EtherCAT USER’S MANUAL

For ProNet

Version update information

Date	version	description	Author
2011/3/10	1.00	Draft finished	Liu Yan
2012/3/23	1.00	English version	Alps Qi

Content

1 Brief introduction of EtherCAT

1.1 What is EtherCAT

EtherCAT is an open network based on Ethernet to achieve real time control. It could support high spnded and synchronized control. By using efficient network topology, the network structure with too many concentrator and complicated connections are avoided. It is very suitable to use this protocol in motion control and other factory automation applications.

1.2 EtherCAT general introduction

EtherCAT technology breaks the limits of normal internet solution. Through this technology, we don't need to receive Ethernet data, decode the data, and then copy the process data to different devices. EtherCAT slave device could read the data marked with this device's address information when message passes this device. As the same time, some data will be written into the message when data message pass the device. In this way, data reading and data writing could be done within several nanoseconds.

EtherCAT uses standard Ethernet technology and support almost kinds of topologies, including the line type, tree type, star type and so on. Its physical layer could be 100 BASE-TXI twisted-pair wire, 100BASE-FX fiber or LVDS (low voltage differential signaling). It could also be done through exchangers or media converters or in order to achieve the combination of different Ethernet structure.

Relying on the especial chips for EtherCAT in the slave and DMA technology that reads network interface data, the processing of the protocol is done in the hardware. EtherCAT system could update the information for 1000 I/O within 30μs. It could exchange a frame as big as 1486 bytes within 300μs. This is almost like 12000 digital output or input. Controlling one servo with 100 8-byte I/O data only takes 100μs. Within this period, the system could update the actual positions and status presented by command value and controlling data. Distribute clock technology could make the cycle synchronization error lower than 1μs.

1.3 Product introduction

Pronet servo drive achieves EtherCAT communication through EC100 network module. It is a real time Ethernet communication and the application layer applies CANopen Drive Profile(CiA 402).

Besides supporting the PV, PP, IP and other control mode defined in CANopen DS402, this module also supports CSP control mode. Clients could switch the control mode by changing correspondent parameters. It is available from simple velocity control to high speed high precision position control,

CoE term

The tables below lists the terms used in CANopen and EtherCAT.

Abbreviation	Description
APRD	Automatic physical reading: Choosing the storage space of the slave according to the position that the slave stays in the network to read
APWR	Automatic physical writing: Choosing the storage space of the slave according to the position that the slave stays in the network to write.
APRW	Automatic physical reading/writing in one slave
ARMW	Automatic physical reading/writing in multiple slaves.
BRD	Broadcast reading: Reading physical storage area in all slaves in the network.
BRW	Broadcasting writing: Writing physical storage area in all slaves in the network
CiA	CAN in Automation
CoE	CANopen over EtherCAT
DC	Distribute Clock which is used to make all the slaves obtain the same time
ECAT	EtherCAT
EEPROM	electric removable read only memory
ESC	EtherCAT Slave Controller
ESM	EtherCAT status machine
ETG	EtherCAT Technology Group
EtherCAT	One real time industrial network standard based on Ethernet
FMMU	Filed bus memory management unit
INIT	One EtherCAT status machine: Initialization.
LRD	Read data from one or several slave's storage space according to the selection of logical address
LWR	Write data into one or several slave's storage space according to the selection of logical address
LRW	read or write data from or into the slave's memory according to the selection of logical address
OP	EtherCAT status machine：operating
OD	Object dictionary
PDO	Process data object
PREOP	EtherCAT status machine：pre-operation
RXPDO	Receive PDO
SAFEOP	EtherCAT status machine： safety operation mode
SDO	Service data object
SyncManager	Sync manager this is used to control the visit to applied storage area.
TXPDO	Transfer PDO

1.4 data type

The table below lists all the data types and their range that will be used in this manual

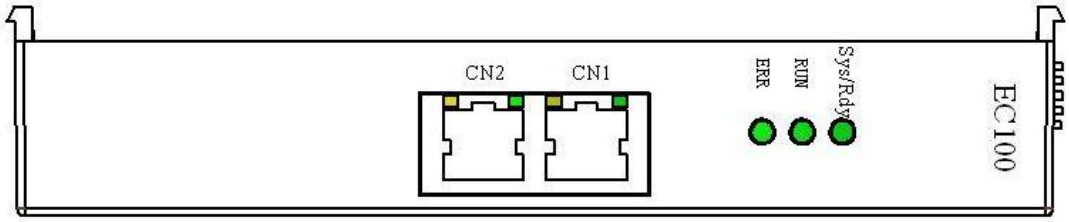
Code	Data type	Range
UINT8	Unsigned integer 8	0 to 255
INT8	Integer 8	–128 to +127
UINT16	Unsigned integer 16	0 to 65535
INT16	Integer 16	–32768 to +32767
UINT32	Unsigned integer 32	0 to 4294967295
INT32	Signed integer 32	–2147483648 to +2147483627
STR	string	—

1.5 Communication introduction

EtherCAT communication	applied communication standard	IEC 61158 Type12, IEC 61800-7 CiA402 Drive Profile
	Physical layer	100BASE-TX (IEEE802.3)
	Interface	CN4 (RJ45): EtherCAT Signal IN CN5 (RJ45): EtherCAT Signal OUT
	Wiring	Level-5 twisted pair wire
	communication distance	distance between nodes below 100 meters
	SyncManager	SM0: output mailbox, SM1: input mailbox SM2: input process data SM3: Output process data
	FMMU	FMMU0:.mapping to output area of process data(RXPDO) FMMU1:.Mapping to transmit area of process data(TxPDO) FMMU2: mapping to mailbox status
	EtherCAT Commands (Data Link Layer)	APRD, FPRD, BRD, LRD, APWR, FPWR, BWR, LWR, ARMW, FRMW Note: APRW, FPRW, BRW, LRW Commands are not supported.
	PDO data	Dynamic PDO mapping
	Mailbox (CoE)	Emergency Message, SDO Request, SDO Response, SDO information Note: Don't support TXPDO/RxPDO and remote TxPDO/RxPDO.
	Distribute data （DC）	Free-run ， DC mode （ activated by configuration）

		supported DC cycle time：250us—8ms
	SII	256 bytes(read only）
	LED light	EtherCAT system indicator（SYS）×1 EtherCAT run indicator（RUN）×1 EtherCAT error indicator（ERR）×1
CiA402 Drive Profile		• Homing mode • Profile position mode • Interpolated position mode • Profile velocity mode • Cyclic synchronous position mode

