ECM Starter Kit

ECM-SK

User Guide

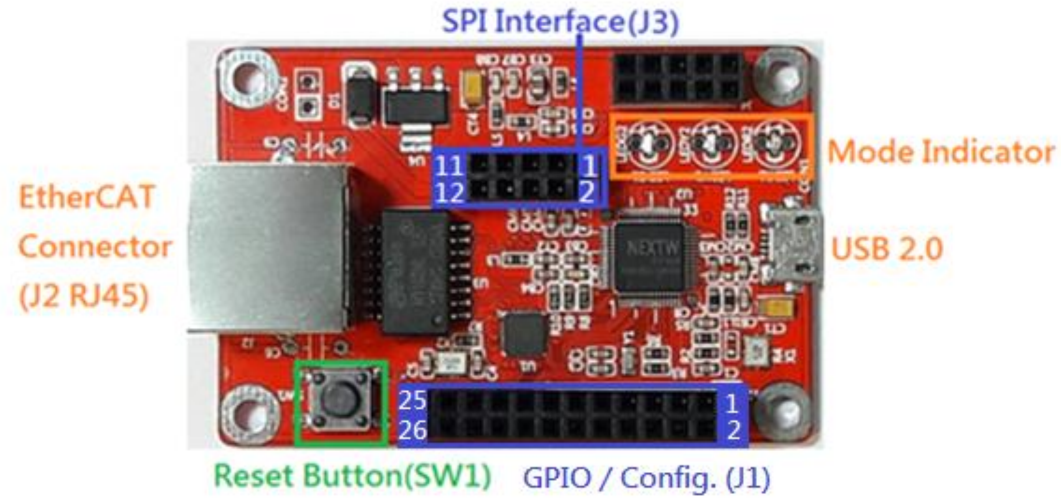
Versions: V.1.7.4

Date: 2022.05

Contents

Chapter 1 Usage of steps

1.1 Board Pin Description (V1.0)



1.1.1 GPIO / CONFIG (J1) pin description

Pin 11	**Pin 9**	**Pin 7**	**Pin 5**	**Pin 3**	**Pin 1**
Input 5	Output 0	Input 3	Reserved	Reserved	3.3V Out
Pin 12	**Pin 10**	**Pin 8**	**Pin 6**	**Pin 4**	**Pin 2**
CONFIG0	Output 1	Input 1	Input 2	Reserved	Reserved

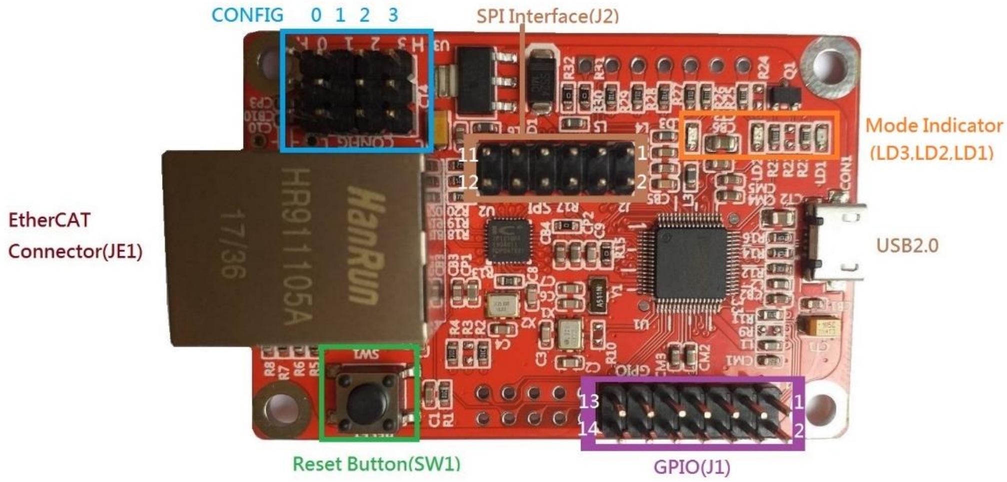
Pin 25	**Pin 23**	**Pin 21**	**Pin 19**	**Pin 17**	**Pin 15**	**Pin 13**
Reserved	Output 5	Output 4	CONFIG3	CONFIG2	Output 2	CONFIG1
Pin 26	**Pin 24**	**Pin 22**	**Pin 20**	**Pin 18**	**Pin 16**	**Pin 14**
GND	Reserved	Input 4	Input 0	Reserved	Busy (Output)	Output 3

* Note: The J1 Input / Output pin is directly provided by the ECM IC. Please design an appropriate isolation circuit to avoid IC damage. 3.3V is High level and 0V is Low level. Please refer to the EC01M data sheet.

1.1.2 SPI(J3) pin description

Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
/RESET	SPI_MISO	SPI_/SS	Busy (Output)	Power In (3.3V)	Reserved
Pin 12	**Pin 10**	**Pin 8**	**Pin 6**	**Pin 4**	**Pin 2**
Reserved	SPI_MOSI	SPI_CLK	Reserved	Reserved	GND

1.2 Board Pin Description (V1.1)



1.2.1 GPIO (J1) pin description

Pin 13	Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
Input 5	Input 4	Input 3	Input 2	Input 1	Input 0	3.3V Out
Pin 14	**Pin 12**	**Pin 10**	**Pin 8**	**Pin 6**	**Pin 4**	**Pin 2**
Output 5	Output 4	Output 3	Output 2	Output 1	Output 0	GND

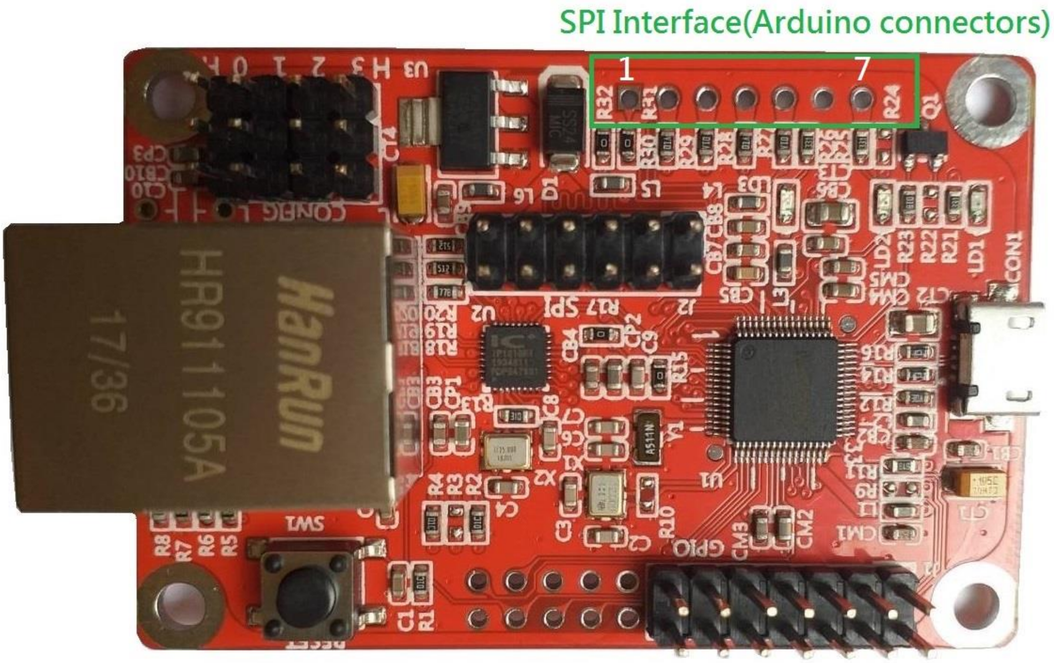
* Note: The J1 Input / Output pin is directly provided by the ECM IC. Please design an appropriate isolation circuit to avoid IC damage. 3.3V is High level and 0V is Low level. Please refer to the EC01M data sheet.

1.2.2 SPI(J2) pin description

Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
/RESET	SPI_MISO	SPI_/SS	Busy (Output)	Power In (3.3V)	Reserved

Pin 12	Pin 10	Pin 8	Pin 6	Pin 4	Pin 2
Reserved	SPI_MOSI	SPI_CLK	Reserved	Reserved	GND

1.2.3 SPI(Arduino Connectors) pin description



Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7
Busy (Output)	SPI_/SS	SPI_MOSI	SPI_MISO	SPI_CLK	GND	Power In (3.3V)

* Note：SPI (Arduino Connectors) has the same function as SPI (J2). Do NOT operate at the same time.

* Pin7 Power In: It is recommended to provide 3.3V more than 200mA.

1.3 LED and CONFIG pin Description

1.3.1 LED indicator description

Component mark (Color)	Meaning
LEDR1(Red)	Power
LEDY1(Yellow)	Status LED Y
LEDG1(Green)	Status LED G

1.3.2 LED status description

LEDY1（Yellow）	LEDG1（Green）	Status
ON	ON	No EtherCAT slave found or slave did not reach the expected state
OFF	OFF	Init State or Pre-Operational State
ON	OFF	Safe-Operational State
OFF	ON	Operational State

1.3.3 Network (RJ45) LED status description

Network (RJ45) LED Y (Yellow): ON means the network speed is 100MHz. (normal)

Network (RJ45) LED G (Green): Flashing means data transmission.

1.3.4 CONFIG Pin status description

CONFIG	Status	Description
CONFIG0	L	Host interface is set as USB
	X	Host interface is set as SPI (Default)
	H	
CONFIG1	X	Normal mode (Default)
	L	
	H	Test mode
CONFIG2	X	Data size of each slave is 12 Bytes (Default)
	L	
	H	Data size of each slave is 16 Bytes
CONFIG3	L	FIFO abandon disable
	X	FIFO abandon enable (Default)
	H	

X: Floating, L: Low (0V), H: High (3.3V)

CONFIG0 Interface selection with controller
CONFIG1 Mode selection
Normal Mode –Control according to the user's command

Test Mode – After power-on, it will automatically enter the OP state according to the default value (Slave Type: IO, Cycle Time: 1ms), and periodically send different output commands to all slaves.

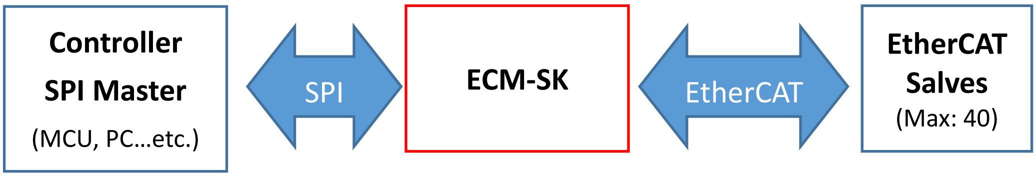
CONFIG2 Data length selection for a single slave

CONFIG3 Selection of the longest transmission time for a single command

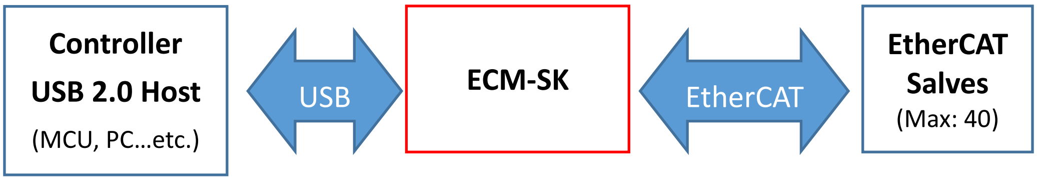
L: disable (Unlimited transmission time for a single command)

H: enable (The maximum transmission time of a single command is 100 cycles. If it exceeds, the whole command will be abandoned)

1.4 System Connection



Minimum communication cycle 250us



Minimum communication cycle 500us

*** The transmission interface (SPI or USB) determined by CONFIG0 (J1 Pin12).**

*** USB communication speed depends on USB host chip performance.**

1.5 Wiring between ECM-SK SPI and Controller

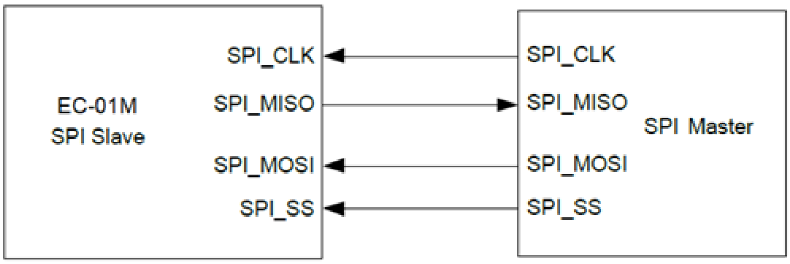


Table1.1 SPI pin name and meaning

Name	J3 Pin	Meaning	Description
Busy	Pin5	ECM Busy	ECM is busy and does not accept new SPI transfers.
SCLK	Pin8	Serial Clock	Generated and controlled by the SPI Master, with minimum clock requirements for controlling the EtherCAT communication cycle time.
MOSI	Pin10	Master out and slave in	SPI master data out and SPI slave data in.
MISO	Pin9	Master in and slave out	SPI master data in and SPI slave data out.
/SS	Pin7	slave select	The selection signal is controlled by the Master. The slave will respond when the /SS signal is low.

