ECM-XF

EtherCAT Master Chip
User Manual

NEXTW Technology Co., Ltd.

`

2024/05/22
Ver.050

Introduction

ECM-XF is a cost-effective EtherCAT master chip that connects to the host through the SPI interface to help users achieve EtherCAT communication, achieving a minimum DC cycle time of 125us and the ability to connect up to 128 slaves. It has a variety of peripheral IO, interrupt and application functions, and is suitable for PLC controllers, robot controllers and various automation controllers.
ECM-XF can upgrade its firmware via USB. **If you want to use USB to control EtherCAT slave devices, you need to use ECM-XFU chip instead.**

Support DC function	Yes
Maximum number of slaves	128
Minimum cycle time	125us
Industrial Ethernet	EtherCAT
Mailbox protocol	CoE / FoE
Peripheral IO	GPIO / QEI / ADC / DAC
App Features	CiA402 state machine control FIFO Buffer

interface

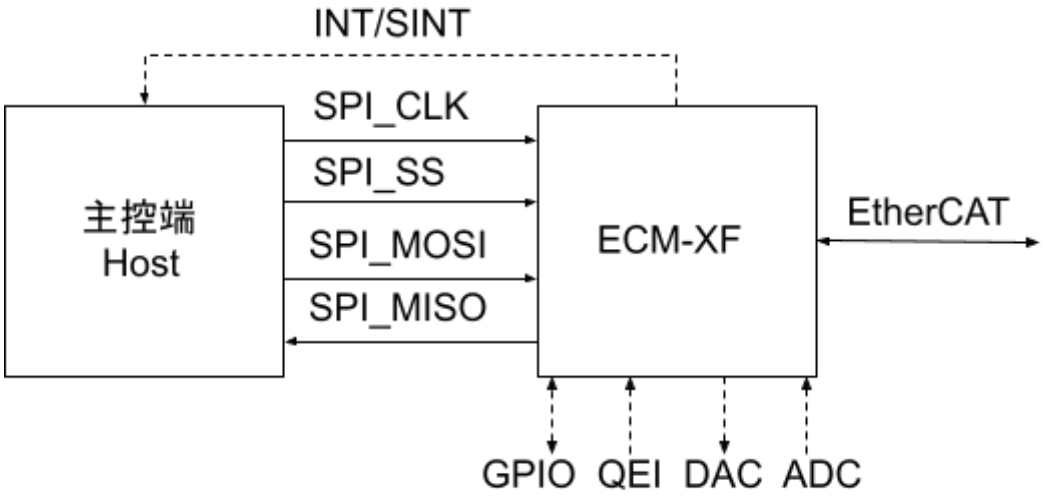


Figure 1. ECM-XF interface connection diagram

*** PHY Address needs to be set to 1**

SPI Mode

name	Foot position	significance	illustrate
SPI_CLK	Pin38	Frequency signal	Generated and controlled by the SPI Master, there is a minimum frequency requirement for controlling the EtherCAT communication cycle
SPI_MOSI	Pin40	Master out, slave in	SPI Master data output, SPI Slave data receiving
SPI_MISO	Pin39	Master in, slave out	SPI Master data receiving, SPI Slave data output
/SPI_SS	Pin37	Chip Enable	The selection signal is controlled by the Master. The Slave will only respond to the Master's operation instructions when the SPI_SS signal is low.

<table>
<tr><td>Duplex Mode</td><td colspan="3">Full duplex</td></tr>
<tr><td>Transfer rate</td><td colspan="3">96Mbps(max.)</td></tr>
<tr><td rowspan="3">Timing Mode</td><td colspan="3">SPI_CLK idle low, rising edge latches data, falling edge sends data (Mode 0)</td></tr>
<tr><td>POL=0</td><td colspan="2">PHA=0</td></tr>
<tr><td>POL=0</td><td>TXNEG=1</td><td>RXNEG=0</td></tr>
</table>

SPI Mode Transmission Description

The SPI of ECM is in Slave mode. It is low voltage when idle. It sends on the falling edge and receives on the rising edge. The high-order data is transmitted first (MSB). Please refer to the following figure for explanation.

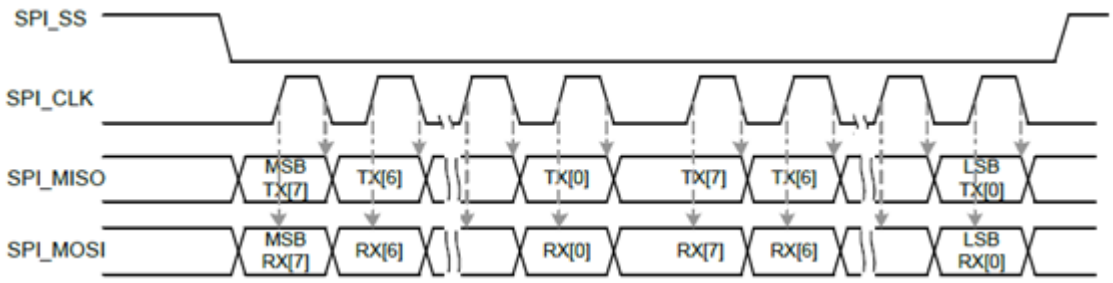


Figure SPI Timing Diagram

The SPI on the upper controller side is in Master mode, which needs to generate CLK and provide it to the SPI Slave, and send it on the rising edge and receive it on the falling edge.SPI transmission is in Byte units, and each transmission starts from the low address and goes to the highest address in sequence. In other words, SPI transmission starts from Byte0, and then transmits Byte1, Byte2, etc. in sequence until the last Byte. When SPI transmits a single Byte, it adopts MSB mode, that is, the high bit is transmitted first.

SPI command transmission and response reception instructions

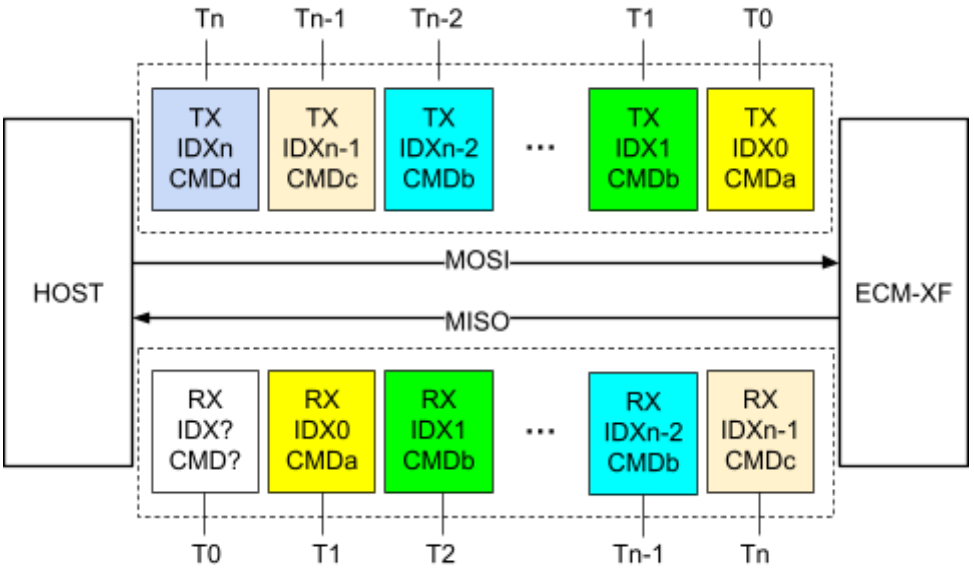


Figure 2: SPI full-duplex data exchange packet sequencing diagram

Notice
1. After the host completes sending the command packet via SPI, it will receive the response packet of the previous command
2. After ECM-XF is started, the first response packet sent back to the host does not contain a command response because there is no previous command.
3. If the index numbers and command codes of consecutive command packets are

the same, except for the first command ECM-XF, which will be considered a valid command, the remaining commands will be considered ECM_CMD_INFO_UPDATE_OP commands.

Packet format

1. The packet consists of four parts:**Header, data segment, checksum, end-of-packet word**
2. The command header contains the command, parameter information, system-related controls, etc.
3. The response header includes the command return value, return data, error status, system status, etc.
4. The data segment content is command related data or PDO data. It can be configured with sufficient size according to actual needs.
5. There are two formats for the check word to choose from (Fixed value 0x12345678 check、CRC-32)
6. The packet end word is a fixed value (0x56575859) and is used to identify packet integrity.
7. Data segment length32bytes~1408bytes, the default value is 112 bytes

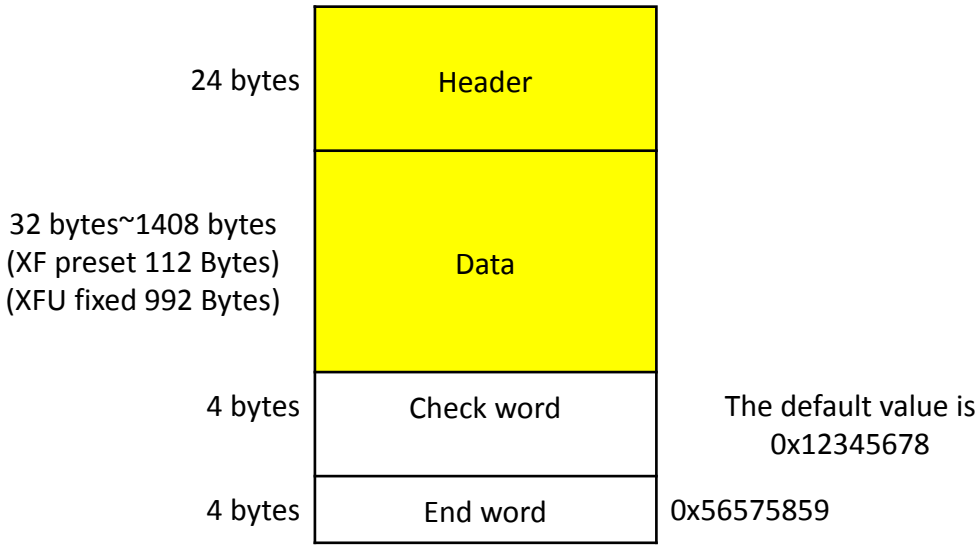


Figure 3 SPI packet format

Header

Command Header: Header of the ECM-XF packet sent by the host

3	2	1	0	
Packet start word 0xA1A2A3A4				0x00
Command information 3	Command information 2	Command information 1	Command information 0	0x04
Index Number	Control Characters	reserve		0x08
Valid data segment length		Command Parameters	**Command code**	0x0c
Interrupt clear				0x10
GPIO output 1	GPIO output 0	GPIO interrupt clear 1	GPIO interrupt clear 0	0x14

Figure 4 Command header

Command code association

Interrupt related

System or hardware related

Response Header: The header of the packet sent back by ECM-XF to the host

3	2	1	0	
Packet start word 0xA1A2A3A4				0x00
RxPDO number to be transmitted	TxPDO number to be received	WKC Error Count	CRC error count	0x04

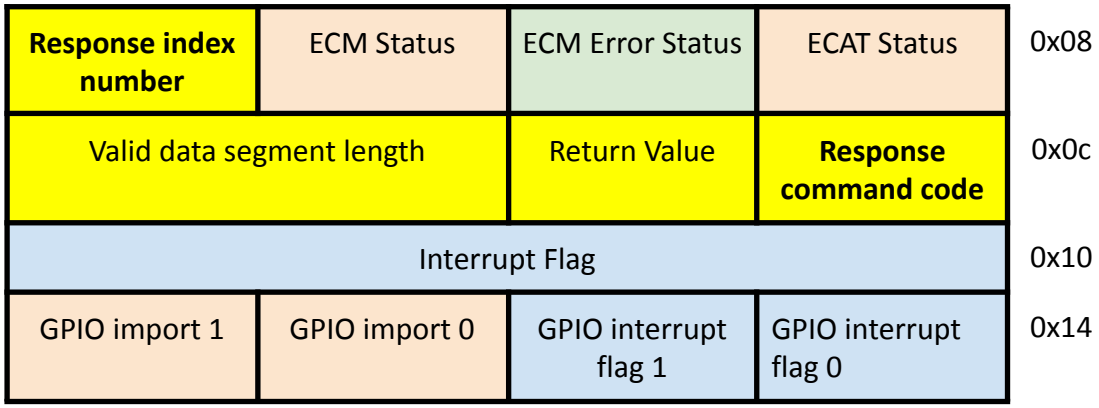
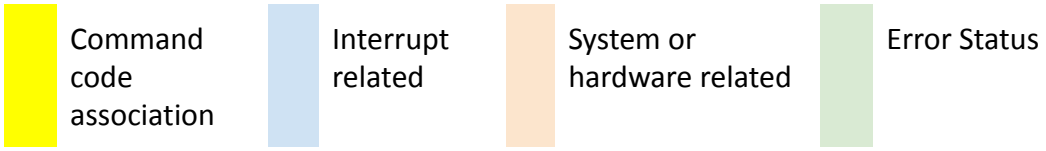


