



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

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

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



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





TMC8460-BI

Integrated EtherCAT Slave Controller with Enhanced Functionality

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

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

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

Focus

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

Features

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

Table of Contents

TABLE OF CONTENTS.....	2
LIST OF FIGURES.....	5
LIST OF TABLES.....	6
1 ABBREVIATIONS.....	10
2 PRINCIPLES OF OPERATION.....	11
2.1 KEY CONCEPTS.....	11
2.1.1 <i>General Information on EtherCAT</i>	11
2.1.2 <i>EtherCAT Slave Controller (ESC)</i>	11
2.1.3 <i>Trinamic Multi-Function and Control IO Block</i>	12
2.2 CONFIGURATION OPTIONS.....	13
2.3 CONTROL INTERFACES.....	13
2.3.1 <i>Ethernet Interface</i>	13
2.3.2 <i>Process Data Interface</i>	13
2.3.3 <i>Multi-Function and Control IO Block Interface</i>	13
2.3.4 <i>SPI Bus Sharing</i>	13
2.3.5 <i>Configuration Inputs</i>	14
2.3.6 <i>EEPROM Interface</i>	14
2.4 SOFTWARE VIEW.....	14
3 DEVICE USAGE AND HANDLING.....	20
3.1 SAMPLE BLOCK DIAGRAMS.....	20
3.1.1 <i>Typical EtherCAT Slave architecture</i>	20
3.1.2 <i>MFCIO block based Microcontroller Architecture</i>	20
3.1.3 <i>Device Emulation Example</i>	21
3.2 SAMPLES CIRCUITS.....	22
3.2.1 <i>IC supply</i>	22
3.2.2 <i>PDI interface</i>	22
3.2.3 <i>Miscellaneous signals</i>	22
3.2.4 <i>MII 1 (EtherCAT Input)</i>	23
3.2.5 <i>MII 2 (EtherCAT Output)</i>	23
3.2.6 <i>MFC I/Os</i>	24
3.3 PINOUT AND PIN DESCRIPTION.....	25
3.4 ETHERNET PHYs.....	33
3.4.1 <i>Ethernet PHY MII interface and MI interface</i>	33
3.4.2 <i>PHY Configuration Pins</i>	34
3.5 PDI SPI.....	34
3.5.1 <i>SPI protocol description</i>	35
3.5.2 <i>Timing example</i>	38
3.6 MFC CTRL SPI.....	39
3.6.1 <i>Timing example</i>	41
3.6.2 <i>Sharing Bus Lines with PDI SPI</i>	41
3.7 EEPROM INTERFACE.....	42
3.8 VENDOR ID, ESC TYPE, ESC REVISION AND BUILD HISTORY.....	43
3.9 ELECTRICAL CHARACTERISTICS.....	44
3.9.1 <i>Operating Conditions</i>	44
3.9.2 <i>External CLK Source</i>	44
3.9.3 <i>IO Characteristics</i>	44
3.9.4 <i>Power Consumption</i>	45
3.9.5 <i>Package Thermal Behavior</i>	45
3.10 MARKING AND ORDER CODES.....	46
3.11 PACKAGE DIMENSIONS.....	46
3.12 LAYOUT CONSIDERATIONS.....	48

3.12.1	<i>Example layout of the TMC8460-Eval</i>	49
3.13	SOLDERING PROFILE.....	50
4	ETHERCAT ADDRESS SPACE OVERVIEW.....	51
5	ETHERCAT REGISTER DESCRIPTION.....	56
5.1	TYPE (0x0000).....	56
5.2	REVISION (0x0001).....	56
5.3	BUILD (0x0002:0x0003).....	56
5.4	FMMUs SUPPORTED (0x0004).....	56
5.5	SYNCMANAGERS SUPPORTED (0x0005).....	56
5.6	RAM SIZE (0x0006).....	56
5.7	PORT DESCRIPTOR (0x0007).....	57
5.8	ESC FEATURES SUPPORTED (0x0008:0x0009).....	57
5.9	CONFIGURED STATION ADDRESS (0x0010:0x0011).....	58
5.10	CONFIGURED STATION ALIAS (0x0012:0x0013).....	58
5.11	WRITE REGISTER ENABLE (0x0020).....	59
5.12	WRITE REGISTER PROTECTION (0x0021).....	59
5.13	ESC WRITE ENABLE (0x0030).....	59
5.14	ESC WRITE PROTECTION (0x0031).....	60
5.15	ESC RESET ECAT (0x0040).....	60
5.16	ESC RESET PDI (0x0041).....	60
5.17	ESC DL CONTROL (0x0100:0x0103).....	61
5.18	PHYSICAL READ/WRITE OFFSET (0x0108:0x0109).....	62
5.19	ESC DL STATUS (0x0110:0x0111).....	63
5.20	AL CONTROL (0x0120:0x0121).....	65
5.21	AL STATUS (0x0130:0x0131).....	65
5.22	AL STATUS CODE (0x0134:0x0135).....	66
5.23	RUN LED OVERRIDE (0x0138).....	66
5.24	ERR LED OVERRIDE (0x0139).....	66
5.25	PDI CONTROL (0x0140).....	67
5.26	ESC CONFIGURATION (0x0141).....	68
5.27	PDI INFORMATION (0x014E:0x014F).....	69
5.28	PDI CONFIGURATION (0x0150:0x0153).....	69
5.28.1	<i>PDI SPI Slave Configuration</i>	69
5.28.2	<i>Sync/Latch[1:0] PDI Configuration</i>	70
5.29	ECAT EVENT MASK (0x0200:0x0201).....	71
5.30	PDI AL EVENT MASK (0x0204:0x0207).....	71
5.31	ECAT EVENT REQUEST (0x0210:0x0211).....	72
5.32	AL EVENT REQUEST (0x0220:0x0223).....	72
5.33	RX ERROR COUNTER (0x0300:0x0307).....	74
5.34	FORWARDED RX ERROR COUNTER (0x0308:0x030B).....	74
5.35	ECAT PROCESSING UNIT ERROR COUNTER (0x030C).....	74
5.36	PDI ERROR COUNTER (0x030D).....	74
5.37	SPI PDI ERROR CODE (0x030E).....	74
5.38	LOST LINK COUNTER (0x0310:0x0313).....	75
5.39	WATCHDOG DIVIDER (0x0400:0x0401).....	75
5.40	WATCHDOG TIME PDI (0x0410:0x0411).....	75
5.41	WATCHDOG TIME PROCESS DATA (0x0420:0x0421).....	75
5.42	WATCHDOG STATUS PROCESS DATA (0x0440:0x0441).....	76
5.43	WATCHDOG COUNTER PROCESS DATA (0x0442).....	76
5.44	WATCHDOG COUNTER PDI (0x0443).....	76
5.45	SII EEPROM INTERFACE (0x0500:0x050F).....	76
5.45.1	<i>EEPROM emulation with TMC8460</i>	80
5.46	MII MANAGEMENT INTERFACE (0x0510:0x0515).....	80
5.47	PARAMETER RAM (0x0580:0x05AB) FOR TMC8460 MFCIO BLOCK CONFIGURATION.....	85
5.48	FMMU (0x0600:0x06FF).....	86