1.6 SPI Specification and Transmission Mode

The SPI is used between the ECM and the host controller. The wiring between the ECM and the host controller can be directly connected to the DuPont line at low frequencies (below 10 MHz). At high frequencies, it is recommended to connect directly with silver wire to avoid noise interference. SCLK is provided by the host controller and has the minimum clock requirement for different communication cycles. SCLK supports up to 24MHz. The ECM supports 40 slaves (each slave contains 12-byte data), or supports 30 slaves (each slave contains 16-byte data). Minimum requirement for the SCLK is shown in Table 1.2.

Table 1.2 Minimum requirement for the SCLK

EtherCAT Cycle Time	Minimum SCLK	Wiring
250us	24 MHz	Silver wire or PCB
1ms	6 MHz	DuPont line

SPI transfer instructions

The SPI of the ECM is in slave mode. When idle, it is low. It is sent at the falling edge and received at the rising edge. The high-order data is transmitted first (MSB). Please refer to the following figure.

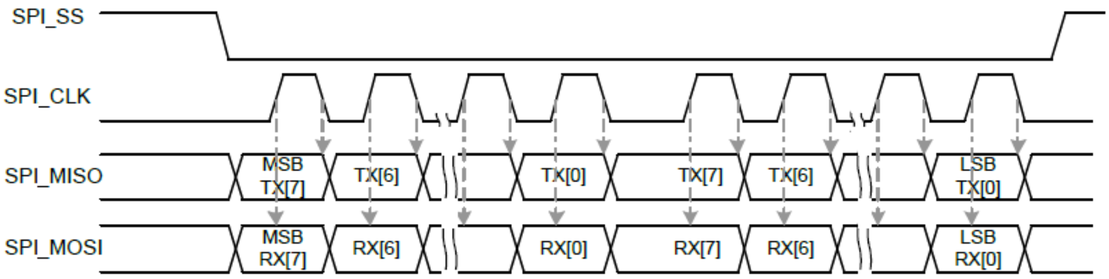


Figure 1.1 SPI Timing

The SPI of the host controller is the master mode, which needs to generate SCLK and provide it to the SPI slave, and send it on the rising edge and receive it at the falling edge. The SPI transmission is in Bytes. Each time it will be transmitted from the lower address byte to the highest address in sequence. In other words, the SPI transmission will start from Byte0, Byte1, Byte2... to the last Byte. The MSB mode is adopted. The high-order element is transmitted first when the SPI transmits a single Byte.

1.7 Description of SPI Transmission Timing

The host controller can communicate with the ECM through the SPI. When the host controller detects that the SPI BUSY pin (J1 Pin16 or J3 Pin5) is low, the SPI communication can be started. When the SPI communication is started, the SPI BUSY pin rises to high immediately. The SPI Busy signal will return to low level when the ECM has processed the command packet. The host controller can send the next command via the SPI when BUSY pin return to low level. The detailed timing figure is as follows:

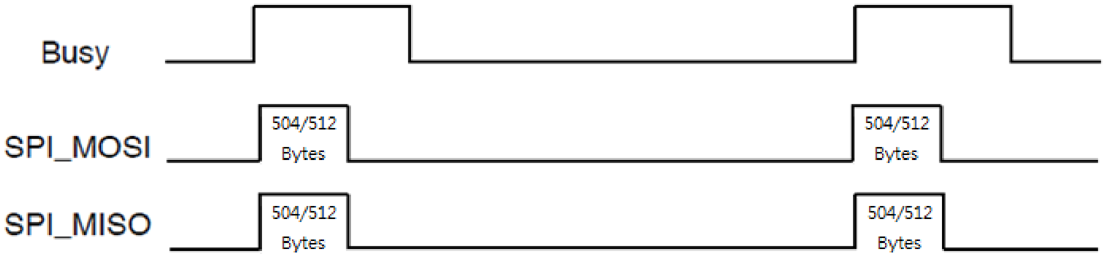


Figure 1.2 SPI Signal Timing Figure

To achieve the shortest cycle 250us communication requirement, the SPI transmission frequency must be 24MHz. When the host controller detects that BUSY pin is low, it must exchange the next command immediately. If the application has a slower communication period (for example, slower than 1ms), the host controller can be delayed by a fixed time (for example, 800us) and periodically transmitted command, so that the BUSY signal could be ignored, which makes the design simpler.

1.8 EtherCAT State Flow

EtherCAT can be divided into 4 execution states, Init, PreOP, SafeOP, and OP. Different states can execute different commands. In the PreOP state, the slave type and the drive type can be set; the SafeOP state can perform non-periodic data exchange, such as reading and writing Service Data Objects (SDO); the OP state can perform periodic data exchange, that is, exchange fixed data in a fixed period. The OP state also can perform non-periodic data exchange. The user can complete all operations only in the PreOP state and the OP state.

The switching time between EtherCAT states will vary depending on the EtherCAT slaves (related to the slave brand, performance, quantity and others). The user can confirm the current status from the return value of the GET_STATUS command. The detailed process and executable commands are shown below:

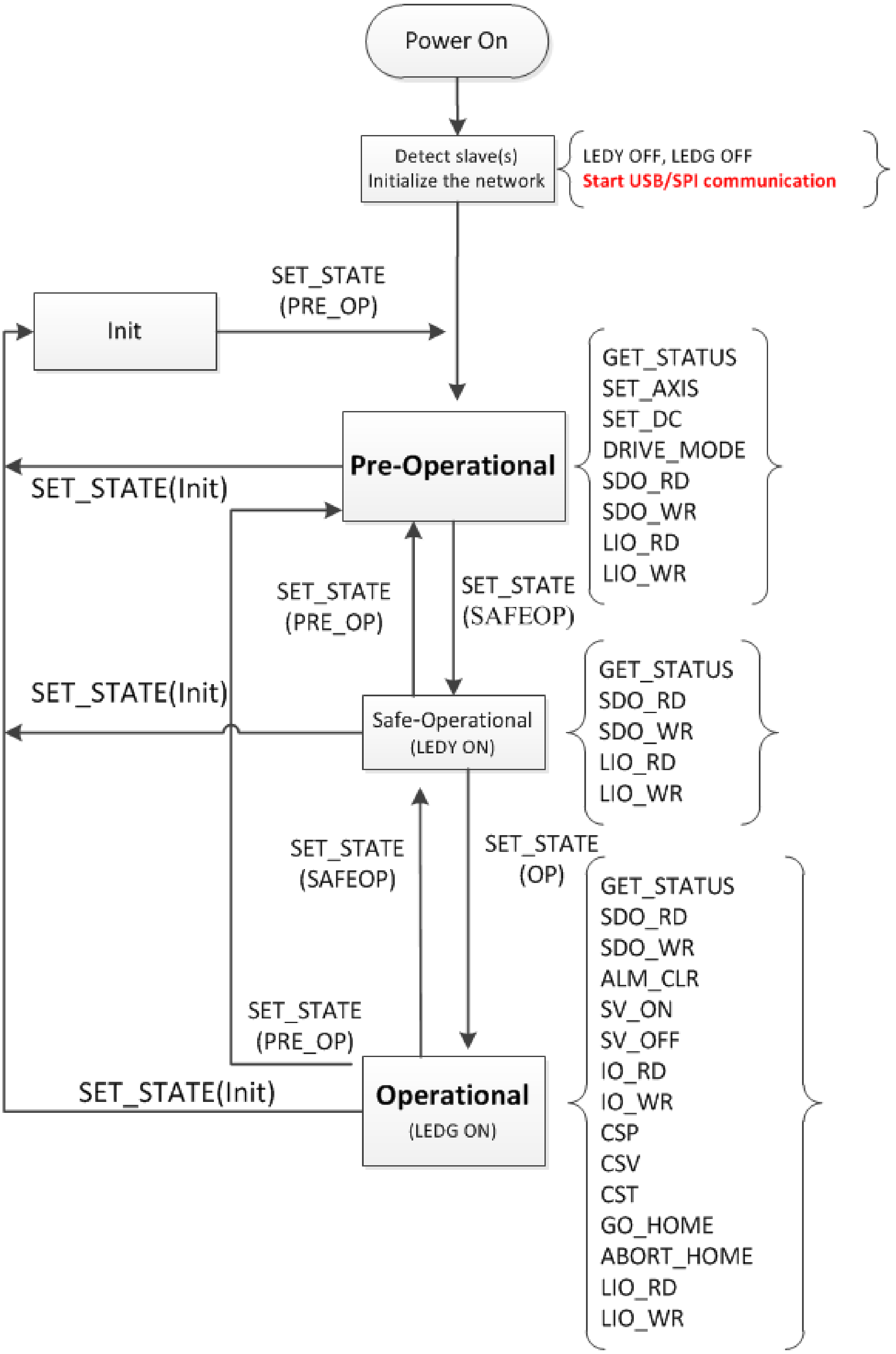


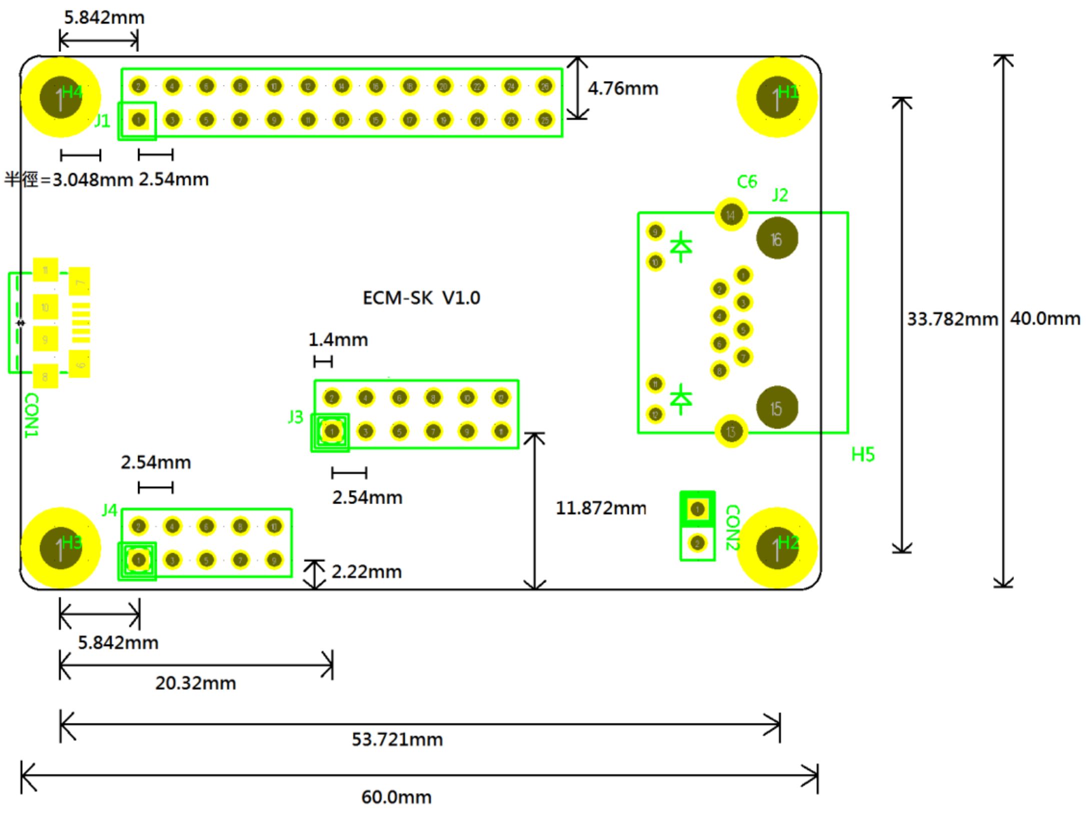
Figure 1.3 EtherCAT Sates Flow

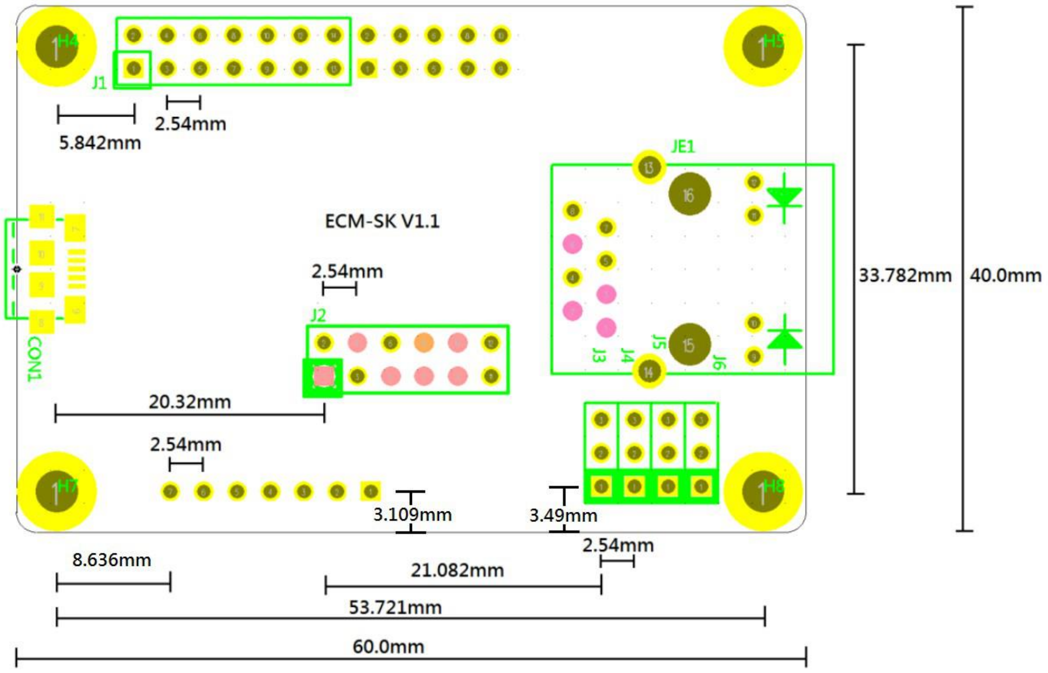
Table 1.3 Command List

Cmd. ID	Command	Operation	ECM	Drive	IO	Init	PreOP	SafeOP	OP	Ref.
0x00	GET_STATUS	Get Status	V	V	V	V	V	V	V	2.1.1
0x01	SET_STATE	Set EtherCAT state	V			V	V	V	V	2.1.2
0x02	SET_AXIS	Set type of slave	V			V	V			2.1.3
0x03	SET_DC	Set DC Mode	V			V	V			2.1.4
0x04	SET_EX	Set Extensions (CRC)	V			V	V	V		2.1.5
0x05	SET_FIFO	Set FIFO	V						V	2.1.6
0x06	DRIVE_MODE	Set mode of servo drive		V		V	V			2.1.7
0x07	SDO_RD	Service data object read		V	V		V	V	V	2.1.8
0x08	SDO _WR	Service data object write		V	V		V	V	V	2.1.9
0x10	ALM_CLR	Alarm clear		V					V	2.1.10
0x11	SV_ON	Servo ON		V					V	2.1.11
0x12	SV_OFF	Servo OFF		V					V	2.1.12
0x13	IO_RD	Digital Input			V				V	2.1.13
0x14	IO_WR	Digital Output			V				V	2.1.14
0x15	CSP	Position control		V					V	2.1.15
0x16	CSV	Velocity control		V					V	2.1.16
0x17	CST	Torque control		V					V	2.1.17
0x18	GO_HOME	Start Homing Procedure		V					V	2.1.18
0x19	ABORT_HOME	Abort Homing Procedure		V					V	2.1.19
0x21	LIO_RD	Read ECM IC input	V			V	V	V	V	2.1.20

0x22	LIO_WR	Write ECM IC output	V			V	V	V	V	2.1.21
0xBB	SW_RESET	Software Reset	V			V	V	V	V	2.1.22

1.9 Dimensions and holes

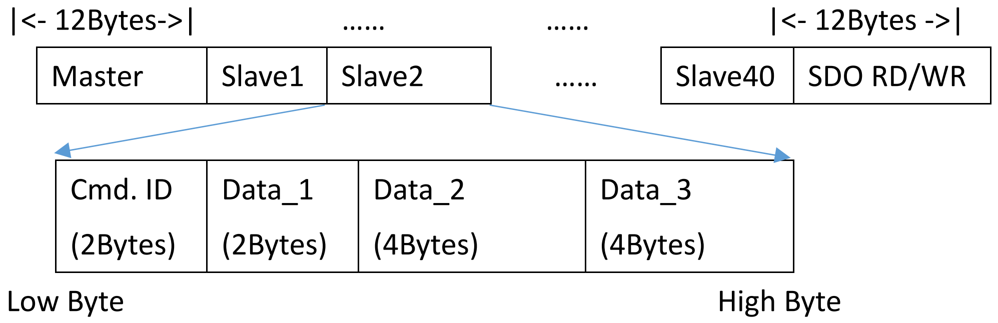




Chapter 2 Command and Response

2.1 Command and Response Data Structure

The host controller can send commands to each slave to control or obtain the status in different states. Each command is in the form of 12Bytes or 16Bytes according to the setting of CONFIG 2 (J1 Pin17). The first command corresponds to the ECM, the second command corresponds to the first slave, and the third command corresponds to the second slave, and so on. If the data length is 12 Bytes, 40 slaves can be controlled. If it is 16 Bytes, 30 slaves can be controlled. The last command corresponds to a single slave for non-periodic parameter reading and writing (SDO Read/Write). When CONFIG 2 is Low, the ECM01 can connect up to 40 slaves, so each time the command is sent, it will pass (40+2) * 12 =504 Bytes of data, and will get (40+2) * 12 =504 Bytes of response.



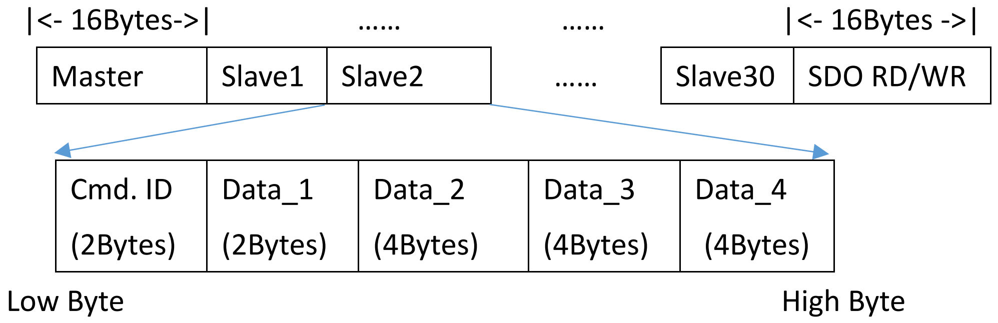
|<- 12Bytes->| …… …… |<- 12Bytes ->|

Master	Slave1	Slave2	……	Slave40	SDO RD/WR

Cmd. ID (2Bytes)	Data_1 (2Bytes)	Data_2 (4Bytes)	Data_3 (4Bytes)

Low Byte High Byte

When CONFIG 2 is High, the ECM01 can connect up to 30 slaves, so each time the command is sent, it will pass (30+2) * 16 =514 Bytes of data, and will get (30+2) * 16 =512 Bytes of response.



|<- 16Bytes->| …… …… |<- 16Bytes ->|

Master	Slave1	Slave2	……	Slave30	SDO RD/WR

Cmd. ID (2Bytes)	Data_1 (2Bytes)	Data_2 (4Bytes)	Data_3 (4Bytes)	Data_4 (4Bytes)

Low Byte High Byte

If the data is stored in Little Endian, the low byte is placed at the lower address. For example, Cmd. ID is 2-Byte data. If the value is 0x0001, the low byte 0x01 will be placed in Byte0, and the high byte 0x00 will be placed in Byte1. As another example, Data_2 is 4-Byte data, and assumed value is 0x87 65 43 21. The lowest byte value 0x21 will be placed at the lowest address Byte0, then 0x43, 0x65, and the highest byte 0x87 is placed at the highest address Byte3.

The SPI transmission is in Bytes. Each time it will be transmitted from the lower address, and will be transmitted to the highest address in sequence. In other words, the SPI transfer will start from Byte0, and then pass Byte1, Byte2... to the last Byte. When the SPI transmits a single Byte, the MSB mode is adopted, that is, the high-order element is transmitted first.

The result of this command will be returned at the next command. However, some commands execution may exceed one cycle time (such as SET_STATE, SDO_RD, SDO_WR, GO_HOME, etc.). The result can be obtained using the GET_STATUS command after several cycles.

2.1.1 GET_STATUS Command

- Command Description ： Get Status
- Command ID : 0x0000
- State ： All state
- Category ： All category

Table 2.1 GET_STATUS Command

BYTE	Definition	Command	Response
0	Command ID	0x0000	0x00 or 0xBF
1			
2	Reserved	--	Pram
3	CRC8 Value**	CRC8 Value	
4	Reserved	--	Data1
5			
6			
7			
8	Reserved	--	Data2

9			
10			
11			
12	Reserved*	--	Data3
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

Hint: Byte 0 of Response is 0xBF in the OP state, and is 0x00 in the other states. Byte1~Byte11 will update data (OP State) or retain the result (other states). For more information, please refer to 2.2 Response Data.

2.1.2 SET_STATE Command

- Command Description：Set EtherCAT State
- Command ID : 0x0001
- State：All State
- Category：ECM(The First Command)

Table 2.2 SET_STATE Command

BYTE	Definition	Command	Response
0	Command ID	0x0001	0x01
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	EtherCAT State	Requested STATE	--
5	-INIT: 0x01		
6	-PRE_OP: 0x02		
7	-SAFE_OP: 0x04 -OP: 0x08		
8	Reserved	--	--
9			
10			
11			

12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

Error Code CRC error count

Current State

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

Hint: Changing the EtherCAT state takes several cycles and is related to the number of connected slaves. The result can be obtained using the GET_STATUS command after several cycles.

2.1.3 SET_AXIS Command

- Command Description：Set Type of Slave (IO, Drive or HSP). Supports up to 40 axes.
- Command ID : 0x0002
- State：Pre-Operational State
- Category：ECM(The First Command)

Table 2.3 SET_AXIS Command

BYTE	Definition	Command	Response
0	Command ID	0x0002	0x02
1			Error Code
2	Group (0~4)	Group (Default:0)	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Topology -Drive : 0x0 -IO : 0x1	Topology (Default:0x1)	SlaveCount - Number of detected slaves
5			
6			

7	-HSP : 0x2		
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

Group

An ECM can connect up to 40 slaves (Drive, IO or NEXTW HSP). 8 slaves can be set at a time. Group 0 represents set slave 1~8, Group 1 represents set slave 9~16, and so on. Value of Group can be 0~4.

SlaveCount

Number of detected slaves. It returns 0 in the Init state, and returns the number of detected slaves in the Pre-Operational state.

Topology

Bit28	Bit24	Bit20	Bit16	Bit12	Bit8	Bit4	Bit0
Slave n+8	Slave n+7	Slave n+6	Slave n+5	Slave n+4	Slave n+3	Slave n+2	Slave n+1

n = Group * 8

0: Drive

1: IO

2: NEXTW HSP (High Speed Pulse)

Error Code CRC error count

Current State

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

2.1.4 SET_DC Command

- Command Description：Set ECM Cycle Time
- Command ID : 0x0003
- State：Pre-Operational State
- Category：ECM(The First Command)

Table 2.4 SET_DC Command

BYTE	Definition	Command	Response
0	Command ID	0x0003	0x03
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Cycle Time (us) Should be 250 x 2^n (Minimum : 250)	Cycle Time (Default:1000)	--
5			
6			
7			
8	**FIX Value**	**0xFFFF** (Default:0xFFFF)	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

FIX Value

Fix value. Always 0xFFFF0000.

Error Code CRC error count

Current State

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

2.1.5 SET_EX Command

- Command Description：Set CRC (It will take effect after the next boot)
  - Command ID : 0x0004
  - State：Init State、Pre-Operational State、Safe-Operational State
  - Category：ECM(The First Command)

Table 2.5 SET_EX Command

BYTE	Definition	Command	Response
0	Command ID	0x0004	0x04
1			Error Code
2	Reserved	--	Current State
3			CRC8 Value
4	Extension Setting 1~2	Extension Setting Value	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

* It will take effect after the next boot

Extension Setting

Bit28	Bit24	Bit20	Bit16	Bit12	Bit8	Bit4	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Setting2	Setting1

Setting 1: Command CRC Verification

0: Disable Command CRC Verification

1: Enable Command CRC Verification

Setting 2: Response CRC Verification

0: Disable Response CRC Verification

1: Enable Response CRC Verification

2.1.6 SET_FIFO Command

- Command Description：Set FIFO
- Command ID : 0x0005
- State：Operational State
- Category：ECM(The First Command)

Table 2.6 SET_FIFO Command

BYTE	Definition	Command	Response
0	Command ID	0x0005	0x05
1			Error Code
2	Action	Action(1~2)	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Value	Value (>=0)	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			

15			

* BYTE12-15 is available only when CONFIG 2 is High level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

Action

0x0001 : Clear FIFO content and FIFO threshold setting.