Figure 5 Response header



Header Field Description

Table 1 Command Packet Header Field List

Command Header			
Location	length	name	illustrate
0x00	4	Packet start word	Fixed value 0xA1A2A3A4
0x04	1	Command information 0	Passing parameters or data related to the command
0x05	1	Command information 1	Passing parameters or data related to the command
0x06	1	Command information 2	Passing parameters or data related to the command
0x07	1	Command information 3	Passing parameters or data related to the command
0x08	2	reserve	Bit 15: 1 indicates that the GPIO output 0 (0x16) and GPIO output 1 (0x17) of the header file are valid

0x0A	1	Control Characters	Control or clear system errors bit0: header IO output field is valid bit1: Update ECAT status from slave (non-immediate command) bit3:Non-immediate command error clearing bit4: SPI communication error clear bit5: FIFO error clear bit6: SPI CRC error clear bit7: command error clear
0x0B	1	Index Number	Mainly used for identification
0x0C	1	Command code	Please refer to the command list
0x0D	1	Command parameters	Command code related parameters
0x0E	2	Valid data segment length	Command code related parameters or effective data segment length
0x10	4	Interrupt clear	Bit 31: EtherCAT data packet interrupt clear Bit 29: Encoder interrupt clear Bit 0 ~ 28, 30: Reserved
0x14	1	GPIO interrupt clear 0	Clear GPIO low bit interrupt flag
0x15	1	GPIO interrupt clear 1	Clear GPIO high interrupt flag
0x16	1	GPIO output 0	GPIO low bit output value (GPIO 0~7) (0x08 bit15 must be 1)
0x17	1	GPIO output 1	GPIO high bit output value (GPIO 8~11) (0x08 bit15 must be 1)

- To control GPIO 12 to 15, please use CMD ID 32 (ECM_GPIO_FUNC_OP)
- Version 0x1E and later can be controlledbit12~15, but bit12 controls GPIO13, bit13 controls GPIO12

Table 2 Response packet header field list

Response Headers			
Location	length	name	illustrate

0x00	4	Packet start word	Fixed value 0xA1A2A3A4
0x04	1	CRC error count	SPI communication CRC error times
0x05	1	WKC Error Count	PDO communication working count number of errors
0x06	1	TxPdoFIFO number	The number of TxPDOs not yet read in FIFO
0x07	1	RxPdoFifo number	Number of RxPDOs not yet transmitted in FIFO
0x08	1	ECAT Status	0x00 : NONE (uninitialized state) 0x01: INIT (initialization state) 0x02 : PRE_OP (pre-operation state) 0x03: BOOT (boot status) 0x04 : SAFE_OP (safe operation state) 0x08 : OPERATIONAL (Operational status) 0x10 : ERROR (error status)
0x09	1	Error/Status	bit0: Check the connection status between the network port and ECM-XF PIN 52 LINK (When connected, the value is 1 or 0 depending on the hardware circuit design) bit1: Software confirms PHY connection status during Ecat_Init (When connected, the value is 1 or 0 depending on the hardware circuit design) bit2: reserved bit3: Non-immediate command returns an error bit4: SPI communication error bit5: FIFO error bit6: SPI CRC error bit7:Command Error Note: Bit 7 is 1, indicating a command error, including an error in the command parameters or a new non-immediate command received while the non-immediate command is busy.
0x0A	1	ECM Status	bit1-bit0: 0: Fixed value check 1: Undefined 2: Undefined 3: CRC32 check bit2: DC cycle time stability flag bit3: Network port line initialization completed flag bit4: ECAT PDO configuration completed flag bit5: The response packet is NOP flag

			bit6: FIFO start flag bit7: Non-immediate command busy flag
0x0B	1	Index Number	Response command header index number
0x0C	1	Command code	Response command header command code
0x0D	1	Command return value	Echo the command execution return value
0x0E	2	Data segment length	Valid data length in the data segment
0x10	4	Interrupt Flag	Bit 31: EtherCAT data packet interrupt flag Bit 29: Encoder interrupt flag Bit 0 ~ 28, 30: Reserved
0x14	1	GPIO interrupt flag 0	GPIO low bit interrupt flag
0x15	1	GPIO interrupt flag 1	GPIO high bit interrupt flag
0x16	1	GPIO import 0	GPIO low bit input value
0x17	1	GPIO import 1	GPIO high bit input value

Command code list

Commands are divided into two categories according to response time.
1. Immediate command (NON-BLOCKING) - After receiving the command, the operation is completed and returned immediately
2. Non-immediate command (BLOCKING) - After receiving the command, it takes a certain amount of time to complete and return
3. ECM-XF can only execute one non-immediate command at a time Therefore, before using non-immediate command operations, you need to make sure that the ECM status non-immediate command busy flag is 0 before sending non-immediate commands.
4. Non-immediate command operation can be performed simultaneously with immediate command or other application operations
Please refer to Figure 6 for the command and response receiving and sending flow chart

Table 3 Command code list

Non-immediate commands	Immediate Commands
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 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 List

ECM_CMD_INFO_UPDATE_OP	Update header information
ECM_CMD_ECAT_INIT_OP	Initialize the EtherCAT network and slaves. You must initialize the EtherCAT network before using the EtherCAT function.
ECM_CMD_ECAT_RECONFIG_OP	After re-planning PDO, this command is required to make ECM-XF reconfigure the memory space.
ECM_CMD_ECAT_PDO_WC_GET	ReadPDOInImport informationNumber of slave stations,Export InformationNumber of slaves or currentWorking countervalue
ECM_CMD_ECAT_PDO_DATA_FIFO_OP	Access PDO data to FIFO
ECM_CMD_ECAT_PDO_DATA_OP	Read PDO data
ECM_CMD_ECAT_PDO_CONFIG_SET	Configure slave PDO
ECM_CMD_ECAT_PDO_CONFIG_REQ	Request to execute PDO configuration read. After the operation is completed, use ECM_CMD_ECAT_PDO_CONFIG_GET to retrieve the configuration
ECM_CMD_ECAT_PDO_CONFIG_GET	Read slave PDO configuration
ECM_CMD_ECAT_SDO_REQ	Request to execute SDO read/write command. The data segment of the write operation instruction is the write data. After the read operation is completed, the read data is retrieved through the ECM_CMD_ECAT_SDO_GET instruction.
ECM_CMD_ECAT_SDO_GET	Read back the ECM_CMD_ECAT_SDO_REQ read operation result
ECM_CMD_ECAT_STATE_SET	Changing EtherCAT Status
ECM_CMD_ECAT_STATE_GET	Retrieve EtherCAT status
ECM_CMD_ECAT_SLV_INFO_GET	Read slave information
ECM_CMD_ECAT_SLV_CNT_GET	Read the number of slaves
ECM_CMD_FIFO_ENABLE	Enable RxPDO FIFO output. ECM-XF enables RxPDO FIFO output by default. Users can use this command to disable RxPDO FIFO periodic output, which can be used for pre-storage output.

ECM_CMD_FIFO_PACK_SIZE_GET	Read the length of PDO data. One data in FIFO is one PDO.
ECM_CMD_SPI_PACK_SIZE_GET	Read SPI data length
ECM_CMD_SPI_RECONFIG_OP	Reset SPI data length
ECM_CMD_CRC_ERR_CNT_CLR	Clear the accumulated CRC check errors
ECM_CMD_CRC_TYPE_SET	Set the checksum type
ECM_CMD_402_CONFIG_SET	Specify the slave control word and status word offset
ECM_CMD_402_STATE_SET	Switch the specified slave 402 status
ECM_CMD_402_STATE_GET	Read the specified slave 402 status
ECM_CMD_402_CTL_SET	ECM-XF internal 402 state machine control byte setting
ECM_CMD_402_CTL_GET	ECM-XF internal 402 state machine control byte read
ECM_GPIO_CONFIG_SET (ECM_GpioSetMode)	Set GPIO mode
ECM_GPIO_CONFIG_SET (ECM_GpioEnableDebounce)	Enable/disable GPIO bounce
ECM_GPIO_CONFIG_SET (ECM_GpioSetDebounceClock)	Set the bounce 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 (Extended functions, available only in versions 9 and later)
ECM_GPIO_FUNC_OP (ECM_GpioGetValue)	Get GPIO value
ECM_GPIO_FUNC_OP (ECM_GpioExtGetValue)	Get GPIO value (Extended functions, available only in

	versions 9 and later)
ECM_GPIO_FUNC_OP (ECM_GpioGetIntFlag)	Get GPIO interrupt flag
ECM_GPIO_FUNC_OP (ECM_GpioExtGetIntFlag)	Get GPIO interrupt flag (Extended functions, available only in versions 9 and later)
ECM_QEI_FUNC_OP (ECM_EncOpen)	Enable 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 count
ECM_DAC_FUNC_OP (ECM_DacOpen)	Enable DAC function
ECM_DAC_FUNC_OP (ECM_DacClose)	Turn off DAC function
ECM_DAC_FUNC_OP (ECM_DacSetDelayTime)	Setting the 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)	Enable ADC function
ECM_ADC_FUNC_OP (ECM_AdcClose)	Disable ADC function
ECM_ADC_FUNC_OP (ECM_AdcConfigSampleModule)	Set the ADC trigger source
ECM_ADC_FUNC_OP (ECM_AdcGetDataValidFlag)	Get ADC valid data flag
ECM_ADC_FUNC_OP (ECM_AdcStartConv)	Start ADC conversion

ECM_ADC_FUNC_OP (ECM_AdcGetConvData)	Get ADC conversion data
ECM_EEPROM_REQ	Request to execute EEPROM read and write commands. The data segment of the write operation instruction is the write data. After the read operation is completed, the read data is retrieved through the ECM_EEPROM_GET instruction.
ECM_EEPROM_GET	Read back the ECM_EEPROM_REQ read operation result
ECM_CMD_ECAT_STATE_CHECK	Update and confirm slave status
ECM_CMD_ECAT_DCSYNC	Set the DC sync signal source, cycle time and offset time
ECM_CMD_FIFO_CLR_OP	Clear FIFO
ECM_CMD_FIFO_SET_TX_CNT	Set Tx FIFO number
ECM_CMD_FIFO_GET_TX_CNT	Get Tx FIFO number
ECM_CMD_FIFO_SET_RX_CNT	Set the number of Rx FIFOs
ECM_CMD_FIFO_GET_RX_CNT	Get the number of Rx FIFOs
ECM_CMD_FW_VERSION_GET	Read the firmware version number
ECM_CMD_ECAT_STATE_UPDATE	Update ECAT Status
ECM_CMD_ECAT_INT_SET_ENABLE	Set the interrupt enable mask
ECM_CMD_ECAT_INT_GET_ENABLE	Get interrupt enable mask
ECM_CMD_FIFO_INIT	Initialize FIFO
ECM_CMD_ECAT_SDO_ABORTCODE_GET	Get SDO Abort Code (SDO error code)
ECM_CMD_ECAT_CONFIG_SM	Configure Sync Manager
ECM_CMD_ECAT_CONFIG_FMMU	Setting FMMU
ECM_CMD_ECAT_CONFIG_MAP	Set PDO mapping
ECM_CMD_ASYNCCMD_RESET	Reset non-periodic command
ECM_CMD_WARM_RESET	Reset ECM-XF
ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_GET	Read the maximum working error value

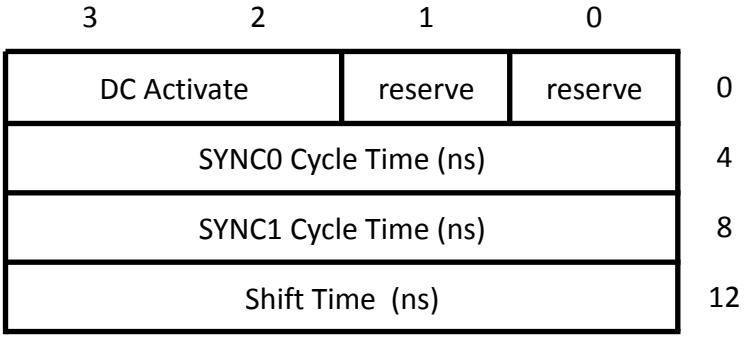
ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_SET	Set the maximum working error value

Command Description

This chapter mainly introduces the command code related header fields and data segment contents. Please refer to Figure 3 Figure 4 Figure 5
1. Yellow is the column related to the command code. According to the command code, the relevant parameters and information are given.
2. The index number is used to identify the response packet
3. The index numbers and command codes of consecutive command packets are consistent. Except for the first command, which is considered a valid command, the remaining commands are considered ECM_CMD_INFO_UPDATE_OP commands.