5.49	SYNCMANAGER (0x0800:0x087F)	88
5.50	DISTRIBUTED CLOCKS (0x0900:0x09FF).....	92
5.50.1	Receive Times.....	92
5.50.2	Time Loop Control Unit.....	94
5.50.3	Cyclic Unit Control.....	98
5.50.4	SYNC Out Unit.....	99
5.50.5	Latch In unit.....	102
5.50.6	SyncManager Event Times.....	106
5.51	ESC SPECIFIC PRODUCT AND VENDOR ID.....	107
5.52	USER RAM (0x0F80:0x0FFF).....	107
5.53	PROCESS DATA RAM (0x1000:0xFFFF).....	108
5.53.1	MFCIO Block ECAT Write Data Memory Block (0x4000:0x405F).....	108
5.53.2	MFCIO Block ECAT Read Data Memory Block (0x4800:0x4823).....	110
6	MFCIO BLOCK REGISTER AND FUNCTIONAL DESCRIPTION	111
6.1	MFCIO BLOCK GENERAL INFORMATION	111
6.2	MFCIO BLOCK ADDRESS SPACE OVERVIEW.....	112
6.3	MFCIO BLOCK EEPROM PARAMETER MAP.....	113
6.4	MFCIO REGISTER CONFIGURATION	114
6.5	MFCIO BLOCK EXAMPLE CONFIGURATION AND EXAMPLE XML/ESI FILE.....	116
6.5.1	Example 1.....	116
6.5.2	Example 2.....	118
6.6	MFCIO INCREMENTAL ENCODER UNIT	121
6.6.1	Incremental Encoder Unit Signals.....	121
6.6.2	Incremental Encoder Unit Register Set.....	121
6.6.3	ENC_MODE.....	122
6.6.4	ENC_STATUS.....	122
6.6.5	ENC_X (W).....	122
6.6.6	ENC_X (R).....	122
6.6.7	ENC_CONST.....	123
6.6.8	ENC_LATCH.....	123
6.7	MFCIO SPI MASTER UNIT	125
6.7.1	SPI Master Unit Signals.....	125
6.7.2	SPI Master Unit Register Set.....	125
6.7.3	SPI_RX_DATA.....	126
6.7.4	SPI_TX_DATA.....	126
6.7.5	SPI_CONF.....	126
6.7.6	SPI_STATUS.....	127
6.7.7	SPI_LENGTH.....	127
6.7.8	SPI_TIME.....	127
6.7.9	SPI Examples.....	127
6.8	MFCIO STEP DIRECTION UNIT.....	130
2.1.1	Step Direction Unit Timing.....	130
2.1.2	Step Direction Unit Signals.....	131
2.1.3	Step Direction Unit and Register Set.....	131
6.8.1	Step Direction Accumulation Constant.....	132
6.8.2	Step Counter.....	133
6.8.3	Step Target.....	133
6.8.4	Step Length.....	133
6.8.5	Step-to-Direction Delay.....	133
6.8.6	Step Direction Unit Configuration.....	133
6.8.7	Interrupt Output Signal.....	133
6.9	MFCIO PWM UNIT	134
2.1.4	PWM Unit Signals.....	135
2.1.5	PWM Unit and Register Set.....	135
6.9.1	PWM_MAXCNT Configuration Register.....	138

6.9.2	<i>PWM_CHOPMODE Configuration Register</i>	138
6.9.3	<i>PWM_ALIGNMENT Configuration Register</i>	139
6.9.4	<i>POLARITIES Configuration Register</i>	139
6.9.5	<i>PWM Value Registers</i>	140
6.9.6	<i>PULSE_A Configuration Register</i>	140
6.9.7	<i>PULSE_B Configuration Register</i>	140
6.9.8	<i>PULSE_LENGTH Configuration Register</i>	140
6.9.9	<i>Asymmetric PWM Configuration Registers</i>	140
6.9.10	<i>Brake-Before-Make (BBM)</i>	140
6.9.11	<i>BBM_H Configuration Register</i>	140
6.9.12	<i>BBM_L Configuration Registers</i>	140
6.9.13	<i>Emergency Switch Input Off-State</i>	141
6.10	MFCIO GENERAL PURPOSE IO UNIT	142
6.10.1	<i>GPIO Registers</i>	142
6.10.2	<i>General purpose Inputs (GPI)</i>	142
6.10.3	<i>General purpose Outputs (GPO)</i>	142
6.10.4	<i>Emergency Switch Input State</i>	142
6.11	MFCIO WATCHDOG UNIT	143
6.11.1	<i>General Function</i>	143
6.11.2	<i>Watchdog Register Set</i>	143
6.12	MFCIO IRQ UNIT AND REGISTER SET	147
2.1.6	<i>IRQ_CFG Register</i>	147
2.1.7	<i>IRQ_FLAGS Register</i>	147
6.13	MFCIO EMERGENCY SWITCH INPUT	148
6.13.1	<i>Activation & Usage</i>	148
6.13.2	<i>Re-Activation</i>	148
6.14	AUXILIARY CLOCK OUTPUT	148
6.15	AL-STATE OVERRIDE CONFIGURATION	149
7	ESD SENSITIVE DEVICE	150
8	DISCLAIMER	150
9	REVISION HISTORY	151

List of Figures

FIGURE 1 - TMCL-IDE WITH DIRECT REGISTER ACCESS TO THE TMC8460-BI ON ITS EVALUATION BOARD.....	15
FIGURE 2 - WIZARD START SCREEN.....	16
FIGURE 3 - WIZARD DEVICE SELECTION AND FEATURE SELECTION.....	17
FIGURE 4 - WIZARD REGISTER SELECTION AND CONFIGURATION VIEW.....	18
FIGURE 5 - WIZARD OUTPUT VIEW WITH EEPROM CONFIGURATION STRING AND FIRMWARE C-CODE SNIPPETS.....	19
FIGURE 6 - APPLICATION DIAGRAM USING ONLY THE LOCAL APPLICATION CONTROLLER TO INTERFACE THE APPLICATION20	
FIGURE 7 - APPLICATION DIAGRAM USING THE MFCIO BLOCK FEATURES TO REDUCE SOFTWARE OVERHEAD AND PROVIDE REAL-TIME HARDWARE SUPPORT TO THE MCU. OTHER APPLICATION PARTS MAY STILL BE CONNECTED TO THE MCU.	21
FIGURE 8 - APPLICATION DIAGRAM WITHOUT MCU. THE TMC8460 IS USED IN DEVICE EMULATION MODE. SPI SLAVE CHIPS AND OTHER APPLICATION PERIPHERALS CAN BE CONNECTED TO THE MFCIO BLOCK. THE ETHERCAT MASTER CAN DIRECTLY CONTROL ALL THE APPLICATION FUNCTIONS.	21
FIGURE 9 : MII INTERFACE SIGNALS.....	33
FIGURE 10 - PDI SPI INTERFACE SIGNALS.....	35
FIGURE 11 - 2 BYTE ADDRESSING MODE.....	36
FIGURE 12 - 3 BYTE ADDRESSING MODE.....	36
FIGURE 13 - PDI SPI TIMING EXAMPLE.....	38
FIGURE 14 - MFC CTRL SPI INTERFACE SIGNALS.....	39
FIGURE 15 - 2-BYTE MFC REGISTER ACCESS.....	40
FIGURE 16 - 3-BYTE MFC REGISTER ACCESS.....	40
FIGURE 17 - MFC CONTROL SPI TIMING EXAMPLE.....	41

FIGURE 18 - SHARED SPI BUS CONFIGURATION.....	42
FIGURE 19 - EEPROM INTERFACE SIGNALS.....	42
FIGURE 20 - RECOMMENDED LAND PATTERN MEASUREMENTS.....	48
FIGURE 21 - TOP LAYER (1).....	49
FIGURE 22 - INNER LAYER (2).....	49
FIGURE 23 - INNER LAYER (3).....	49
FIGURE 24 - INNER LAYER (4).....	49
FIGURE 25 - INNER LAYER (5).....	49
FIGURE 26 - BOTTOM LAYER (6).....	49
FIGURE 27 - SOLDERING PROFILE.....	50
FIGURE 28 - MFCIO BLOCK INTERFACES TO ESC PDRAM.....	111
FIGURE 29 - CONNECTIONS TO TMC8460 IN EXAMPLE 1.....	116
FIGURE 30 - CONNECTIONS TO TMC8460 IN EXAMPLE 2.....	118
FIGURE 31 - BLOCK STRUCTURE OF THE INCREMENTAL ENCODER UNIT.....	121
FIGURE 32 - BLOCK STRUCTURE OF SPI MASTER UNIT.....	125
FIGURE 33 - STEP DIRECTION UNIT BLOCK DIAGRAM.....	130
FIGURE 34 - STEP-DIRECTION TIMING.....	130
FIGURE 35 - PWM BLOCK DIAGRAM.....	134
FIGURE 36 - PWM TIMING (CENTERED PWM).....	136
FIGURE 37 - PWM TIMING (LEFT ALIGNED PWM).....	137
FIGURE 38 - PWM TIMING (RIGHT ALIGNED PWM).....	137
FIGURE 39 - CHOPPER MODES (OFF, LOW SIDE ON, HIGH SIDE ON, LOW SIDE CHOPPER, HIGH SIDE CHOPPER, COMPLEMENTARY LOW SIDE AND HIGH SIDE CHOPPER).....	139
FIGURE 40 - CENTERED PWM WITH PWM#2 SHIFTED FROM CENTER (EXAMPLE).....	139
FIGURE 41 - BRAKE BEFORE MAKE (BBM) TIMING (INDIVIDUAL PROGRAMMABLE FOR LOW SIDE AND HIGH SIDE).....	140
FIGURE 6.42 - STRUCTURE OF THE WATCHDOG UNIT.....	145