0x0002 : Set FIFO threshold to **Value**.

In the OP state, when the remaining space of the FIFO is less than the Value, it starts to fetch a piece of data for execution in each cycle, and automatically reset the minimum value to 160 after the conditions are met.

2.1.7 DRIVE_MODE Command

- Command Description：Set Mode of Servo Drive
- Command ID : 0x06
- State：Pre-Operational State
- Category：Drive or NEXTW HSP

Table 2.7 DRIVE_MODE Command

BYTE	Definition	Command	Response
0	Command ID	0x06	0x06
1		Command Index	Command Index
2	Reserved	--	--
3			
4	OP_Mode	OP_Mode	--
5	-CSP: 0x08	(Default:0x08)	
6	-CSV: 0x09		
7	-CST: 0x0A		
8	DRIVE_TYPE	DRIVE_TYPE	--
9	- FREERUN: 0x00	(Default:0x01)	
10	- DCSYNC: 0x01		
11			
12	Reserved*	--	--
13			
14			

15			

* BYTE12-15 is available only when CONFIG 2 is High level.

- Command Index：The user-defined number, the Command Index of the command and response will be the same.
- OP_MODE：

  CSP(Cyclic Synchronous Position Mode)：

  The ECM periodically sends PDOs. In CSP mode the target absolute position command is transmitted to all the slaves when PDO is transmitted.

  CSV(Cycle Synchronized Velocity Mode)：

  The ECM periodically sends PDOs. In CSV mode the target speed command is transmitted to all the slaves when PDO is transmitted.

  CST (Cyclic Synchronous Torque Mode)

  The ECM periodically sends PDOs. In CST mode the target torque command is transmitted to all the slaves when PDO is transmitted.

- DRIVE_TYPE:

  FREE RUN

  Each slave is asynchronous, and each slave processes EtherCAT data according to its own local time. It is independent of master time, the other slaves, and the arrival time of the EtherCAT data.

  DCSYNC

  DCSYNC requires master to have strong real-time performance. DCSYNC is a high-precision time synchronization mode. In DCSYNC mode all slaves are synchronized with the first slave. The synchronization signal of all slaves is generated by taking the first slave local time as the reference time, plus the time delay such as transmission delay and jitter.

2.1.8 SDO_RD Command

- Command Description：Read SDO (Service Data Objects)
- Command ID : 0x0007
- State：Pre-Operational State, Safe-Operational State or Operational State
- Category：SDO RD/WR (The Last Command)

Table 2.8 SDO _RD Command

BYTE	Definition	Command	Response
0	Command ID	0x0007	0x0007
1			
2	Slave Index	Slave Index(1~40)	Slave Index
3			
4	Object Index	Object Sub Index	Object Sub Index
5			
6		Object Index	Object Index
7			
8	Value	--	Value
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Index

Bit16 Bit0

Object Index	Object Sub Index

Hint: SDO_RD command takes several cycle time. The result can be obtained using the GET_STATUS command after several cycles.

2.1.9 SDO_WR Command

- Command Description：Write SDO (Service Data Objects)
- Command ID : 0x0008
- State：Pre-Operational State、Safe-Operational State、Operational State
- Category：SDO RD/WR (The Last Command)

Table 2.9 SDO_WR Command

BYTE	Definition	Command	Response
0	Command ID	0x0008	0x0008
1			
2	Slave Index	Slave Index (1~40)	Slave Index
3	Size	Object Size	
4	Object Index	Object Sub Index	Object Sub Index
5			
6		Object Index	Object Index
7			
8	Value	Object Value	Object Value
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Index

Bit16 Bit0

Object Index	Object Sub Index

Hint: Hint: SDO_WR command takes several cycle time. The result can be obtained using the GET_STATUS command after several cycles.

2.1.10 ALM_CLR Command

- Command Description：Clear Alarm of Slave
- Command ID : 0x10
- State：Operational State
- Category：Drive

Table 2.10 ALM_CLR Command

BYTE	Definition	Command	Response
0	Command ID	0x10	0x10

1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Hint: Not all alarms can be cleared. Please refer to the Slave Manual.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

2.1.11 SV_ON Command

- Command Description：Set Servo ON
- Command ID : 0x11
- State：Operational State
- Category：Drive、NEXTW HSP

Table 2.11 SV_ON Command

BYTE	Definition	Command	Response
0	Command ID	0x11	0x11
1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			

8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

Hint: The SV_ON command will be sent to the slave in one communication cycle, but the time required for servo on may be different for each slave. The user can get the latest Statusword through GET_STATUS command. The Statusword can determine the status of the Servo on/off of the slave.

2.1.12 SV_OFF Command

- Command Description：Set Servo OFF
- Command ID : 0x12
- State：Operational State
- Category：Drive、NEXTW HSP

Table 2.12 SV_OFF Command

BYTE	Definition	Command	Response
0	Command ID	0x0012	0x12
1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			

11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

Hint: The SV_OFF command will be sent to the slave in one communication cycle, but the time required for servo off may be different for each slave. The user can get the latest Statusword through GET_STATUS command. The Statusword can determine the status of the Servo on/off of the slave.

2.1.13 IO_RD Command

- Command Description：Read Digital Input Status
- Command ID : 0x0013
- State：Operational State
- Category：IO Slave

Table 2.13 IO_RD Command

BYTE	Definition	Command	Response
0	Command ID	0x0013	0x13
1			--
2	Reserved	--	--
3			
4	Input - Input Status	--	Input
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--

13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

2.1.14 IO_WR Command

- Command Description：Write Digital Output Values
- Command ID : 0x14
- State：Operational State
- Category：IO Slave

Table 2.14 IO_WR Command

BYTE	Definition	Command	Response
0	Command ID	0x14	0x14
1		Command Index	Command Index
2	Reserved	--	--
3			
4	Output - Output Value	Output	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

2.1.15 CSP Command

- Command Description：Set Cyclic Synchronous Position Command

- Command ID : 0x15
- State：Operational State
- Category：Drive or NEXTW HSP (OP_Mode set to CSP)

Table 2.15 CSP Command for Drive and 1 channel HSP

BYTE	Definition	Command	Response	Master Cyclic Response
0	Command ID	0x15	0x15	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive status	--	Statusword	Current State
3				CRC8 Value
4	Position	Target Position (Absolute Position)	Current Position	--
5				
6				
7				
8	Status	Output Value	Current Torque **	FIFO Remaining
9				
10			ALM Code**	FIFO Full Count (The number of commands rejected due to FIFO full)
11				
12	Reserved*	--	--	--
13				
14				
15				

* BYTE12-15 is available only when CONFIG 2 is High level.

** The HSP (High Speed Pulse) slave return Input status due to HSP has no ALM Code. The Drive slave return Current Torque and ALM Code.

Cmd. Index：The user-defined number, the Command Index of the command and response will be the same.

Error Code CRC error count

Current State: Should be 0x08 (OP).

Output Value Set Output inside the Drive or HSP slave. (Some drive slaves may NOT support.)

Hint: The CSP command is to set the target absolute position of the slave. It is recommended that users plan the acceleration and deceleration. Please send a CSP command at least in one communication cycle. The user can send two or more CSP commands in one communication cycle. Commands that have not been sent to the slave are temporarily stored in the FIFO. The space of the FIFO is limited. **Please note that if the FIFO has no space to store, the command will be ignored.** The user should check the FIFO Remaining in the master response. If the remaining space is too small, it is recommended to suspend the communication after several communication cycles.

Table 2.16 CSP Command for 2-channel HSP

BYTE	Definition	Command	Response	Master Cyclic Response
0	Command ID	0x0015	0x15	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive status	--	Statusword	Current State
3				CRC8 Value
4	Position In/Out status	Target Position 1 (Absolute Position)	Current Position 1	--
5				
6				
7				
8		Output Value 1	Input Status 1	FIFO Remaining
9				
10		Target Position 2(Absolute Position)	Current Position 2	FIFO Full Count (The number of commands rejected due to FIFO full)
11				
12				
13				
14		Output Value 2	Input Status 2	
15				

Hint: To output two sets of Pulse, CONFIG 2 must set to High level.

2.1.16 CSV Command

- Command Description：Set Cyclic Synchronous Velocity Command
- Command ID : 0x16

- State：Operational State
- Category：Drive or NEXTW HSP (OP_Mode set to CSV)

Table 2.17 CSV Command for Drive and 1 channel HSP

BYTE	Definition	Command	Response	Master Cyclic Response
0	Command ID	0x16	0x16	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword –	--	Statusword	Current State
3	Drive status			CRC8 Value
4	Velocity –Target	Velocity	Current Position	--
5	Velocity			
6				
7	Current Position			
8	ALM Code -	--	Current Torque **	FIFO Remaining
9	Alarm Code			
10			ALM Code**	FIFO Full Count (The number of commands rejected due to FIFO full)
11	Current Torque			
12	Reserved*	--	--	--
13				
14				
15				

* BYTE12-15 is available only when CONFIG 2 is High level.

** The HSP (High Speed Pulse) slave return Input status due to HSP has no ALM Code. The Drive slave return Current Torque and ALM Code.

Cmd. Index：The user-defined number, the Command Index of the command and response will be the same.
Error Code CRC error count
Current State: Should be 0x08 (OP).

Hint: The CSV command is to set the target velocity of the slave. It is recommended that users plan the acceleration and deceleration. Please send a CSV command at least in one communication cycle. The user can send two or more CSV commands in one communication cycle. Commands that have not been sent to the slave are temporarily stored in the FIFO. The space of the FIFO is limited. **Please note that if the FIFO has no space to store, the command will be ignored.** The user should check the FIFO Remaining in the master response. If the remaining space is too small, it is recommended to suspend the communication after several communication cycles.

Table 2.18 CSV Command for 2 channel HSP

BYTE	Definition	Command	Response	Master Cyclic Response
0	Command ID	0x16	0x16	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive status	--	Statusword	Current State
3				CRC8 Value
4	Target Velocity	Target Velocity 1	Current Position 1	--
5				
6	Current Position			
7				
8	Output Value	Output Value 1	Input Status 1	FIFO Remaining
9				
10	Input Status	Target Velocity 2	Current Position 2	FIFO Full Count (The number of commands rejected due to FIFO full)
11				
12				
13				
14		Output Value 2	Input Status 2	
15				

Hint: To output two sets of Pulse, CONFIG 2 must set to High level.

Cmd. Index：The user-defined number, the Command Index of the command and response will be the same.

2.1.17 CST Command

- Command Description：Set Cyclic Synchronous Torque Command
- Command ID : 0x17
- State：Operational State

- Category：Drive (OP_Mode set to CST)

Table 2.19 CST Command

BYTE	Definition	Command	Response	Master Cyclic Response
0	Command ID	0x17	0x17	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive status	--	Statusword	Current State
3				CRC8 Value
4	Torque – Target Torque Command Current Position	Torque	Current Position	
5				
6				
7				
8	ALM Code - Alarm Code Current Torque	--	Current Torque	FIFO Remaining
9				
10			ALM Code	FIFO Full Count (The number of commands rejected due to FIFO full)
11				
12	Reserved*	--	--	--
13				
14				
15				

* BYTE12-15 is available only when CONFIG 2 is High level.

Cmd. Index：The user-defined number, the Command Index of the command and response will be the same.

Error Code CRC error count

Current State: Should be 0x08 (OP).

Hint: The CST command is to set the target torque of the slave. It is recommended that users plan the continuous torque output. Please send a CST command at least in one communication cycle. The user can send two or more CST commands in one communication cycle. Commands that have not been sent to the slave are temporarily stored in the FIFO. The space of the FIFO is limited. **Please note that if**

the FIFO has no space to store, the command will be ignored. The user should check the FIFO Remaining in the master response. If the remaining space is too small, it is recommended to suspend the communication after several communication cycles.

2.1.18 GO_HOME Command

- Command Description：Start the Homing Procedure
- Command ID : 0x18
- State：Operational State
- Category：Drive or NEXTW HSP

Table 2.20 GO_HOME Command

BYTE	Definition	Command	Response
0	Command ID	0x18	0x18
1		Command Index	Command Index
2	Statusword –	--	Statusword
3	Drive status		
4	Current Position	--	Current Position
5			
6			
7			
8	ALM Code - Alarm	--	Current Torque
9	Code		
10	Current Torque		ALM Code
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

Hint 1: The modes of operation must be specified by SDO_WR before executing GO_HOME command. Please refer to the slave instructions and confirm the Homing Method type.

Index	SubIndex	Name	Size	Value	Description

6060h	0	Modes of Operation	1 Byte	6	Homing Mode
6098h	0	Homing Method	1 Byte	0~35	Set Homing Method

It can also set the parameters through SDO_WR, including:

Index	SubIndex	Name	Size
607Ch	0	Home Offset	4 Bytes
6099h	1	Speed during search for switch	4 Bytes
6099h	2	Speed during search for zero	4 Bytes
609Ah	0	Homing Acceleration	4 Bytes

Hint 2: It will switch the operation mode back to the original setting mode (CSP / CSV / CST) automatically when go home finished.