ECM_CMD_INFO_UPDATE_OP Update header information			
Command Packet			
Header	1	Command code	0
	2	Index Number	Custom
	3	Data length	0

ECM_CMD_ECAT_INIT_OP Initialize the EtherCAT network and slaves. You must initialize the EtherCAT network before using the EtherCAT function.			
Command Packet			
Header	1	Command code	1
	2	Index Number	Custom
	3	Data length	16
	4	Command parameters	48: Special slave Others: General slaves
	5	reserve	bit 11: 0 means disable LRW command (version 0x22 and later)

<table>
<tr><td></td><td></td><td></td><td>bit 12: 0 represents master shift mode, 1 represents bus shift mode (after version 0x22)</td></tr>
<tr><td>Data section</td><td>1</td><td></td><td>


<table>
<tr><td>Location</td><td>length</td><td>name</td><td>illustrate</td></tr>
<tr><td>0x00</td><td>1</td><td>reserve</td><td>Fixed value 0xFF</td></tr>
<tr><td>0x01</td><td>1</td><td>reserve</td><td>Fixed value 0x81
FIFO is disabled by default
If you need to use the FIFO function, enter OP and then enable the FIFO function.</td></tr>
<tr><td>0x02</td><td>2</td><td>DC Activate</td><td>Please refer to the AssignActivate field in ESI</td></tr>
<tr><td>0x04</td><td>4</td><td>SYNC0 Cycle Time</td><td>SYNC0 cycle time (in ns)</td></tr>
<tr><td>0x08</td><td>4</td><td>SYNC1 Cycle Time</td><td>SYNC1 cycle time (in ns)</td></tr>
<tr><td>0x12</td><td>4</td><td>Shift Time</td><td>Cycle time offset (in ns)</td></tr>
</table>
</td></tr>
</table>

*When setting enable LRW command, if there is a slave station that does not support LRW command, LRW command will still be disabled.
*Master shift mode distributed clocks use the first slave clock as the standard clock; bus shift mode distributed clocks use the master clock as the standard clock

<table>
<tr><td colspan="4">ECM_CMD_ECAT_RECONFIG_OP
After re-planning PDO, this command is required to make ECM-XF reconfigure the memory space.</td></tr>
<tr><td colspan="4">Command Packet</td></tr>
<tr><td rowspan="2">Header</td><td>1</td><td>Command code</td><td>2</td></tr>
<tr><td>2</td><td>Index Number</td><td>Custom</td></tr>
</table>

	3	Data length	0

* If the slave uses ECM_CMD_ECAT_RECONFIG_OP to restore the Slave's PDO configuration to the default, please use ECM_CMD_ECAT_CONFIG_MAP (command code 73) instead.

ECM_CMD_ECAT_INIT_DC_OP Initialize decentralized clocks			
Command Packet			
Header	1	Command code	3
	2	Index Number	Custom
	3	Data length	0

*If you want to specify different DC Activate between different slaves, you can change the DC Active Code of a single slave through ECM_CMD_ECAT_DCSYNC after ECM_CMD_ECAT_INIT_OP. After the change, you must use ECM_CMD_ECAT_INIT_DC_OP to initialize the distributed clock.

ECM_CMD_ECAT_PDO_WC_GET ReadPDOInImport informationNumber of slave stations,Export InformationNumber of slaves or currentWorking countervalue			
Command Packet			
Header	1	Command code	4
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	1 : haveImport informationNumber of slaves 2 : haveExport InformationNumber of slaves 3 : Working counter值
Response Packet			
Header	1	Command code	4
	2	Index Number	Same as command packet index number

	3	Data length	4
Data section	1	Read quantity return value	

Working counter theoretical value =haveImport informationNumber of slaves *1 + withExport InformationNumber of slave stations*2

ECM_CMD_ECAT_PDO_DATA_FIFO_OP Access PDO data to FIFO			
Command Packet			
Header	1	Command code	5
	2	Index Number	Custom
	3	Data length	RxPDO length
	4	Command Parameters	Number of PDOs in the data segment
	5	Command information 0	Read and write opcodes bit0: write operation bit1: read operation
Data section	1	PDO Information	RxPDO Information
Response Packet			
Header	1	Command code	5
	2	Index Number	Same as command packet index number

	3	Data length	TxPDO length
	4	Return Value	Read and write opcodes bit0: write operation bit1: read operation
Data section	1	TxPDO Information	

ECM_CMD_ECAT_PDO_DATA_OP Read PDO data			
Command Packet			
Header	1	Command code	6
	2	Index Number	Custom
	3	Data length	RxPDO length
	4	Command information 0	Operation Code bit1: read operation bit2: 1 Read and write operation codes are valid. If this bit is 0, read and write operations are performed.
Data section	1	RxPDO Information	
Response Packet			
Header	1	Command code	6
	2	Index Number	Same as command packet index number

	3	Data length	TxPDO length
Data section	1	TxPDO Information	

* If you want to write to PDO, please use ECM_CMD_ECAT_PDO_DATA_FIFO_OP instead

ECM_CMD_ECAT_PDO_CONFIG_SET
Configure slave PDO
Note: This API is designed to accommodate only 3 PDOs with 8 Objects each. If more than 8 Objects are available, please use SDO to set them directly.

Command Packet

Header	1	Command code	7
	2	Index Number	Custom
	3	Data length	112

Data section 1

3	2	1	0	
PDO specified index number		Number of PDOs	Slave station number	0x00
2nd PDO Mapping		1st PDO Mapping		0x04
1st PDO object quantity		3rd PDO Mapping		0x08
3rd PDO object number		2nd PDO object quantity		0x0C
1st PDO property 0				0x10
.....				
1st PDO property 7				
2nd PDO property 0				0x30
.....				

2nd PDO property 7
3rd PDO property 0
.....
3rd PDO property 7

0x50

Location	length	name	illustrate
0x00	1	Slave station number	0~127
0x01	1	Number of PDOs	0~127
0x02	2	PDO specified index number	TxPDO or RxPDO specified index number
0x04	2	1st PDO Mapping	The first PDO mapping index
0x06	2	2nd PDO Mapping	The second PDO mapping index number
0x08	2	3rd PDO Mapping	The third PDO mapping index number
0x0A	2	1st PDO object quantity	The number of the first PDO object
0x0C	2	2nd PDO object quantity	The number of the second PDO object
0x0E	2	3rd PDO object number	The number of the third PDO object
0x10	4	First PDO object 0 3 2 1 0 Index Number | Sub-index number | Length (bits)	
...		The first PDO object N (N=1~7)	
0x30	4	Second PDO object 0	
...		The second PDO object N (N=1~7)	
0x50	4	The third PDO object 0	
...		The third PDO object N (N=1~7)	

* If PDO configuration cannot be completed using ECM_CMD_ECAT_PDO_CONFIG_SET or the number of PDO objects does not meet the requirements, SDO can be used to read and write slave-related objects.
*Some slaves do not support master configuration PDO, please refer to the slave instructions.

<table>
<tr><td colspan="5">ECM_CMD_ECAT_PDO_CONFIG_REQ
Request to execute PDO configuration read. After the operation is completed, use ECM_CMD_ECAT_PDO_CONFIG_GET to retrieve the configuration</td></tr>
<tr><td colspan="5">Command Packet</td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>Command code</td><td colspan="2">8</td></tr>
<tr><td>2</td><td>Index Number</td><td colspan="2">Custom</td></tr>
<tr><td>3</td><td>Data length</td><td colspan="2">4</td></tr>
<tr><td>Data section</td><td>1</td><td colspan="3">3 2 1 0
PDO specified index number | reserve | Slave station number
0x00</td></tr>
<tr><td></td><td></td><td>Location</td><td>length</td><td>name</td><td>illustrate</td></tr>
<tr><td></td><td></td><td>0x00</td><td>1</td><td>Slave station number</td><td>0~127</td></tr>
<tr><td></td><td></td><td>0x01</td><td>1</td><td>reserve</td><td></td></tr>
<tr><td></td><td></td><td>0x02</td><td>2</td><td>PDO specified index number</td><td>TxPDO or RxPDO specified index number</td></tr>
</table>

<table>
<tr><td colspan="5">ECM_CMD_ECAT_PDO_CONFIG_GET
Read slave PDO configuration</td></tr>
<tr><td colspan="5">Command Packet</td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>Command code</td><td colspan="2">9</td></tr>
<tr><td>2</td><td>Index Number</td><td colspan="2">Custom</td></tr>
<tr><td>3</td><td>Data length</td><td colspan="2">0</td></tr>
</table>

Response Packet			
Header	1	Command code	9
	2	Index Number	Same as command packet index number
	3	Data length	112

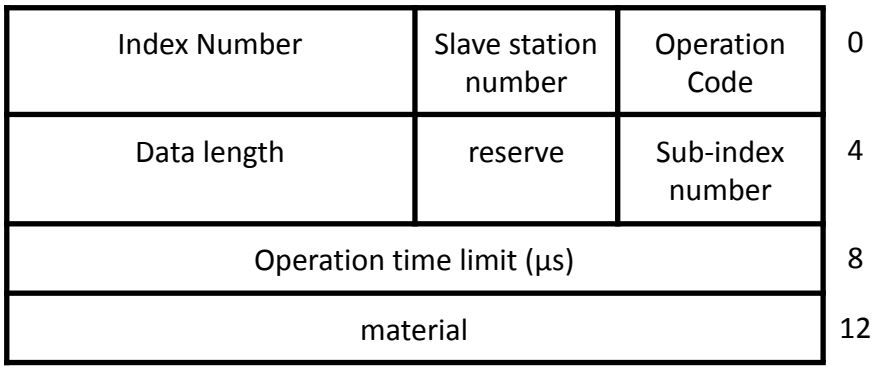
Data section 1

3	2	1	0	
PDO specified index number		Number of PDOs	Slave station number	0x00
2nd PDO Mapping		1st PDO Mapping		0x04
1st PDO object quantity		3rd PDO Mapping		0x08
3rd PDO object number		2nd PDO object quantity		0x0C
1st PDO property 0				0x10
.....				
1st PDO property 7				
2nd PDO property 0				0x30
.....				
2nd PDO property 7				
3rd PDO property 0				0x50
.....				
3rd PDO property 7				

Location	length	name	illustrate
0x00	1	Slave station number	0~127
0x01	1	Number of PDOs	0~127
0x02	2	PDO specified index number	TxPDO or PxPDO specified index number
0x04	2	1st PDO Mapping	The first PDO mapping index

<table>
<tr><td></td><td></td><td>0x06</td><td>2</td><td>2nd PDO Mapping</td><td>The second PDO mapping index number</td></tr>
<tr><td></td><td></td><td>0x08</td><td>2</td><td>3rd PDO Mapping</td><td>The third PDO mapping index number</td></tr>
<tr><td></td><td></td><td>0x0A</td><td>2</td><td>1st PDO object quantity</td><td>The number of the first PDO object</td></tr>
<tr><td></td><td></td><td>0x0C</td><td>2</td><td>2nd PDO object quantity</td><td>The number of the second PDO object</td></tr>
<tr><td></td><td></td><td>0x0E</td><td>2</td><td>3rd PDO object number</td><td>The number of the third PDO object</td></tr>
<tr><td></td><td></td><td>0x10</td><td>4</td><td colspan="2">First PDO object 0
3 2 1 0
Index Number | Sub-index number | Length (bits)</td></tr>
<tr><td></td><td></td><td>...</td><td></td><td colspan="2">The first PDO object N (N=1~7)</td></tr>
<tr><td></td><td></td><td>0x30</td><td>4</td><td colspan="2">Second PDO object 0</td></tr>
<tr><td></td><td></td><td>...</td><td></td><td colspan="2">The second PDO object N (N=1~7)</td></tr>
<tr><td></td><td></td><td>0x50</td><td>4</td><td colspan="2">The third PDO object 0</td></tr>
<tr><td></td><td></td><td>...</td><td></td><td colspan="2">The third PDO object N (N=1~7)</td></tr>
</table>

<table>
<tr><td colspan="4">ECM_CMD_ECAT_SDO_REQ
Request to execute SDO read/write command. The data segment of the write operation instruction is the write data. After the read operation is completed, the read data is retrieved through the ECM_CMD_ECAT_SDO_GET instruction.</td></tr>
<tr><td colspan="4">Command Packet</td></tr>
<tr><td rowspan="4">Header</td><td>1</td><td>Command code</td><td>10</td></tr>
<tr><td>2</td><td>Index Number</td><td>Custom</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Data length</td><td>Read operations: 12</td></tr>
<tr><td>Write operation: 12 + length of the object value to be written</td></tr>
</table>