List of Tables

TABLE 1 : MII INTERFACE SIGNAL DESCRIPTION AND CONNECTION.....	33
TABLE 2 : PDI SPI INTERFACE SIGNAL DESCRIPTION AND CONNECTION.....	35
TABLE 3 : PDI-SPI COMMANDS.....	35
TABLE 4 : MFC CTRL SPI INTERFACE SIGNAL DESCRIPTION AND CONNECTION.....	39
TABLE 5 : ABSOLUTE MAXIMUM RATINGS.....	44
TABLE 6 : RECOMMENDED OPERATING CONDITIONS.....	44
TABLE 7 : TMC8460 POWER CONSUMPTION.....	45
TABLE 8 : POWER CONSUMPTION BY RAIL.....	45
TABLE 9 : TMC8460 PACKAGE THERMAL BEHAVIOR.....	45
TABLE 10 : SOLDERING PROFILE PARAMETERS.....	50
TABLE 11 : TMC8460 ADDRESS SPACE.....	51
TABLE 12: REGISTER TYPE (0x0000).....	56
TABLE 13: REGISTER REVISION (0x0001).....	56
TABLE 14: REGISTER BUILD (0x0002:0x0003).....	56
TABLE 15: REGISTER FMMUS SUPPORTED (0x0004).....	56
TABLE 16: REGISTER SYNCMANAGERS SUPPORTED (0x0005).....	56
TABLE 17: REGISTER RAM SIZE (0x0006).....	56
TABLE 18: REGISTER PORT DESCRIPTOR (0x0007).....	57
TABLE 19: REGISTER ESC FEATURES SUPPORTED (0x0008:0x0009).....	57
TABLE 20: REGISTER CONFIGURED STATION ADDRESS (0x0010:0x0011).....	58
TABLE 21: REGISTER CONFIGURED STATION ALIAS (0x0012:0x0013).....	58
TABLE 22: REGISTER WRITE REGISTER ENABLE (0x0020).....	59
TABLE 23: REGISTER WRITE REGISTER PROTECTION (0x0021).....	59
TABLE 24: REGISTER ESC WRITE ENABLE (0x0030).....	59
TABLE 25: REGISTER ESC WRITE PROTECTION (0x0031).....	60
TABLE 26: REGISTER ESC RESET ECAT (0x0040).....	60

TABLE 27: REGISTER ESC RESET PDI (0x0041)	60
TABLE 28: REGISTER ESC DL CONTROL (0x0100:0x0103)	61
TABLE 29: REGISTER PHYSICAL READ/WRITE OFFSET (0x0108:0x0109)	62
TABLE 30: REGISTER ESC DL STATUS (0x0110:0x0111)	63
TABLE 31: DECODING PORT STATE IN ESC DL STATUS REGISTER 0x0111 (TYPICAL MODES ONLY)	64
TABLE 32: REGISTER AL CONTROL (0x0120:0x0121)	65
TABLE 33: REGISTER AL STATUS (0x0130:0x0131)	65
TABLE 34: REGISTER AL STATUS CODE (0x0134:0x0135)	66
TABLE 35: REGISTER RUN LED OVERRIDE (0x0138)	66
TABLE 36: REGISTER ERR LED OVERRIDE (0x0139)	67
TABLE 37: REGISTER PDI CONTROL (0x0140)	67
TABLE 38: REGISTER ESC CONFIGURATION (0x0141)	68
TABLE 39: REGISTER PDI INFORMATION (0x014E:0x014F)	69
TABLE 40: REGISTER PDI SPI SLAVE CONFIGURATION (0x0150)	69
TABLE 41: REGISTER PDI SPI SLAVE EXTENDED CONFIGURATION (0x0152:0x0153)	70
TABLE 42: REGISTER SYNC/LATCH[1:0] PDI CONFIGURATION (0x0151)	70
TABLE 43: REGISTER ECAT EVENT MASK (0x0200:0x0201)	71
TABLE 44: REGISTER PDI AL EVENT MASK (0x0204:0x0207)	71
TABLE 45: REGISTER ECAT EVENT REQUEST (0x0210:0x0211)	72
TABLE 46: REGISTER AL EVENT REQUEST (0x0220:0x0223)	72
TABLE 47: REGISTER RX ERROR COUNTER PORT Y (0x0300+Y*2:0x0301+Y*2)	74
TABLE 48: REGISTER FORWARDED RX ERROR COUNTER PORT Y (0x0308+Y)	74
TABLE 49: REGISTER ECAT PROCESSING UNIT ERROR COUNTER (0x030C)	74
TABLE 50: REGISTER PDI ERROR COUNTER (0x030D)	74
TABLE 51: REGISTER SPI PDI ERROR CODE (0x030E)	74
TABLE 52: REGISTER LOST LINK COUNTER PORT Y (0x0310+Y)	75
TABLE 53: REGISTER WATCHDOG DIVIDER (0x0400:0x0401)	75
TABLE 54: REGISTER WATCHDOG TIME PDI (0x0410:0x0411)	75
TABLE 55: REGISTER WATCHDOG TIME PROCESS DATA (0x0420:0x0421)	75
TABLE 56: REGISTER WATCHDOG STATUS PROCESS DATA (0x0440:0x0441)	76
TABLE 57: REGISTER WATCHDOG COUNTER PROCESS DATA (0x0442)	76
TABLE 58: REGISTER WATCHDOG COUNTER PDI (0x0443)	76
TABLE 59: SII EEPROM INTERFACE REGISTER OVERVIEW	76
TABLE 60: REGISTER EEPROM CONFIGURATION (0x0500)	77
TABLE 61: REGISTER EEPROM PDI ACCESS STATE (0x0501)	77
TABLE 62: REGISTER EEPROM CONTROL/STATUS (0x0502:0x0503)	78
TABLE 63: REGISTER EEPROM ADDRESS (0x0504:0x0507)	79
TABLE 64: REGISTER EEPROM DATA (0x0508:0x050F [0x0508:0x050B])	79
TABLE 65: REGISTER EEPROM DATA FOR EEPROM EMULATION RELOAD (0x0508:0x050F)	80
TABLE 66: MII MANAGEMENT INTERFACE REGISTER OVERVIEW	80
TABLE 67: REGISTER MII MANAGEMENT CONTROL/STATUS (0x0510:0x0511)	82
TABLE 68: REGISTER PHY ADDRESS (0x0512)	83
TABLE 69: REGISTER PHY REGISTER ADDRESS (0x0513)	83
TABLE 70: REGISTER PHY DATA (0x0514:0x0515)	83
TABLE 71: REGISTER MII MANAGEMENT ECAT ACCESS STATE (0x0516)	83
TABLE 72: REGISTER MII MANAGEMENT PDI ACCESS STATE (0x0517)	84
TABLE 73: REGISTER PHY PORT Y (PORT NUMBER Y=0 TO 3) STATUS (0x0518+Y)	84
TABLE 74: MFCIO REGISTER CONFIGURATION (0x0580+Y)	85
TABLE 75: FMMU REGISTER OVERVIEW	86
TABLE 76: REGISTER LOGICAL START ADDRESS FMMU Y (0x06Y0:0x06Y3)	86
TABLE 77: REGISTER LENGTH FMMU Y (0x06Y4:0x06Y5)	86
TABLE 78: REGISTER START BIT FMMU Y IN LOGICAL ADDRESS SPACE (0x06Y6)	86
TABLE 79: REGISTER STOP BIT FMMU Y IN LOGICAL ADDRESS SPACE (0x06Y7)	86
TABLE 80: REGISTER PHYSICAL START ADDRESS FMMU Y (0x06Y8-0x06Y9)	87
TABLE 81: REGISTER PHYSICAL START BIT FMMU Y (0x06YA)	87
TABLE 82: REGISTER TYPE FMMU Y (0x06YB)	87