Hint 3: The HSP second axis related object index is the first axis index + 0x800. (Please refer to the NEXTW HSP User Manual for details.)

Hint 4: Before the next GO_HOME command, ALL homing slaves must be completed (or terminated) the homing procedure.

2.1.19 ABORT_HOME Command

- Command Description：Terminate GO_HOME Procedure of **ALL** the Slaves
- Command ID : 0x19
- State：Operational State
- Category：Drive or NEXTW HSP

Table 2.21 ABORT_HOME Command

BYTE	Definition	Command	Response
0	Command ID	0x19	0x19
1		Command Index	Command Index
2	Statusword – Drive status	--	Statusword
3			
4	Current Position	--	Current Position
5			
6			
7			
8	ALM Code - Alarm Code Current Torque	--	Current Torque
9			
10			ALM Code
11			

12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Command Index：The user-defined number, the Command Index of the command and response will be the same.

Hint 1: Statusword bit10 and bit12 returned by the slave can know the homing status. Go home procedure has been completed when bit10 and bit12 are both 1. **All** slaves will be forced to terminate the homing procedure using the ABORT_HOME command.

Hint 2: ABORT_HOME command will terminate the ALL slaves homing procedure. (Not terminate single slave)

2.1.20 LIO_RD Command

- Command Description：Read Input Status on the ECM IC
- Command ID : 0x0021
- State：All state
- Category：ECM(The First Command)

Table 2.22 LIO_RD Command

BYTE	Definition	Command	Response
0	Command ID	0x0021	0x21
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Input Status	--	Input Status
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--

13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Note: The LIO pin is directly provided by the ECM IC. Please design an isolation circuit to avoid IC damage. 3.3V is High level and 0V is Low level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

2.1.21 LIO_WR Command

- Command Description：Write Output Values on the ECM IC
- Command ID : 0x0022
- State：All state
- Category：ECM(The First Command)

Table 2.23 LIO_WR Command

BYTE	Definition	Command	Response
0	Command ID	0x0022	0x22
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Output Value	Output Value	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* BYTE12-15 is available only when CONFIG 2 is High level.

Note: The LIO pin is directly provided by the ECM IC. Please design an isolation circuit to avoid IC damage. 3.3V is High level and 0V is Low level.

** For the CRC8 Value, please refer to 2.2 Response Data. To enable the CRC check or not can be set through 2.1.5 SET_EX

2.1.22 SW_RESET Command

- Command Description：Reset
- Command ID : 0x00BB
- State：All state
- Category：ECM(The First Command)

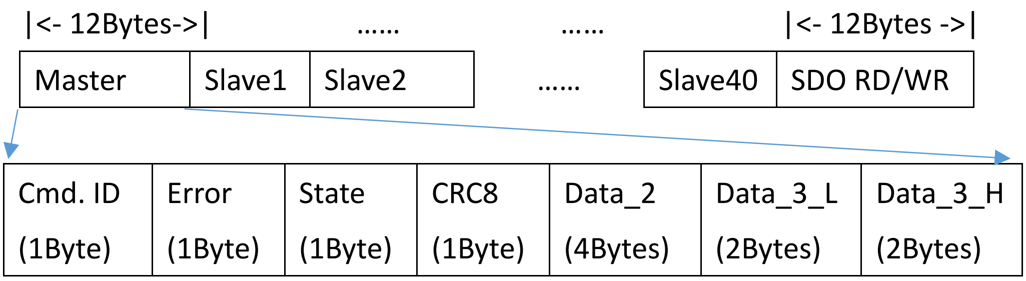
Table 2.24 SW_RESET Command

BYTE	Definition	Command	Response
0	Command ID	0x00BB	--
1			
2	Reserved	--	--
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* After Reset, all variables will be reset, and the opened USB connection will be interrupted and closed. To continue the transfer, the USB connection must be reopened.

2.2 Response Data

Command and response data are delivered simultaneously. Due to the data transfer cycle, the response data received in the current time is the result of the last command. Some commands (such as SET_STATE, SDO_RD, SDO_WR, etc.) will respond after several cycles. The first 12 Bytes of the response data are the Master information. The first 4 Bytes of the Master information are Cmd. ID, Error Code, Current State and CRC8 Value. When CONFIG 2 is Low level, the response data of each slave is 12 bytes, and its meaning is as follows:


|<- 12Bytes->| …… …… |<- 12Bytes ->|

Master | Slave1 | Slave2 …… Slave40 | SDO RD/WR

Cmd. ID	Error	State	CRC8	Data_2	Data_3_L	Data_3_H
(1Byte)	(1Byte)	(1Byte)	(1Byte)	(4Bytes)	(2Bytes)	(2Bytes)

Cmd. ID Return the command ID (such as SET_STATE, SET_AXIS, SET_DC, etc.). It will return 0xBF in OP state.

Error Command rejected code due to invalid format.

State Current state

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

CRC8 Cyclic Redundancy Check value of response data. The method is as follows:

STEP 1: Record the CRC8 value first, and fill the field with 0.

STEP 2: A total of 504 Bytes (or 512 Bytes) of the response data are using following CRC8 polynomial: CRC-8: $X^8 + X^2 + X + 1$, Initial value is 0x5A

STEP 3: Calculate the CRC8 value, which should match the value recorded in STEP 1.

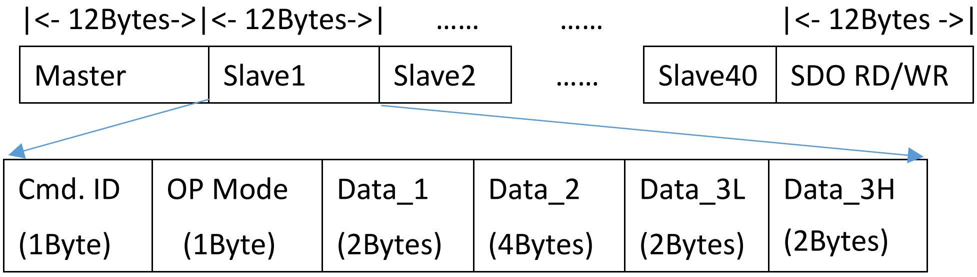
(This value is sent or not can be set by 2.1.5 SET_EX command)

Data_2 Result of command (different meanings depending on the command).

Data_3_L Show the FIFO remaining in OP state.

Data_3_H Show the number of commands rejected due to FIFO full in OP state.

From the 13th Byte to the 492Byte of the response data are the Slaves information. The first 2 Bytes of the drive slave is Cmd. ID and OP Mode. When CONFIG 2 is Low level, the response data of each slave is 12 bytes, and its meaning is as follows:



|<- 12Bytes->|	|<- 12Bytes->|	……	……	|<- 12Bytes ->|	
Master	Slave1	Slave2	……	Slave40	SDO RD/WR

Cmd. ID (1Byte)	OP Mode (1Byte)	Data_1 (2Bytes)	Data_2 (4Bytes)	Data_3L (2Bytes)	Data_3H (2Bytes)

Cmd. ID Return the command ID

OP_Mode Drive slave operating mode in the OP state. This is the standard of CoE Object 0x6060, the meaning is as following:

-Homing: 0x06
-CSP: 0x08
-CSV: 0x09
-CST: 0x0A

Data_1 Drive slave Status Word in the OP state. This is the standard of CoE Object 0x6041. For the definitions please refer to the slave instructions.

Data_2 Drive slave Current Position in the OP state. This is the standard of CoE Object 0x6064.

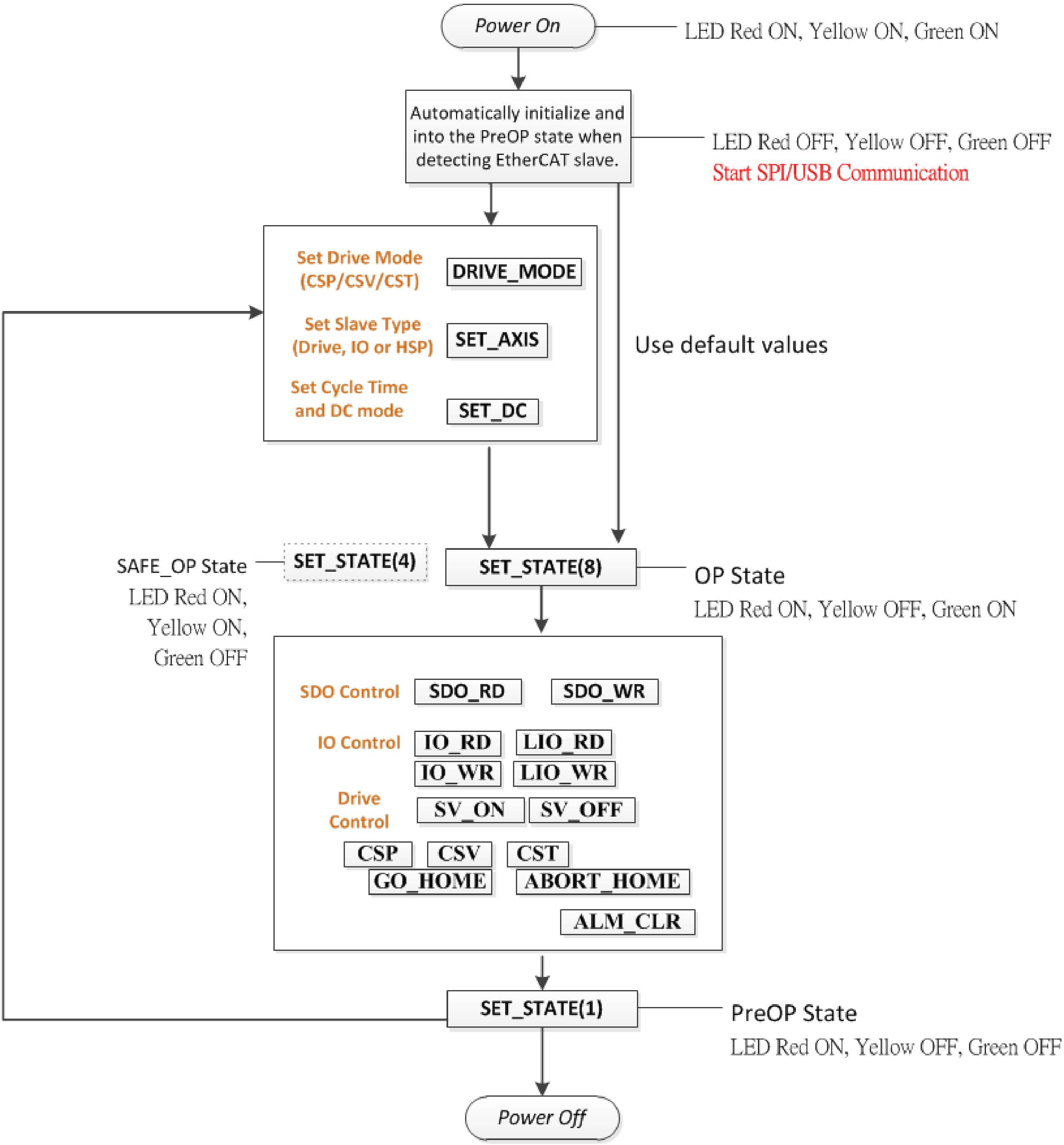
Data_3_L Drive slave Current Torque in the OP state. This is the standard of CoE Object 0x6077.

Data_3_H Drive slave Error Code in the OP state. This is the standard of CoE Object 0x603F.

When CONFIG 2 is High level, the response data of each slave is 16 bytes. The first 12 bytes are the same as above. Only the NEXTW HSP slave will use the last 4 Bytes in the CSP/CSV command.

|<- 16Bytes->|	……		……	|<- 16Bytes ->|	
Master	Slave1	Slave2	……	Slave30	SDO RD/WR

Chapter 3 Operational Process Example


Power On
LED Red ON, Yellow ON, Green ON
Automatically initialize and into the PreOP state when detecting EtherCAT slave.
LED Red OFF, Yellow OFF, Green OFF
Start SPI/USB Communication
Set Drive Mode (CSP/CSV/CST)
DRIVE_MODE
Set Slave Type (Drive, IO or HSP)
SET_AXIS
Set Cycle Time and DC mode
SET_DC
Use default values
SAFE_OP State
LED Red ON, Yellow ON, Green OFF
SET_STATE(4)
SET_STATE(8)
OP State
LED Red ON, Yellow OFF, Green ON
SDO Control
SDO_RD
SDO_WR
IO Control
IO_RD
LIO_RD
IO_WR
LIO_WR
Drive Control
SV_ON
SV_OFF
CSP
CSV
CST
GO_HOME
ABORT_HOME
ALM_CLR
SET_STATE(1)
PreOP State
LED Red ON, Yellow OFF, Green OFF
Power Off

Chapter 4 Dynamic Library

4.1 Introduction to Dynamic Library

"NEXTWUSBLib.dll" is a dynamic library for the ECM-SK via USB port in the Windows operating system. Support languages include C++, Visual Basic .NET, C# .NET, etc. "NEXTWUSBLib.dll" must be placed in the same directory as the .exe file.