Data section	1	3 2 1 0 

Location	length	name	illustrate
0x00	1	Operation Code	0: Write operation 1: Read operation
0x01	1	Slave station number	0~127
0x02	2	Index Number	CoE object index
0x04	1	Sub-index number	CoE object sub-index
0x05	1	reserve	
0x06	2	Data length	Length of the object to be read/written (Byte)
0x08	4	Operation time limit	SDO maximum waiting time (μs)
0x12	256	material	Write operation is valid, write object value

ECM_CMD_ECAT_SDO_GET Read back the ECM_CMD_ECAT_SDO_REQ read operation result			
Command Packet			
Header	1	Command code	11
	2	Index Number	Custom
	3	Data length	0

Response Packet			
Header	1	Command code	11
	2	Index Number	Same as command packet index number
	3	Data length	Data segment effective length
Data section	1	The data read back by the previous ECM_CMD_ECAT_SDO_REQ read operation	

ECM_CMD_ECAT_STATE_SET Changing EtherCAT Status			
Command Packet			
Header	1	Command code	12
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0~127: slave station number 0xFF: All slaves
	5	Command information 0	EtherCAT Status 0x01 : EC_STATE_INIT 0x02 : EC_STATE_PRE_OP 0x03 : EC_STATE_BOOT 0x04 : EC_STATE_SAFE_OP 0x08 : EC_STATE_OPERATIONAL

ECM_CMD_ECAT_STATE_GET Retrieve EtherCAT status from XF			
Command Packet			
H	1	Command	13

ea d er		code	
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0~127: slave station number 0xFF: All slaves
Response Packet			
H ea d er	1	Command code	13
	2	Index Number	Same as command packet index number
	3	Data length	0
	4	Return Value	EtherCAT Status 0x01 : EC_STATE_INIT 0x02 : EC_STATE_PRE_OP 0x03 : EC_STATE_BOOT 0x04 : EC_STATE_SAFE_OP 0x08 : EC_STATE_OPERATIONAL

* To update the EtherCAT status of the slave, first useUse this command after updating ECM_CMD_ECAT_STATE_UPDATE

ECM_CMD_ECAT_SLV_INFO_GET Read slave information			
Command Packet			
H ea d er	1	Command code	15
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	0~127: slave station number
	5	Command information 0	Slave station information code 0 : Manufacturer code 1: Manufacturer product code 2: Product version number

			3 : Product Name 4: Configuration location 5 : Location alias 6 : Status (1: Init / 2: PreOP / 4: SafeOP / 8: OP) 7: AL status (0: no error / other: please refer to slave AL status code description) 8: Output data length (RxPDO Size) 9: Input data length (TxPDO Size)
Response Packet			
Header	1	Command code	15
	2	Index Number	Same as command packet index number
	3	Data length	Length of slave station information
Data section	1	From the station information	

* For more information about the slave station, please refer to the definition of ECM_SLV_INFO in EcmDriver.h. The data read back is the data on the slave station, and the XF master station cannot determine the correctness.

ECM_CMD_ECAT_SLV_CNT_GET Read the number of slaves			
Command Packet			
Header	1	Command code	16
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	16
	2	Index	Same as command packet index number

		Number	
	3	Data length	0
	4	Return Value	Number of slaves

ECM_CMD_FIFO_ENABLE Enable RxPDO FIFO output. ECM-XF enables RxPDO FIFO output by default. Users can use this command to disable RxPDO FIFO periodic output, which can be used for pre-storage output.			
Command Packet			
Header	1	Command code	18
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0 : Off 1 : Start

ECM_CMD_FIFO_PACK_SIZE_GET Read the length of PDO data. One data in FIFO is one PDO.			
Command Packet			
Header	1	Command code	19
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Read RxPDO data length 1: Read TxPDO data length
Response Packet			
Header	1	Command code	19
	2	Index Number	Same as command packet index number

	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 Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	20
	2	Index Number	Same as command packet index number
	3	Data length	2
Data section	1	SPI data length	

ECM_CMD_SPI_RECONFIG_OP Reset SPI data packet length			
Command Packet			
Hea	1	Command code	23

der	2	Index Number	Custom
	3	Data length	Data field length in SPI packet (32~1408)
Response Packet			
Header	1	Command code	23
	2	Index Number	Same as command packet index number
	3	Data length	0

Note: When using the ECM_CMD_SPI_RECONFIG_OP command, the CRC field is always given a fixed value (0x12345678). Due to the change in SPI Data Size, the CRC field may change, resulting in an increase in the number of CRC errors in the response header. This is normal.
Note: ECM-XFU SPI Data Size is fixed to 992Bytes and cannot be changed

ECM_CMD_CRC_ERR_CNT_CLR Clear the accumulated CRC check errors			
Command Packet			
Header	1	Command code	24
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	24
	2	Index Number	Same as command packet index number
	3	Data length	0
	4	Return Value	Clear the accumulated CRC check errors

ECM_CMD_CRC_TYPE_SET Set the check code type

Command Packet			
Header	1	Command code	25
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0 : Fixed value check: $0x12345678$ 1 : Undefined 2 : Undefined 3: CRC-32: (calculated from the first bit of the header to the bit before the checksum field, excluding the tail) $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$ (Poly = 0x04C11DB7) Init = 0xFFFFFFFF, RefIn = True, RefOut = True, XorOut = 0xFFFFFFFF

ECM_CMD_402_CONFIG_SET Specify the slave control word and status word offset			
Command Packet			
Header	1	Command code	26
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Motor number (the first axis is number 0)
	5	Command information 0	Control word offset (byte)
	6	Command information 1	Status word offset (byte)
	7	Command information 2	Internal 402 state machine control byte setting bit3: Slave state machine error single clear bit4: State machine control start bit7: Slave state machine error multiple clears Enabling this function will automatically control the control word of

			the axis to achieve functions such as switching the slave 402 status, fault reset, etc. Bit3 is 1 for single error clear (Fault Reset), bit4 is 1 for enable (Enable), bit7 is 1 for multiple error clear (Fault auto reset)
	8	Command information 3	Slave station number (the first station is number 0)

*After the state machine control is started, XF will overwrite bits 0~3, 7 of the slave control word. The user's designation of bits 0~3, 7 of the slave control word is invalid.
* Error clearing mechanism is divided into single clearing (bit 3) and multiple clearing (bit 7). XF will overwrite bit 7 of the slave control word.

ECM_CMD_402_STATE_SET Switch the specified slave 402 status			
Command Packet			
Header	1	Command code	27
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Motor number (the first axis is number 0)
	5	Command information 0	402 Status 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

- You must first use ECM_CMD_402_CONFIG_SET to specify the offset of the slave control word/status word
- OPERATION ENABLED is the enabled (Servo on) state.

ECM_CMD_402_STATE_GET Read the specified slave 402 status
Command Packet

Header	1	Command code	28
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Motor number (the first axis is number 0)
Response Packet			
Header	1	Command code	28
	2	Index Number	Same as command packet index number
	3	Data length	0
	4	Return Value	Current 402 status

- You must first use ECM_CMD_402_CONFIG_SET to specify the offset of the slave control word/status word

ECM_CMD_402_CTL_SET ECM-XF internal 402 state machine control byte setting			
Command Packet			
Header	1	Command code	29
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Motor number (the first axis is number 0)
	5	Command information 0	bit3: Slave state machine error single clear bit4: State machine control start bit7: Slave state machine error multiple clears
	6	Command information 1	Fixed value 0x5A

	7	Command information 2	Total number of motors
	8	Command information 3	Slave number

* You must first use ECM_CMD_402_CONFIG_SET to specify the offset of the slave control word/status word
* After the state machine control is started, XF will overwrite bits 0~3, 7 of the slave control word. The user's designation of bits 0~3, 7 of the slave control word is invalid.
* Error clearing mechanism is divided into single clearing (bit 3) and multiple clearing (bit 7). XF will overwrite bit 7 of the slave control word.

ECM_CMD_402_CTL_GET ECM-XF internal 402 state machine control byte read			
Command Packet			
Header	1	Command code	30
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Motor number (the first axis is number 0)
Response Packet			
Header	1	Command code	30
	2	Index Number	Same as command packet index number
	3	Data length	0
	4	Return Value	402 state machine control setting value

ECM_GPIO_CONFIG_SET(ECM_GpioSetMode) Set GPIO mode			
Command Packet			
H	1	Command	31

ea d er		code	
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0
	5	Command information 0	Bit n: 1 sets the nth GPIO mode (n = 0 to 7)
	6	Command information 1	Bit n: 1 sets the n+8th GPIO mode (n = 0 to 7)
	7	Command information 2	GPIO mode definition 0x0: Input mode 0x1: Output mode 0x2: Open-Drain mode 0x3: Quasi-bidirectional mode
	8	Command information 3	GPIO pull-up setting definition 0x0: Disable pull-up setting 0x1: Pull-up mode 0x2: Pull-down mode

ECM_GPIO_CONFIG_SET(ECM_GpioSetMode) Set GPIO mode (GPIO expansion function, only applicable to specific versions)			
Command Packet			
H ea d er	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	8
	5	Command information 0	Bit n: 1 sets the n+16th GPIO mode (n = 0 to 3)

	6	Command information 1	Reserved, set to 0
	7	Command information 2	GPIO mode definition 0x0: Input mode 0x1: Output mode 0x2: Open-Drain mode 0x3: Quasi-bidirectional mode
	8	Command information 3	GPIO pull-up setting definition 0x0: Disable pull-up setting 0x1: Pull-up mode 0x2: Pull-down mode

ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce) Enable/disable GPIO bounce reduction (only required for Input)			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	2
	5	Command information 0	Bit n: 1 sets the nth GPIO bounce function (n = 0 to 7)
	6	Command information 1	Bit n: 1 sets the n+8th GPIO bounce function (n = 0 to 7)
	7	Command information 2	Bounce Elimination 0: Disable 1: Enable
	8	Command information 3	reserve

ECM_GPIO_CONFIG_SET(ECM_GpioEnableDebounce) Enable/disable GPIO bounce reduction (only required for Input) (GPIO expansion function, only applicable to specific versions)			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	9
	5	Command information 0	Bit n: 1 sets the n+16th GPIO bounce function (n = 0 to 3)
	6	Command information 1	Reserved, set to 0
	7	Command information 2	Bounce Elimination 0: Disable 1: Enable
	8	Command information 3	reserve

ECM_GPIO_CONFIG_SET(ECM_GpioSetDebounceClock) Set bounce clock			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	3
	5	Command	Bit 0 ~ 3: Bounce clock cycle, valid value is 0 to 15

		information 0	Bit 4: Clock source, 0 is 192M, 1 is 10K The sampling period is 2 to the power of (bounce period clock)*(clock time)

ECM_GPIO_CONFIG_SET(ECM_GpioIntEnable) Enable/disable GPIO interrupt			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	1
	5	Command information 0	Channel (0~15) Specific version (0~19)
	6	Command information 1	Interrupt Type 0x0: Disable interrupts 0x1: rising voltage or falling voltage enable interrupt 0x2: Falling voltage enable interrupt 0x3: Rising voltage enable interrupt

ECM_GPIO_CONFIG_SET(ECM_GpioIntClear) Clear GPIO interrupt flag			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	4
	5	Command information 0	Bit n: Clear the nth GPIO interrupt flag (n = 0 ~ 7)

	6	Command information 1	Bit n: Clear the n+8th GPIO interrupt flag (n = 0 ~ 7)

ECM_GPIO_CONFIG_SET(ECM_GpioIntClear) Clear GPIO interrupt flag (GPIO expansion function, only available after version 9)			
Command Packet			
Header	1	Command code	31
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	6
	5	Command information 0	Bit n: Clear the nth GPIO interrupt flag (n = 0 ~ 7)
	6	Command information 1	Bit n: Clear the n+8th GPIO interrupt flag (n = 0 ~ 7)
	7	Command information 2	Bit n: Clear the n+16th GPIO interrupt flag (n = 0 ~ 3) Only applicable to specific versions

ECM_GPIO_FUNC_OP(ECM_GpioSetValue) Set GPIO value (Output mode only)			
Command Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	0

	5	Command information 0	Bit n: Set 1 as the nth GPIO high bit, 0 as the low bit (n = 0 ~ 7)
	6	Command information 1	Bit n: Set 1 as the high bit of the n+8th GPIO, 0 as the low bit (n = 0 ~ 7)

ECM_GPIO_FUNC_OP(ECM_GpioExtSetValue) Set GPIO value (GPIO expansion function, only available after version 9)			
Command Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	2
	5	Command information 0	Bit n: Set 1 as the nth GPIO high bit, 0 as the low bit (n = 0 ~ 7)
	6	Command information 1	Bit n: Set 1 as the high bit of the n+8th GPIO, 0 as the low bit (n = 0 ~ 7)
	7	Command information 2	Bit n: Set 1 as the high bit of the n+16th GPIO, 0 as the low bit (n = 0 ~ 3) Only available in certain versions