TABLE 83: REGISTER ACTIVATE FMMU Y (0x06YC).....	87
TABLE 84: REGISTER RESERVED FMMU Y (0x06YD:0x06YF)	87
TABLE 85: SYNCMANAGER REGISTER OVERVIEW.....	88
TABLE 86: REGISTER PHYSICAL START ADDRESS SYNCMANAGER Y (0x0800+Y*8:0x0801+Y*8).....	88
TABLE 87: REGISTER LENGTH SYNCMANAGER Y (0x0802+Y*8:0x0803+Y*8).....	88
TABLE 88: REGISTER CONTROL REGISTER SYNCMANAGER Y (0x0804+Y*8).....	89
TABLE 89: REGISTER STATUS REGISTER SYNCMANAGER Y (0x0805+Y*8).....	90
TABLE 90: REGISTER ACTIVATE SYNCMANAGER Y (0x0806+Y*8)	91
TABLE 91: REGISTER PDI CONTROL SYNCMANAGER Y (0x0807+Y*8).....	92
TABLE 92: REGISTER RECEIVE TIME PORT 0 (0x0900:0x0903).....	92
TABLE 93: REGISTER RECEIVE TIME PORT 1 (0x0904:0x0907).....	92
TABLE 94: REGISTER RECEIVE TIME ECAT PROCESSING UNIT (0x0918:0x091F).....	92
TABLE 95: REGISTER SYSTEM TIME (0x0910:0x0913 [0x0910:0x0917]).....	94
TABLE 96: REGISTER SYSTEM TIME OFFSET (0x0920:0x0923 [0x0920:0x0927]).....	94
TABLE 97: REGISTER SYSTEM TIME DELAY (0x0928:0x092B).....	95
TABLE 98: REGISTER SYSTEM TIME DIFFERENCE (0x092C:0x092F)	95
TABLE 99: REGISTER SPEED COUNTER START (0x0930:0x0931)	95
TABLE 100: REGISTER SPEED COUNTER DIFF (0x0932:0x0933)	95
TABLE 101: REGISTER SYSTEM TIME DIFFERENCE FILTER DEPTH (0x0934)	96
TABLE 102: REGISTER SPEED COUNTER FILTER DEPTH (0x0935)	96
TABLE 103: REGISTER RECEIVE TIME LATCH MODE (0x0936)	97
TABLE 104: REGISTER CYCLIC UNIT CONTROL (0x0980).....	98
TABLE 105: REGISTER ACTIVATION REGISTER (0x0981).....	99
TABLE 106: REGISTER PULSE LENGTH OF SYNC SIGNALS (0x0982:0x0983).....	99
TABLE 107: REGISTER ACTIVATION STATUS (0x0984)	99
TABLE 108: REGISTER SYNC0 STATUS (0x098E)	100
TABLE 109: REGISTER SYNC1 STATUS (0x098F)	100
TABLE 110: REGISTER START TIME CYCLIC OPERATION (0x0990:0x0993 [0x0990:0x0997]).....	100
TABLE 111: REGISTER NEXT SYNC1 PULSE (0x0998:0x099B [0x0998:0x099F]).....	101
TABLE 112: REGISTER SYNC0 CYCLE TIME (0x09A0:0x09A3).....	101
TABLE 113: REGISTER SYNC1 CYCLE TIME (0x09A4:0x09A7).....	101
TABLE 114: REGISTER LATCH0 CONTROL (0x09A8)	102
TABLE 115: REGISTER LATCH1 CONTROL (0x09A9)	102
TABLE 116: REGISTER LATCH0 STATUS (0x09AE).....	103
TABLE 117: REGISTER LATCH1 STATUS (0x09AF).....	103
TABLE 118: REGISTER LATCH0 TIME POSITIVE EDGE (0x09B0:0x09B3 [0x09B0:0x09B7]).....	104
TABLE 119: REGISTER LATCH0 TIME NEGATIVE EDGE (0x09B8:0x09BB [0x09B8:0x09BF])	104
TABLE 120: REGISTER LATCH1 TIME POSITIVE EDGE (0x09C0:0x09C3 [0x09C0:0x09C7])	105
TABLE 121: REGISTER LATCH1 TIME NEGATIVE EDGE (0x09C8:0x09CB [0x09C8:0x09CF])	105
TABLE 122: REGISTER ETHERCAT BUFFER CHANGE EVENT TIME (0x09F0:0x09F3).....	106
TABLE 123: REGISTER PDI BUFFER START EVENT TIME (0x09F8:0x09FB)	106
TABLE 124: REGISTER PDI BUFFER CHANGE EVENT TIME (0x09FC:0x09FF).....	106
TABLE 125: REGISTER PRODUCT ID (0xE00:0xE07)	107
TABLE 126: REGISTER VENDOR ID (0xE08:0xE0F)	107
TABLE 127: USER RAM (0xF80:0xFFFF)	107
TABLE 128: PROCESS DATA RAM (0x1000:0x4FFF).....	108
TABLE 129: MFCIO BLOCK ECAT WRITE DATA MEMORY BLOCK (0x4000:0x405F).....	108
TABLE 130: PADDING BYTES	109
TABLE 131: MFCIO BLOCK ECAT READ DATA MEMORY BLOCK (0x4800:0x4823)	110
TABLE 132: PADDING BYTES	110
TABLE 133 : MFCIO BLOCK REGISTER OVERVIEW	112
TABLE 134 : EEPROM PARAMETER MAP & ESC RAM ADDRESS MAPPING FOR TMC8460-BI.....	113
TABLE 135 : MFCIO REGISTER CONFIGURATION BYTE	114
TABLE 136 : MFCIO REGISTER SHADOW TRIGGER SOURCE CONFIGURATION.....	114
TABLE 137 : MFCIO INCREMENTAL ENCODER UNIT SIGNALS.....	121
TABLE 138 : MFCIO INCREMENTAL ENCODER UNIT REGISTER SET.....	121

TABLE 139 : LIST OF TYPICAL ENCODER CONSTANTS.....	123
TABLE 140 : MFCIO MASTER UNIT SIGNALS.....	125
TABLE 141 : MFCIO SPI MASTER UNIT REGISTER SET.....	125
TABLE 142 : MFCIO SPI MASTER UNIT SPI MODE CONFIGURATION.....	126
TABLE 143 : MFCIO S/D UNIT SIGNALS	131
TABLE 144 : MFCIO S/D UNIT REGISTER SET	131
TABLE 145 : MFCIO S/D UNIT REAL WORLD UNIT RELATIONS.....	132
TABLE 146 : MFCIO PWM UNIT SIGNALS.....	135
TABLE 147 : MFCIO PWM UNIT REGISTER SET	135
TABLE 148 : MFCIO PWM BLOCK REAL WORLD UNIT RELATIONS	138
TABLE 149 : PWM CHOPPER MODE SELECTION	138
TABLE 150 : MFCIO GPIO UNIT REGISTERS.....	142
TABLE 151 : MFCIO WATCHDOG UNIT REGISTERS	143
TABLE 152 : WATCHDOG UNIT OUTPUT PORT CONFIGURATION ASSIGNMENT	145
TABLE 153 : WATCHDOG UNIT INPUT PORT CONFIGURATION ASSIGNMENT	146
TABLE 154 : MFCIO IRQ UNIT REGISTERS.....	147
TABLE 155 : TMC8460 AUXILIARY CLOCK PINS.....	148
TABLE 156 : AL-STATE OVERRIDE REGISTER.....	149
TABLE 157: DOCUMENTATION REVISIONS.....	151