1.6 EC-100 module structure

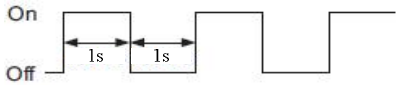


model structure

1.7 LED indicator

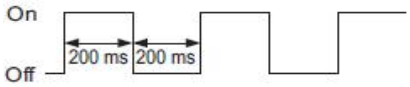
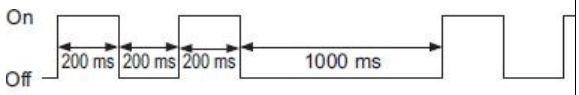
·SYS

EC-100 module indicate light, used to show the software status in the module.

LED light（green/yellow）		introduction
status	description	
distinguish	Distinguish for long time	no power supply or reset status
flashing(yellow)		Boot mode
light on (green)	.light is on for long time	Module's internal program has finished initiation and operates well.

·RUN

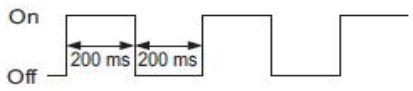
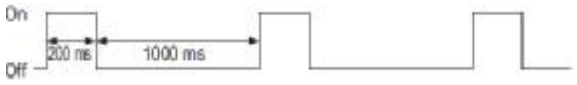
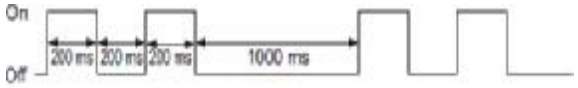
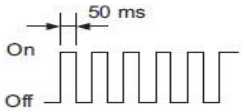
RUN light is used to inidtcate the communication status of EtherCAT

LED indicator（green）		Introduction
status	description	
distinguish	Distinguish for long time	System initiation
flashing		pre-operation status
Double flashing		safety operation mode

Light on	Light on	Operation status

• ERR

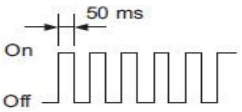
ERP light is used to indicate the error in EtherCAT communication.

LED light（red）		introduction
status	description	
distinguish	Distinguish for long time	No error
flashing		Due to register problem or object configuration problem, the status changing required by the master couldn't be achieve.
single flashing		Sync error. Communication data error
Double flashing		Application program supervision overtime. SyncManager watchdog overtime
flashing light circle		Initiating error
Light on	Light is on for long time	PDI supervision overtime

·LINK/ACT (green light on RJ45 COM1/COM2)

LINK/ACT light is used to indicate the physical communication and if there is data exchange.

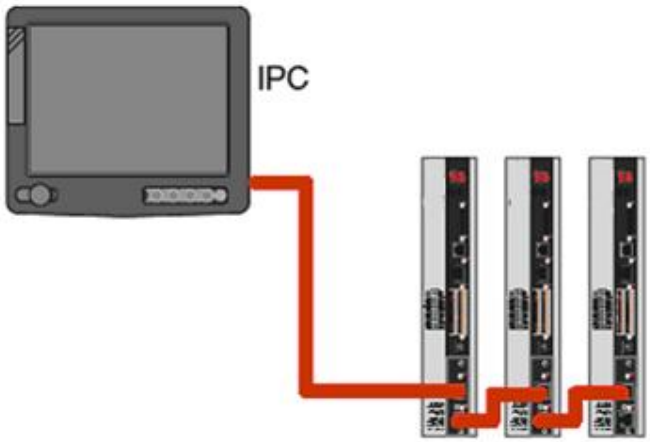
LED light（green）		introduction
status	description	
distinguish	Light off for long time	Physical level communication has not been started. EtherCAT controller has not been

		started.
flashing light circle		slave is exchanging data
Light on	Light is on for long time	There is connection in link layer but there is no date exchange

2 Installation and connection

2.1 Installation and connection

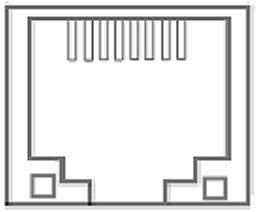
EtherCAT network is normally composed of one master (for example, industrial PC) and some slaves(for example, masters, filed bus terminals and so on). Every EtherCAT slave has two standard Ethernet interfaces.



EtherCAT network

2.2 EtherCAT interface specification

EtherCAT interface should be connected by twisted pair wire
·Electrical feature： according to IEEE802.3 standard
Interface: RJ45 8 pin modularize connector(According to ISO 8877)