ECM_GPIO_FUNC_OP(ECM_GpioGetValue) Get GPIO value			
Command Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	0

	4	Command parameters	1
Response Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	2
Data section			Bit n: When the nth GPIO is in input mode, it is 1 when the input is high When the nth GPIO is in input mode, the input is 0 when it is low. (n = 0 ~ 15)

ECM_GPIO_FUNC_OP(ECM_GpioExtGetValue) Get GPIO value (GPIO expansion function, only available after version 9)			
Command Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	3
Response Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	4
Dat			Bit n: When the nth GPIO is in input mode, it is 1 when the input is high

a section			When the nth GPIO is in input mode, the input is 0 when it is low. (n = 0 ~ 15), specific versions only (n = 0 ~ 19)

ECM_GPIO_FUNC_OP(ECM_GpioGetIntFlag) Get GPIO interrupt flag			
Command Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	5
Response Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	2
Data section			Bit n: nth GPIO interrupt flag (n = 0 ~ 15)

ECM_GPIO_FUNC_OP(ECM_GpioExtGetIntFlag) Get GPIO interrupt flag (GPIO expansion function, only applicable to specific versions)			
Command Packet			
Hea	1	Command code	32

der	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	7
Response Packet			
Header	1	Command code	32
	2	Index Number	Custom
	3	Data length	4
Data section			Bit n: nth GPIO interrupt flag (n = 0 ~ 19)

ECM_QEI_FUNC_OP(ECM_EncOpen) Enable and set encoder mode			
Command Packet			
Header	1	Command code	33
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	4
Data section			model 0x000: X4 free step counting mode 0x100: X2 free step counting mode 0x200: X4 comparison pedometer mode 0x300: X2 comparison pedometer mode

ECM_QEI_FUNC_OP(ECM_EncStart) Start encoder step counting			
Command Packet			
Header	1	Command code	33
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	5

ECM_QEI_FUNC_OP(ECM_EncStop) Stop encoder step counting			
Command Packet			
Header	1	Command code	33
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	6

ECM_QEI_FUNC_OP(ECM_EncGetCount) Get the encoder step value			
Command Packet			
Header	1	Command code	33
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	21
Response Packet			

Header	1	Command code	33
	2	Index Number	Custom
	3	Data length	4
Data section			Encoder information

ECM_DAC_FUNC_OP(ECM_DacOpen) Enable DAC mode			
Command Packet			
Header	1	Command code	34
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	20
Data section			model 0x0000: Use ECM_DacSetData() to trigger DAC output 0x0010: Use ECM_DacStartConv() to trigger DAC output 0x0030: Triggered when the DAC_ST pin is low 0x1030: Triggered when the DAC_ST pin is high 0x2030: Triggered when the DAC_ST pin is at a rising voltage 0x3030: Triggered when the DAC_ST pin voltage drops slightly

ECM_DAC_FUNC_OP(ECM_DacClose) Turn off DAC mode			
Command Packet			
Head	1	Command code	34

er	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	21

ECM_DAC_FUNC_OP(ECM_DacSetDelayTime) Setting the DAC delay time			
Command Packet			
Header	1	Command code	34
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	22
Data section			The delay range is 0 ~ 1023 (unit: 1/96 us) It takes at least 8us for DAC to change from 0x000 to 0xFFF, so this value should be greater than 768.

ECM_DAC_FUNC_OP(ECM_DacSetData) Set DAC data			
Command Packet			
Header	1	Command code	34
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	11
D			Data Value

ata section			Range 0 ~ 4095 corresponds to 0 ~ 3.3V

ECM_DAC_FUNC_OP(ECM_DacStartConv) Start DAC conversion			
Command Packet			
Header	1	Command code	34
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0

ECM_ADC_FUNC_OP(ECM_AdcOpen) Enable ADC function			
Command Packet			
Header	1	Command code	35
	2	Index Number	Custom
	3	Data length	4
	4	Command parameters	40
Data section			0x000: Single-ended mode 0x100: Differential mode

ECM_ADC_FUNC_OP(ECM_AdcClose) Disable ADC function			
Command Packet			
Header	1	Command code	35
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	41

ECM_ADC_FUNC_OP(ECM_AdcConfigSampleModule) Set ADC trigger source			
Command Packet			
Header	1	Command code	35
	2	Index Number	Custom
	3	Data length	12
	4	Command Parameters	42
Data section			3 2 1 0 Reserved, set to 0 — 0x0 0: Software trigger — 0x4 Reserved, set to 0 — 0x8

ECM_ADC_FUNC_OP(ECM_AdcGetDataValidFlag) Get ADC valid data flag			
Command Packet			
H	1	Command	35

ea d er		code	
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	17
D at a se cti o n			Fixed value, 0x1000
Response Packet			
H ea d er	1	Command code	35
	2	Index Number	Custom
	3	Data length	4
D at a se cti o n			Valid flags When the valid flag is non-zero, the conversion data is valid

ECM_ADC_FUNC_OP(ECM_AdcStartConv) Start ADC conversion			
Command Packet			
H ea d er	1	Command code	35
	2	Index Number	Custom
	3	Data length	4
	4	Command	12

		Parameters	
Data section			Fixed value, 0x1000

ECM_ADC_FUNC_OP(ECM_AdcGetConvData) Get ADC conversion data			
Command Packet			
Header	1	Command code	35
	2	Index Number	Custom
	3	Data length	4
	4	Command Parameters	15
Data section			Fixed value, 12
Response Packet			
Header	1	Command code	35
	2	Index Number	Custom
	3	Data length	4
Data secti			Convert data 0 ~ 4095 corresponds to 0 ~ 3.3V

o n			

ECM_EEPROM_REQ
Request to execute EEPROM read and write commands. The data segment of the write operation instruction is the write data. After the read operation is completed, the read data is retrieved through the ECM_EEPROM_GET instruction.

Command Packet

Header	1	Command code	38
	2	Index Number	Custom
	3	Data length	12

Data section 1

3	2	1	0	
Slave station number		Operation Code		0
Write data		Read and write position		4
Operation time limit				8

Location	length	name	illustrate
0x00	2	Operation Code	0 : Read 1 : Write
0x02	2	Slave station number	0~127
0x04	2	Read and write position	EEPROM Location
0x06	2	Write data	Write operation is valid
0x08	4	Operation time limit	Maximum waiting time (μs)

ECM_EEPROM_GET
Read back the ECM_EEPROM_REQ read operation result

Command Packet

Header	1	Command code	39
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	39
	2	Index Number	Same as command packet index number
	3	Data length	Data segment effective length
	4	Return Value	Operation error code
Data section	1	Retrieve ECM_EEPROM_REQ read operation to read back data	

ECM_CMD_ECAT_STATE_CHECK Update and confirm slave status			
Command Packet			
Header	1	Command code	41
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0~127: slave station number 0xFF: All slaves
	5	Command information 0	EtherCAT Status

* This command is composite command of ECM_CMD_ECAT_STATE_SET + ECM_CMD_ECAT_STATE_UPDATE + ECM_CMD_ECAT_STATE_GET

ECM_CMD_ECAT_DCSYNC Set the DC SYNC signal source, cycle time, and offset time			
Command Packet			
Header	1	Command code	50
	2	Index Number	Custom
	3	Data length	16
Data section			

3	2	1	0	
DC Activate		reserve	Slave station number	0
SYNC0 Cycle Time (ns)				4
SYNC1 Cycle Time (ns)				8
Shift Time (ns)				12

Response Packet			
Header	1	Command code	50
	2	Index Number	Same as command packet index number
	3	Data length	0

*If you want to specify different DC Activate between different slaves, you can change the DC Active Code of a single slave through ECM_CMD_ECAT_DCSYNC after ECM_CMD_ECAT_INIT_OP. After the change, you must use ECM_CMD_ECAT_INIT_DC_OP to initialize the distributed clock.

ECM_CMD_FIFO_CLR_OP Clear FIFO contents			
Command Packet			
Hea	1	Command code	51

der	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Clear TxFIFO and RxFIFO 1: Clear TxFIFO only 2: Clear RxFIFO only

ECM_CMD_FIFO_SET_TX_CNT Set Tx FIFO number			
Command Packet			
Header	1	Command code	52
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	Tx FIFO number

* The number of TxFIFOs is preset to 64, and the total space of RxFIFO and TxFIFO is 0x4000 Bytes
* RxFIFO quantity x RxPDOSize + TxFIFO quantity x TxPDOSize <= 0x4000 Bytes
* When TxFIFO is full, the oldest data will be discarded (that is, the latest data will always be kept)

ECM_CMD_FIFO_GET_TX_CNT Get Tx FIFO setting quantity			
Command Packet			
Header	1	Command code	53
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	53
	2	Index	Same as command packet index number

		Number	
	3	Data length	0
	4	Return Value	Tx FIFO setting quantity

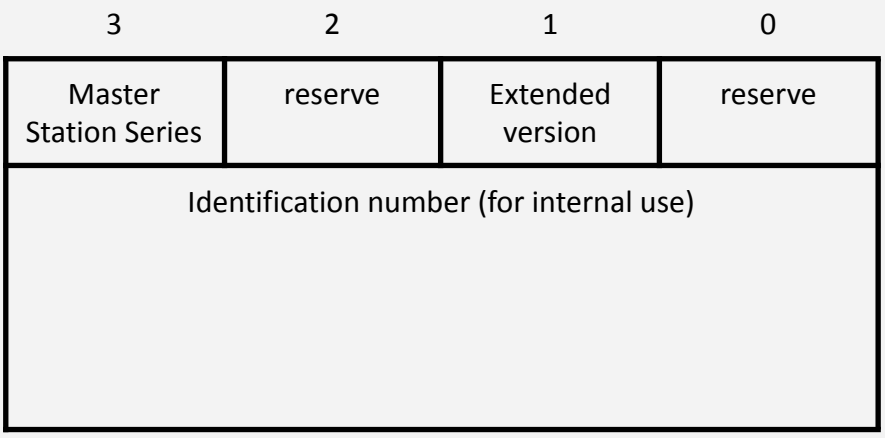
ECM_CMD_FIFO_SET_RX_CNT Set the number of Rx FIFOs			
Command Packet			
Header	1	Command code	54
	2	Index Number	Custom
	3	Data length	0
	4	Command parameters	Rx FIFO number

* The number of RxFIFOs is preset to 64, and the total space of RxFIFO and TxFIFO is 0x4000 Bytes
* RxFIFO quantity x RxPDOSize + TxFIFO quantity x TxPDOSize <= 0x4000 Bytes
* When RxFIFO is full, the data will be discarded (that is, the latest data cannot be put into FIFO)

ECM_CMD_FIFO_GET_RX_CNT Get the Rx FIFO setting number			
Command Packet			
Header	1	Command code	55
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	55
	2	Index Number	Same as command packet index number

	3	Data length	0
	4	Return Value	Rx FIFO setting quantity

ECM_CMD_FW_VERSION_GET Read the firmware version number			
Command Packet			
Header	1	Command code	56
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	56
	2	Index Number	Same as command packet index number
	3	Data length	20
	4	Return Value	Firmware version number
Data section			



The master station series includes ECM-XF, ECM-XFL, ECM-XFU, ECM-XM and other different series of master stations

ECM_CMD_ECAT_STATE_UPDATE Update slave ECAT status to XF			
Command Packet			
Header	1	Command code	57
	2	Index Number	Custom
	3	Data length	0

ECM_CMD_ECAT_INT_SET_ENABLE Set the interrupt enable mask			
Command Packet			
Header	1	Command code	58
	2	Index Number	Custom
	3	Data length	6
	4	Command Parameters	u8INTActiveHigh: INT0/INT1 interrupt signal polarity BIT0: INT1 signal polarity (0 is low voltage start, 1 is high voltage start), default is 0 BIT1: INT0 signal polarity (0 is low voltage start, 1 is high voltage start), default is 1
Data section		INT0 indicates whether XF is processing commands and not accepting new commands. INT1 interrupt source 3 2 1 0 u32CompIntEnable Comprehensive interrupt (0) u8PeripIntEnable Input interrupt and QEI interrupt (4) u8GpioIntEnable GPIO input interrupt u32CompIntEnable : comprehensive interrupt	

		BIT31: EtherCAT packet receiving BIT25: RxFIFO low limit number BIT24: TxFIFO high limit BIT23 : CRC error BIT22 : EtherCAT counter error u8GpioIntEnable : GPIO0~7 input interrupt BIT0 : GPIO00 INT BIT1 : GPIO01 INT BIT2 : GPIO02 INT BIT3 : GPIO03 INT BIT4 : GPIO04 INT BIT5 : GPIO05 INT BIT6 : GPIO06 INT BIT7 : GPIO07 INT u8PeripIntEnable : GPIO8~11 & QEI interrupt BIT0 : GPIO08 INT BIT1 : GPIO09 INT BIT2 : GPIO10 INT BIT3 : GPIO11 INT BIT4 : QEI index INT BIT5 : QEI compares INT BIT6 : QEI Over/Under flow INT

