

ECM-XF

EtherCAT Master Chip User Manual

NEXTW Technology CO., LTD.

10/15/2020

Ver.031

Introduction	5
Interface	5
SPI Mode	6
Packet Format	8
Header	9
Command Code Table	12
ECM_CMD_INFO_UPDATE_OP	17
ECM_CMD_ECAT_INIT_OP	17
ECM_CMD_ECAT_RECONFIG_OP	18
ECM_CMD_CRC_TYPE_SET	18
ECM_CMD_ECAT_PDO_WC_GET	19
ECM_CMD_ECAT_PDO_DATA_FIFO_OP	20
ECM_CMD_ECAT_PDO_DATA_OP	20
ECM_CMD_ECAT_PDO_CONFIG_SET	21
ECM_CMD_ECAT_PDO_CONFIG_REQ	23
ECM_CMD_ECAT_PDO_CONFIG_GET	24
ECM_CMD_ECAT_SDO_REQ	26
ECM_CMD_ECAT_SDO_GET	27
ECM_CMD_ECAT_STATE_SET	27
ECM_CMD_ECAT_STATE_GET	28
ECM_CMD_ECAT_SLV_INFO_GET	28
ECM_CMD_ECAT_SLV_CNT_GET	29
ECM_CMD_FIFO_ENABLE_OP	30
ECM_CMD_FIFO_PACK_SIZE_GET	30
ECM_CMD_SPI_PACK_SIZE_GET	30
ECM_CMD_SPI_RECONFIG_OP	31
ECM_CMD_CRC_ERR_CNT_CLR	31
ECM_CMD_402_CONFIG_SET	32
ECM_CMD_402_STATE_SET	32
ECM_CMD_402_STATE_GET	33
ECM_CMD_402_CTL_SET	33
ECM_CMD_402_CTL_GET	34
ECM_GPIO_CONFIG_SET(ECM_GpioSetMode)	34
ECM_GPIO_CONFIG_SET(ECM_GpioSetMode)	35
ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce)	35
ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce)	36
ECM_GPIO_CONFIG_SET(ECM_GpioSetDebounceClock)	37

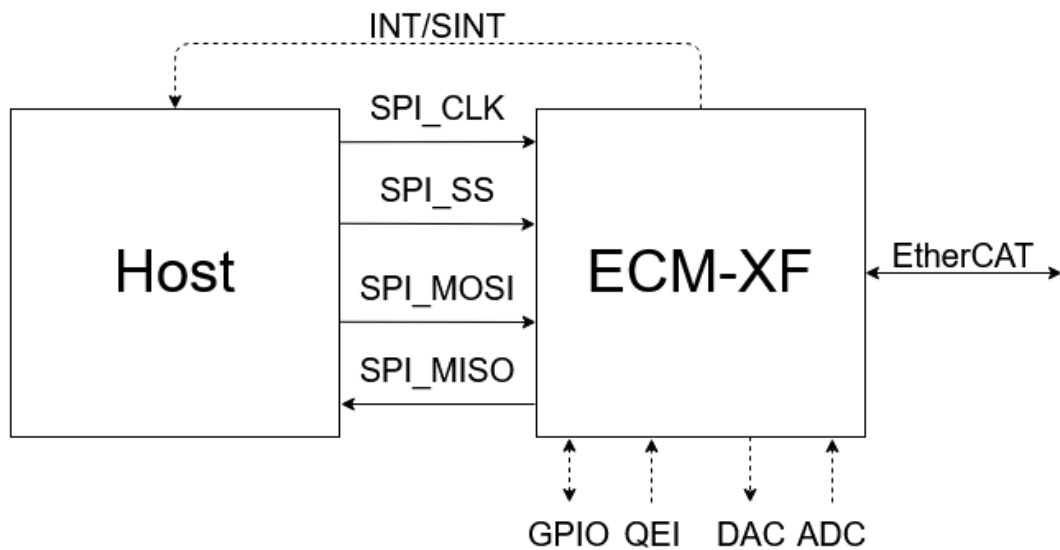
ECM_GPIO_CONFIG_SET(ECM_GpioIntEnable)	37
ECM_GPIO_CONFIG_SET(ECM_GpioIntClear)	37
ECM_GPIO_CONFIG_SET(ECM_GpioIntClear)	38
ECM_GPIO_FUNC_OP(ECM_GpioSetValue)	38
ECM_GPIO_FUNC_OP(ECM_GpioExtSetValue)	39
ECM_GPIO_FUNC_OP(ECM_GpioGetValue)	39
ECM_GPIO_FUNC_OP(ECM_GpioExtGetValue)	40
ECM_GPIO_FUNC_OP(ECM_GpioGetIntFlag)	41
ECM_GPIO_FUNC_OP(ECM_GpioExtGetIntFlag)	41
ECM_QEI_FUNC_OP(ECM_EncOpen)	42
ECM_QEI_FUNC_OP(ECM_EncStart)	42
ECM_QEI_FUNC_OP(ECM_EncStop)	42
ECM_QEI_FUNC_OP(ECM_EncGetCount)	43
ECM_DAC_FUNC_OP(ECM_DacOpen)	43
ECM_DAC_FUNC_OP(ECM_DacClose)	44
ECM_DAC_FUNC_OP(ECM_DacSetDelayTime)	44
ECM_DAC_FUNC_OP(ECM_DacSetData)	44
ECM_DAC_FUNC_OP(ECM_DacStartConv)	45
ECM_ADC_FUNC_OP(ECM_AdcOpen)	45
ECM_ADC_FUNC_OP(ECM_AdcClose)	45
ECM_ADC_FUNC_OP(ECM_AdcConfigSampleModule)	46
ECM_ADC_FUNC_OP(ECM_AdcGetDataValidFlag)	46
ECM_ADC_FUNC_OP(ECM_AdcStartConv)	47
ECM_ADC_FUNC_OP(ECM_AdcGetConvData)	47
ECM_EEPROM_REQ	48
ECM_EEPROM_GET	49
ECM_CMD_ECAT_STATE_CHECK	49
ECM_CMD_ECAT_DCSYNC	49
ECM_CMD_FIFO_CLR_OP	50
ECM_CMD_FIFO_SET_TX_CNT	50
ECM_CMD_FIFO_GET_TX_CNT	51
ECM_CMD_FIFO_SET_RX_CNT	51
ECM_CMD_FIFO_GET_RX_CNT	51
ECM_CMD_FW_VERSION_GET	52
ECM_CMD_ECAT_INT_SET_ENABLE	52
ECM_CMD_ECAT_INT_GET_ENABLE	53
ECM-XF Command and Return Access Diagram	53
ECM-XF EtherCAT Network Initialization and Slave Configuration	54
ECM-XF Interrupts	57
Update Log	63

Introduction

ECM-XF is a cost effective EtherCAT Master chip connect with Control master by SPI interface. Provide users with minimum 125 us DC cycle time and 128 slaves. Support multiple surrounding IO, interrupt and application. Suitable for PLC controller, robot controller or multi-automation controller.

Support DC	Yes
Maximum Slaves	128
Minimum Cycle Time	125us
Mailbox Protocol	CoE/FoE
Surrounding IO	GPIO / QEI / ADC / DAC
Application function	CiA402 state machine control FIFO buffer

Interface



Pic. 1 、 ECM-XF interface connection

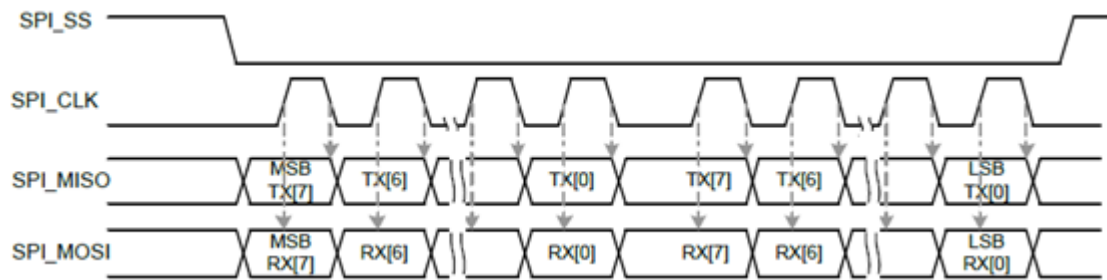
SPI Mode

Name	Pin Number	Meaning	Description
SPI_CLK	Pin 38	Signal Frequency	The minimum frequency generate and control by SPI Master to connect EtherCAT communication cycle.
SPI_MOSI	Pin 40	Master Out, Slave In	SPI Master data output, SPI Slave data input
SPI_MISO	Pin 39	Master In, Slave Out	SPI Master data input, SPI Slave data output
SPI_SS	Pin 37	SPI Slave Select	Option signal, control by master. The slave only react with master when the level in low.

Duplex Mode	Full duplex		
Transmission Rate	96Mbps(max.)		
Timing Mode	SPI_CLK at low level when idling, upward edges receive data, downward edges transmit data		
	POL=0	PHA=0	
	POL=0	TXNEG=1	RXNEG=0

SPI Mode Transmission

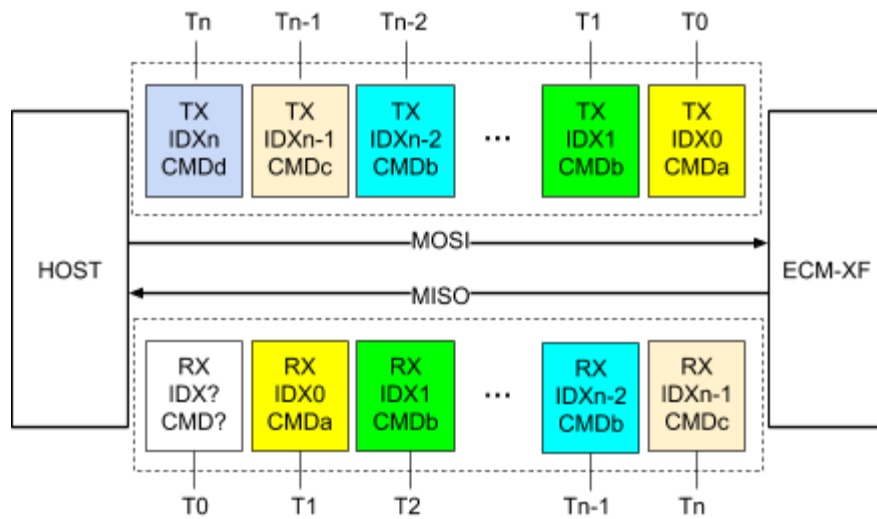
In ECM SPI Slave mode, SPI_CLK at low level when idling, upward edges receive data, downward edges transmit data, following MSB like picture shows below.