"NEXTWUSBLib.lib" is a reference file for developing C++ programs. "NEXTWUSB_dotNET.dll" is a function prototype file for .NET languages. It should be added to the project during program development.

4.2 NEXTWUSBLib Library

The NEXTWUSBLib library supports C++ and .NET series programming languages such as Visual Basic .NET, C# .NET, etc. It is referenced during the execution phase of the program, so it must be placed in the same directory as the executable file. Table 4.1 is the main function of the "NEXTWUSBLib.dll" dynamic library.

Table 4.1 「NEXTWUSBLib.dll」 main function

Function Name	Function Description	Ref.
OpenECMUSB()	Open USB connection with ECM-SK	4.2.1
CloseECMUSB()	Close USB connection with ECM-SK	4.2.2
ECMUSBWrite()	Write command to ECM-SK via USB	4.2.3
ECMUSBRead()	Read data from ECM-SK via USB	4.2.4

4.2.1 OpenECMUSB Function

Prototype	`bool OpenECMUSB()`
Argument	Void
Return Type	bool (Indicates whether the open connection is successful or not.)

Description Open USB connection with ECM-SK. If the return value is False, there are the following possible reasons:

- ECM-SK is not connected to any USB port.
- ECM-SK CONFIG 0 is not set correctly.
- ECM-SK detect no EtherCAT Slave.
- ECM-SK has been opened by other programs.
- The power supply of the USB port is insufficient, resulting in the ECM-SK not working properly.

4.2.2 CloseECMUSB Function

Prototype `void CloseECMUSB()`

Argument Void

Return Type Void

Description Close the USB connection with ECM-SK and release resources.

4.2.3 ECMUSBWrite Function

Prototype `ECMUSBWrite(unsigned char * data, unsigned long dwlength)`

Argument data unsigned char array pointer. The array stores the command data to be written.

dwlength 4 bytes unsigned long. Indicate the number of written bytes. Since each exchange of data is 504 or 512 bytes, this value should be 504 or 512.

Return Type bool (Indicates whether the write command is successful or not.)

Description Write command to ECM-SK via USB. If the return value is False, there are the following possible reasons:

- ECM-SK is not opened, please call OpenECMUSB to open the connection.
- The connection is invalid. The USB port may be failed or removed.

4.2.4 ECMUSBRead Function

Prototype	ECMUSBRead(unsigned char * data, unsigned long dwlength)
Argument	data — unsigned char array pointer. The array will store the response data. dwlength — 4 bytes unsigned long. Indicate the number of read bytes. Since each exchange of data is 504 or 512 bytes, this value should be 504 or 512.
Return Type	bool (Indicates whether the read response is successful or not.)
Description	Read response data from ECM-SK via USB. If the return value is False, there are the following possible reasons: ● ECM-SK is not opened, please call OpenECMUSB to open the connection. ● The connection is invalid. The USB port may be failed or removed.

4.3 NEXTWUSB_dotNET Library

The NEXTWUSB_dotNET library supports .NET series programming languages such as Visual Basic .NET and C# .NET. Provides constant definitions, structures, and functions related to ECM-SK in the .NET environment. This library needs to be added to the project reference during the design phase. The implementation phase will still refer to "NEXTWUSBLib.dll". "NEXTWUSBLib.dll" will still be placed in the same directory as the executable file. Table 4.2 is the main function of the "NEXTWUSB_dotNET_XXB.dll" dynamic library, and Table 4.3 is the "NEXTWUSB_dotNET_12B.dll" dynamic library command structure.

Table 4.2 「NEXTWUSB_dotNET_XXB.dll」main function

Group	Constant Name	Type	Value	Description
Maximum number of axes	DEF_MA_MAX	Int32	42	NEXTWUSB_dotNET_12B.dll includes 1 ECM, 40 Slaves, and 1 SDO RD/WR field.
			32	NEXTWUSB_dotNET_16B.dll includes 1 ECM, 30 Slaves,

				and 1 SDO RD/WR field.
EthetCAT State	NIC_INIT	Int32	0	Initial network State
	STATE_INIT	Int32	1	EtherCAT Init State
	STATE_PRE_OP	Int32	2	EtherCAT PreOP State
	STATE_SAFE_OP	Int32	4	EtherCAT SafeOP State
	STATE_OPERATIONAL	Int32	8	EtherCAT OP State
Drive Mode	CSP_MODE	Int32	8	CSP Mode
	CSV_MODE	Int32	9	CSV Mode
	CST_MODE	Int32	10	CST Mode
DC Mode	FREERUN	Int32	0	Free Run
	DCSYNC	Int32	1	DC Synchronize
Slave Type	DRIVE	Int16	0x0	Drive
	IO	Int16	0x1	IO
	HSP	Int16	0x2	HSP
	None	Int16	0xF	None
Command ID	GET_STATUS	Int16	0x00	Get Status
	SET_STATE	Int16	0x01	Set ECM State
	SET_AXIS	Int16	0x02	Set Slaves Type
	SET_DC	Int16	0x03	Set ECM Cycle Time
	DRIVE_MODE	Int16	0x06	Set Mode of Servo Drive
	SDO_RD	Int16	0x07	Read Service Data Objects
	SDO_WR	Int16	0x08	Write Service Data Objects
Command ID	ALM_CLR	Int16	0x10	Clear Alarm of Slave
	SV_ON	Int16	0x11	Set Servo ON
	SV_OFF	Int16	0x12	Set Servo OFF
	IO_RD	Int16	0x13	Read Digital Input Status
	IO_WR	Int16	0x14	Write Digital Output Values
	CSP	Int16	0x15	Set Cyclic Synchronous Position Command
	CSV	Int16	0x16	Set Cyclic Synchronous Velocity Command
	CST	Int16	0x17	Set Cyclic Synchronous

				Torque Command
	GO_HOME	Int16	0x18	Start the Go Home Procedure
	ABORT_HOME	Int16	0x19	Force Termination of the slave GO_HOME Procedure
Command ID	LIO_RD	Int16	0x21	Read Input status on the ECM IC
	LIO_WR	Int16	0x22	Write Output Values on the ECM IC

Table 4.3 「NEXTWUSB_dotNET_12B.dll」 Structure

Structure	Member Name	Type
transData	CMD	Int16
	Pram	Int16
	Data1	Int32
	Data2	Int32

"NEXTWUSB_dotNET_12B.dll" contains two public variables (cmdData and respData). These two public variables are array type with a length of 42, which include 1 ECM and 40 slave data, plus a SDO RD/WR field. Table 4.4 is the "NEXTWUSB_dotNET_16.dll" dynamic library command structure.

Table 4.4 「NEXTWUSB_dotNET_16B.dll」 Structure

Structure	Member Name	Type
transData	CMD	Int16
	Pram	Int16
	Data1	Int32
	Data2	Int32
	Data3	Int32

"NEXTWUSB_dotNET_16B.dll" contains two public variables (cmdData and respData). These two public variables are array type with a length of 32, which include 1 ECM and 30 slave data, plus a SDO RD/WR field. Table 4.5 is the "NEXTWUSB_dotNET_XX.dll" main function.

Table 4.5 「NEXTWUSB_dotNET_XXB.dll」 main function

Function Name	Function Description	Ref.
OpenECMUSB()	Open USB connection with ECM-SK	4.3.1
CloseECMUSB()	Close USB connection with ECM-SK	4.3.2
ECMUSBWrite()	Write command to ECM-SK via USB	4.3.3
ECMUSBRead()	Read data from ECM-SK via USB	4.3.4
ClearCmdData()	Clear the CmdData	4.3.5

4.3.1 OpenECMUSB Function

Prototype `bool OpenECMUSB()`

Argument Void

Return Type bool (Indicates whether the open connection is successful or not.)

Description Open USB connection with ECM-SK. If the return value is False, there are the following possible reasons:

- ECM-SK is not connected to any USB port.
- ECM-SK CONFIG 0 is not set correctly.
- ECM-SK detect no EtherCAT Slave.
- ECM-SK has been opened by other programs.
- The power supply of the USB port is insufficient, resulting in the ECM-SK not working properly.

4.3.2 CloseECMUSB Function

Prototype `void CloseECMUSB()`

Argument Void

Return Type Void

Description Close the USB connection with ECM-SK and release resources.

4.3.3 ECMUSBWrite Function

Prototype `bool ECMUSBWrite()`

Argument Void

Return Type bool (Indicates whether the write command is successful or not.)

Description Write command to ECM-SK via USB. If the return value is False, there are the following possible reasons:

- ECM-SK is not opened, please call OpenECMUSB to open the connection.
- The connection is invalid. The USB port may be failed or removed.

4.3.4 ECMUSBRead Function

Prototype `bool ECMUSBRead()`

Argument Void

Return Type bool (Indicates whether the read response is successful or not.)

Description Read response data from ECM-SK via USB. If the return value is False, there are the following possible reasons:

- ECM-SK is not opened, please call OpenECMUSB to open the connection.
- The connection is invalid. The USB port may be failed or removed.
- CRC8 value verification failed.

4.3.5 ClearCmdData Function

Prototype `void ClearCmdData()`

Argument Void

Return Type Void

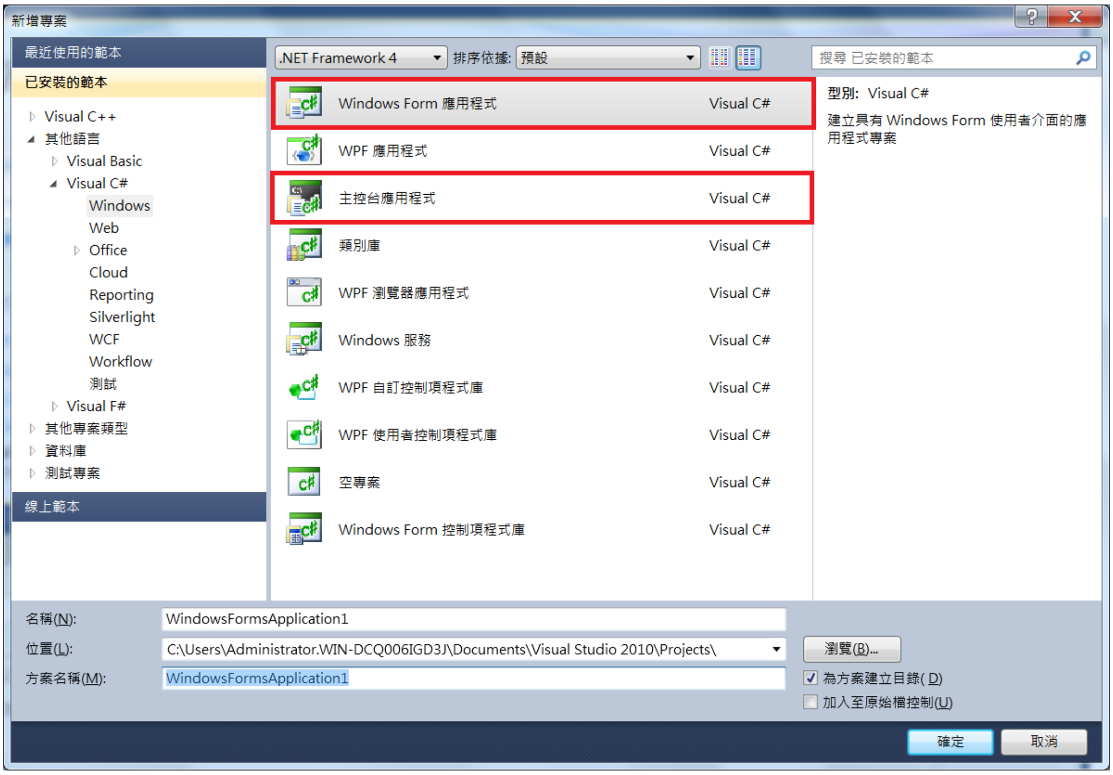
Description Clear the contents of cmdData so that all the contents value is 0.

4.4 Visual Studio Environment Setting

This section takes Microsoft's Visual Studio 2010 as an example to explain the steps of the environment setting.

4.4.1 C# .NET Environment Setting

Step 1: New Project –Select Windows Form App. or Console App. according to your needs.