* Generally speaking, the order of interruption is [EtherCAT packet receiving] -> [TxFIFO high limit] -> [RxFIFO low limit]

ECM_CMD_ECAT_INT_GET_ENABLE Get interrupt enable mask			
Command Packet			
Header	1	Command code	59
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	59
	2	Index Number	Custom

<table>
<tr><td rowspan="2"></td><td>3</td><td>Data length</td><td>6</td></tr>
<tr><td>4</td><td>Return Value</td><td>u8INTActiveHigh: INT0/INT1 interrupt signal polarity
BIT0: INT1 signal polarity (0 is low voltage start, 1 is high voltage start)
BIT1: INT0 signal polarity (0 is low voltage start, 1 is high voltage start)</td></tr>
<tr><td>Data section</td><td></td><td colspan="2">
INT1 interrupt source
<table>
<tr><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td colspan="3">u32CompIntEnable
Comprehensive interrupt</td><td>0</td></tr>
<tr><td></td><td>u8PeripIntEnable
Input interrupt and QEI interrupt</td><td>u8GpioIntEnable
GPIO input interrupt</td><td>4</td></tr>
</table>
u32CompIntEnable : comprehensive interrupt
BIT31: EtherCAT packet receiving
BIT25 : RxFIFO low limit
BIT24: TxFIFO high limit
BIT23 : CRC error
BIT22 : EtherCAT counter error

u8GpioIntEnable : GPIO0~7 input interrupt
BIT0 : GPIO00 INT
BIT1 : GPIO01 INT
BIT2 : GPIO02 INT
BIT3 : GPIO03 INT
BIT4 : GPIO04 INT
BIT5 : GPIO05 INT
BIT6 : GPIO06 INT
BIT7 : GPIO07 INT

u8PeripIntEnable : GPIO8~11 & QEI interrupt
BIT0 : GPIO08 INT
BIT1 : GPIO09 INT
BIT2 : GPIO10 INT
BIT3 : GPIO11 INT
BIT4 : QEI index INT
BIT5 : QEI compares INT
BIT6 : QEI Over/Under flow INT
</td></tr>
</table>

ECM_CMD_FIFO_INIT Initialize FIFO and build FIFO space according to FIFO quantity and PDO size			
Command Packet			
Header	1	Command code	66
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	66
	2	Index Number	Custom
	3	Data length	0

- The maximum space is 0x4000Byte

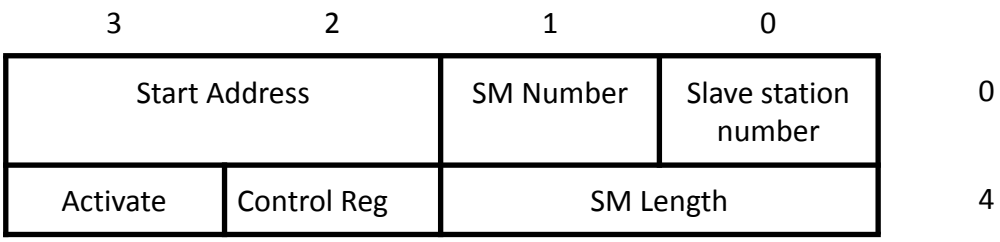
ECM_CMD_ECAT_SDO_ABORTCODE_GET Get SDO Abort Code (SDO error code)			
Command Packet			
Header	1	Command code	69
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	69
	2	Index Number	Custom
	3	Data length	4
Data	1	Byte0~3: SDO Abort Code The Abort Code is transmitted by the slave station according to the ETG definition as	

section		follows, for reference only 0x00000000 No SDO error 0x05030000 Toggle bit not changed 0x05040000 SDO timeout 0x05040001 Command specifier unknown 0x05040005 Out of memory 0x06010000 Unsupported Access 0x06010001 Write only entry 0x06010002 Read only entry 0x06010003 Entry can not be written because Subindex0 is not 0 0x06010004 The object can not be accessed via complete access 0x06020000 Object not existing 0x06040041 Object can not be mapped to PDO 0x06040042 Mapped Object exceeds PDO 0x06040043 Parameter is incompatible 0x06040047 Device incompatibility 0x06060000 Hardware error 0x06070010 Parameter length error 0x06070012 Parameter is too long 0x06070013 Parameter is too short 0x06090011 Subindex (Entry) not exists 0x06090030 Value exceeds 0x06090031 Value is too great 0x06090032 Value is too small 0x06090033 Detected Module Ident List (0xF030) and Configured Module Ident list (0xF050) does not match 0x06090036 Value is less than minimum value 0x08000000 General error 0x08000020 Data can not be read or written 0x08000021 Data can not be accessed because of local control 0x08000022 Data can not be read or written in the current state 0x08000023 Object is not in the object dictionary

ECM_CMD_ECAT_SET_FIFO_TH Set the TxFIFO upper limit and RxFIFO lower limit. When the set limit is reached, an interrupt is issued via INT1			
Command Packet			
Header	1	Command code	70
	2	Index Number	Custom
	3	Data length	0

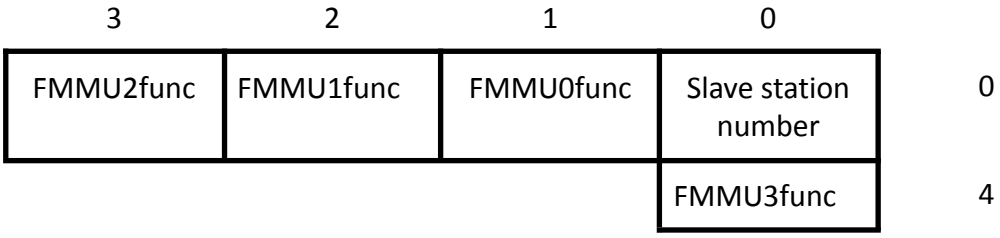
	4	Command Parameters	Disable: 0 Enable: 1 Bit 0 TxFIFO high limit Bit 1 RxFIFO low limit
	5	Command information 0	TxFIFO high limit
	6	Command information 1	RxFIFO low limit

- Assume that the TxFIFO high limit is set to 10. When the TxFIFO changes from 9 to 10, an interrupt is triggered.
- Assume that the RxFIFO low limit is set to 10. When the RxFIFO changes from 11 to 10, an interrupt is triggered.

ECM_CMD_ECAT_CONFIG_SM Set up Sync Manager (please follow the instructions on the slave station)			
Command Packet			
Header	1	Command code	71
	2	Index Number	Custom
	3	Data length	8
Data section			
Response Packet			
Header	1	Command code	71
	2	Index Number	Same as command packet index number
	3	Data length	0

* ECM-XF will read the slave memory to set. If the parameter to be set is different

from the memory value in the slave, you need to use this command to manually configure it.

<table>
<tr><td colspan="4">ECM_CMD_ECAT_CONFIG_FMMU
Set FMMU (FMMU is configured by the master station, this command is invalid)</td></tr>
<tr><td colspan="4">Command Packet</td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>Command code</td><td>72</td></tr>
<tr><td>2</td><td>Index Number</td><td>Custom</td></tr>
<tr><td>3</td><td>Data length</td><td>5</td></tr>
<tr><td>Data section</td><td colspan="3"> 
</td></tr>
<tr><td colspan="4">Response Packet</td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>Command code</td><td>72</td></tr>
<tr><td>2</td><td>Index Number</td><td>Same as command packet index number</td></tr>
<tr><td>3</td><td>Data length</td><td>0</td></tr>
</table>

* ECM-XF will configure the slave FMMU. Even if the user uses this command to configure, it will still be overwritten. This command is invalid.

<table>
<tr><td colspan="4">ECM_CMD_ECAT_CONFIG_MAP
Set PDO mapping</td></tr>
<tr><td colspan="4">Command Packet</td></tr>
<tr><td rowspan="2">Header</td><td>1</td><td>Command code</td><td>73</td></tr>
<tr><td>2</td><td>Index Number</td><td>Custom</td></tr>
</table>

	3	Data length	0
Response Packet			
Header	1	Command code	73
	2	Index Number	Custom
	3	Data length	0

ECM_CMD_402_GET_STATUSWORD Get the status word of CiA402 slave			
Command Packet			
Header	1	Command code	74
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	Motor number (the first axis is number 0)
Response Packet			
Header	1	Command code	74
	2	Index Number	Custom
	3	Data length	2
Data section		Status word value	

* It is recommended to use SDO directly to read the slave's 0x6041 object.

ECM_CMD_ASYNCCMD_RESET Reset non-periodic command

Command Packet			
Header	1	Command code	75
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	1: Clear the AsyncBusy flag 2: Skip the network port initialization wait 4: Abandon the previous Async command

ECM_CMD_MDIO Read and write MDIO interface			
Command Packet			
Header	1	Command code	81
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Write 1: Read
	5	Command information	3 2 1 0 Write data (when writing) Retain (when reading) / PHY Address / PHY Registers
Response Packet			
Header	1	Command code	81
	2	Index Number	Custom
	3	Data length	0 (when writing) 2 (when reading)

Data section		MDIO read data

ECM_CMD_DIRECT_ASSIGN Manually specify the process data length			
Command Packet			
Header	1	Command code	82
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Enable or disable the function 1: Specify the process data length
	5	Command information	When the command parameter is 0: 3 2 1 0 reserve | 0 means not enabled 1 is enabled | Slave address When the command parameter is 1: 3 2 1 0 Process data length | 0 is the specified output 1 is the specified input | Slave address
Response Packet			
Head	1	Command code	82

er	2	Index Number	Custom
	3	Data length	0

ECM_CMD_COE_EMERGENCY List the latest n CoE alarms			
Command Packet			
Header	1	Command code	83
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	n (0<n≤8)
Response Packet			
Header	1	Command code	83
	2	Index Number	Custom
	3	Data length	12*n

Data section		3 2 1 0 Alarm number* Note 2 | Slave station number | 0 or 0xff* Note 1 — 0 error data | error register | error code — 4 error data — 8 Alarm number* Note 2 | Slave station number | 1 or 0xff* Note 1 — 12 error data | error register | error code — 16 error data — 20 … Alarm number* Note 2 | Slave station number | n-1 or 0xff* Note 1 — 12*(n-1) error data | error register | error code — 12*(n-1)+4 error data — 12*(n-1)+8 *Note 1: If it is 0xff, it means there is no alarm *Note 2: This number indicates the number of alarms read after the ECM starts operating.

ECM_CMD_WARM_RESET Reset ECM-XF			
Command Packet			
Header	1	Command code	127
	2	Index Number	Custom
	3	Data length	0

* After using ECM_CMD_WARM_RESET, ECM-XF will be reset and initialized, and all states will return to the initial preset state. The next command must wait until it is initialized before it can be accepted

ECM_CMD_ECAT_WKC_ACCOUNTS_ERR_CNT_GET Read the working error value			
Command Packet			
Header	1	Command code	130
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	130
	2	Index Number	Custom
	3	Data length	0
	4	Return Value	Current wkc error count

ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_GET Read the maximum working error value			
Command Packet			
Header	1	Command code	131
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	131
	2	Index Number	Custom
	3	Data length	0
	4	Return	0~254: Set the maximum value

		Value	255: Set the maximum value to infinity

ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_SET Set the maximum working error value When the working count is greater than the maximum value, PDO data exchange will stop, and Status bit4/bit6 in the Head will be 0			
Command Packet			
Header	1	Command code	132
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	Set Maximum Value The default value is 255 (255 means the maximum value is infinite), and the setting value can be 1~254

ECM_CMD_RAW_ECAT_FUNC_REQ Send any EtherCAT datagram			
Command Packet			
Header	1	Command code	137
	2	Index Number	Custom
	3	Data length	20+n
Data section			

3	2	1	0	
Datagram length		Reserved bits	OP code	0
DECORATION		APD		4
Logical address (used by LRD, LWR, LRW instructions)				8
Timeout (Unitms)				12

		Reserved bits	16
		n-byte data (n≤256)	20
		OP code can be the following values 1: APRD; 2: APWR; 4: FPRD; 5: FPWR; 7: BRD; 8: BWR; 10: LRD; 11: FLOOR; 12: LRW; 13: ARMS; 14: FRMW	
Response Packet			
Header	1	Command code	137
	2	Index Number	Custom
	3	Data length	0

ECM_CMD_RAW_ECAT_FUNC_GET Read EtherCAT datagram feedback data Need to be used with the ECM_CMD_RAW_ECAT_FUNC_REQ command			
Command Packet			
Header	1	Command code	138
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	138
	2	Index Number	Custom
	3	Data length	4+n
Data sectio		3 2 1 0 Datagram length n | Reserved bits | OP code	0

n		DECORATION / APD	4
		Logical address (used by LRD, LWR, LRW instructions)	8
		Timeout (in μs)	12
		Working counter	16
		n-byte data (n≤256)	20