RJ45 connector

·RJ45 connector

connector	description
CN1	EtherCAT IN port
CN2	EtherCAT OUT port

·Pin layout

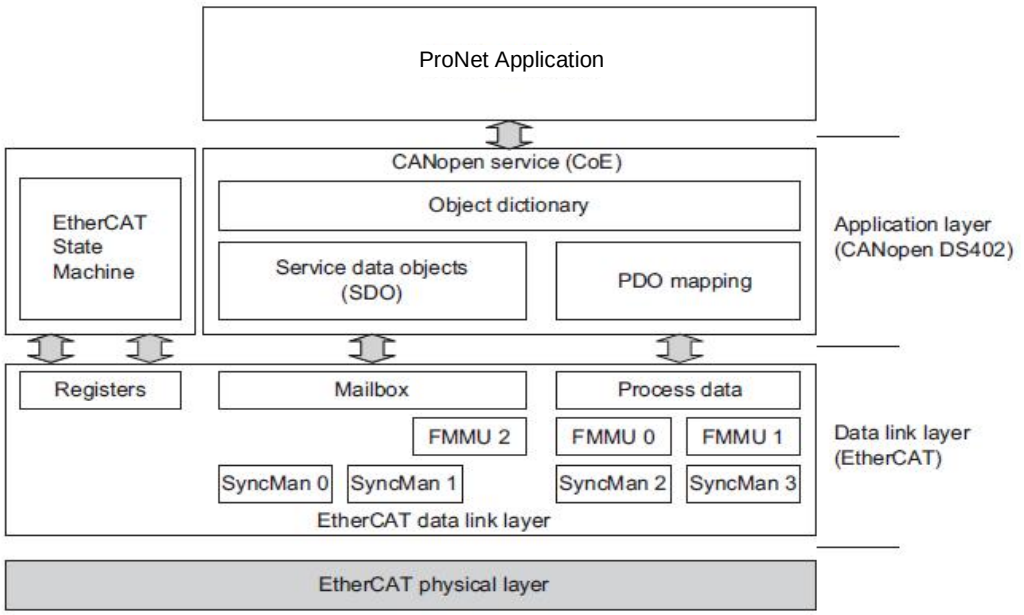
Pin No.	Signal name	abbreviation	signal transmit direction
1	Data transmit＋	TD＋	Output
2	Data transmit－	TD-	Output
3	Data receive ＋	RD+	Input
4	Not used	－	－
5	Not used	－	－
6	Data receive －	RD-	Input
7	Not used	－	－
8	Not used	－	－
Interface grounding	grounding	FG	－

2.3 wire specification

·Level 5 or above.
·Shield

3 EtherCAT-EC information

3.1 CANopen over EtherCAT model



Communication model

EtherCAT (CoE) network model is composed of two parts: data link layer and application layer. Data link layer is mainly in charge of EtherCAT communication protocol. Application layer is mainly oriented to CanOpen drive profiles(DS402) communication protocol. Object dictionary in CoE includes parameters, application data and PDO mapping information.

Process data object (PDO) is composed of objects in the object dictionary that could operate PDO mapping. The content of PDO data is defined by PDO mapping. PDO data's read and write are periodical without checking OD. However, mail communication(SDO) is not periodical. When they are read or written, it is necessary to check OD.

Note: To decode SDO data and PDO data on EtherCAT data link layer correctly, we need to configure FMMU and Sync Manager as below

·Sync Manager configuration

Sync Manager	Assignment（Fixed）	Size	Start Address（Fixed）
Sync Manager 0	Assigned to Receive Mailbox	128byte（Fixed）	0x1000
Sync Manager 1	Assigned to Transmit Mailbox	128byte（Fixed）	0x1080

Sync Manager 2	Assigned to Receive PDO	0 to 200byte	0x1100
Sync Manager 3	Assigned to Transmit PDO	0 to 200byte	0x1358

·FMMU Settings

FMMU	Settings
FMMU 0	Mapped to Receive PDO
FMMU 1	Mapped to Transmit PDO
FMMU 2	Mapped to Fill Status of Transmit Mailbox

3.2 EtherCAT slave information

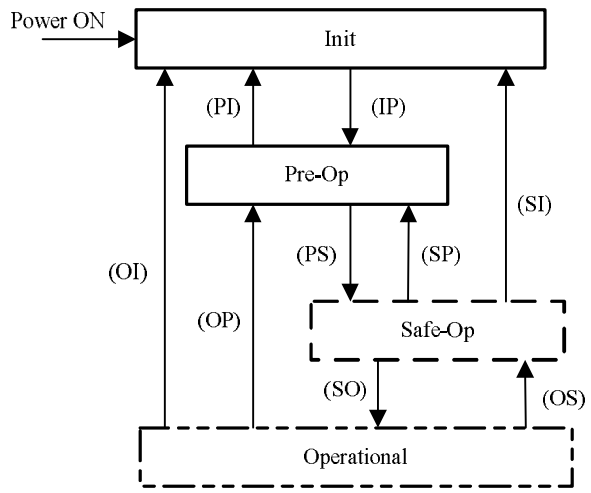
EtherCAT slave information (XML document) could be read by the master to build the master-slave configuration. ESTUN Pronet servo drive offers document as below ....

·ESTUN_ProNet_CoE.xml

EtherCAT network status machine

EtherCAT status machine is used to describe the statuses that one slave applies and the status change. Status change request is normally launched by the master and answered by the slave.

The graph below describes the slave's status machine.



Init:
Initiation, no communication

Pre-Op:
Mailbox service is activated. Slaves could achieve specific initiation setting and data exchange through mailbox.

Safe-Op:
Master receives the engineering data from the slave and the output process data is set as safety status and ready for synchronization.

OP:
Communication is good and I/O data is transmitted correctly.

Status	Description
Init	·No mailbox communication No process data communication
Init to Pre-Op	·Master configures data link layer address and initiate mailbox communication .Master initiates DC time clock synchronization. ·Master asks to change into Pre-op status. ·Master sets AL control register. ·Slave checks if mailbox initiation is good.
Pre-Operation （Pre-Op）	·,mailbox communication is activated. ·Process data communication is not available.
Pre-Op to Safe-Op	• Master configures SyncManager channels and FMMU channels for process data. • Master configures PDO mapping and the sync manager PDO assignment parameters via SDO. • Master requests 'Safe-Operational' state. • Slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct.