Pic. 2 SPI Timing

SPI Master generates clock to SPI slaves, transmitting data at the edges of high levels, receiving data at the edge of low levels. SPI transmission packs on Byte. Transmitting from the lowest address to the highest address. It means it starts at Byte 0, Byte 1, Byte 2 to the last Byte. But the SPI transmitting Byte are followed MSB.

SPI Command Transmit and Receive



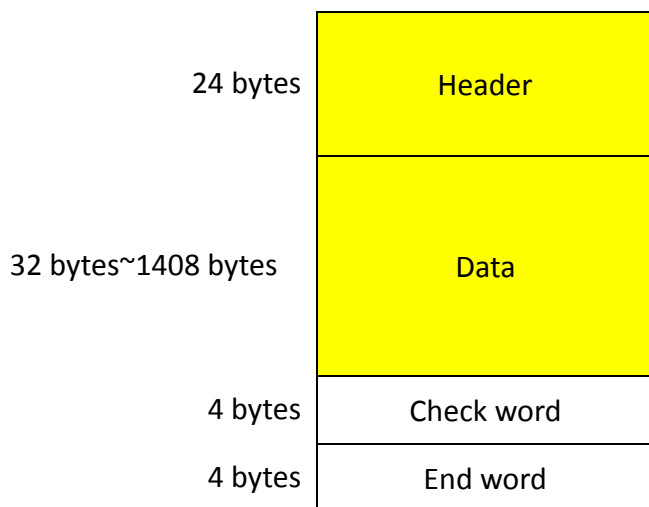
Pic 3. SPI full duplex data packet exchange

Warning

1. Master will receive the last data packet when transmitting the SPI command.
2. When ECM-XF enabled, the first return packet does not include the last respond packet.
3. If command packets are same at index code and command code, the commands will be treated as ECM_CMD_INFO_UPDATE_OP except the first one.

Packet Format

1. Command packet includes 4 parts: header, data, check word and end word.
2. Command header include commands, parameters, and system control, etc.
3. Return header include command return values, return data, error states and system state, etc.
4. Data section include PDO data or including data. The data size could be modified by application.
5. Check word has 4 different types for users as following: Fixed value, CRC-8, CRC-CCITT, CRC 32.
6. End word is a fixed value(0x56575859) for checking packet completeness
7. Data size is 32 bytes~1408 bytes, default is 112 bytes.



Pic 3. SPI packet format

Header

Command Header: Master to ECM-XF packet header

3	2	1	0	
Packet Header				0x00
Parametric data 3	Parametric data 2	Parametric data 1	Parametric data 0	0x04
Index	Control Word	Reserved		0x08
Data Length		Command Parameter	Command Code	0x0c
Interrupt Control				0x10
GPIO Output 1	GPIO Output 0	Interrupt Erase 1	Interrupt Erase 0	0x14

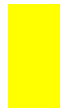
Pic 4 Command Header



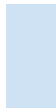
Response Header: The header from ECM-XF to master

3	2	1	0	
Packet Header				0x00
RxPDO waiting to transmit quantity	TxPDO waiting to receive quantity	WKC Error Count	CRC Error Count	0x04
Return Index	ECM State	ECM Error State	ECAT State	0x08
Data Length		Return Value	Return Command Code	0x0c
Interrupt Flag				0x10
GPIO Input 1	GPIO Input 0	GPIO Interrupt Flag 1	GPIO Interrupt Flag 0	0x14

Pic 5 Return Header



Command Code relating



Interrupt relating



System or Hardware relating



Error State

Header Column Explanation

Table 1. Header column in command packet

Command Header			
Location	Length	Name	Description
0x00	4	Packet Start Word	Fixed Value(0xA1A2A3A4)
0x04	1	Parametric Data 0	Transmit command parameter or data
0x05	1	Parametric Data 1	Transmit command parameter or data
0x06	1	Parametric Data 2	Transmit command parameter or data
0x07	1	Parametric Data 3	Transmit command parameter or data
0x08	2	Reserved	
0x0A	1	Control Word	Control or erase system interrupt event bit0: Valid header IO output column bit3: Erase FIFO error bit4: Erase SPI communication error bit5: Erase SPI CRC error bit6: Erase blocking command bit7: Erase non-blocking command
0x0B	1	Index	Identify main application
0x0C	1	Command Code	Refer to command table
0x0D	1	Command Parameter	command parameter
0x0E	2	Data	command parameter or data transmission

		Length	
0x10	4	Interrupt Control	Interrupt control column
0x14	1	Erase Interrupt 0	Erase GPIO low level interrupt flag
0x15	1	Erase Interrupt 1	Erase GPIO high level interrupt flag
0x16	1	GPIO output 0	GPIO low level output value
0x17	1	GPIO output 1	GPIO high level output value

Table 2. Header column in return packet

Return Header			
Location	Length	Name	Description
0x00	4	Packet Start Word	Fixed value(0xA1A2A3A4)
0x04	1	CRC Error Count	Number of CRC error count in SPI communication
0x05	1	WKC Error Count	Number of working count error in PDO communication
0x06	1	Number of TxPdoFIFO	Number of TxPDO which are not received in FIFO
0x07	1	Number of RxPdoFifo	Number of RxPDO which are not received in FIFO
0x08	1	ECAT state	0x00 : NONE(non initialize state) 0x01 : INIT(initialized state) 0x02 : PRE_OP(pre-operate state) 0x03 : BOOT(boot state) 0x04 : SAFE_OP(safe operate state) 0x08 : OPERATIONAL(operation state)

			0x10 : ERROR(error state)
0x09	1	Error State	bit3: blocking command return error bit5: FIFO error bit6: SPI CRC error bit7: blocking command busy error
0x0A	1	State	bit1-bit0 : 0 : fixed value check 1: CRC8 check 2: CRC-CCITT check 3: CRC32 check bit2 : DC cycle time stable flag bit3 : ethernet initialized flag bit4 : ECAT PDO configured flag bit5 : return packet is NOP flag bit6 : FIFO enable flag bit7 : blocking time command busy flag
0x0B	1	Index	Return command header index
0x0C	1	Command Code	Return command header command code
0x0D	1	Command Return Value	Return value from return command
0x0E	2	Data length	command parameter or data valid length
0x10	4	Interrupt flag	interrupt flag
0x14	1	GPIO interrupt flag 0	GPIO low level interrupt flag
0x15	1	GPIO interrupt flag 1	GPIO high level interrupt flag
0x16	1	GPIO input 0	GPIO low level input value
0x17	1	GPIO input 1	GPIO high level input value

Command Code Table

Commands are divided into two types by response time.

1. NON-BLOCKING - Received commands and operate immediately and return parameters or values.
2. BLOCKING - Received commands and wait a period time to operate and return parameters or values.
3. ECM-XF can only operate a blocking command once. Check the blocking command busy flag of ECM state is zero before transmit next blocking command.
4. When blocking command operating, transmit non-blocking command and other operation is allowed.

Please refer to Pic 6 Command and Return flowchart diagram

Table 3. Command Table

Blocking Command	Non-blocking Command
ECM_CMD_ECAT_INIT_OP ECM_CMD_ECAT_RECONFIG_OP ECM_CMD_ECAT_DCSYNC ECM_CMD_ECAT_PDO_CONFIG_SET ECM_CMD_ECAT_STATE_SET	ECM_CMD_INFO_UPDATE_OP ECM_CMD_FW_VERSION_GET ECM_CMD_ECAT_PDO_DATA_FIFO_OP ECM_CMD_ECAT_PDO_DATA_OP ECM_CMD_ECAT_STATE_GET ECM_CMD_ECAT_PDO_WC_GET ECM_CMD_ECAT_SLV_INFO_GET ECM_CMD_ECAT_SLV_CNT_GET ECM_CMD_FIFO_ENABLE_OP ECM_CMD_FIFO_PACK_SIZE_GET ECM_CMD_SPI_PACK_SIZE_GET ECM_CMD_SPI_RECONFIG_OP ECM_CMD_CRC_ERR_CNT_CLR ECM_CMD_CRC_TYPE_SET ECM_CMD_402_CONFIG_SET ECM_CMD_402_STATE_SET ECM_CMD_402_STATE_GET ECM_CMD_402_CTL_SET ECM_CMD_402_CTL_GET
ECM_CMD_ECAT_PDO_CONFIG_REQ ECM_CMD_ECAT_SDO_REQ ECM_EEPROM_REQ	ECM_CMD_ECAT_PDO_CONFIG_GET ECM_CMD_ECAT_SDO_GET ECM_EEPROM_GET