1 Abbreviations

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

2 Principles of Operation

2.1 Key Concepts

2.1.1 General Information on EtherCAT

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

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

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

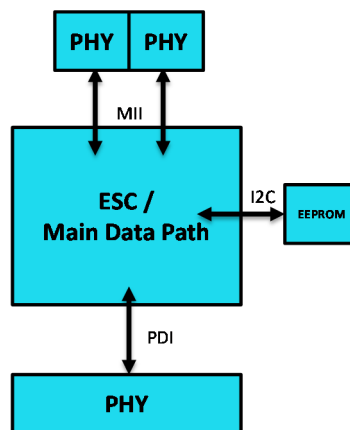
The ETG takes care for standardization activities and conformance testing. EtherCAT is integrated into the following major standards: IEC 61158 (protocols and services), IEC 61784-2 (communication profiles for devices), IEC 61800-7 (drive profiles and communication), SEMI standard E54.20 (since 2007).

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

2.1.2 EtherCAT Slave Controller (ESC)

The TMC8460 is a standard-conform dedicated EtherCAT Slave Controller providing EtherCAT MAC layer functionality to EtherCAT slaves.

It connects via standard of-the-shelf Ethernet PHYs to the physical medium and provides a digital control interface to a local application controller while also providing the option for standalone operation.



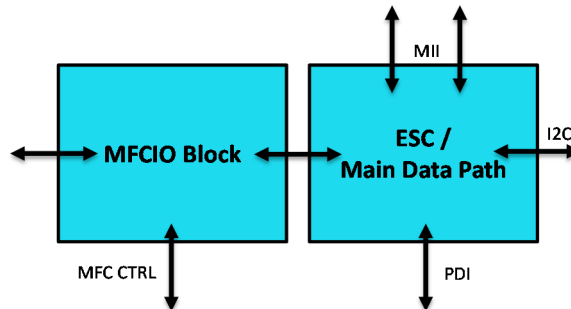
MAJOR ETHERCAT FEATURES

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

2.1.3 Trinamic Multi-Function and Control IO Block

Besides the proven EtherCAT functionality and the main EtherCAT data path, TMC8460 comes with a dedicated hardware block providing a configurable set of complex real-time IO functions to smart embedded systems. This IO functionality is called Multi-Function Control and IO block – MFCIO. Its special focus is on motor and motion control applications and systems while it is not limited to this application area.

The MFCIO block combines various functional sub-blocks that are helpful in an embedded design to reduce complexity, simplify the bill of materials (BOM), and to provide hardware acceleration to compute intensive tasks or time critical tasks. These functions can be used from the local application controller using a dedicated SPI interface or can directly be mapped into the Process Data RAM for direct access by the EtherCAT master.



GENERAL PURPOSE IOS

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

INCREMENTAL ENCODER UNIT

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

STEP & DIRECTION UNIT

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

3-CH PWM

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

SPI MASTER INTERFACE

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

IRQ / EVENT OUTPUT

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

WATCHDOG

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

EMERGENCY SWITCH INPUT

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

2.2 Configuration Options

The TMC8460 must be configured after power-up for proper operation. The EtherCAT part is automatically configured using configuration data from the connected I2C EEPROM.

The MFCIO block can also be configured using EEPROM configuration data. The EEPROM must therefore contain additional configuration data with category '1', which is automatically copied to ESC configuration RAM at addresses 0x0580:0x05FF.

Another way to configure the MFCIO block is to directly write the configuration bits to this RAM area using the ECAT or the PDI interface.

The MFCIO block can be used directly by a local application controller independent of the EtherCAT data path. Therefore, no upfront configuration is required since the MFCIO blocks comes with a dedicated SPI interface allowing complete access to its functions and local register set.

2.3 Control Interfaces

2.3.1 Ethernet Interface

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

2.3.2 Process Data Interface

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

2.3.3 Multi-Function and Control IO Block Interface

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

2.3.4 SPI Bus Sharing

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

2.3.5 Configuration Inputs

External package pins allow for selection of configuration options that typically do not change during operation by directly connecting them to 3V3 or ground. These package pins can also be controlled by GPIOs of the local application controller.

These options are for example external EEPROM's size, PHY addressing and MII TX clock shift configuration, polarities of the PHYs' link indicator, device emulation mode and enabling of the 16MHz clock output.

2.3.6 EEPROM Interface

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

2.4 Software View

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

As seen from the local application controller, the TMC8460-BI is a peripheral SPI slave device with a number of control and status registers that are accessible using the PDI SPI interface for the main EtherCAT data path or the MFC CTRL SPI interface for the MFCIO block.

For proper EtherCAT functionality, the local application controller runs the EtherCAT slave stack to process master requests regarding state changes in the state machine for example using the PDI SPI interface. In device emulation mode the PDI SPI interface is not used since master requests are handled inside the ESC.

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

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

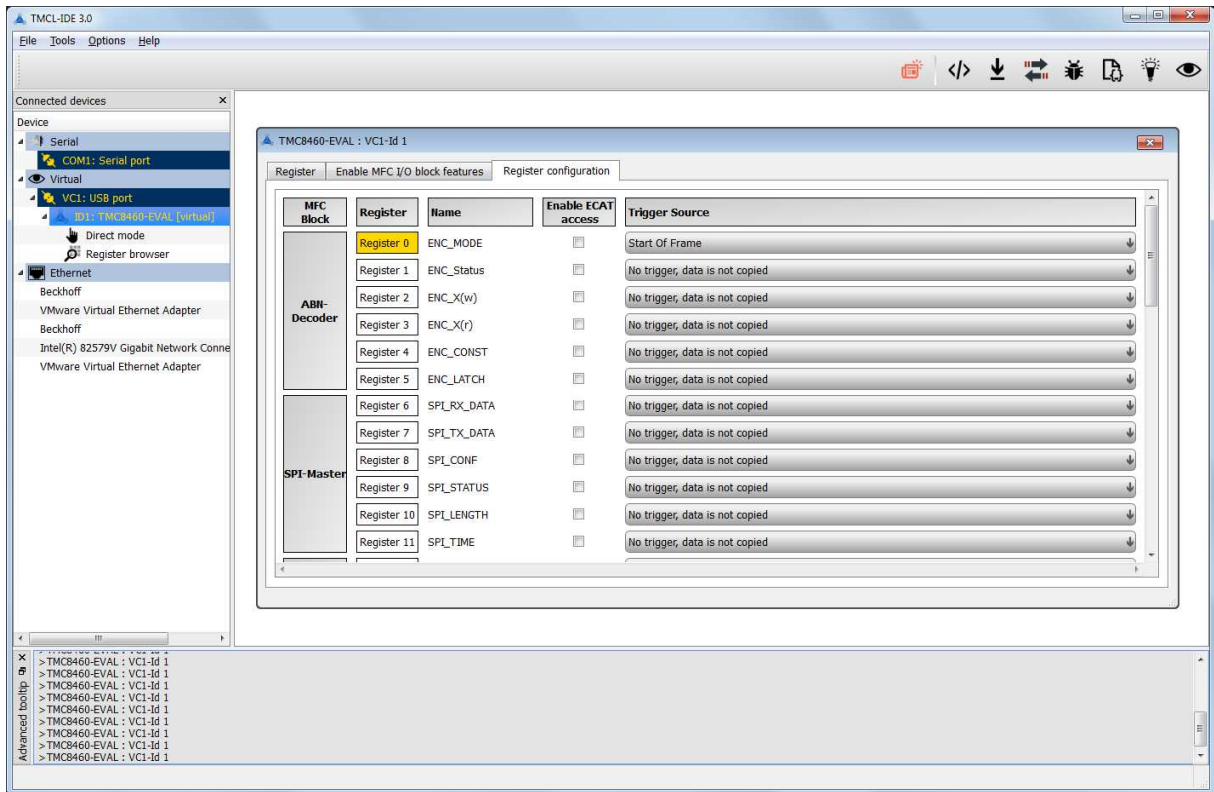


Figure 1 - TMCL-IDE with direct register access to the TMC8460-BI on its evaluation board