ECM_CMD_FOE_FUNC_REQ FoE (File over EtherCAT) Commands			
Command Packet			
Header	1	Command code	139
	2	Index Number	Custom
	3	Data length	20+n (n: file name length or data length when writing, unit byte)

Data section

FoE upload/download commands:

3	2	1	0	
Reserved bits	OP code 1: Upload 2: Download	Reserved bits	Slave number	0
FoE file password				4
FoE file length (bytes)				8
Timeout (in μs)				12
The starting location of the file in the flash memory (0≤starting position<0x40000)				16
n-byte FoE file name				20

Write/Read Flash Memory Commands:

		3 2 1 0 Reserved bits | OP code 3: Read 4: Write | Reserved bits | Reserved bits — 0 Reserved bits — 4 Write/read data length n (in bytes, n≤256) — 8 Reserved bits — 12 The starting location of the file in the flash memory (0≤starting position<0x40000) — 16 n bytes of data (only required for write flash memory command) — 20

ECM_CMD_FOE_FUNC_GET Read FoE uploaded data Requires ECM_CMD_FOE_FUNC_REQ command			
Command Packet			
Header	1	Command code	140
	2	Index Number	Custom
	3	Data length	0
Response Packet			
Header	1	Command code	140
	2	Index Number	Custom
	3	Data length	n
Data sectio		After the flash memory is read using the ECM_CMD_FOE_FUNC_REQ command, n bytes of data are returned.	

n		

ECM_CMD_ENABLE_LRW Enable or disable the LRW instruction			
Command Packet			
Header	1	Command code	142
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Disable LRW instruction (use LRD/LWR instead of LRW instruction) 1: If the slave supports LWR command, use LRW command (default)
Response Packet			
Header	1	Command code	142
	2	Index Number	Custom
	3	Data length	0

ECM_CMD_RECV_MAILBOX Read the mailbox data received from the slave station			
Command Packet			
Header	1	Command code	143
	2	Index Number	Custom
	3	Data length	0
	4	Command Parameters	0: Send a request to read the mailbox 1: Read email information
	5	Command information	When the command parameter is 0:

<table>
<tr><td></td><td></td><td></td><td>
3 2 1 0
<table>
<tr><td colspan="3">reserve</td><td>Slave address</td></tr>
</table>
When the command parameter is 1:
3 2 1 0
<table>
<tr><td colspan="2">Read data length</td><td colspan="2">Read data starting point</td></tr>
</table>
</td></tr>
<tr><td colspan="4">Response Packet</td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>Command code</td><td>143</td></tr>
<tr><td>2</td><td>Index Number</td><td>Custom</td></tr>
<tr><td>3</td><td>Data length</td><td>n (only when the command parameter is 1, n = command data 3*64 + command data 2)
0 (only when the command parameter is 0)</td></tr>
<tr><td>Data section</td><td></td><td colspan="2">n bytes of mailbox data (only when command parameter is 1)
The starting position of the data is the [command data 1*64 + command data 0]th byte of the mailbox</td></tr>
</table>

ECM_CMD_MBX_STAT Read the slave receiving mailbox status			
Command Packet			
Header	1	Command code	144
	2	Index Number	Custom
	3	Data length	0 (when the command parameter is 0 or 2) n (When the command parameter is 1, n = ceil (number of slaves/8)
	4	Command Parameters	0: Read the slave receiving mailbox status in OP state 1: Enable or disable the master station polling to receive mailbox status 2. Check whether the master station polling receiving mailbox status is enabled
Data section	When the command parameter is 1: n byte whether to enable polling receiving mailbox The kth bit of the nth byte represents the [8*n+k]th slave 1: Enable polling mailbox 0: Do not enable polling mailbox		
Response Packet			
Header	1	Command code	144
	2	Index Number	Custom
	3	Data length	n (When the command parameter is 0 or 2, n = ceil (number of slaves/8))
Data section	When the command parameter is 0: n-byte mailbox status The kth bit of the nth byte represents the status of the [8*n+k]th slave 1: The receiving mailbox has information 0: No data in the receiving mailbox When the command parameter is 2: n byte whether to enable polling receiving mailbox The kth bit of the nth byte represents the [8*n+k]th slave 1: Enable polling mailbox 0: Do not enable polling receiving mailbox		

* After enabling the polling receive mailbox function with ECM_CMD_MBX_STAT, you

need to send the ECM_CMD_ECAT_RECONFIG_OP command again for the change to take effect.

ECM-XF command and response sending and receiving process description

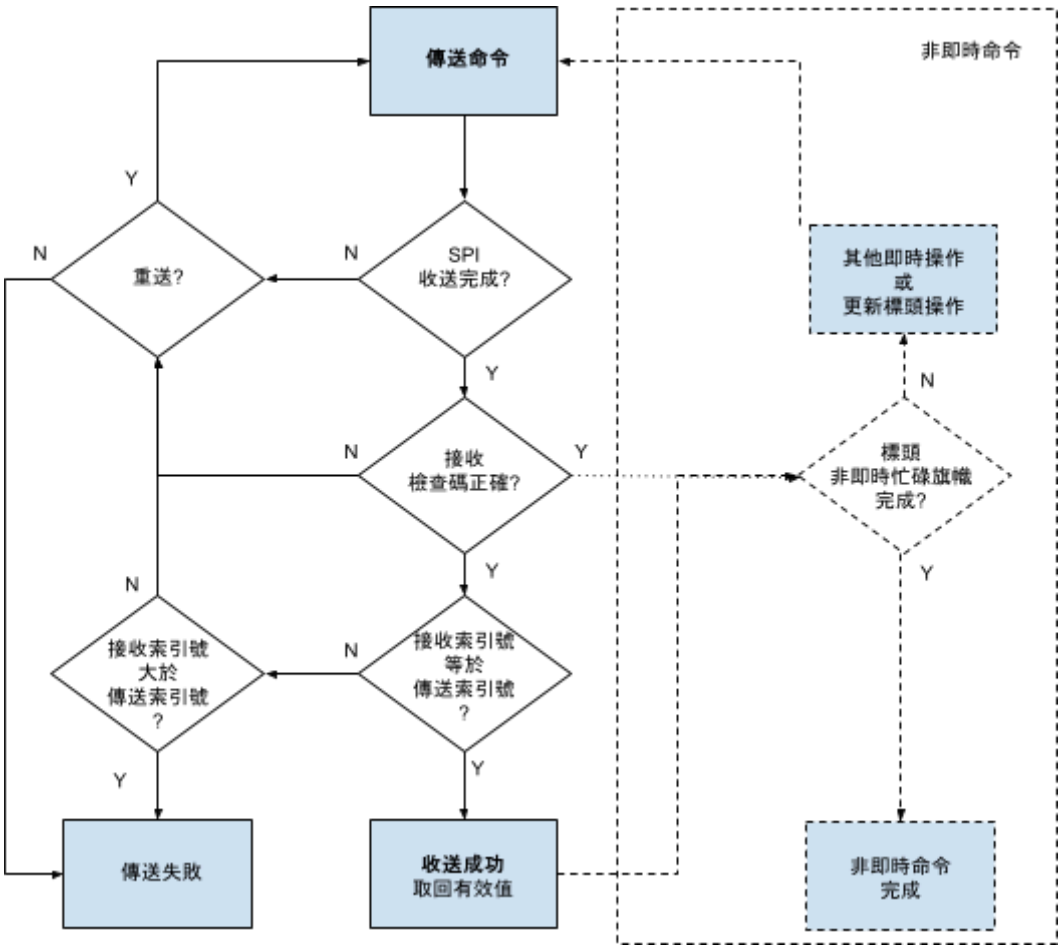


Figure 6 Command and response sending and receiving process

ECM-XF Initialize EtherCAT network and slave configuration instructions

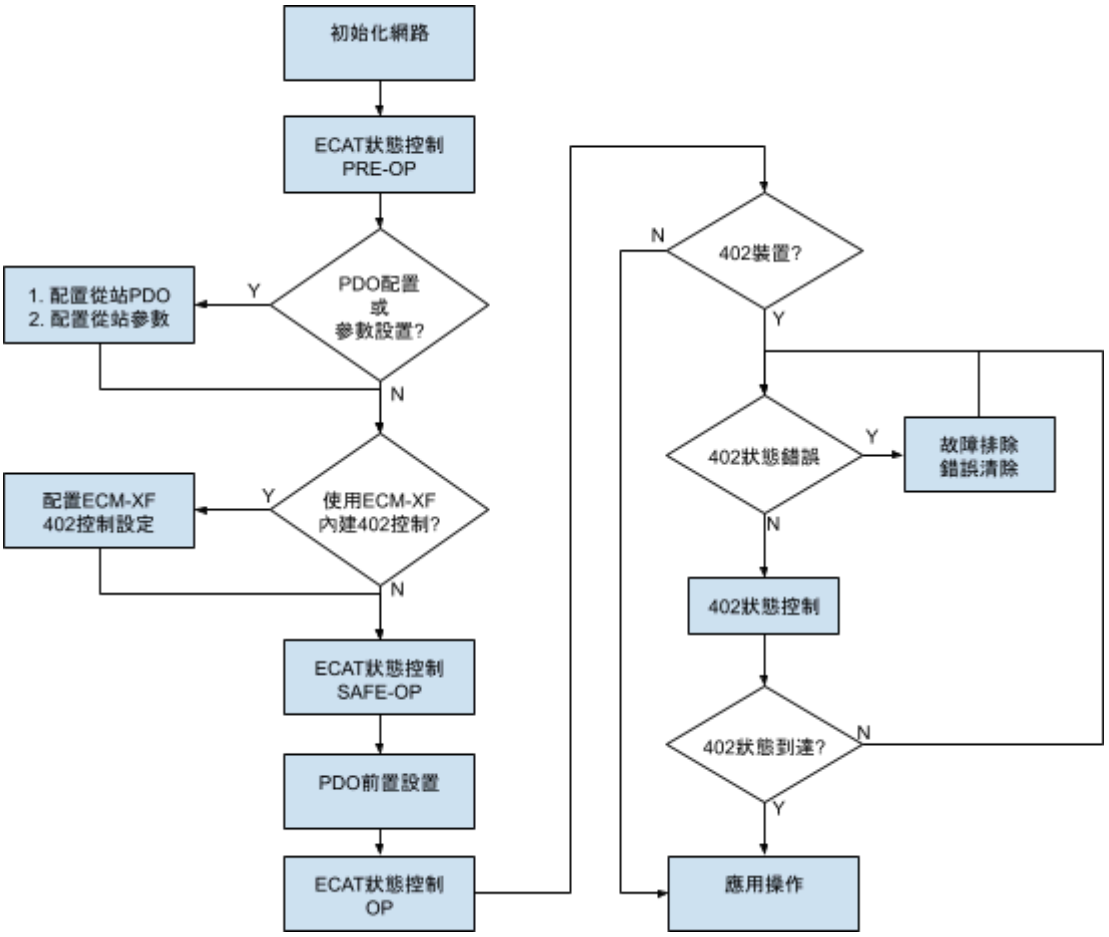


Figure 7 EtherCAT network initialization and configuration process

Process Status	Related commands	illustrate
Initialize the network	ECM_CMD_ECAT_INIT_OP	Set DC mode and time
ECAT status control	ECM_CMD_ECAT_STATE_SET ECM_CMD_ECAT_STATE_GET	Toggle ECAT status
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	Specify or configure PDO
Configure slave parameters	ECM_CMD_ECAT_SDO_REQ ECM_CMD_ECAT_SDO_GET	Using SDO Configuration Parameters ex. Control mode Torque limit...
Configuring 402 Control	ECM_CMD_402_CONFIG_SET	Built-in 402 control Setting Error Clear
402 Status Control	ECM_CMD_402_STATE_SET ECM_CMD_402_STATE_GET	Toggle 402 status
PDO pre-settings	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 Operations	ECM_CMD_ECAT_PDO_DATA_FIFO_OP ECM_CMD_ECAT_SDO_REQ ECM_CMD_ECAT_SDO_GET ECM_CMD_INFO_UPDATE_OP	Exchange PDO data via FIFO Read and write data using SDO Status Update

GPIO

The default GPIO settings of ECM-XF are all floating inputs.

ECM-XF Interrupt

ECM-XF has two interrupt outputs. When SPI is ready, the INT0 pin is preset to high. The polarity can be changed through the ECM_CMD_ECAT_INT_SET_ENABLE instruction. Under normal circumstances, after the SPI transmission is completed, ECM-XF will need a short time to process. It is recommended to use a delay of 150us. The general processing time is < 150us. If the cycle time is very short (such as 250us), you can refer to the INT0 signal to reduce the delay time.