Table 4. Command Description

<u>ECM_CMD_INFO_UPDATE_OP</u>	Refresh header data
<u>ECM_CMD_ECAT_INIT_OP</u>	Initialize EtherCAT network and slaves before EtherCAT application
<u>ECM_CMD_ECAT_RECONFIG_OP</u>	After configuring PDO, reconfiguring memory size for ECM-XF
<u>ECM_CMD_CRC_TYPE_SET</u>	Set checking type
<u>ECM_CMD_ECAT_PDO_WC_GET</u>	Read the slaves quantity of the PDO data contents outputs/inputs or working counter value.
<u>ECM_CMD_ECAT_PDO_DATA_FIFO_OP</u>	Access PDO data to FIFO
<u>ECM_CMD_ECAT_PDO_DATA_OP</u>	Access PDO data
<u>ECM_CMD_ECAT_PDO_CONFIG_SET</u>	Configure PDO slaves
<u>ECM_CMD_ECAT_PDO_CONFIG_REQ</u>	Request PDO configuration read. Use ECM_CMD_PDO_CONFIG_GET to return configuration.
<u>ECM_CMD_ECAT_PDO_CONFIG_GET</u>	Read PDO slave configuration.
<u>ECM_CMD_ECAT_SDO_REQ</u>	Request operating SDO write/read commands. Use ECM_CMD_ECAT_SDO_GET to read return data
<u>ECM_CMD_ECAT_SDO_GET</u>	Return ECM_CMD_ECAT_SDO_REQ data
<u>ECM_CMD_ECAT_STATE_SET</u>	Set EtherCAT state
<u>ECM_CMD_ECAT_STATE_GET</u>	Return EtherCAT state
<u>ECM_CMD_ECAT_SLV_INFO_GET</u>	Read slaves information
<u>ECM_CMD_ECAT_SLV_CNT_GET</u>	Read slave quantity
<u>ECM_CMD_FIFO_ENABLE_OP</u>	Enable RxPDO output. ECM-XF enables RxPDO FIFO by default. Use this command can disable RxPDO FIFO manually and pre-save output.
<u>ECM_CMD_FIFO_PACK_SIZE_GET</u>	Read PDO data length, a PDO counts one in FIFO
<u>ECM_CMD_SPI_PACK_SIZE_GET</u>	Read SPI data packet length

ECM_CMD_SPI_RECONFIG_OP	Reset SPI data packet length
ECM_CMD_CRC_ERR_CNT_CLR	Erase CRC error counter
ECM_CMD_402_CONFIG_SET	Assign slaves control word and status word offset
ECM_CMD_402_STATE_SET	Switch 402 state of assigned slaves
ECM_CMD_402_STATE_GET	Read 402 state of assigned slaves
ECM_CMD_402_CTL_SET	Set control byte in ECM-XF internal 402 state
ECM_CMD_402_CTL_GET	Read control byte in ECM-XF internal 402 state
ECM_GPIO_CONFIG_SET (ECM_GpioSetMode)	Set GPIO mode
ECM_GPIO_CONFIG_SET (ECM_GpioEnableDebounce)	Enable/disable GPIO debounce
ECM_GPIO_CONFIG_SET (ECM_GpioSetDebounceClock)	Set debounce clock
ECM_GPIO_CONFIG_SET (ECM_GpioIntEnable)	Enable/Disable GPIO interrupt
ECM_GPIO_CONFIG_SET (ECM_GpioIntClear)	Clear GPIO interrupt flag
ECM_GPIO_FUNC_OP (ECM_GpioSetValue)	Set GPIO value
ECM_GPIO_FUNC_OP (ECM_GpioExtSetValue)	Set GPIO value (Extension function, only work for ver. 9 or later)
ECM_GPIO_FUNC_OP (ECM_GpioGetValue)	Get GPIO value
ECM_GPIO_FUNC_OP (ECM_GpioExtGetValue)	Get GPIO value (Extension function, only work for ver. 9 or later)
ECM_GPIO_FUNC_OP (ECM_GpioGetIntFlag)	Get GPIO Interrupt flag
ECM_GPIO_FUNC_OP (ECM_GpioExtGetIntFlag)	Get GPIO Interrupt flag (Extension function, only work for ver. 9 or later)

ECM_QEI_FUNC_OP (ECM_EncOpen)	Open and set encoder mode
ECM_QEI_FUNC_OP (ECM_EncStart)	Start encoder counting
ECM_QEI_FUNC_OP (ECM_EncStop)	Stop encoder counting
ECM_QEI_FUNC_OP (ECM_EncGetCount)	Get encoder counter value
ECM_DAC_FUNC_OP (ECM_DacOpen)	Open DAC function
ECM_DAC_FUNC_OP (ECM_DacClose)	Close DAC function
ECM_DAC_FUNC_OP (ECM_DacSetDelayTime)	Set DAC delay time
ECM_DAC_FUNC_OP (ECM_DacSetData)	Set DAC data
ECM_DAC_FUNC_OP (ECM_DacStartConv)	Start DAC conversion
ECM_ADC_FUNC_OP (ECM_AdcOpen)	Open ADC function
ECM_ADC_FUNC_OP (ECM_AdcClose)	Close ADC function
ECM_ADC_FUNC_OP (ECM_AdcConfigSampleModule)	Configure ADC trigger source
ECM_ADC_FUNC_OP (ECM_AdcGetDataValidFlag)	Get ADC data valid flag
ECM_ADC_FUNC_OP (ECM_AdcStartConv)	Start ADC conversion
ECM_ADC_FUNC_OP (ECM_AdcGetConvData)	Get ADC converted data
ECM_EEPROM_REQ	Request EEPROM write/read command. Use ECM_EEPROM_GET to return reading data.
ECM_EEPROM_GET	Read ECM_EEPROM_REQ data

ECM_CMD_ECAT_STATE_CHECK	Reflash and check slaves state
ECM_CMD_ECAT_DCSYNC	Set DC sync signal source, cycle time, and shift time
ECM_CMD_FIFO_CLR_OP	Clear FIFO
ECM_CMD_FIFO_SET_TX_CNT	Set Tx FIFO count
ECM_CMD_FIFO_GET_TX_CNT	Get Tx FIFO count
ECM_CMD_FIFO_SET_RX_CNT	Set Rx FIFO count
ECM_CMD_FIFO_GET_RX_CNT	Get Rx FIFO count
ECM_CMD_FW_VERSION_GET	Read firmware version
ECM_CMD_ECAT_INT_SET_ENABLE	Set interrupt enable mask
ECM_CMD_ECAT_INT_GET_ENABLE	Get interrupt enable mask

Command Description

In this section, introducing commands and header relating columns and data content. Please refer to pic 3 to pic 5.

1. Yellow background are command relating, according to command code parametric data
2. Index codes for identify return packet
3. When the command packets have same index code, the commands will be treated as ECM_CMD_INFO_UPDATE_OP except the first one.

ECM_CMD_INFO_UPDATE_OP Refresh header data			
Command Packet			
Header	1	Command Code	0
	2	Index Code	Custom Defined
	3	Data Length	0

ECM_CMD_ECAT_INIT_OP Initialize EtherCAT network and slaves before EtherCAT application			
Command Packet			
Header	1	Command	1

		Code																			
	2	Index Code	Custom Defined																		
	3	Data Length	16																		
Data Section	1	<div style="text-align: center;"> 3 2 1 0 </div> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 40px; text-align: center;">DC activate</td> <td style="width: 40px; text-align: center;">Reserved</td> <td style="width: 40px; text-align: center;">Reserved</td> <td style="width: 20px; text-align: center;">0</td> </tr> <tr> <td colspan="3" style="text-align: center;">Cycle time 0</td> <td style="text-align: center;">4</td> </tr> <tr> <td colspan="3" style="text-align: center;">Cycle time 1</td> <td style="text-align: center;">8</td> </tr> <tr> <td colspan="3" style="text-align: center;">Time offset</td> <td style="text-align: center;">12</td> </tr> </table>				DC activate	Reserved	Reserved	0	Cycle time 0			4	Cycle time 1			8	Time offset			12
		DC activate	Reserved	Reserved	0																
		Cycle time 0			4																
		Cycle time 1			8																
	Time offset			12																	
		Location	Length	Name	Description																
		0x00	1	Reserved	Fixed Value 0xFF																
		0x01	1	Reserved	Fixed Value 0																
		0x02	2	DC activate	Reference AssignActivate in ESI																
	0x04	4	Cycle Time 0	SYNC0 Cycle Time(ns)																	
	0x08	4	Cycle Time 1	SYNC1 Cycle Time(ns)																	
	0x12	4	Time Offset	Cycle Time Offset(ns)																	

ECM_CMD_ECAT_RECONFIG_OP				
After configuring PDO, reconfiguring memory size for ECM-XF				
Command Packet				
Header	1	Command Code	2	
	2	Index Code	Custom Defined	
	3	Data Length	0	

ECM_CMD_CRC_TYPE_SET				
Set checking type				

Command Code			
Header	1	Command Code	3
	2	Index	Custom Defined
	3	Data Length	0
	4	Data Parameter	0 : Fixed Value Check: 0x12345678 1 : CRC-8: $X^8 + X^2 + X + 1$ (Poly = 0x07) Init = 0x00, RefIn = False, RefOut = False, XorOut = 0x00 2 : CRC-CCITT-FALSE: $X^{16} + X^{12} + X^5 + 1$ (Poly = 0x1021) Init = 0xFFFF, RefIn = False, RefOut = False, XorOut = 0x0000 3 : CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7$ + $X^5 + X^4 + X^2 + X + 1$ Init = 0xFFFFFFFF, RefIn = True, RefOut = True, XorOut = 0xFFFFFFFF

ECM_CMD_ECAT_PDO_WC_GET			
Read the slaves quantity of the PDO data contents outputs/inputs or working counter value.			
Command Packet			
Header	1	Command Code	4
	2	Index	Custom Defined
	3	Data Length	0
	4	Command Parameter	1 : Input Slave Quantity 2 : Output Slave Quantity 3 : Working Counter Quantity
Return Packet			
Header	1	Command Code	4
	2	Index	Same as Command Packet Index
	3	Data Length	4
Data Section	1	Read return quantity value	