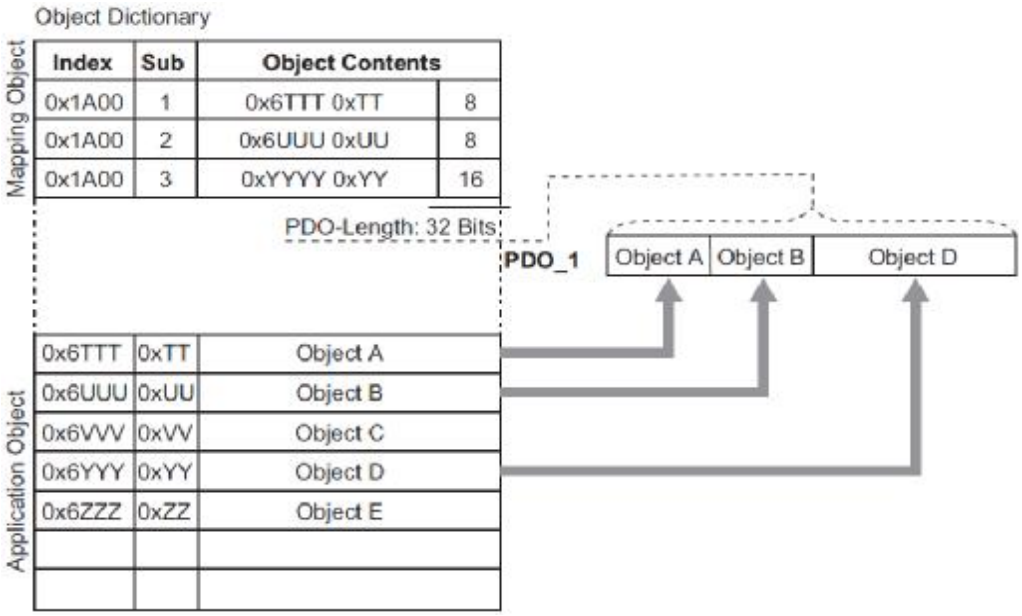
Safe-Operation （Safe-Op）	Slave's program will transmit actual input data and will not execute output. Output is set as safety status.
Safe-Op to Op	·Master transmits effective output data. Master asks to change into OP status.
Operational（Op）	Process data communication is available now.

3.3 PDO process data mapping

1. PDO mapping

PDO mapping is related to the mapping from object dictionary to PDO's application objects (real time process data).

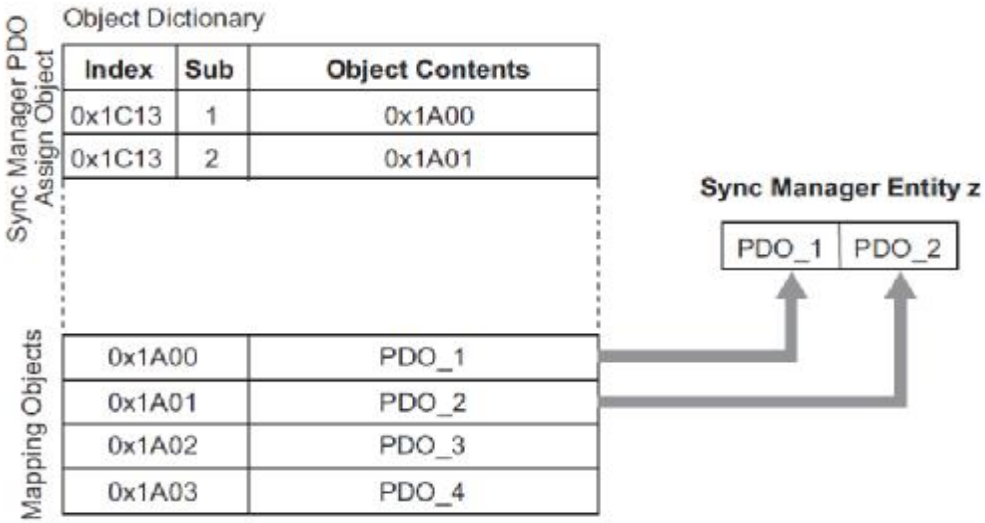
The index 0x1600 and 0x1A00 in object dictionary are separately reserved in the mapping tables of RXPDO and TxPDOs. The graph as below is one example ….



PDO mapping example

2. PDO configuration

Sync manager object (SMCO) is composed of multiple PDOs. SM-PDO-Assign object (0x1C12 and 0x1C13) describes the relationship between PDOs and Sync Manager as below



PDO configuration example

Note: The PDO mapping objects (index 1600h to 1603h, 1A00h to 1A03h) and the Sync Manager PDO assign objects
(index 1C12h and 1C13h) can be written only in Pre-Operation state.
Please note:

3. PDO mapping process

1) Stop PDO allocating function (set the sub-index 0 of 0x1c12 and 0x1c13 into 0).
2) Stop PDO mapping function(set sub-index 0 of 0x1600～0x1603 and 0x1A00～0x1A03 into 0).
3) Set the number of mapping entries in PDO mapping objects (Set subindex 0 of object 1600h to 1603h/1A00h to 1A03h).
4) Set the assignment of the Sync manager and PDO (Set subindex 1 of object 1C12h and 1C13h)
5) Enable the assignment of the Sync manager and PDO (Set subindex 0 of object 1C12h and 1C13h to 1).

3.4 Emergency message

When the servo drive generates an alarm, Coe will activate an emergency message and inform consumers the current servo drive model number and error code.

Emergency message structure：

6 bytes	2 bytes	2 bytes	1 byte	5 bytes	1...n bytes
Mailbox Header	CoE Header	ErrorCode	Error Register	Data	Data1