型別: Visual C#

建立具有 Windows Form 使用者介面的應用程式專案

名稱(N): WindowsFormsApplication1

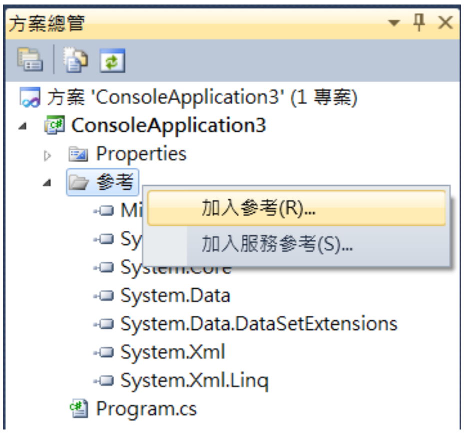
位置(L): C:\Users\Administrator.WIN-DCQ006IGD3J\Documents\Visual Studio 2010\Projects\

方案名稱(M): WindowsFormsApplication1

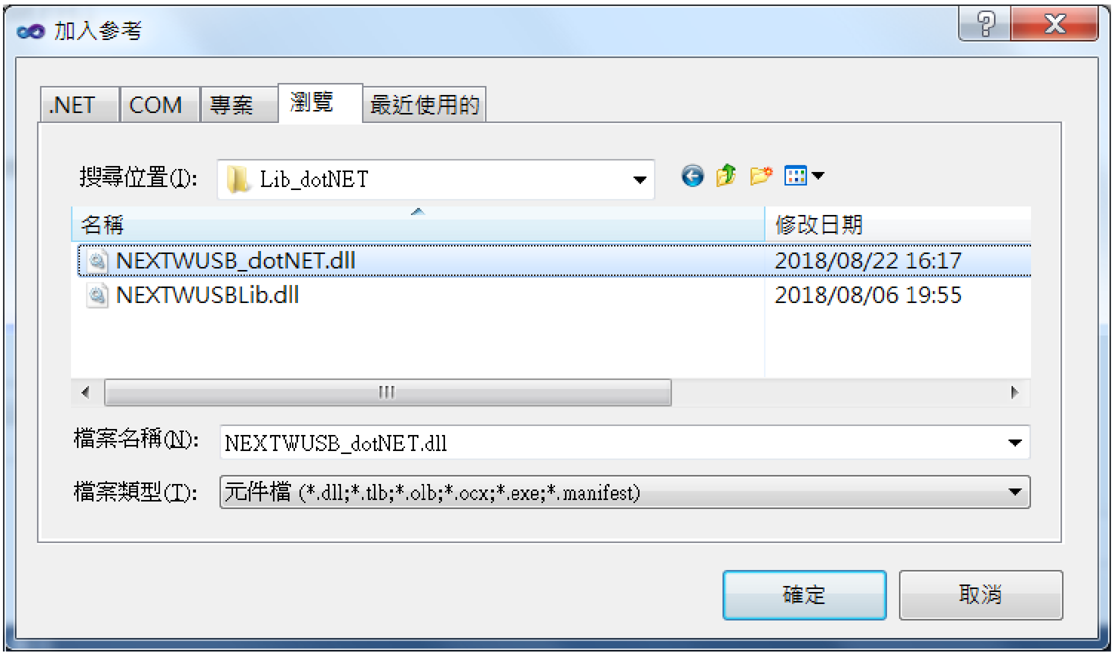
☑ 為方案建立目錄(D)

□ 加入至原始檔控制(U)

Step 2 – Solution Explorer -> Add -> Reference



Step 3 –Browse... -> Select「NEXTWUSB_dotNET_XXB.dll」and press OK button.



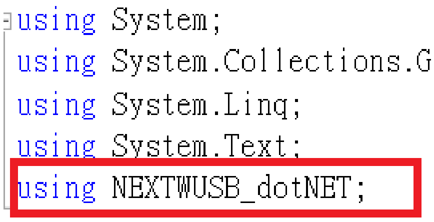
搜尋位置(I): Lib_dotNET

名稱	修改日期
NEXTWUSB_dotNET.dll	2018/08/22 16:17
NEXTWUSBLib.dll	2018/08/06 19:55

檔案名稱(N): NEXTWUSB_dotNET.dll

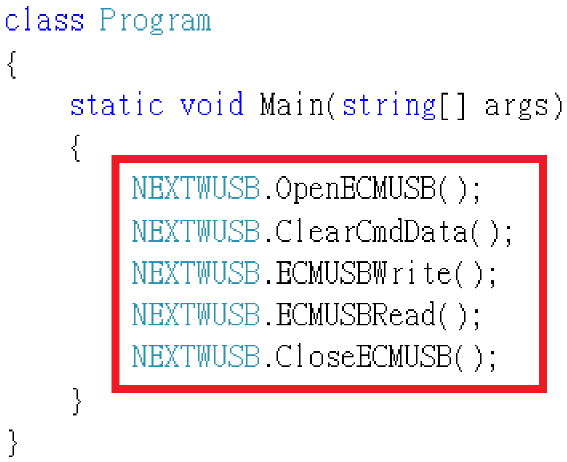
檔案類型(T): 元件檔 (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

Step 4 –Using Name Space「NEXTWUSB_dotNET_XXB」



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using NEXTWUSB_dotNET;
```

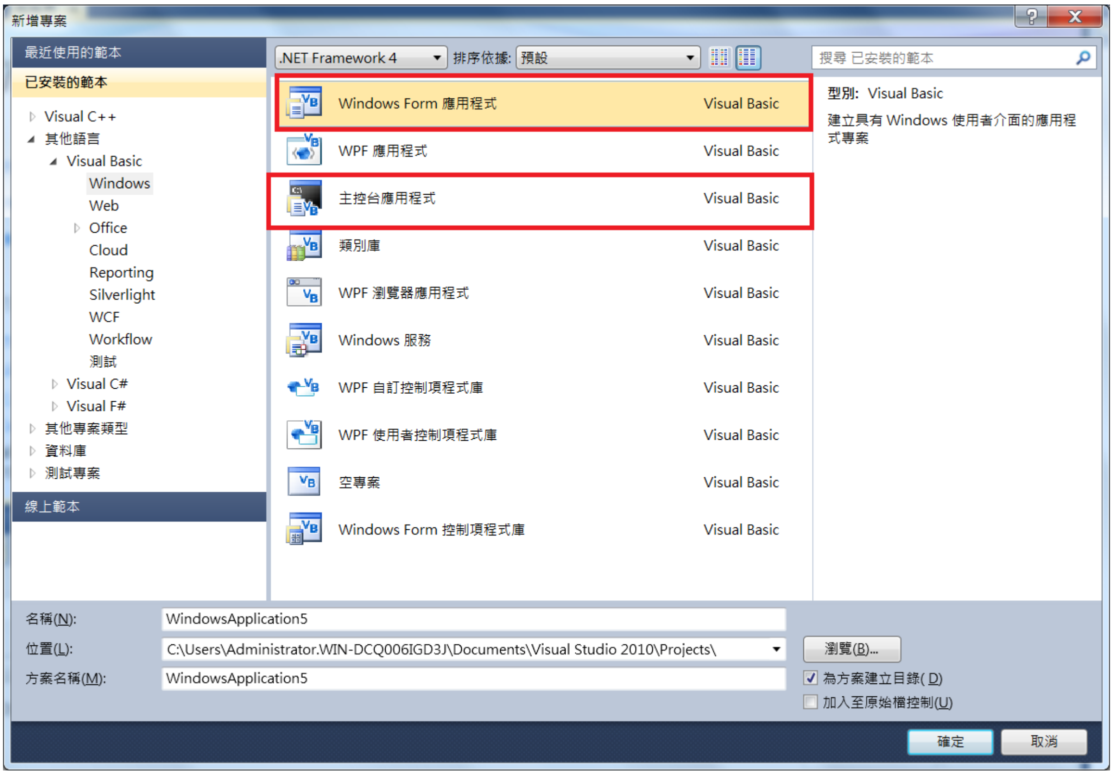
Step 5 –Use the "NEXTWUSB_dotNET.dll" dynamic library function



```
class Program
{
    static void Main(string[] args)
    {
        NEXTWUSB.OpenECMUSB();
        NEXTWUSB.ClearCmdData();
        NEXTWUSB.ECMUSBWrite();
        NEXTWUSB.ECMUSBRead();
        NEXTWUSB.CloseECMUSB();
    }
}
```

4.4.2 Visual Basic .NET Environment Setting

Step 1: New Project –Select Windows Form App. or Console App. according to your needs.



型別: Visual Basic

建立具有 Windows 使用者介面的應用程式專案

名稱(N): WindowsApplication5

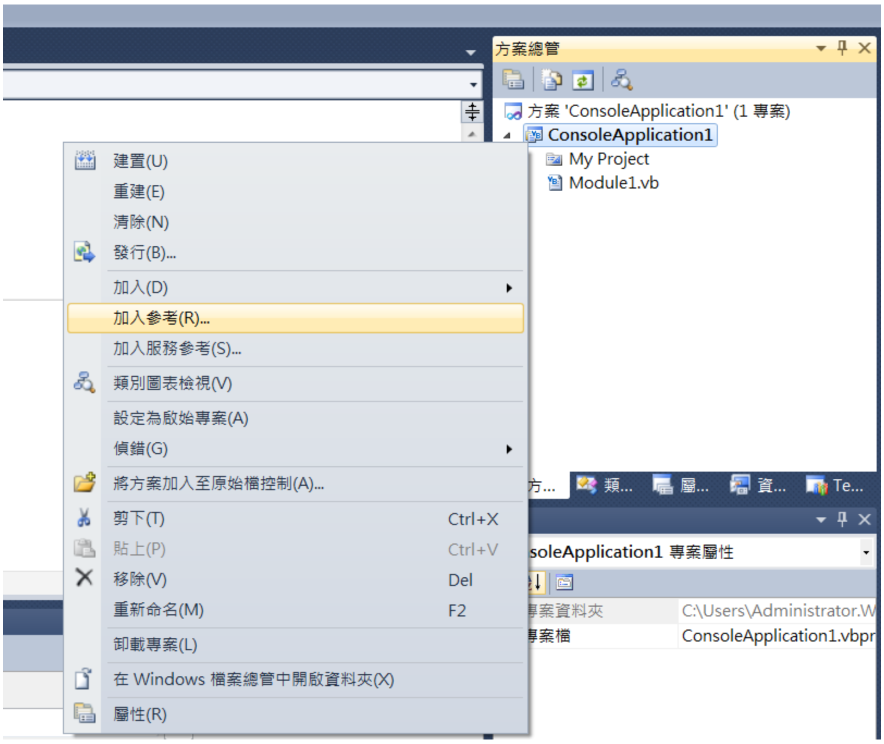
位置(L): C:\Users\Administrator.WIN-DCQ006IGD3J\Documents\Visual Studio 2010\Projects\

方案名稱(M): WindowsApplication5

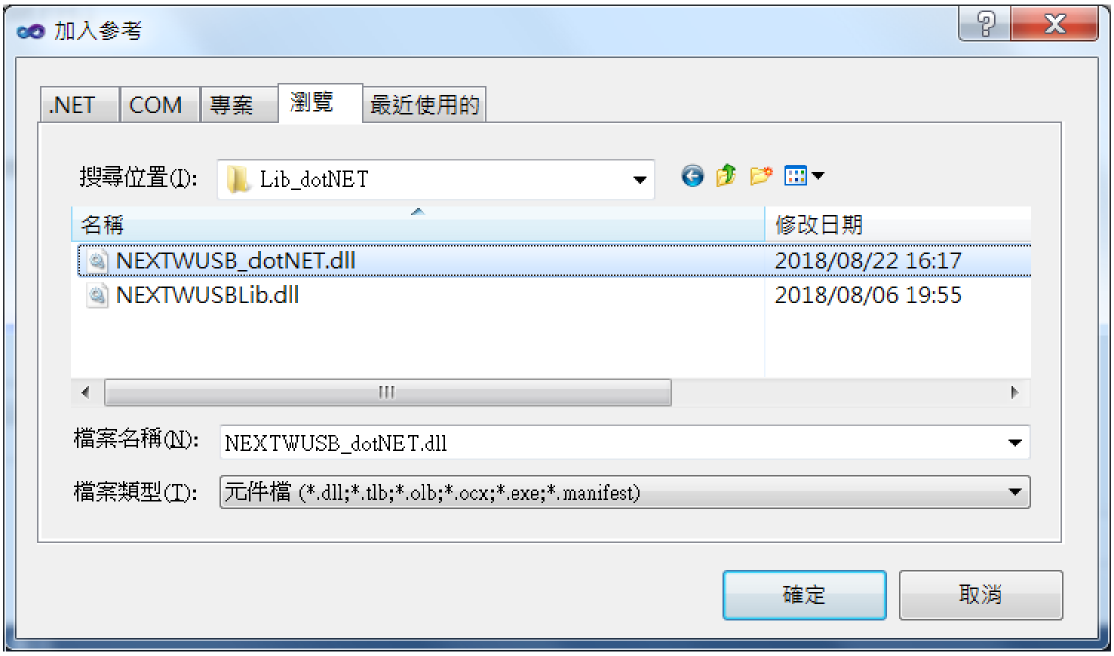
☑ 為方案建立目錄(D)

□ 加入至原始檔控制(U)

Step 2 –Solution Explorer -> Add -> Reference



Step 3 –Browse... -> Select「NEXTWUSB_dotNET_XXB.dll」and press OK button.



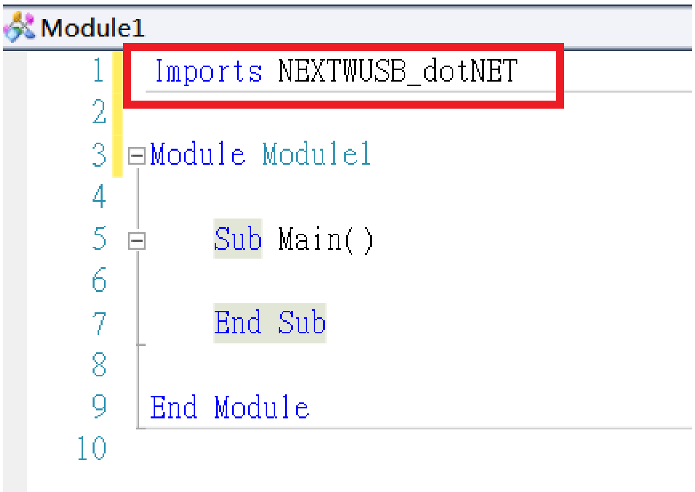
搜尋位置(I): Lib_dotNET

名稱	修改日期
NEXTWUSB_dotNET.dll	2018/08/22 16:17
NEXTWUSBLib.dll	2018/08/06 19:55

檔案名稱(N): NEXTWUSB_dotNET.dll

檔案類型(T): 元件檔 (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

Step 4 – Imports Name Space「NEXTWUSB_dotNET_XXB」



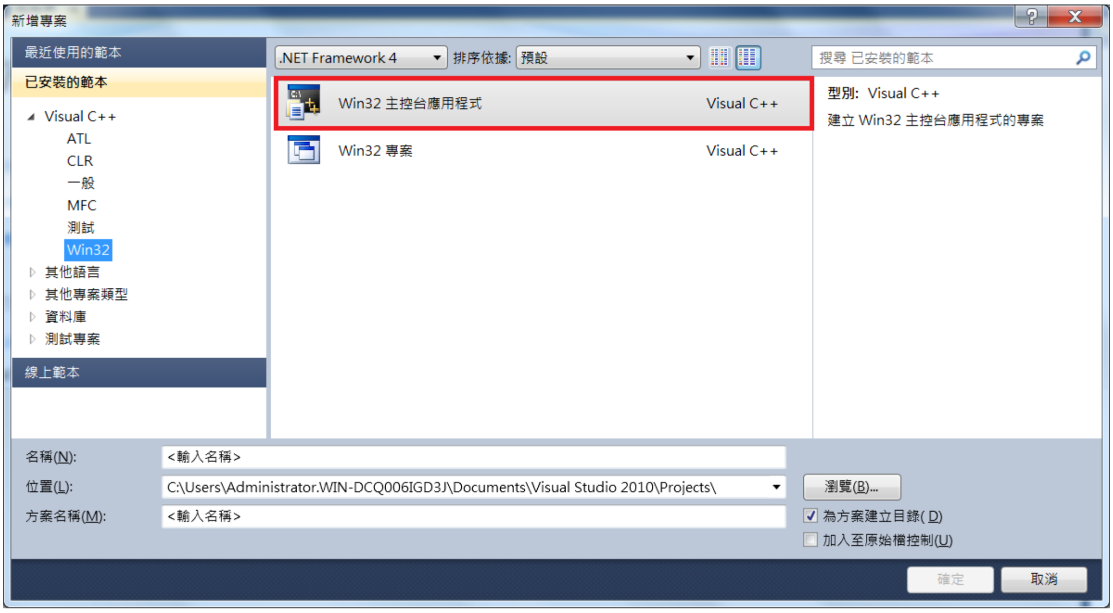
```
Imports NEXTWUSB_dotNET