Working counter logical value = (input slave quantity)*1 + (output slave quantity)*2

ECM_CMD_ECAT_PDO_DATA_FIFO_OP Access PDO data to FIFO			
Command Packet			
Header	1	Command Code	5
	2	Index Code	Custom Defined
	3	Data Length	RxPDO Length
	4	Command Parameter	PDO Data quantity
	5	Command Data 0	Write/Read Operation Code bit 0: Write operation bit 1: Read operation
Data Section	1	PDO Data	RxPDO Data
Return Packet			
Header	1	Command Code	5
	2	Index Code	Same as Command Packet Index
	3	Data Length	TxPDO Length
	4	Return Value	Write/Read Operation Code bit 0: Write operation bit 1: Read operation
Data Section	1		TxPDO Data

ECM_CMD_ECAT_PDO_DATA_OP Access PDO data			
Command Packet			
Header	1	Command Code	6
	2	Index Code	Custom Defined

	3	Command Parameter	RxPDO Length
	4	Command Data 0	Write/Read Operation Code bit 0: Write operation bit 1: Read operation bit 2: 1 presents operation valid. If bit value is 0, force it operates write and read operation
Data Section	1	RxPDO Data	
Return Packet			
Header	1	Command Code	6
	2	Index Code	Same as Command Packet Index
	3	Data Length	TxPDO Length
	4	Return Value	Write/Read Operation Code bit 0: Write operation bit 1: Read operation bit 2: 1 presents operation valid. If bit value is 0, force it operates write and read operation
Data Section	1	TxPDO Data	

ECM_CMD_ECAT_PDO_CONFIG_SET Set PDO configuration			
Command Packet			
Header	1	Command Code	7
	2	Index	Custom Defined
	3	Data Length	112

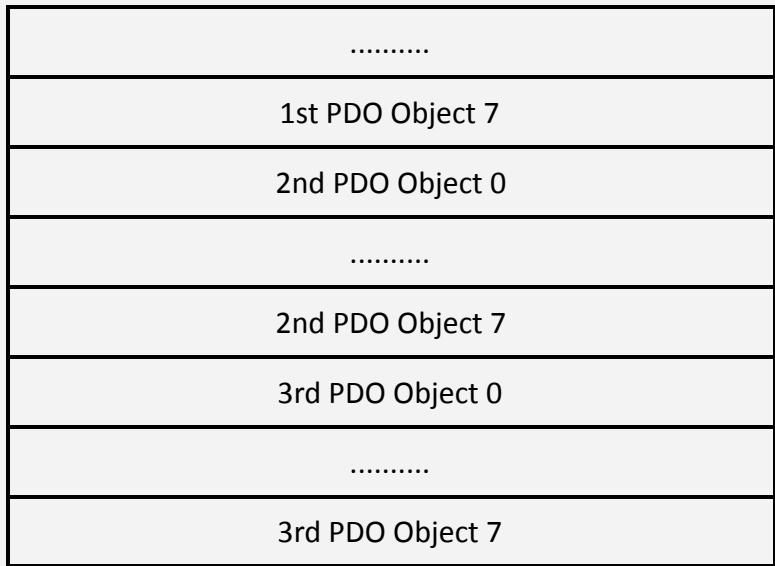
Data Section	1					
		3	2	1	0	
		PDO Assigned Index		PDO quantity	Slave Number	0x00
		2nd PDO Mapping		1st PDO Mapping		0x04
		1st PDO Object Quantity		3rd PDO Mapping		0x08
		3rd PDO Object Quantity		2nd PDO Object Quantity		0x0C
		1st PDO Object 0				0x10
					
		1st PDO Object 7				
		2nd PDO Object 0				0x30
					
		2nd PDO Object 7				
		3rd PDO Object 0				0x50
					
		3rd PDO Object 7				
		Location	Length	Name	Description	
		0x00	1	Slave Number	0~127	
		0x01	1	PDO Quantity	0~127	
		0x02	2	PDO Assigned Index	TxPDO/RxPDO Assigned Index	
		0x04	2	1st PDO Mapping	The 1st PDO Mapping Index	
0x06	2	2nd PDO Mapping	The 2nd PDO Mapping Index			
0x08	2	3rd PDO	The 3rd PDO Mapping Index			

			Mapping				
	0x0A	2	1st PDO Object Quantity	The 1st PDO Object Quantity			
	0x0C	2	2nd PDO Object Quantity	The 2nd PDO Object Quantity			
	0x0E	2	3rd PDO Object Quantity	The 3rd PDO Object Quantity			
	0x10	4	The 1st PDO Object 0 3 2 1 0 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 30px;">Index</td> <td style="width: 30px;">Sub-index</td> <td style="width: 30px;">Length(bits)</td> </tr> </table>		Index	Sub-index	Length(bits)
Index	Sub-index	Length(bits)					
	...		The 1st PDO Object N(N=1~7)				
	0x30	4	The 2nd PDO Object 0				
	...		The 2nd PDO Object N(N=1~7)				
	0x50	4	The 3rd PDO Object 0				
	...		The 3rd PDO Object N(N=1~7)				

ECM_CMD_ECAT_PDO_CONFIG_REQ								
Request execute PDO configuration reading								
Use ECM_CMD_ECAT_PDO_CONFIG_GET to get configuration								
Command Packet								
Header	1	Command Code	8					
	2	Index	Custom Defined					
	3	Data Length	4					
Data Section	1	3 2 1 0 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 30px;">PDO Assigned Index</td> <td style="width: 30px;">Reserved</td> <td style="width: 30px;">Slave</td> </tr> </table>			PDO Assigned Index	Reserved	Slave	0x00
		PDO Assigned Index	Reserved	Slave				

				Number
	Location	Length	Name	Description
	0x00	1	Slave Number	0~127
	0x01	1	Reserved	
	0x02	2	PDO Assigned Index	TxPDO/RxPDO Assigned Index

ECM_CMD_ECAT_PDO_CONFIG_GET																												
Read slave PDO configuration																												
Command Packet																												
Header	1	Command Code	9																									
	2	Index	Custom Defined																									
	3	Data Length	0																									
Return Packet																												
Header	1	Command Code	9																									
	2	Index	Same as Command Packet Index																									
	3	Data Length	112																									
Data Section	1	<table border="1"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>PDO Assigned Index</td> <td>PDO Quantity</td> <td>Slave Number</td> <td>0x00</td> </tr> <tr> <td>2nd PDO Mapping</td> <td colspan="2">1st PDO Mapping</td> <td>0x04</td> </tr> <tr> <td>1st PDO Object Quantity</td> <td colspan="2">3rd PDO Mapping</td> <td>0x08</td> </tr> <tr> <td>3rd PDO Object Quantity</td> <td colspan="2">2nd PDO Object Quantity</td> <td>0x0C</td> </tr> <tr> <td colspan="3">1st PDO Object 0</td> <td>0x10</td> </tr> </table>			3	2	1	0	PDO Assigned Index	PDO Quantity	Slave Number	0x00	2nd PDO Mapping	1st PDO Mapping		0x04	1st PDO Object Quantity	3rd PDO Mapping		0x08	3rd PDO Object Quantity	2nd PDO Object Quantity		0x0C	1st PDO Object 0			0x10
		3	2	1	0																							
		PDO Assigned Index	PDO Quantity	Slave Number	0x00																							
		2nd PDO Mapping	1st PDO Mapping		0x04																							
		1st PDO Object Quantity	3rd PDO Mapping		0x08																							
		3rd PDO Object Quantity	2nd PDO Object Quantity		0x0C																							
1st PDO Object 0			0x10																									



0x30

0x50

Location	Length	Name	Description
0x00	1	Slave Number	0~127
0x01	1	PDO Quantity	0~127
0x02	2	PDO Assigned Index	TxPDO/PxPDO Assigned Index
0x04	2	1st PDO Mapping	The 1st PDO Mapping Index
0x06	2	2nd PDO Mapping	The 2nd PDO Mapping Index
0x08	2	3rd PDO Mapping	The 3rd PDO Mapping Index
0x0A	2	1st PDO Object Quantity	The 1st PDO Object Quantity
0x0C	2	2nd PDO Object Quantity	The 2nd PDO Object Quantity
0x0E	2	3rd PDO Object Quantity	The 3rd PDO Object Quantity

	0x10	4	The 1st PDO Object 0			
			3	2	1	0
			Index		Sub-index	Length (bits)
	...		The 1st PDO Object N(N=1~7)			
	0x30	4	The 2nd PDO Object 0			
	...		The 2nd PDO Object N(N=1~7)			
0x50	4	The 3rd PDO Object 0				
...		The 3rd PDO Object N(N=1~7)				

ECM_CMD_ECAT_SDO_REQ
Request SDO write/read command.
Use ECM_CMD_ECAT_SDO_GET to read return data.

Command Packet

Header	1	Command Code	10			
	2	Index Code	Custom Defined			
	3	Data Length	Read Operation : 12 Write Operation : 12+ Valid command data			
Data Section	1					
		3	2	1	0	
		Index Code		Slave Number	Operation Code	0
		Data Length		Reserved	Sub-index Code	4
		Operation Timeout				8
Data				12		
		Location	Length	Name	Description	

	0x00	1	Operation Code	0:Write Operation 1:Read Operation
	0x01	1	Slave Number	0~127
	0x02	2	Index	CoE Object Index
	0x04	1	Sub-index	CoE Object Sub-index
	0x05	1	Reserved	
	0x06	2	Data Length	Valid Write Operation, Valid Command Length
	0x08	4	Timeout	SDO Maximum Timeout(ns)
	0x12	256	Data	Valid Write Operation

ECM_CMD_ECAT_SDO_GET Read the result from ECM_CMD_ECAT_SDO_REQ				
Command Packet				
Header	1	Return Code	11	
	2	Index Code	Custom Defined	
	3	Data Length	0	
Return Packet				
Header	1	Command Code	11	
	2	Index Code	Same as Command Packet Index	
	3	Data Length	Valid Data Length	
Data Section	1	Last ECM_CMD_ECAT_SDO_REQ Read Data		

ECM_CMD_ECAT_STATE_SET Set EtherCAT state				
Command Packet				
Header	1	Command Code	12	

	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	Slave Location
	5	Command Data 0	EtherCAT State

ECM_CMD_ECAT_STATE_GET Return EtherCAT state			
Command Packet			
Header	1	Command Code	13
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	Slave Location
Return Packet			
Header	1	Command Code	13
	2	Index Code	Same as Command Packet Index
	3	Data Length	0
	4	Return Value	EtherCAT State

ECM_CMD_ECAT_SLV_INFO_GET Read Slave Information			
Command Packet			
Header	1	Command Code	15
	2	Index	Custom Defined
	3	Data Length	0
	4	Command Parameter	Slave No.