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



Figure 2 - Wizard Start Screen

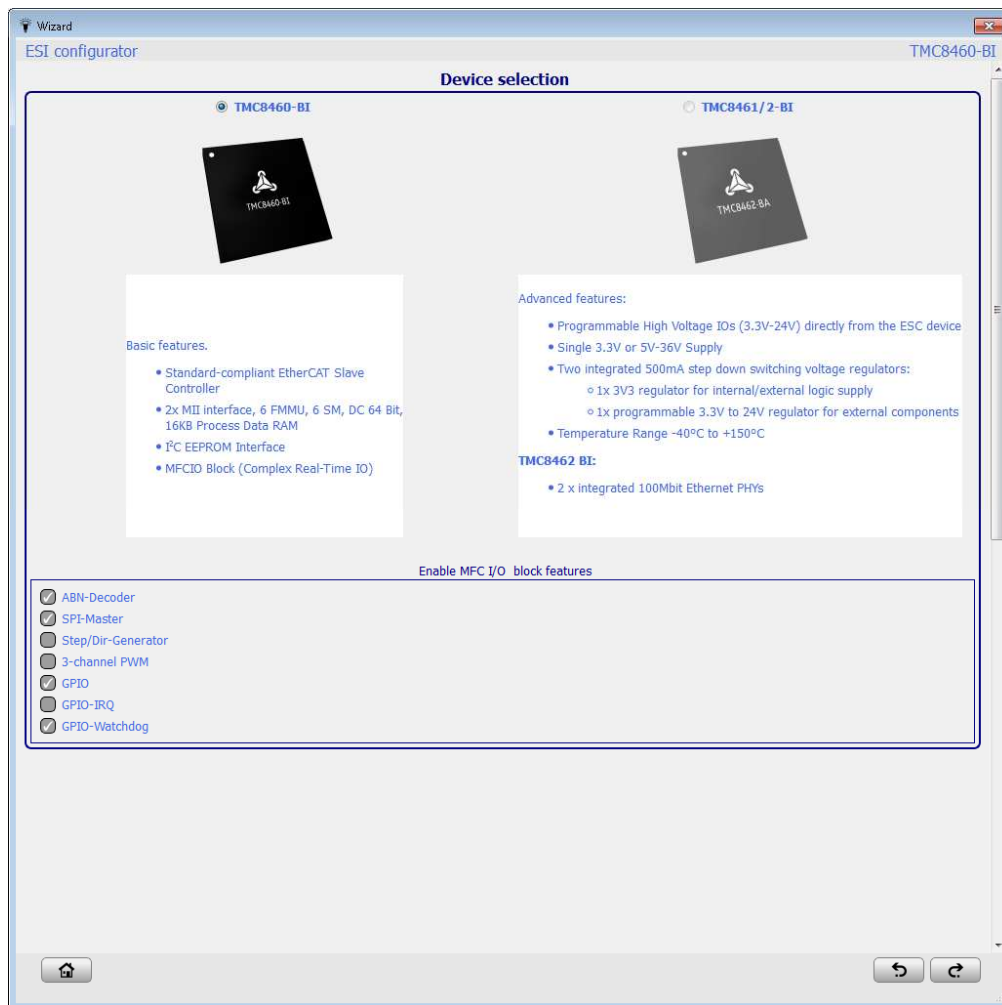


Figure 3 - Wizard Device Selection and Feature Selection

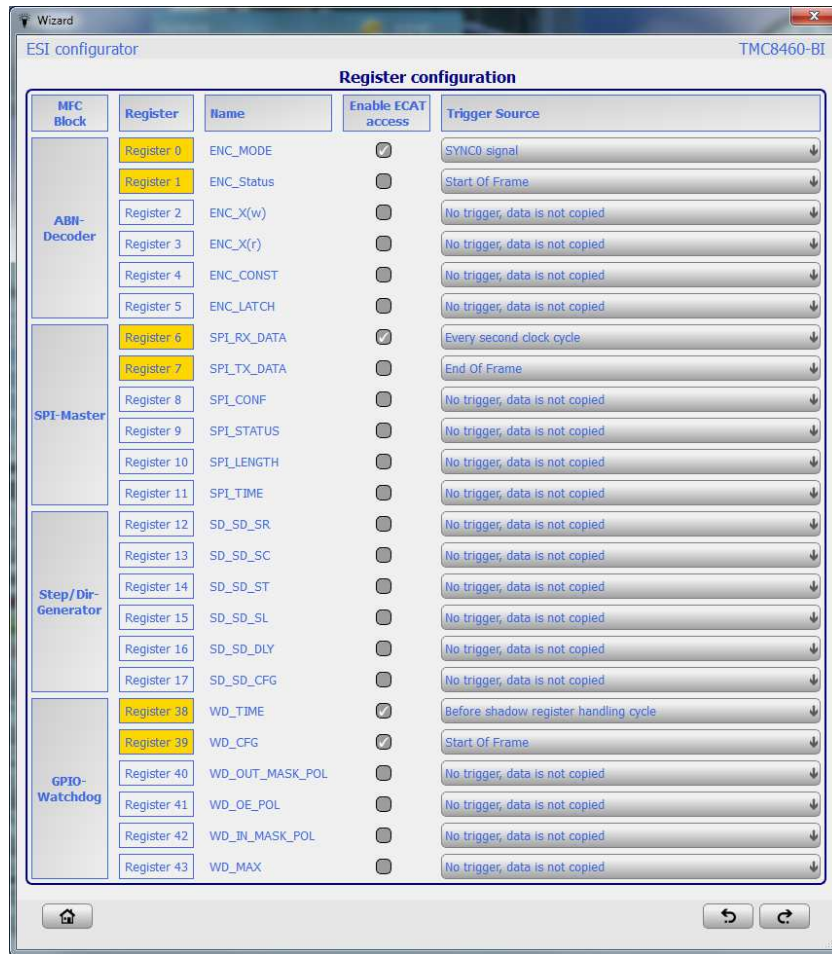


Figure 4 - Wizard Register Selection and Configuration View

```

ESI configurator
TMC8460-BI

ESI generation

Mark & Copy the <Data> entry to your esi file <Eeprom> section

<Category>
<CatNo>2049</CatNo>
<Data>030301000002021003040410101010100C0000020101010505010411121103111010000000000000000000000</Data>
</Category>

<Category>
<CatNo>2049</CatNo>
<Data>030301000002021003040410101010100C0000020101010505010411121103111010000000000000000000000</Data>
</Category>

//Prototype of spi write function
void HW_MfcWrite(UINT16 Address,UINT32 Size,UCHAR *Data);

//Address mapping definitions
#define ENC_MODE 0x0000
#define ENC_Status 0x0010
#define ENC_Xw 0x0020
#define ENC_LATCH 0x0050
#define SPI_RX_DATA 0x0060
#define SPI_CONF 0x0080
#define SPI_STATUS 0x0090
#define SPI_LENGTH 0x00A0
#define SD_DLY 0x0100
#define PWM_CHOPMODE 0x0130
#define PWM_ALIGNMENT 0x0140
#define PWM_POLARITIES 0x0150
#define PWM_VALUE_1 0x0160
#define PWM_VALUE_2 0x0170
#define PWM_VALUE_3 0x0180
#define PWM_CNTRSHIFT_1 0x0190
#define PWM_CNTRSHIFT_2 0x01A0
#define PWM_PULSE_LENGTH 0x01E0

//Convenience access via arrays
UINT16 spiAddress[]={ENC_MODE, ENC_Status, ENC_Xw, ENC_LATCH, SPI_RX_DATA, SPI_CONF, SPI_STATUS,
SPI_LENGTH, SD_DLY, PWM_CHOPMODE, PWM_ALIGNMENT, PWM_POLARITIES, PWM_VALUE_1, PWM_VALUE_2,
PWM_VALUE_3, PWM_CNTRSHIFT_1, PWM_CNTRSHIFT_2, PWM_PULSE_LENGTH};
UINT16 *spiAddress_ptr = &spiAddress[0];

UCHAR spiData[]={0x03, 0x03, 0x01, 0x02, 0x02, 0x03, 0x04, 0x04, 0x0C, 0x02, 0x01, 0x01, 0x01,
0x05, 0x05, 0x01, 0x04, 0x03};
UCHAR*spiData_ptr = &spiData[0];

//Datagram send via SPI to given address
while(spiData_ptr != &spiData[sizeof(spiData)-1])
MfcWrite (*spiAddress_ptr++, *spiData_ptr++);

```

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

3 Device Usage and Handling

3.1 Sample Block Diagrams

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

The following examples show typical system architectures using the TMC8460.