standard data frame head　　standard CANopen urgent event message　　optional

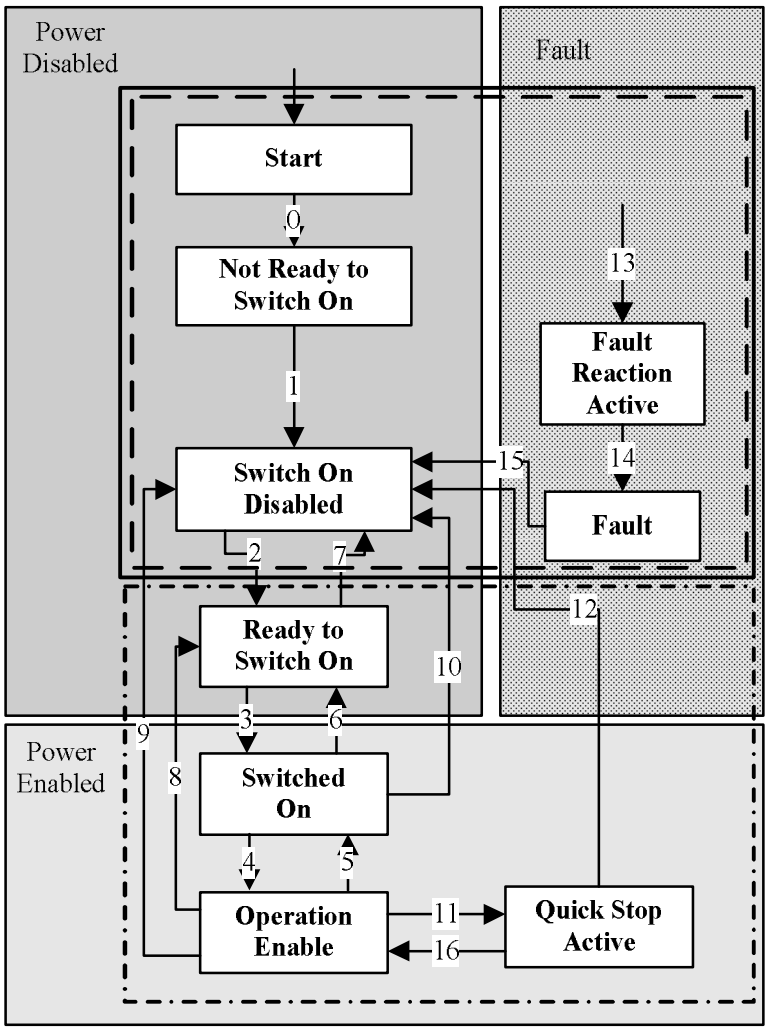
Byte	0	1	2	3	4	5	6	7

Data	Emergency Error Code	Error Register (Object 1001h)	Reserved	Manufacturer Specific Error Field	
				ProNet Alarm/Warning Code*2	Reserved

4 CiA402 device protocol

Pronet's device control is used mainly to achieve the motion control in different control modes. The master controls the servo drive through control word and knows the status of the servo drive by reading the servo drive's status word.

4.1 CANopen over EtherCAT(CoE) status machine



Status word (0x6041) could only be updated and delivered in OP status and Safe-Op status.

Control word (0x6040) could be read from slaves only in OP status.

PIC CANopen status machine

As above, the status machines could be divided into 3 parts: "power disabled", "power enabled" and "fault". All the states will be into "Fault" status after alarm. After power enabled, servo drive will finish initiating and then enter SWITECH_ON_DISA status. Now we could configure the servo drive, for example, set the working mode of the servo drive as position profile mode. At this time, the main

power supply is still shut down and the servo motor is now disabled. After the state transition 2, 3 and 4, the servo drive will be in OPERATION ENABLE mode. At this time, the main power will be activated and servo drive starts to control the servo motor according to the configured working mode. So, before this state, we have to be sure to configure the servo drive's parameters correctly. State Transition 9 will be used to shut down the main power supply. Once alarm happens to the servo drive, the servo drive's state will be in FAULT state.

states	description
Not Ready to Switch On	Servo drive is initiating.
Switch On Disabled	Initiation completed.
Ready to Switch On	Servo drive enters Switch On state. The servo motor is not servo-on yet.
Switched On	Servo drive ready and main power is on
Operation Enable	Servo on and control the servo motor according to the control mode.
Quick Stop Active	Servo drive stops in pre-defined method
Fault Reaction Active	Servo drive detects alarm and stop according to pre-defined method. Servo motor is still on.
Fault	Servo off

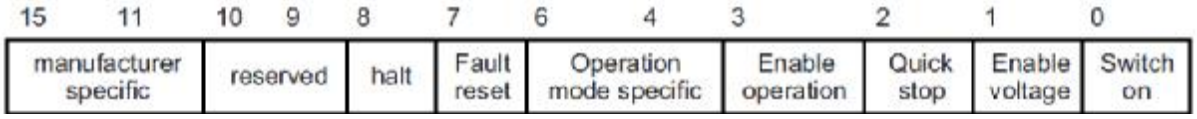
4.2 Parameters for device control.

Index	Object	Name	Type	Attr.
6040 h	VAR	Controlword	UINT16	RW
6041 h	VAR	Statusword	UINT16	RO
605A h	VAR	Quick stop option code	INT16	RW
605B h	VAR	Shutdown option code	INT16	RW
605C h	VAR	Disabled operation option code	INT16	RW
605D h	VAR	Halt option code	INT16	RW
605E h	VAR	Fault reaction option code	INT16	RW

4.2.1 controlword

Index	6040 h
Name	Control word
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	--
Default Value	0

Control word bit description：



Bit0 ~ 3 and Bit7：

The transmission of state machine will be triggered by the command composed by these 5 bits.

device control command list

Command	Bit of the controlword					
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	Transitions
Shutdown	0	×	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	×	×	0	×	7,9,10,12
Quick stop	0	×	0	1	×	7,9,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset	↑	×	×	×	×	15

note: X means this bit could be ignored.

Bit4、5、6、8：

In different control mode, these 4 bits' definition will be different.

Bit	Control mode		
	profile position mode	profile velocity mode	homing mode
4	New set point	reserved	Start homing operation
5	Change set immediately	reserved	reserved
6	abs/rel	reserved	reserved
8	Halt	Halt	Halt

The other bits: All reserved.

4.2.2 statusword

Index	6041 h
Name	statusword
Object Code	VAR
Data Type	UINT16
Access	RO
PDO Mapping	YES
Units	--
Value Range	--
Default Value	--

Statusword bit introduction is as below ...

bit	introduction
0	Ready to switch on
1	Switched on
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7	Warning
9~8	reserved
10	Target reached
11	Internal limit active
13~12	Operation mode specific
15~14	reserved

Bit0 ~ 3 、Bit5 and Bit6：

The combination of these bits represents the status of the servo drive

Value（binary）	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit4： Voltage enabled

When this bit is 1, it means the main power is on.