	5	Command Data 0	Slave Information Code 0 : Manufacturer Code 1 : Manufacturer Product Number 2 : Version Number 3 : Product Name 4 : Configuration Location 5 : Location Alias 6 : State 7 : AL state 8 : Output Data Length 9 : Input Data Length
--	---	----------------	--

Return Packet			
Header	1	Command Code	15
	2	Index	Same as Command Packet Index
	3	Data Length	Slave Information Data Length
Data Section	1	Slave Information	

ECM_CMD_ECAT_SLV_CNT_GET Read slave quantity			
Command Packet			
Header	1	Command Code	16
	2	Index Code	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	16
	2	Index Code	Same as Command Packet Index
	3	Data Length	0
	4	Return Value	Slave quantity

ECM_CMD_FIFO_ENABLE_OP Enable RxPDO output. ECM-XF enables RxPDO FIFO by default. Use this command can disable RxPDO FIFO manually and pre-save output.			
Command Packet			
Header	1	Command Code	18
	2	Index	Custom Defined
	3	Data Length	0
	4	Command Parameter	0 : Disable 1 : Enable

ECM_CMD_FIFO_PACK_SIZE_GET Read PDO data length, a PDO counts 1 in FIFO.			
Command Packet			
Header	1	Command Code	19
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	0 : Read RxPDO Data Length 1 : Read TxPDO Data Length
Return Packet			
Header	1	Command Code	19
	2	Index	Same as Command Packet Index
	3	Data Length	2
Data Section	1	PDO Data Length	

ECM_CMD_SPI_PACK_SIZE_GET Read SPI data packet length			
Command Packet			
Header	1	Command Code	20

	2	Index Code	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	20
	2	Index Code	Same as Command Packet Index
	3	Data Length	2
Data Section	1		PDO data length

ECM_CMD_SPI_RECONFIG_OP Reset SPI data packet length			
Command Packet			
Header	1	Command Code	23
	2	Index Code	Custom Defined
	3	Data Length	SPI packet data length(66 ~ 1442)
Return Packet			
Header	1	Command Code	23
	2	Index Code	Same as Command Packet Index
	3	Data Length	0

ECM_CMD_CRC_ERR_CNT_CLR Erase CRC error check count			
Command Packet			
Header	1	Command Code	24
	2	Index	Custom Defined
	3	Data Length	0
Return Packet			

Header	1	Command Code	24
	2	Index	Same as Command Packet Index
	3	Data Length	0
	4	Return Value	Erase the number of CRC error counting

ECM_CMD_402_CONFIG_SET Assign slave control word and status word offset			
Command Packet			
Header	1	Command Code	26
	2	Index	Custom Defined
	3	Data Length	0
	4	Command Parameter	Slave Number
	5	Command Data 0	Control Word Offset(Byte)
	6	Command Data 1	Status Word Offset(Byte)

ECM_CMD_402_STATE_SET Switch assigned slave 402 state			
Command Packet			
Header	1	Command Code	27
	2	Index	Custom Defined
	3	Data Length	0
	4	Parameter	Slave Number
	5	Parametric Data	402 State NOT READY TO SWITCH ON 0x00 SWITCHED ON DISABLED 0x40 READY TO SWITCH ON 0x21 SWITCHED ON 0x23

			OPERATION ENABLED	0x27
			QUICK STOP ACTIVE	0x07
			FAULT REACTION ACTIVE	0x0F
			FAULT	0x08

ECM_CMD_402_STATE_GET				
Read assigned slave 402 state				
Command Packet				
Header	1	Command Code	28	
	2	Index	Custom Defined	
	3	Data Length	0	
	4	Command Parameter	Slave Number	
Return Packet				
Header	1	Command Code	28	
	2	Index	Same as Command Packet Index	
	3	Data Length	0	
	4	Return Value	Slave current 402 State	

ECM_CMD_402_CTL_SET				
Set control bytes of ECM-XF internal 402 state				
Command Packet				
Header	1	Command Code	29	
	2	Index	Custom Defined	
	3	Data Length	0	
	4	Command Parameter	Slave No.	
	5	Parametric Data 0	bit3 : Erase slave state error bit4 : Enable state machine bit7 : Enable automatic erasing slave state machine error	

ECM_CMD_402_CTL_GET Read control bytes of ECM-XF internal 402 state			
Command Packet			
Header	1	Command Code	30
	2	Index	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	30
	2	Index	Same as Command Packet Index
	3	Data Length	0
	4	Return Value	Set value of 402 state machine

ECM_GPIO_CONFIG_SET(ECM_GpioSetMode) Set GPIO mode			
Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	0
	5	Command Data 0	Bit n: 1 for setting GPIO mode of GPIO _n (n =0 to 7)
	6	Command Data 1	Bit n: 1 for setting GPIO mode of GPIO _{n+8} (n =0 to 7)
	7	Command Data 2	GPIO mode definition 0x0: Input Mode 0x1: Output Mode 0x2: Open-Drain Mode 0x3: Quasi-bidirectional Mode

	8	Command Data 3	GPIO pull select definition 0x0: Pull Select Disable Mode 0x1: Pull-up Mode 0x2: Pull-down Mode
--	---	----------------	--

ECM_GPIO_CONFIG_SET(ECM_GpioSetMode) Set GPIO mode (GPIO Extension function, only work for ver. 9 or later)			
Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	8
	5	Command Data 0	Bit n: 1 for setting GPIO mode of GPIO _n +16 (n = 0 to 3)
	6	Command Data 1	Reserved, set 0.
	7	Command Data 2	GPIO mode definition 0x0: Input Mode 0x1: Output Mode 0x2: Open-Drain Mode 0x3: Quasi-bidirectional Mode
	8	Command Data 3	GPIO pull select definition 0x0: Pull Select Disable Mode 0x1: Pull-up Mode 0x2: Pull-down Mode

ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce) Enable/disable GPIO debounce			
Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0

	4	Command Parameter	2
	5	Command Data 0	Bit n: 1 for enable GPIO debounce of GPIO _n , 0 for disable debounce of GPIO _n . (n =0 to 7)
	6	Command Data 1	Bit n: 1 for enable GPIO debounce of GPIO _{n+8} , 0 for disable debounce of GPIO _{n+8} . (n =0 to 7)
	7	Command Data 2	IO direction 0: disable 1: enable
	8	Command Data 3	Reserved

ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce)
Enable GPIO debounce
(GPIO Extension function, only work for ver. 9 or later)

Command Packet

Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	9
	5	Command Data 0	Bit n: 1 for enable GPIO debounce of GPIO _{n+16} ; 0 for disable debounce of GPIO _{n+16} . (n =0 to 3)
	6	Command Data 1	Reserved, set 0.
	7	Command Data 2	IO direction 0: disable 1: enable
	8	Command Data 3	Reserved

ECM_GPIO_CONFIG_SET(ECM_GpioSetDebounceClock) Set debounce clock			
Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	3
	5	Command Data 0	<p>Bit 0 to 3: Debounce clock cycle. The valid value is 0 to 15.</p> <p>Bit 4: Clock source 0: 192M 1: 10K</p> <p>The sampling cycle is $2^{(Debounce\ clock\ cycle)} * (clock\ period)$</p>

ECM_GPIO_CONFIG_SET(ECM_GpioIntEnable) Enable/Disable GPIO interrupt			
Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	1
	5	Command Data 0	Channels From 0 to 20
	6	Command Data 1	<p>Interrupt type</p> <p>0x0: Disable Interrupt 0x1: Interrupt enable by Rising Edge or Falling Edge 0x2: Interrupt enable by Input Falling Edge 0x3: Interrupt enable by Input Rising Edge</p>

ECM_GPIO_CONFIG_SET(ECM_GpioIntClear) Clear GPIO interrupt flag			
--	--	--	--

Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	4
	5	Command Data 0	Bit n: clear GPIO _n interrupt flag (n =0 to 7)
	6	Command Data 1	Bit n: clear GPIO _{n+8} interrupt flag (n =0 to 7)

ECM_GPIO_CONFIG_SET(ECM_GpioIntClear)
Clear GPIO interrupt flag
(GPIO Extension function, only work for ver. 9 or later)

Command Packet			
Header	1	Command Code	31
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	6
	5	Command Data 0	Bit n: clear GPIO _n interrupt flag (n =0 to 7)
	6	Command Data 1	Bit n: clear GPIO _{n+8} interrupt flag (n =0 to 7)
	7	Command Data 2	Bit n: clear GPIO _{n+16} interrupt flag (n =0 to 3)

ECM_GPIO_FUNC_OP(ECM_GpioSetValue)
Set GPIO value

Command Packet			
Header	1	Command	32

		Code	
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	0
	5	Command Data 0	Bit n: 1 for setting GPIO _n to high, 0 for setting GPIO _n to low. (n =0 to 7)
	6	Command Data 1	Bit n: 1 for setting GPIO _{n+8} to high, 0 for setting GPIO _{n+8} to low. (n =0 to 7)