3.1.1 Typical EtherCAT Slave architecture

The first application diagram shows the TMC8460 in a typical straightforward architecture. The PHYs connect to the TMC8460 using MII interface. Both PHYs and the TMC8460 have the same 25MHz clock source (see Section 3.9.2). The I2C EEPROM is connected to the TMC8460 and contains boot-up configuration required by the ESC after reset or power-cycling. The EEPROM is optionally connected to the μ C to allow EEPROM updates via the MCU's firmware.

The μ C connects to the TMC8460 using an SPI bus interface to the PDI SPI. The local application is connected to the μ C and is controlled by the application layer inside the μ C. The application interface depends on the application and is a generic placeholder in this diagram. In this example the MFCIO IO block is not used.

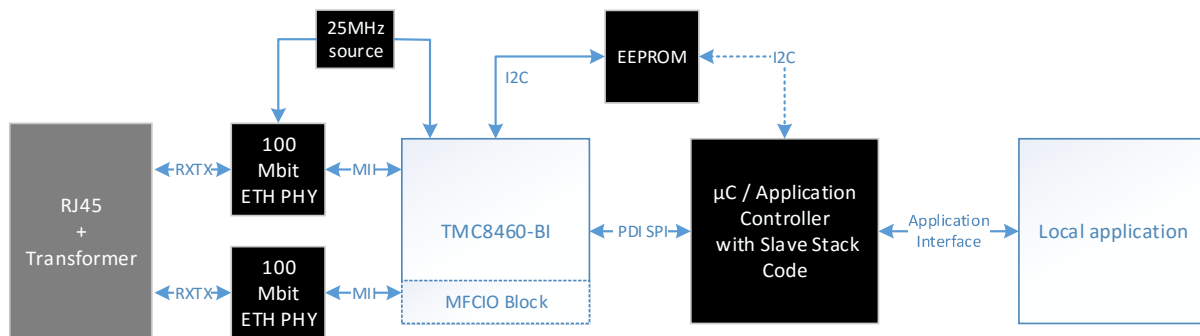


Figure 6 - Application diagram using only the local application controller to interface the application

3.1.2 MFCIO block based Microcontroller Architecture

The second application diagram shows a similar architecture with μ C but with extensive use of the MFCIO block features. The special functions of the MFCIO block allow relocating functionality from the μ C to the TMC8460. That is, certain compute intense and time-critical functions are moved from software to hardware. The application layer in the μ C can focus on interfacing to the real-time bus and for high-level control tasks of the application.

For example, an incremental encoder can directly be connected to the MFCIO block. The μ C only reads back the actual position via the dedicated SPI interface MFC CTRL SPI. Additionally, SPI slave chips are directly connected to the MFCIO and not the μ C, for example Trinamic's dedicated smart stepper motor drivers, dedicated hardware motion controllers, and simple S/D stepper motor drivers. The MFCIO block master SPI interface, the PWM functions, the S/D function, and the 16MHz clock output are used in this case. The application controller does not need to implement these firmware functions and interfaces but uses the available resources of the TMC8460 instead. Software development is simplified. Other application parts not covered by the MFCIO block still connect to the microcontroller.

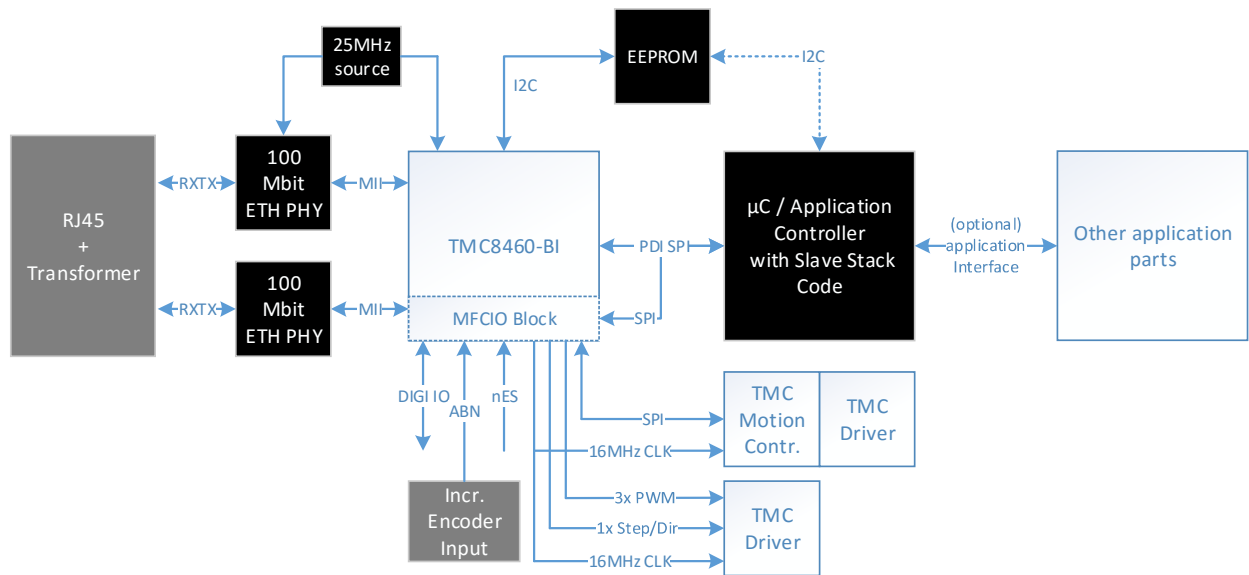


Figure 7 - Application diagram using the MFCIO block features to reduce software overhead and provide real-time hardware support to the MCU. Other application parts may still be connected to the MCU.

3.1.3 Device Emulation Example

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

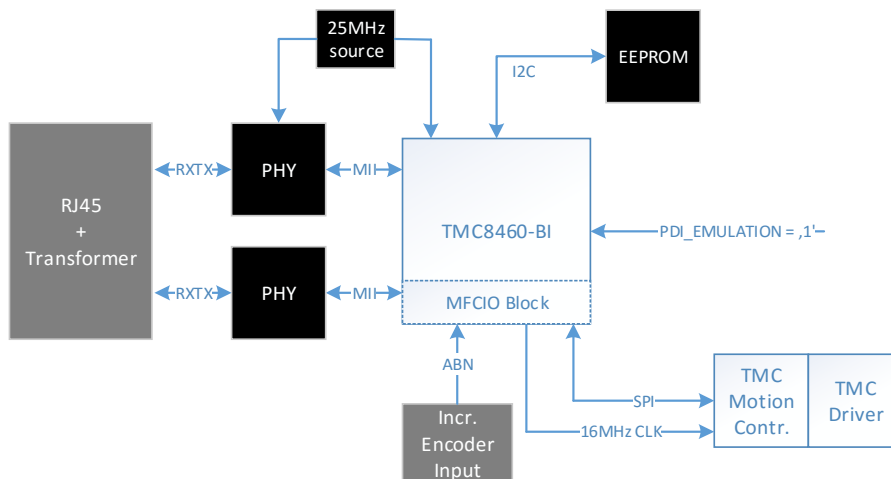
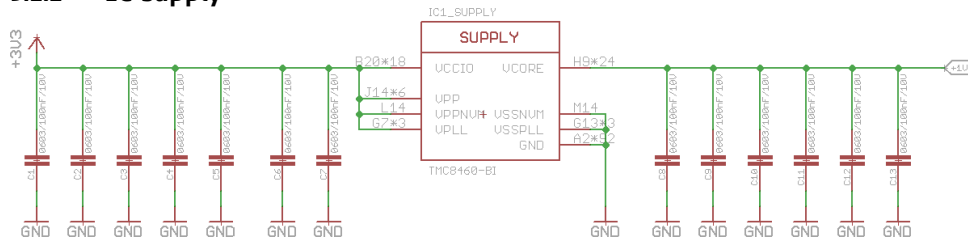


Figure 8 - Application diagram without MCU. The TMC8460 is used in device emulation mode. SPI slave chips and other application peripherals can be connected to the MFCIO block. The EtherCAT master can directly control all the application functions.