INT1 is a general interrupt function pin. When the interrupt enable bit (ECM_CMD_ECAT_INT_SET_ENABLE) is triggered, INT1 is in the Enable state. The example in Figure 8 below shows that when the interrupt mode is low active, EtherCAT accepts changes at t=2 and t=7.

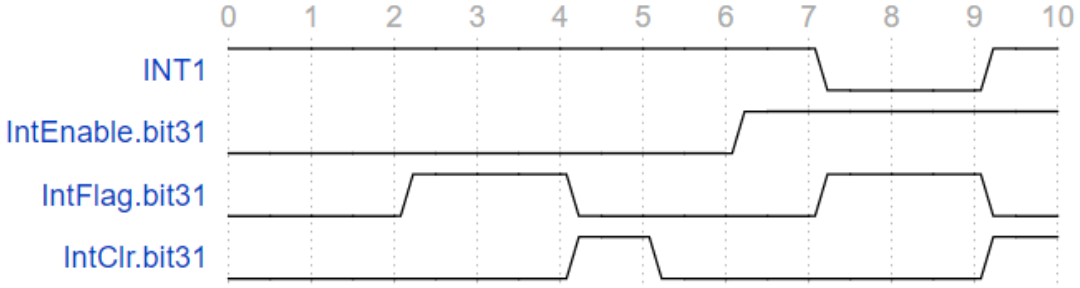
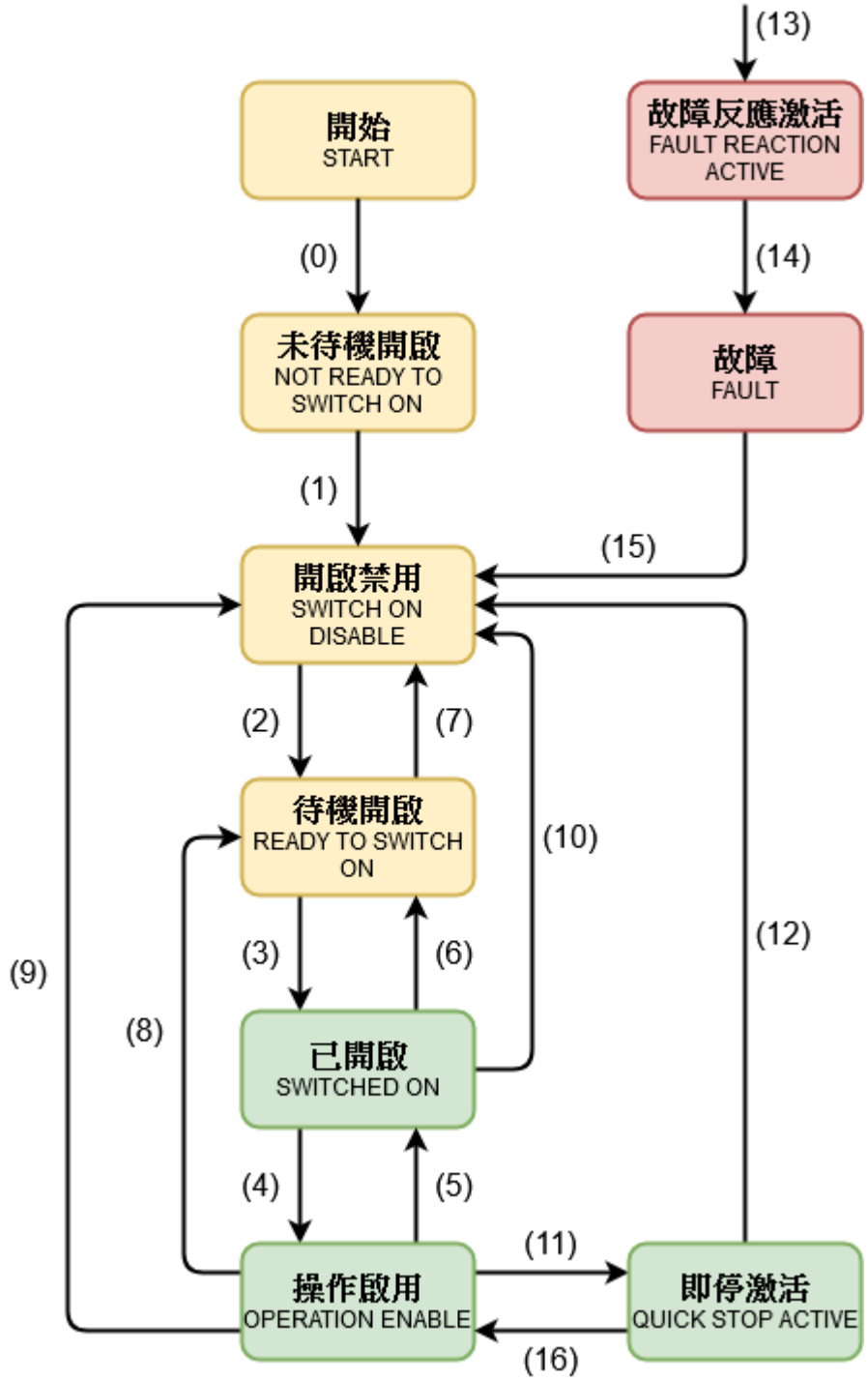


Figure 8. Interrupt example
In the above figure, INT1 is the hardware signal (PIN 7)
IntEnable is the command ECM_CMD_ECAT_INT_SET_ENABLE
IntFlag is located at position 0x10 to 0x13 of the response header
IntClr is located at position 0x10 to 0x13 of the command header

EtherCAT is triggered at t=2 and t=7. IntFlag bit 31 is set at t=2 and t=7, and IntClr (located in the command header 0x10~0x15) is set at t=4. INT1 is only activated when IntFlag (located in the response header 0x10~0x15) and IntEnable (set using ECM_CMD_ECAT_INT_SET_ENABLE) are triggered at the same time, so INT1 is only activated at t=7.

AppendixCiA 402 State Machine



The current CiA402 status can be obtained from Status Word (0x0641:0)

Index	Sub	Name	Data Type	Access	PDO Mapping	Default Value
0x6041	00	Status Word	UINT	RO	AND	*(See 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 shows the various functions that can be activated in different states.
External brake is only supported if present, high power is only supported if the controller has an optional built-in contactor/switch for power use
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 Function	Not on standby Not Ready to Switch On	Enable Disable Switch On Disable	Standby on Ready to Switch On	Enabled Switch On	Operation Enablement Operation Enabled	Stop activation Quick Stop Active	Fault response active Fault Reaction Active	Fault Fault
brake Brake applied, if present	yes Yes	yes Yes	yes Yes	yes Yes	whether(*) Yes/No(*)	whether(*) Yes/No(*)	whether(*) Yes/No(*)	yes Yes
Low power applications Low-level power applied	yes Yes	yes Yes	yes Yes	yes Yes	yes Yes	yes Yes	yes Yes	yes Yes
High power applications High-level power applied	whether(**) Yes/No(**)	whether(**) Yes/No(**)	whether(**) Yes/No(**)	yes Yes	yes Yes	yes Yes	yes Yes	whether(**) Yes/No(**)
Drive function enabled Drive function enabled	no No	no No	no No	no No	yes Yes	yes Yes	yes Yes	no No
Allow configuration Configuration allowed	yes Yes	yes Yes	yes Yes	yes Yes	no No	no No	no No	yes Yes
Diverter Control Enabled Shunt control enabled	no No	no No	no No	no No	yes Yes	yes Yes	yes Yes	no No

(*) In certain conditions, the brakes can be manually activated or deactivated
In some states, brake could be activated and/or deactivated manually.
(**) On some controllers, the high switch is not displayed and may not be deactivated
In some controllers, high-level switch is not present and therefore high level power could not be deactivated.

process Transition	event Event	action Action

0	Automatic process after power-on or reset Automatic transition after power-on or reset application.	Automatic testing or initialization of drive units Drive device self-test and/or self initialization is performed.
1	Automatic process after initialization Automatic transition after initialization.	Communication Activation Communications are activated.
2	Receive "power off command" from control device or local signal Shutdown command received from control device or local signal.	-
3	Receives a "turn on command" from a control device, activation signal or local signal Switch on command received from control device and enable signal is activated (if available) or local signal.	High power on The high-level power is switched on.
4	Receives "operation enable" from a control device, activation enable signal or local signal Enable operation command received from control device and enable signal is activated (if available) or local signal.	Enables the drive function, performs the initial angle determination process and clears all internal set points. The drive function is enabled, Initial angle determination process is executed and all internal set-points cleared.
5	Receive "Operation Disable" from control device, inactive enable signal or local signal Disable operation command received from control device or enable signal is deactivated (if available) or local signal.	Driver function disabled The drive function is disabled.
6	Receives a "power off command" from a control device, a deactivated enable signal or a local signal Shutdown command received from control device or enable signal is deactivated (if available) or local signal.	High power off The high-level power is switched off.

7	Receive "stop" or "disable voltage" from control device or local signal Quick stop or disable voltage command received from control device or local signal.	-
8	Receive "power off" from control device or inactive enable signal Shutdown command received from control device or enable signal is deactivated (if available).	Drive function disabled, high power off The drive function is disabled, and the high-level power is switched off.
9	Receive "Disable Voltage Command" from control device or local signal Disable voltage command from control device or local signal.	Drive function disabled, high power off The drive function is disabled, and the high-level power is switched off.
10	Receive "disable voltage" or "immediate stop" from control device or local signal Disable voltage or quick stop command received from control device or local signal.	High power off The high-level power is switched off.
11	Receive "instant stop" from control device or local signal Quick stop command received from control device or local signal.	Instant stop function starts The quick stop function is started.
12	Automatic process when receiving "Instant Stop" function completion or "Voltage Disable Command" from the control device Automatic transition when the quick stop function is completed or disable voltage command is received from the control device (depending on 0x605A - Quick stop option code).	Drive function disabled, high power off The drive function is disabled, and the high-level power is switched off.
13	Fault signal Fault signal.	Configuring Fault Response Execution The configured fault reaction function is executed.

14	Automatic process Automatic transition.	Drive disabled, high power off The drive function is disabled, and the high-level power is switched off.
15	Receive "Fault Reset Command" from control device or local signal Fault reset command received from control device or local signal.	When no fault occurs, the drive resets the state and executes; when leaving the fault state, the fault reset bit in the control word will be cleared by the control device. 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 controlword is cleared by the control device.
16	Receive "operation command enable" from the control device Enable operation command from control device.	Drive function enabled The drive function is enabled.

Caution:

※When the drive mode is disabled, the motor will not be powered. The target and set point will not work in this state.

When the 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.

※The high power supply can only be turned off in a system with a corresponding contactor or switch.

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

※ The enable signal will only be affected by the available corresponding registers, see Enable/Disable Input for details

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

Change log

Version	Update time	Narrative
03	2020.10.15	Updated version to English version 03 (Update interrupt, ADC, DAC, GPIO, QEI and FIFO clear functions and re-arrange all descriptions)

031	2020.10.15	Update set/get Tx/Rx FIFO count
032	2021.01.18	Update 0x09 error command status Update ECM_CMD_ECAT_PDO_CONFIG_SET command description
033	2021.01.29	renew 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
034	2021.02.18	SPI packet data length correction
035	2021.03.11	Corrected timeout unit (μs), ECM_CMD_CRC_TYPE_SET Added ECM_CMD_ECAT_INIT_DC_OP、 ECM_CMD_ECAT_STATE_UPDATE
036	2021.07.09	Added ECM_CMD_FIFO_INIT Corrected ECM_CMD_ECAT_INIT_OP description Added DAC delay time description
037	2021.07.29	Added description of ECM_CMD_ECAT_INT_SET_ENABLE and ECM_CMD_ECAT_INT_GET_ENABLE Correction Interrupt Chart Description
038	2021.08.04	Added ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_GET and ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_SET
038	2021.08.10	Added ECM_CMD_402_CONFIG_SET internal 402 status switch
039	2021.09.22	New GPIO
040	2021.10.22	Update ECM_CMD_402_CTL_SET
041	2021.12.30	Update response packet header field error/status description Update command packet header field control character description Update ECM_CMD_ECAT_PDO_DATA_OP
042	2022.02.14	Update unit description
043	2022.05.14	Update ECM_CMD_ASYNCCMD_RESET description
044	2022.08.08	Corrected ECM_CMD_ECAT_WKC_CONTI_ERR_MAX_SET description Added header file GPIO field description

047	2023.03.15	Update some command descriptions
048	2024.01.01	Added SDO Abort Code description Added ECM_CMD_ECAT_SET_FIFO_TH description and new command description
049	2024.05.22	Corrected ECM_CMD_ECAT_INT_SET_ENABLE command description