20	LL	IC version time stamp updated
V1.0	2019-AUG-21	LL	UART interface description for TMCL-IDE updated

Table 126: Document Revision