ECM_GPIO_FUNC_OP(ECM_GpioExtSetValue)
Set GPIO value
(Extension function, only work for ver. 9 or later)

Command Packet

Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	2
	5	Command Data 0	Bit n: 1 for setting GPIO _n to high, 0 for setting GPIO _n to low. (n =0 to 7)
	6	Command Data 1	Bit n: 1 for setting GPIO _{n+8} to high, 0 for setting GPIO _{n+8} to low. (n =0 to 7)
	7	Command Data 2	Bit n: 1 for setting GPIO _{n+16} to high, 0 for setting GPIO _{n+16} to low. (n =0 to 3)

ECM_GPIO_FUNC_OP(ECM_GpioGetValue)
Get GPIO value

Command Packet

Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	1
Return Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	2
Data Section			Bit n: 1 if GPIO _n is in input mode and the input is high; 0 if GPIO _n is in input mode and the input is low. (n =0 to 15)

ECM_GPIO_FUNC_OP(ECM_GpioExtGetValue) Get GPIO value (Extension function, only work for ver. 9 or later)			
Command Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	3
Return Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	4
Data Section			Bit n: 1 if GPIO _n is in input mode and the input is high; 0 if GPIO _n is in input mode and the input is low. (n =0 to 19)

ECM_GPIO_FUNC_OP(ECM_GpioGetIntFlag) Get GPIO Interrupt flag			
Command Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	5
Return Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	2
Data Section			Bit n: GPIO n interrupt flag (n = 0 to 15)

ECM_GPIO_FUNC_OP(ECM_GpioExtGetIntFlag) Get GPIO Interrupt flag (Extension function, only work for ver. 9 or later)			
Command Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	7
Return Packet			
Header	1	Command Code	32
	2	Index Code	Custom Defined

	3	Data Length	4
Data Section			Bit n: GPIO interrupt flag (n = 0 to 19)

ECM_QEI_FUNC_OP(ECM_EncOpen) Open and set encoder mode			
Command Packet			
Header	1	Command Code	33
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	4
Data Section			Mode 0x000: X4 free-counting Mode 0x100: X2 free-counting Mode 0x200: X4 compare-counting Mode 0x300: X2 compare-counting Mode

ECM_QEI_FUNC_OP(ECM_EncStart) Start encoder counting			
Command Packet			
Header	1	Command Code	33
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	5

ECM_QEI_FUNC_OP(ECM_EncStop) Stop encoder counting			
Command Packet			
Header	1	Command Code	33

	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	6

ECM_QEI_FUNC_OP(ECM_EncGetCount) Get encoder counter value			
Command Packet			
Header	1	Command Code	33
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	21
Return Packet			
Header	1	Command Code	33
	2	Index Code	Custom Defined
	3	Data Length	4
Data Section			Encoder data

ECM_DAC_FUNC_OP(ECM_DacOpen) Open DAC function			
Command Packet			
Header	1	Command Code	34
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	20
Data	5	Data	Mode

Section			<p>0x0000: Use ECM_DacSetData() function to trigger DAC output. 0x0010: Use ECM_DacStartConv() function to trigger DAC output. 0x0030: DAC is triggered if DAC_ST pin is low. 0x1030: DAC is triggered if DAC_ST pin is high. 0x2030: DAC is triggered if DAC_ST pin detect a rising edge. 0x3030: DAC is triggered if DAC_ST pin detect a falling edge.</p>
---------	--	--	---

ECM_DAC_FUNC_OP(ECM_DacClose) Close DAC function			
Command Packet			
Header	1	Command Code	34
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	21

ECM_DAC_FUNC_OP(ECM_DacSetDelayTime) Set DAC delay time			
Command Packet			
Header	1	Command Code	34
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	22
Data Section		Data	Delay time in microsecond Range 0 to 1023/96

ECM_DAC_FUNC_OP(ECM_DacSetData) Set DAC data			
Command Packet			
Header	1	Command Code	34

	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	11
Data Section		Data	Data value: 0 ~ 4095 corresponding to 0 ~ 3.3V

ECM_DAC_FUNC_OP(ECM_DacStartConv) Start DAC conversion			
Command Packet			
Header	1	Command Code	34
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	0

ECM_ADC_FUNC_OP(ECM_AdcOpen) Open ADC function			
Command Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	40
Data Section		Data	0x000: single-end mode 0x100: differential mode

ECM_ADC_FUNC_OP(ECM_AdcClose) Close ADC function			
Command Packet			

Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	41

ECM_ADC_FUNC_OP(ECM_AdcConfigSampleModule) Configure ADC trigger source																							
Command Packet																							
Header	1	Command Code	35																				
	2	Index Code	Custom Defined																				
	3	Data Length	12																				
	4	Command Parameter	42																				
Data Section		Data	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td colspan="4" style="text-align: center;">Reserved, set 0</td> <td style="text-align: right;">0x0</td> </tr> <tr> <td colspan="4" style="text-align: center;">0: software trigger</td> <td style="text-align: right;">0x4</td> </tr> <tr> <td colspan="4" style="text-align: center;">Reserved, set 0</td> <td style="text-align: right;">0x8</td> </tr> </table>	3	2	1	0		Reserved, set 0				0x0	0: software trigger				0x4	Reserved, set 0				0x8
3	2	1	0																				
Reserved, set 0				0x0																			
0: software trigger				0x4																			
Reserved, set 0				0x8																			

ECM_ADC_FUNC_OP(ECM_AdcGetDataValidFlag) Get ADC data valid flag			
Command Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	17

Data Section		Data	Reserved, set 0x1000.
Return Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
Data Section			Valid flag. The convert data is valid if and only if the valid flag is non-zero.

ECM_ADC_FUNC_OP(ECM_AdcStartConv) Start ADC conversion			
Command Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	12
Data Section			Reserved, set 0x1000.

ECM_ADC_FUNC_OP(ECM_AdcGetConvData) Get ADC converted data			
Command Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
	4	Command Parameter	15
Data Section			Reserved, set 12.

Return Packet			
Header	1	Command Code	35
	2	Index Code	Custom Defined
	3	Data Length	4
Data Section			Convert data. 0 ~ 4095 corresponding to 0 ~ 3.3V

ECM_EEPROM_REQ																				
Request execute EEPROM write/read order Use ECM_EEPROM_GET to read return data from ECM_EEPROM_REQ																				
Command Packet																				
Header	1	Command Code	38																	
	2	Index	Custom Defined																	
	3	Data Length	12																	
Data Section	1	<table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 25%;">3</td> <td style="width: 25%;">2</td> <td style="width: 25%;">1</td> <td style="width: 25%;">0</td> </tr> <tr> <td colspan="2">Slave Number</td> <td colspan="2">Operation Code</td> </tr> <tr> <td colspan="2">Write Data</td> <td colspan="2">Write/Read Location</td> </tr> <tr> <td colspan="4">Timeout</td> </tr> </table>		3	2	1	0	Slave Number		Operation Code		Write Data		Write/Read Location		Timeout				0
		3	2	1	0															
		Slave Number		Operation Code																
		Write Data		Write/Read Location																
		Timeout																		
						4														
						8														
		Location	Length	Name	Description															
0x00	2	Operation Code	0 : Read 1 : Write																	
0x02	2	Slave Number	0~127																	
0x04	2	Write/Read Location	EEPROM Location																	
0x06	2	Write Data	Valid Write Operation																	
0x08	4	Timeout	Maximum Timeout(ns)																	

ECM_EEPROM_GET Read result from ECM_EEPROM_REQ command			
Command Packet			
Header	1	Command Code	39
	2	Index	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	39
	2	Index	Same as Command Packet Index
	3	Data Length	Valid Data Section Length
	4	Return Value	Operation Error Code
Data Section	1	Get ECM_EEPROM_REQ data back	

ECM_CMD_ECAT_STATE_CHECK Reflash and check slave state			
Command Packet			
Header	1	Command Code	41
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	Slave Location
	5	Command Data 0	EtherCAT State

ECM_CMD_ECAT_DCSYNC Set DC sync signal source, cycle time, and shift time			
Command Packet			

Header	1	Command Code	50																											
	2	Index Code	Custom Defined																											
	3	Data Length	16																											
	4	Command Parameter	<table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 25%;">3</td> <td style="width: 25%;">2</td> <td style="width: 25%;">1</td> <td style="width: 25%;">0</td> <td></td> </tr> <tr> <td>DC activate code</td> <td>Reserved</td> <td>Slave Index</td> <td></td> <td>0</td> </tr> <tr> <td colspan="4">SYNCO</td> <td>4</td> </tr> <tr> <td colspan="4"></td> <td>8</td> </tr> <tr> <td colspan="4"></td> <td>12</td> </tr> </table>				3	2	1	0		DC activate code	Reserved	Slave Index		0	SYNCO				4					8				
3	2	1	0																											
DC activate code	Reserved	Slave Index		0																										
SYNCO				4																										
				8																										
				12																										

Return Packet				
Header	1	Command Code	50	
	2	Index Code	Same as Command Packet Index	
	3	Data Length	0	

ECM_CMD_FIFO_CLR_OP Clear FIFO				
Command Packet				
Header	1	Command Code	51	
	2	Index Code	Custom Defined	
	3	Data Length	0	

ECM_CMD_FIFO_SET_TX_CNT Set Tx FIFO count				
Command Packet				
Header	1	Command Code	52	

	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	Tx FIFO count

ECM_CMD_FIFO_GET_TX_CNT Get Tx FIFO count			
Command Packet			
Header	1	Command Code	53
	2	Index Code	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	53
	2	Index	Custom Defined
	3	Data Length	0
	4	Return	Tx FIFO count

ECM_CMD_FIFO_SET_RX_CNT Set Rx FIFO count			
Command Packet			
Header	1	Command Code	54
	2	Index Code	Custom Defined
	3	Data Length	0
	4	Command Parameter	Rx FIFO count