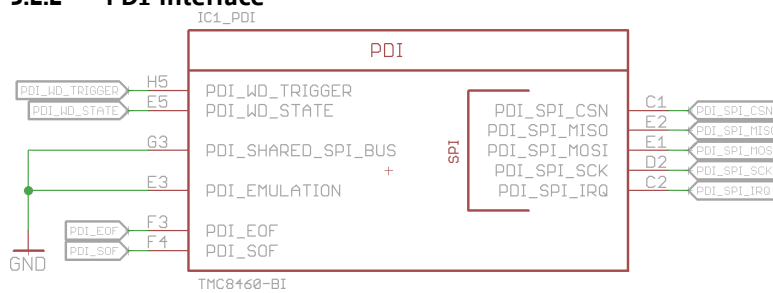
3.2 Samples Circuits

3.2.1 IC supply



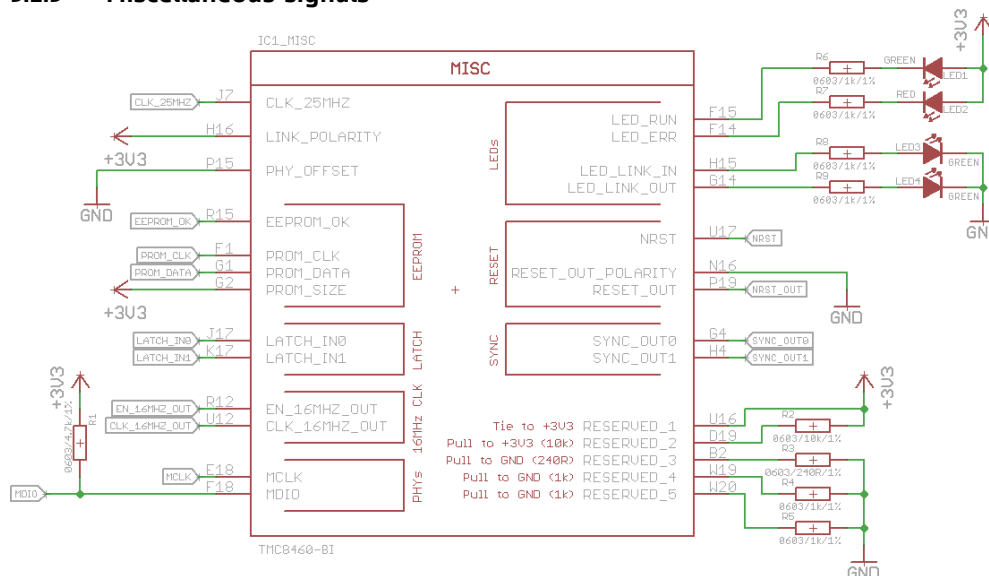
Only a minimal amount of decoupling capacitors is shown here. If possible every supply pin (1.2V and 3.3V) should have a separate 100nF capacitor connected between it and GND as close to the pin as possible. Larger capacitor values can be used on the 3.3V and 1.2V supply rails for increased stability.

3.2.2 PDI interface



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

3.2.3 Miscellaneous signals



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

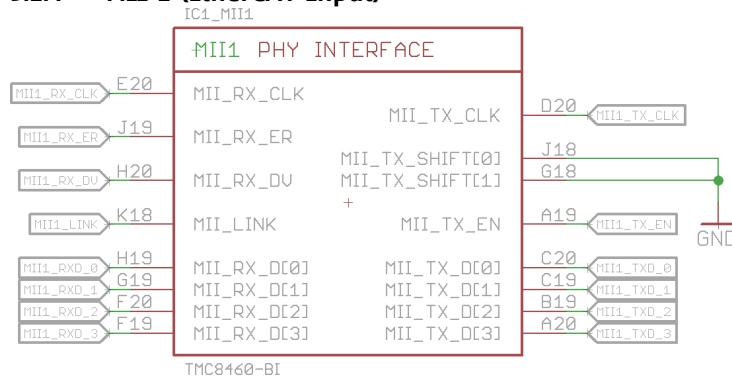
LATCH_IN0, LATCH_IN1, SYNC_OUT0, SYNC_OUT1, NRST_OUT, EN_16MHZ_OUT and CLK_16MHZ_OUT are optional signals that can be used depending on the specific use case.

Configuration Pins with their setting in this example:

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

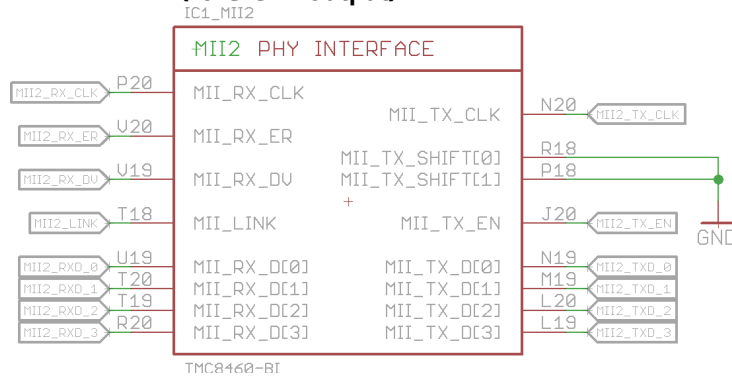
External resistors are required on the MDIO line and for four (4) of the reserved pads.

3.2.4 MII 1 (EtherCAT Input)



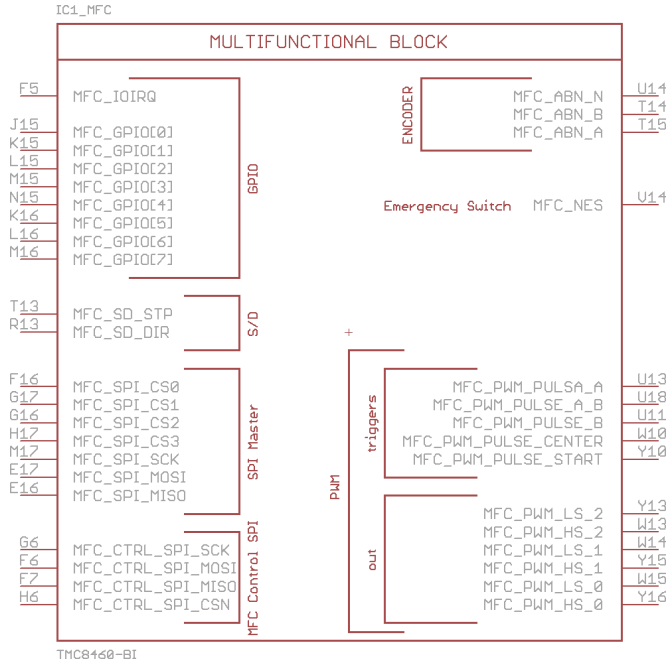
These are the connections to the first PHY, representing the input port of the EtherCAT device.

3.2.5 MII 2 (EtherCAT Output)



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

3.2.6 MFC I/Os



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

3.3 Pinout and Pin Description

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

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

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