Bit5： Quick stop

When this bit is 0, it means the servo drive will stop the servo motor according to the configuration（$605A_h$：quick_stop_option_code）

Bit7： Warning

When the bit is 1, it means the servo drive detects alarm.

Bit10： Target reached

In different control mode, this bit has different meanings.

In Profile Position Mode, when the set position is reached, this bit will be set as 1. When Halt is activated and speed decreases to zero, this bit will be set as 1. When a new position is set, this bit will be cleared.

In Profile Velocity Mode, when the speed reaches the required speed, this bit will be set as 1. When Halt is activated, the speed will decrease to zero and this bit will be set as 1.

Bit11： Internal limit active

\When this bit is 1, it means that the internal torque has surpassed the set value.

Bit12、13：

These two bits in different control mode have different meaning...

Bit	Control mode		
	profile position mode	profile velocity mode	homing mode
12	Set-point acknowledge	Speed	Homing attained
13	Following error	Max slippage error	Homing error

The other bits

All reserved

4.2.3 shutdown_option_code

When Operation Enable mode is transit to Ready to Switch On status, Shutdown_option_code will be used to define how to stop the servo motor.

Index	$605B_h$
Name	Shutdown option code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1
Default Value	0

value	Introduction
0	Shutdown servo excitation signal. Servo motor will stop freely.
1	After the servo motor decelerates and stops,

	the servo excitation signal will be shut down.

4.2.4 disable_operation_option_code

.When the status of Operation Enable transits to Switched On status, Disable operation option code will decide how to halt.

Index	605C h
Name	Disable operation option code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1
Default Value	0

Value	Introduction
0	Shutdown servo excitation signal. Servo motor will stop freely.
1	After the servo motor decelerates and stops, the servo excitation signal will be shut down.

4.2.5 quick_stop_option_code

When the operation enable status transits to Quick Reaction Active status, quick_stop_option_code will define how to stop.

Index	605A h
Name	quick_stop_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1,2,5,6
Default Value	0

Value	Introduction
0	Shutdown servo excitation signal. Servo motor will stop freely.
1	After the servo motor decelerates and stops, the servo excitation signal will be shut down.
2	After servo motor stops urgently, the servo excitation signal will be shut down.
5	After the servo motor decelerates to zero, it will still stay in QuickStop status.
6	After the servo motor stops urgently, it will still

	stay in QuickStop status.

4.2.6 halt_option_code

When bit8 of Controlword is 1, halt option code will define how to halt. .

Index	605D h
Name	halt_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	1,2
Default Value	0

Value	Introduction
1	Servo motor will decelerate gradually to zero
2	Servo motor will decelerate urgently and then stop.

4.2.7 fault_reaction_option_code

When it detects alarm，fault_reaction_option_code will decide how to halt. .

Index	605D h
Name	fault_reaction_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0
Default Value	0

Value	Introduction
0	The servo excitation signal will be shut down and servo motor will stop freely.

4.3 Control mode

Now Pronet servo drive supports 4 control modes...

HOMING MODE
PROFILE VELOCITY MODE
PROFILE POSITION MODE
CYCLIC SYNCHRONIZATION POSITION

This chapter will mainly describe these 4 control methods as above.

4.4 Control mode parameters.

Index	Object	Name	Type	Attr.
6060 h	VAR	modes_of_operation	INT8	RW
6061 h	VAR	modes_of_operation_display	INT8	RO

modes_of_operation

Servo drive's control mode is defined by modes_of_operation.

Index	6060 h
Name	modes_of_operation
Object Code	VAR
Data Type	INT8
Access	RW
PDO Mapping	YES
Units	--
Value Range	1,3,6
Default Value	0

Value	Introduction
0	Not any control mode
1	PROFILE POSITION MODE
3	PROFILE VELOCITY MODE
6	HOMING MODE
8	CYCLIC SYNCHRONIZATION POSITION

modes_of_operation_display

Servo drive's current control mode could be read from the modes_of_operation_display.

Index	6061 h
Name	modes_of_operation_display
Object Code	VAR
Data Type	INT8
Access	RO
PDO Mapping	YES
Units	--
Value Range	1,3,6
Default Value	0

Note: 1. Only through modes_of_operation_display patameters, we could know the control mode of the servo drive.

2. Only in Target Reached status, servo drive's control mode can be transit to configured control mode. And then modes_of_operation_display could be the same as modes_of_operation.

4.5 HOMING MODE

。PRONET servo drive now supports multiple homing methods. Clients could choose the homing method that suits the motor type and application. For example, if the servo drive uses incremental encoder, we could choose C pulse to do the homing. If the servo drive is using serial encoder or resolver, we couldn't use C pulse as the homing method.

Clients can set homing method, homing speed and acceleration. After the servo drive finds the reference point, we could also set the distance between homing position and reference point as much as the value defined by home_offset（607C $_h$）.

Control word of homing mode

15 ~ 9	8	7 ~ 5	4	3 ~ 0
*	Halt	*	home_start_operation	*

*: please referred to previous chapters

Name	Value	Description
Homing operation start	0	Homing mode inactive
	0 → 1	Start homing mode
	1	Homing mode active
	1 → 0	Interrupt homing mode
Halt	0	Execute the instruction of bit 4
	1	Stop axle with homing acceleration

4.5.1 State word of homing mode.

15 ~ 14	13	12	11	10	9 ~ 0
*	homing_error	homing_attained	*	target_reached	*

*: Please refer to the previous chapters

Name	Value	Description
Target reached	0	Halt = 0: Home position not reached Halt = 1: Axle decelerates
	1	Halt = 0: Home position reached Halt = 1: Axle has velocity 0
Homing attained	0	Homing mode not yet completed
	1	Homing mode carried out successfully
Homing error	0	No homing error
	1	Homing error occurred; Homing mode carried out not successfully; The error cause is found by reading the error code

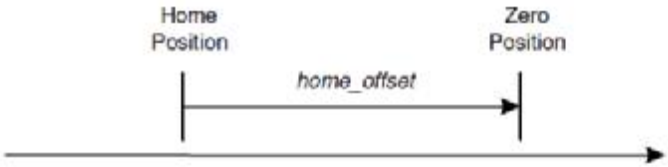
4.5.2 parameters related to homing mode

Index	Object	Name	Type	Attr.
607C $_h$	VAR	home_offset	INT32	RW
6098 $_h$	VAR	homing_method	INT8	RW
6099 $_h$	ARRAY	homing_speeds	UINT32	RW

609A h	VAR	homing_acceleration	INT32	RW

home_offset

home_offset defines the distance between reference position and homing position.