ECM_CMD_FIFO_GET_RX_CNT Get Tx FIFO count			
Command Packet			

Header	1	Command Code	55
	2	Index Code	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	55
	2	Index	Custom Defined
	3	Data Length	0
	4	Return	Rx FIFO count

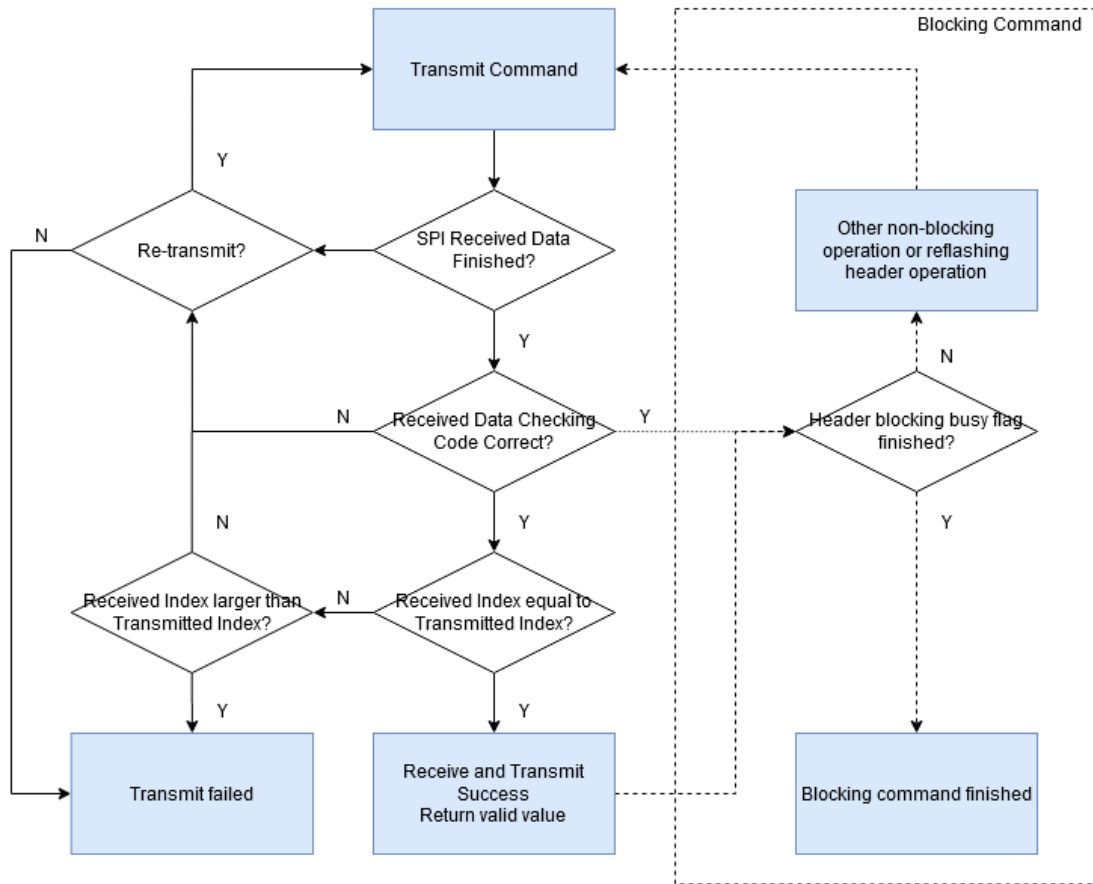
ECM_CMD_FW_VERSION_GET Read firmware version			
Command Packet			
Header	1	Command Code	56
	2	Index	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	56
	2	Index	Same as Command Packet Index
	3	Data Length	0
	4	Return Value	Firmware Version

ECM_CMD_ECAT_INT_SET_ENABLE Set interrupt enable mask			
Command Code			
Header	1	Command Code	58
	2	Index	Custom Defined

	3	Data Length	4
	4	Data Parameter	0 : Low active 1 : High active
Data Section	1	Bit 31 : Enable receive EtherCAT package interrupt Bit 29 : Enable encode interrupt Bit 0-28, 30 : Reserve	

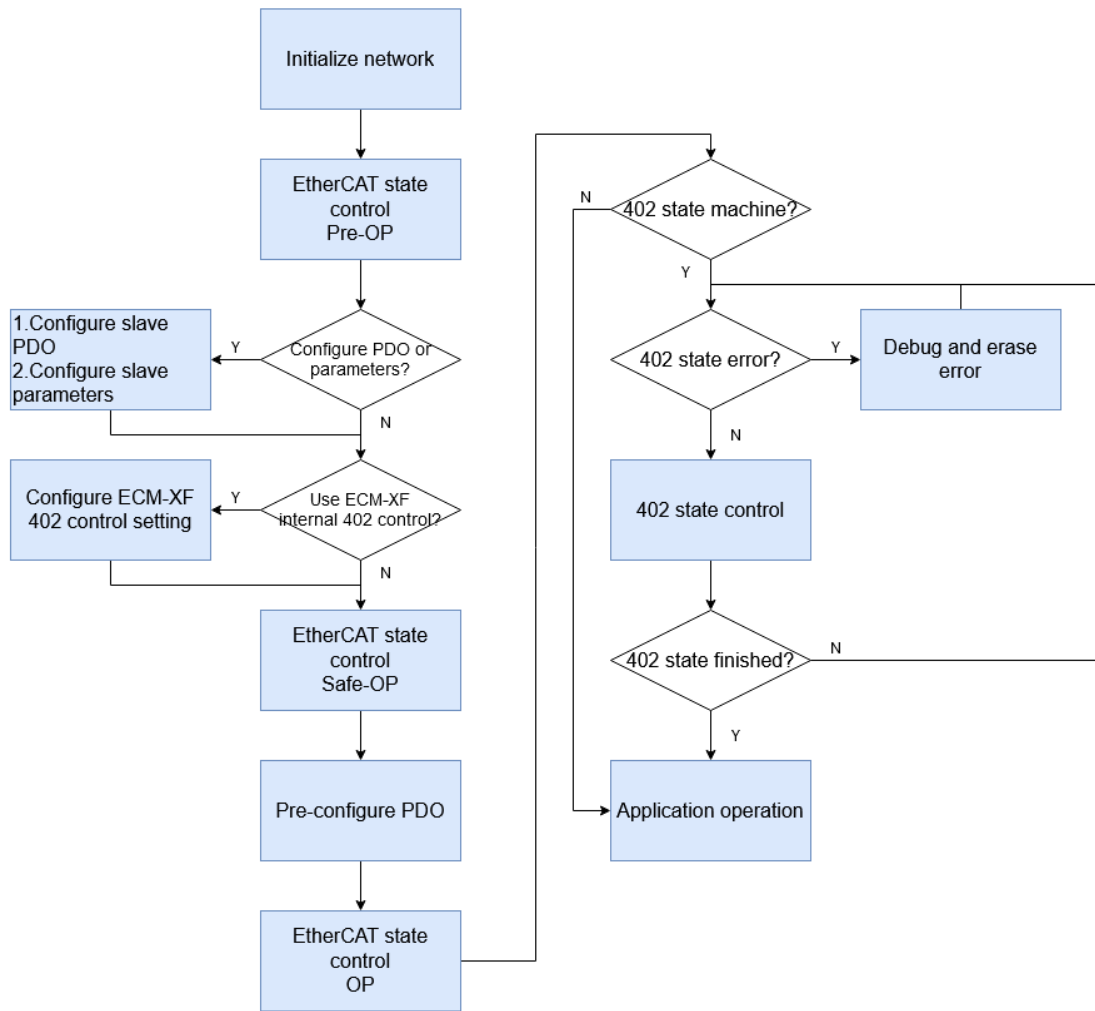
ECM_CMD_ECAT_INT_GET_ENABLE Get interrupt enable mask			
Command Packet			
Header	1	Command Code	59
	2	Index	Custom Defined
	3	Data Length	0
Return Packet			
Header	1	Command Code	59
	2	Index	Same as Command Packet Index
	3	Data Length	4
Data Section	1	Bit 31 : Receive EtherCAT package interrupt enable bit Bit 29 : Encode interrupt enable bit Bit 0-28, 30 : Reserve	

ECM-XF Command and Return Access Diagram



Pic 6 Command and Return access diagram

ECM-XF EtherCAT Network Initialization and Slave Configuration



Pic 7. EtherCAT network initialization and configuration diagram

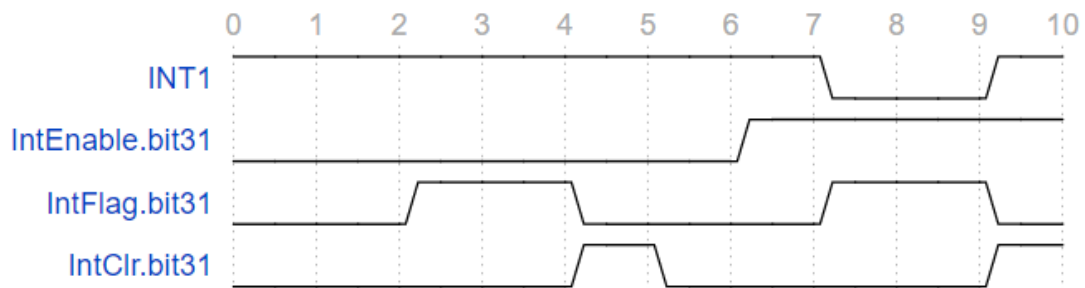
State	Relative Commands	Description
Initialize	ECM_CMD_ECAT_INIT_OP	Set DC Mode and Cycle