Module Module1

    Sub Main()

    End Sub

End Module

```

Step 5 –Use the "NEXTWUSB_dotNET.dll" dynamic library function

```
Sub Main()
    NEXTWUSB.OpenECMUSB()
    NEXTWUSB.ClearCmdData()
    NEXTWUSB.ECMUSBWrite()
    NEXTWUSB.ECMUSBRead()
    NEXTWUSB.CloseECMUSB()
End Sub
```

4.4.3 C++ Environment Setting

Step 1: New Project – Select the project template according to your needs. Here is the example of "Win32 console application".



Win32 主控台應用程式 Visual C++

Win32 專案 Visual C++

型別: Visual C++

建立 Win32 主控台應用程式的專案

名稱(N): <輸入名稱>

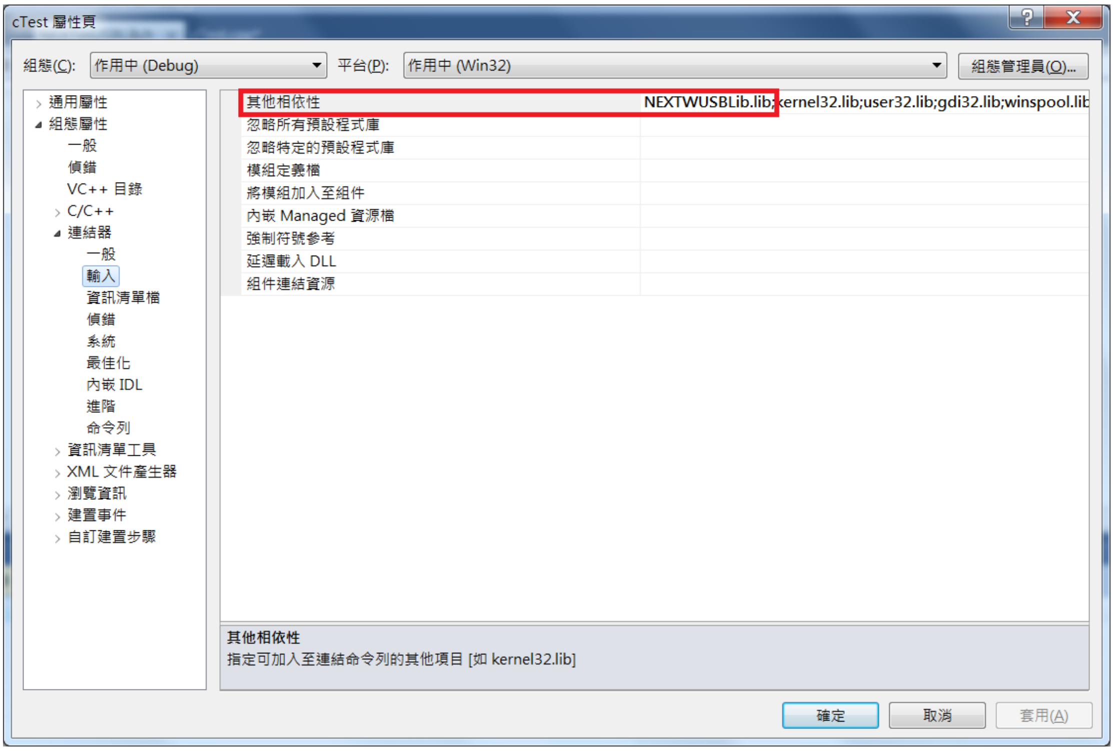
位置(L): C:\Users\Administrator.WIN-DCQ006IGD3J\Documents\Visual Studio 2010\Projects\

方案名稱(M): <輸入名稱>

☑ 為方案建立目錄(D)

□ 加入至原始檔控制(U)

Step 2 – Project Properties -> Linker -> Input -> Addition Dependencies -> Add "NEXTWUSBLib.lib"



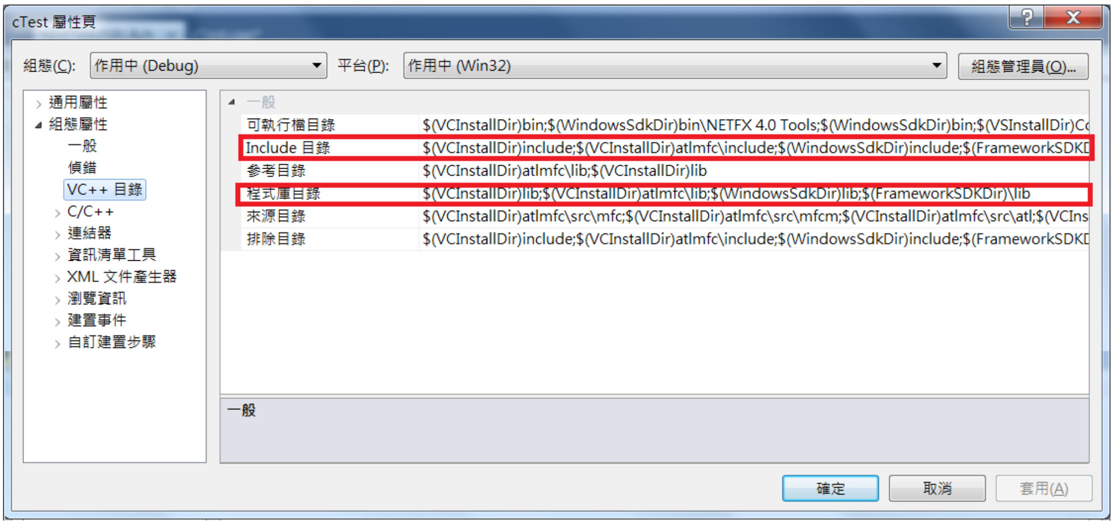
組態(C): 作用中 (Debug) 平台(P): 作用中 (Win32)

其他相依性	NEXTWUSBLib.lib;kernel32.lib;user32.lib;gdi32.lib;winspool.lib
忽略所有預設程式庫	
忽略特定的預設程式庫	
模組定義檔	
將模組加入至組件	
內嵌 Managed 資源檔	
強制符號參考	
延遲載入 DLL	
組件連結資源	

其他相依性

指定可加入至連結命令列的其他項目 [如 kernel32.lib]

Step 3 –Project Properties -> VC++ Directories -> Include Directories / Library Directories Fill in the actual directory of "NEXTWUSBLib.h" and "NEXTWUSBLib.lib".



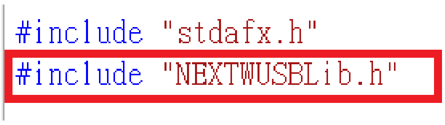
組態(C): 作用中 (Debug)　平台(P): 作用中 (Win32)

一般

可執行檔目錄	$(VCInstallDir)bin;$(WindowsSdkDir)bin\NETFX 4.0 Tools;$(WindowsSdkDir)bin;$(VSInstallDir)Co
Include 目錄	$(VCInstallDir)include;$(VCInstallDir)atlmfc\include;$(WindowsSdkDir)include;$(FrameworkSDKD
參考目錄	$(VCInstallDir)atlmfc\lib;$(VCInstallDir)lib
程式庫目錄	$(VCInstallDir)lib;$(VCInstallDir)atlmfc\lib;$(WindowsSdkDir)lib;$(FrameworkSDKDir)\lib
來源目錄	$(VCInstallDir)atlmfc\src\mfc;$(VCInstallDir)atlmfc\src\mfcm;$(VCInstallDir)atlmfc\src\atl;$(VCIns
排除目錄	$(VCInstallDir)include;$(VCInstallDir)atlmfc\include;$(WindowsSdkDir)include;$(FrameworkSDKD

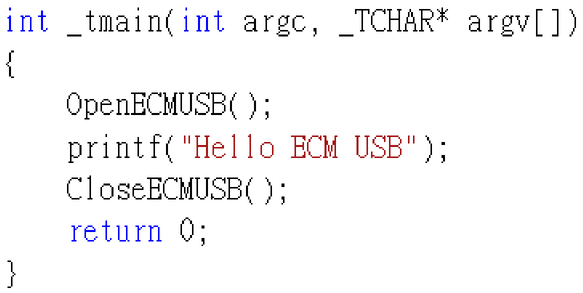
一般

Step 4 – Add Include File「NEXTWUSBLib.h」



```
#include "stdafx.h"
#include "NEXTWUSBLib.h"
```

Step 5 –Use the "NEXTWUSB.dll" dynamic library function



```
int _tmain(int argc, _TCHAR* argv[])
{
    OpenECMUSB();
    printf("Hello ECM USB");
    CloseECMUSB();
    return 0;
}
```