Pic Homing mode

Index	607C h
Name	home_offset
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	0

homing_method

There are 4 signals as homing signals: positive limit switch, negative limit switch, reference position switch and C pulse.

Index	6098 h
Name	homing_method
Object Code	VAR
Data Type	INT8
Access	RW
PDO Mapping	YES
Units	--
Value Range	1,2,3,4,17,18,19,20
Default Value	1

Homing method table

Method	Direction	Target position	reference position	DS402
1	negative	NOT	C pulse	1
2	positive	POT	C pulse	2
3	negative	reference position	C pulse	3

		switch		
4	positive	Reference position switch	C pulse	4
17	negative	NOT	NOT	17
18	positive	POT	POT	18
19	negative	reference position switch	reference position switch	19
20	positive	reference position switch	reference position switch	20

homing_speeds

Two kinds of speed are used in finding the reference position: The speed to find reference position and the speed to find homing position.

Index	6099 $_h$
Name	homing_speeds
Object Code	ARRAY
No. of Elements	2
Data Type	INT32

Sub-Index	01 $_h$
Name	speed_during_search_for_switch
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

Sub-Index	02 $_h$
Name	speed_during_search_for_zero
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

homing_acceleration

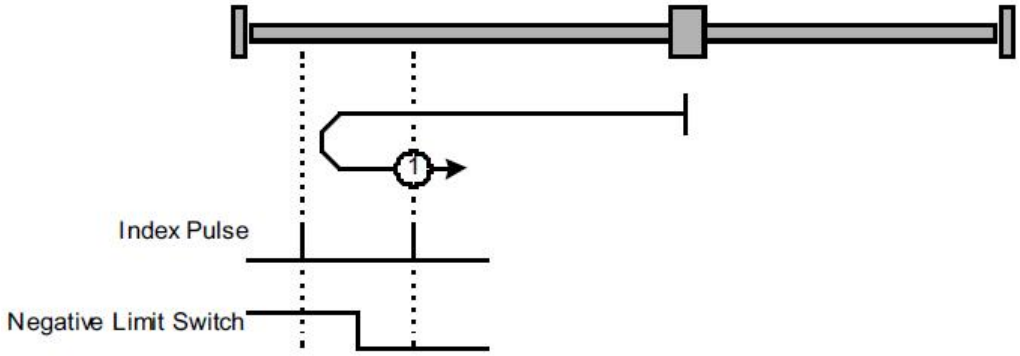
Acceleration and deceleration in homing are all defined by homing_acceleration.

Index	609A $_h$
Name	homing_acceleration
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	0

4.5.3 Homing method

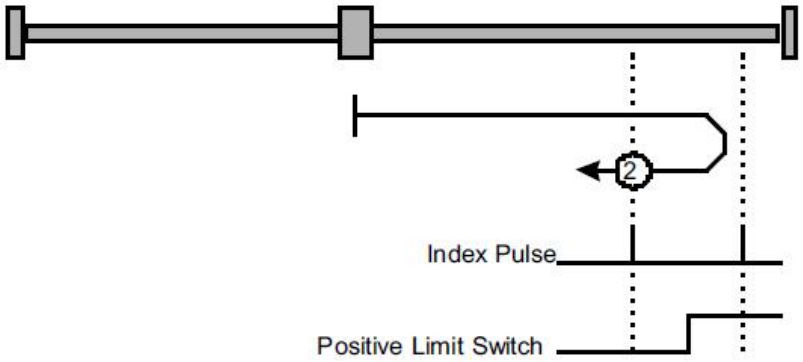
Homing method 1: Use C pulse and negative limit switch

Servo drive needs to move at first toward negative direction fast till hitting the negative limit switch and then decelerate till stop. And then, servo motor will be bounced back slowly and find the target homing position. Under this homing method, the target homing position is the first C pulse away from the limit switch.



Homing method 2: Use C pulse and positive limit switch

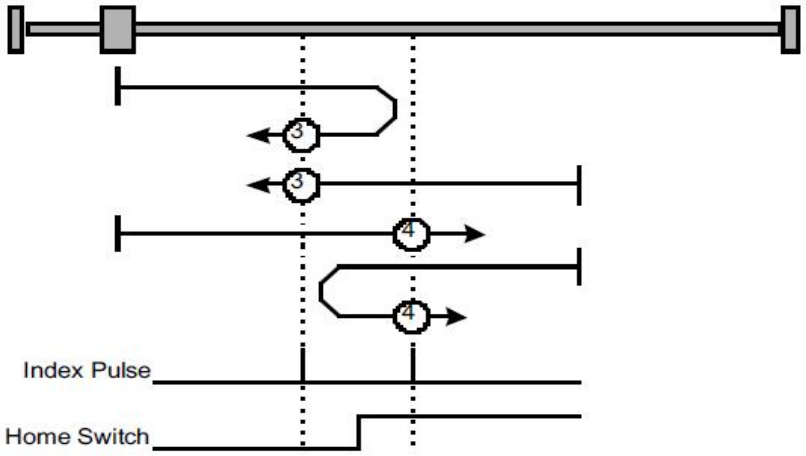
At first servo motor will move fast toward positive direction and decelerate to stop after hitting the positive limit switch. And then servo motor will be bounced back slowly to find homing position. Under this homing method, the target homing position is the first C pulse away from the limit switch.