Network		Time
ECAT State Control	ECM_CMD_ECAT_STATE_SET ECM_CMD_ECAT_STATE_GET	Switch to EtherCAT State
Configure slave PDO	ECM_CMD_ECAT_PDO_CONFIG_SET ECM_CMD_ECAT_PDO_CONFIG_REQ ECM_CMD_ECAT_PDO_CONFIG_GET ECM_CMD_ECAT_RECONFIG_OP	Assign or configure PDO
Configure slave parameters	ECM_CMD_ECAT_SDO_REQ ECM_CMD_ECAT_SDO_GET	Use SDO to configure parameters like: control mode, torque limit, etc.
Configure 402 State	ECM_CMD_402_CONFIG_SET	Enable internal 402 control Configure erasing error
402 State Control	ECM_CMD_402_STATE_SET ECM_CMD_402_STATE_GET	Switch 402 state
Preconfigure PDO	ECM_CMD_FIFO_PACK_SIZE_GET ECM_CMD_ECAT_PDO_WC_GET ECM_CMD_ECAT_PDO_DATA_OP	Get PDO Data Size Get PDO communication WKC value Read TxPDO input Initialize RxPDO output
Application Operation	ECM_CMD_ECAT_PDO_DATA_FIFO_OP ECM_CMD_ECAT_SDO_REQ ECM_CMD_ECAT_SDO_GET ECM_CMD_INFO_UPDATE_OP	PDO data exchange by FIFO Use SDO write/read data Reflash state

ECM-XF Interrupts

ECM-XF has two interrupt outputs. The INTO pin is high if SPI is ready. Not to transmit/receive SPI if the INTO pin is in low state.

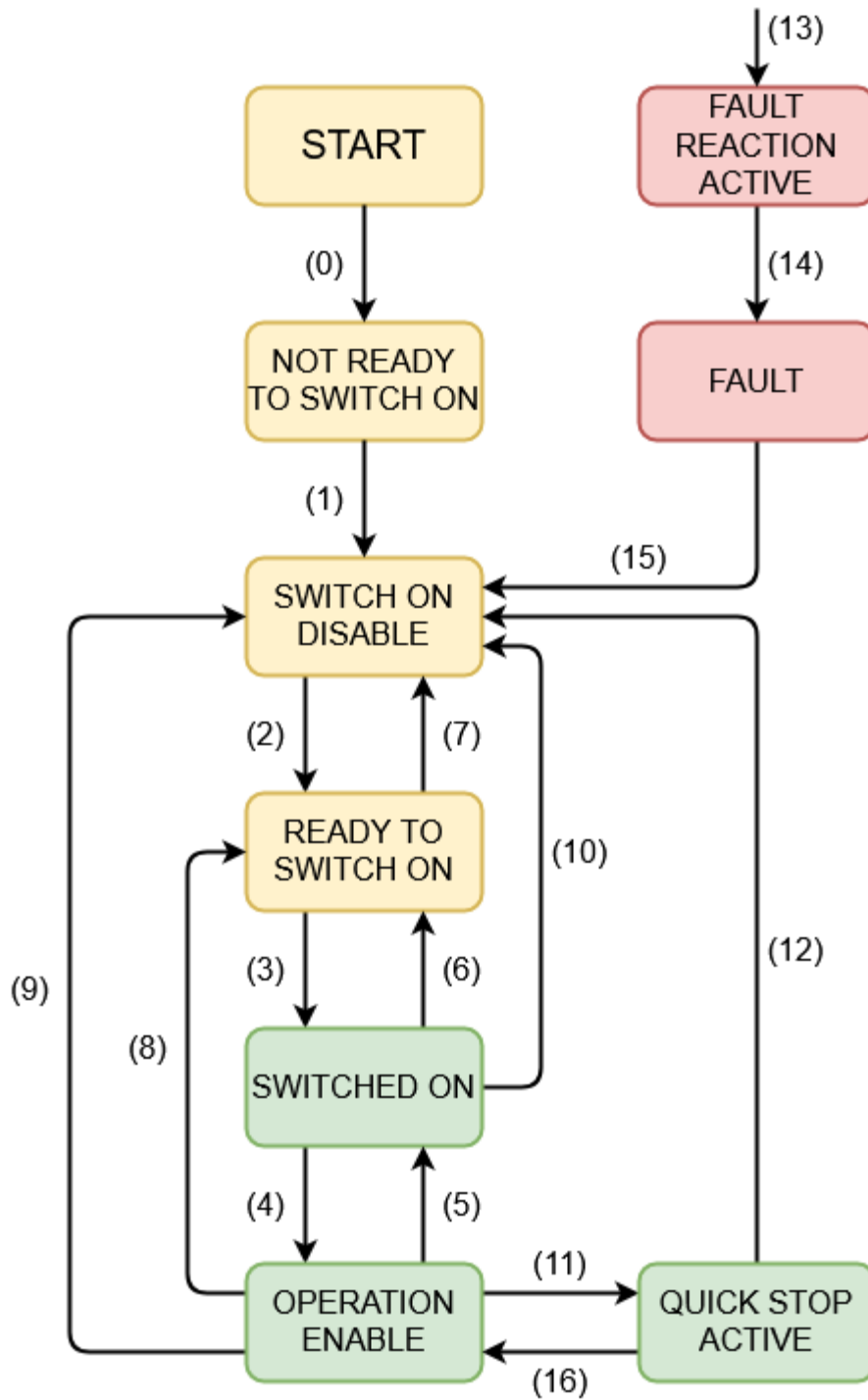
INT1 is a general interrupt function pin. If an interrupt enable bit is set and an interrupt event occurs, the INT1 interrupt is active. Pic 8 shows a scenario that EtherCAT receive events occur in t=2 and t=7; interrupt mode is low active.



Pic 8. An interrupt example.

EtherCAT receive events occur in t=2 and t=7. The bit 31 of IntFlag is set in t=2 and t=7, and is reset if the bit 31 of IntClr is set by SPI master. INT1 is active only if both IntFlag and IntEnable are set. In this case, INT1 is active only in t=7.

Appendix 1. CiA 402 State Machine



Get CiA402 state from transmitting status word(0x0641:0)

Index	Sub	Name	Data Type	Access	PDO Mapping	Default Value
0x6041	00	Status	UINT	RO	Y	*(See

		Word				below)
--	--	-------------	--	--	--	---------------

Bit 0 to 3 and bit 5 to 6: for the current state of the drive

Command	bit 6	bit 5	bit 3	bit 2	bit 1	bit 0
Not ready to switch on	0	0	0	0	0	0
Switch on disabled	1	0	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Quick stop active	0	0	0	1	1	1
Fault reaction active	0	0	1	1	1	1
Fault	0	0	1	0	0	0

The following table indicates which functionalities can be activated on every state. External brake can only applied if it is present, and high-level power applied is only selectable in controllers with an embedded contactor/switch for the power stage.

Function	Not Ready to Switch On	Switch On Disable	Ready to Switch On	Switch On	Operation Enabled	Quick Stop Active	Fault Reaction Active	Fault
Brake applied, if present	Yes	Yes	Yes	Yes	Yes/No(*)	Yes/No(*)	Yes/No(*)	Yes

Low-level power applied	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High-level power applied	Yes/No(**)	Yes/No(**)	Yes/No(**)	Yes	Yes	Yes	Yes	Yes/No(**)
Drive function enabled	No	No	No	No	Yes	Yes	Yes	No
Configuration allowed	Yes	Yes	Yes	Yes	No	No	No	Yes
Shunt control enabled	No	No	No	No	Yes	Yes	Yes	No

(*)In some states, brake could be activated and/or deactivated manually.

(**)In some controllers, high-level switch is not present and therefore high level power could not be deactivated.

Transition	Event	Action
0	Automatic transition after power-on or reset application.	Drive device self-test and/or self initialization is performed.
1	Automatic transition after initialization.	Communications are activated.
2	Shutdown command received from control device or local signal.	-

3	Switch on command received from control device and enable signal is activated (if available) or local signal.	The high-level power is switched on.
4	Enable operation command received from control device and enable signal is activated (if available) or local signal.	The drive function is enabled, Initial angle determination process is executed and all internal set-points cleared.
5	Disable operation command received from control device or enable signal is deactivated (if available) or local signal.	The drive function is disabled.
6	Shutdown command received from control device or enable signal is deactivated (if available) or local signal.	The high-level power is switched off.
7	Quick stop or disable voltage command received from control device or local signal.	-
8	Shutdown command received from control device or enable signal is deactivated (if available).	The drive function is disabled, and the high-level power is switched off.

9	Disable voltage command from control device or local signal.	The drive function is disabled, and the high-level power is switched off.
10	Disable voltage or quick stop command received from control device or local signal.	The high-level power is switched off.
11	Quick stop command received from control device or local signal.	The quick stop function is started.
12	Automatic transition when the quick stop function is completed or disable voltage command is received from control device (depending on 0x605A - Quick stop option code).	The drive function is disabled, and the high-level power is switched off.
13	Fault signal.	The configured fault reaction function is executed.
14	Automatic transition.	The drive function is disabled, and the high-level power is switched off.
15	Fault reset command received from control device or local signal.	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the Fault state, the Fault reset bit in the control word is cleared by the control device.

16	Enable operation command from control device.	The drive function is enabled.
----	---	--------------------------------

Caution:

※When drive function is disabled, no energy will be supplied to the motor. Target or set-point(torque, velocity, position) in that situation are not processed.

※High-level power is switched off only in systems with contractors or switches for this purpose.

※Enable signal will affect only if it is marked as available in the corresponding register. See Enable/Disable input for further information.

Update Log

Version	Update date	Description
022	10.14.2020	Update content copy from Chinese ver. 022
023	10.14.2020	Update interrupt
03	10.14.2020	Update ECM-XF ADC, DAC, GPIO, QEI, and FIFO clear function, resequence all function description
031	10.15.2020	Update set/get Rx/Tx FIFO count