Home method 3 and 4: Use C pulse and reference limit switch

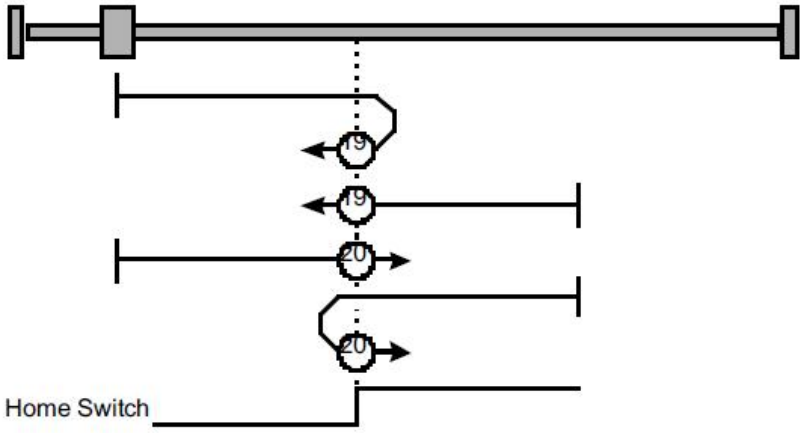
Servo drive's initial moving direction is relied on the status of reference point limit

switch. The target homing position is on the left side or right side of the reference limit switch. The distance between the reference position switch and homing position is one C pulse.

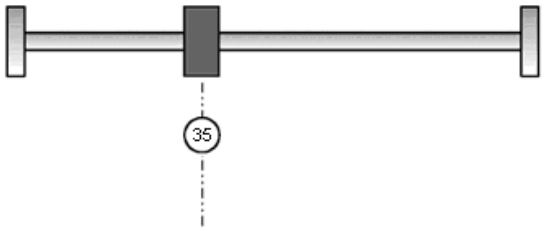


Homing method 17 ~ 20 Not to use C pulse

These 4 homing methods are similar to approach 1-4 but the target homing position is not relied on C pulse any more but on the change of limit switch or reference point. For example, as below, method 19 and method 20 are just similar to method 3 and method 4.



Homing method 35: set current position as the homing point.



PROFILE VELOCITY MODE

4.5.4 Control word of speed mode.

15 ~ 9	8	7 ~ 4	3 ~ 0
*	Halt	*	*

*:Refer to previous chapters

Name	Value	Description
Halt	0	Execute the motion
	1	Stop axle

4.5.5 Status word of speed mode

15 ~ 14	13	12	11	10	9 ~ 0
*	MaxSlippageError	Speed	*	Target reached	*

*: Refer to previous chapters

Name	Value	Description
Target reached	0	Halt = 0: Target position not reached Halt = 1: Axle decelerates
	1	Halt = 0: Target velocity reached Halt = 1: Axle has velocity 0
Speed	0	Speed is not equal 0
	1	Speed is equal 0
Max slippage error	0	Maximum slippage not reached
	1	Maximum slippage reached

4.5.6 Parameters in Velocity control mode

Index	Object	Name	Type	Attr.
6069 h	VAR	velocity_sensor_actual_value	INT32	RO
606B h	VAR	velocity_demand_value	INT32	RO
606C h	VAR	velocity_actual_value	INT32	RO
609D h	VAR	velocity_window	UINT16	RW
606E h	VAR	velocity_window_time	UINT16	RW
606F h	VAR	velocity_threshold	UINT16	RW
6070 h	VAR	velocity_threshold_time	UINT16	RW
60FF h	VAR	target_velocity	INT32	RW

velocity_sensor_actual_value

The master could read the velocity_sensor_actual_value to know the current velocity. The parameter's unit is internal speed unit.

Index	6069 h
Name	velocity_sensor_actual_value
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	0.1rmps (1R/10min)
Value Range	--
Default Value	--

velocity_demand_value

Master can read velocity_demand_value to know the current reference speed value of the servo drive. The unit of this parameter is user's velocity unit.

Index	606B h
Name	velocity_demand_value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	--

velocity_actual_value

The master can read velocity_ actual _value to know the current velocity of the servo motor. The unit of this parameter is user's velocity unit.

Index	606C h
Name	velocity_actual_value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	--

velocity_window

The difference bewteen Velocity_actual_value（606C h）and target_velocity（60FF h）is defined as actual velocity error window. If the actual velocity error window is always smaller than velocity_window(606D h) within the time set by velocity_window_time（606E h ）, then bit 10 of status word (target_reached) will be set as 1 to indicate that the set velocity has been reached.

Index	606D h
Name	velocity_window
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	20 R/10min

velocity_window_time

Velocity window comparator is composed of velocity_window_time and velocity_window.

Index	606E h
Name	velocity_window_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	--
Default Value	0

velocity_threshold

Velocity_threshold indicates a range close to zero speed in order to define if the servo motor has already stopped.

Index	606F $_h$
Name	velocity_threshold
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	10 R/10min

velocity_threshold_time

Velocity_threshold_time is used to set the shorter time when servo motor's speed is under velocity threshold. The unit is: ms. When the time that servo motor's speed is lower than the threshold is more than velocity_threshold_time, status word bit 12(speed is zero) will be set as 1.

Index	6070 $_h$
Name	velocity_threshold_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	--
Default Value	0

target_velocity

target_velocity is reference speed.

Index	60FF $_h$
Name	target_velocity
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

4.6 PROFILE POSITION MODE

4.6.1 control word in profile position mode

15 ~ 9	8	7	6	5	4	3 ~ 0
*	Halt	*	abs / rel	change set immediately	New set-point	*

*: Please refer to previous chapters

Name	Value	Description
New Set-point	0	Does not assume target position
	1	Assume target position
Change set immediately	0	Finish the actual positioning and then start the next positioning
	1	Interrupt the actual positioning and start the next positioning
Abs/rel	0	Target position is an absolute value
	1	Target position is a relative value
Halt	0	Execute positioning
	1	Stop axle with profile deceleration (if not supported with profile acceleration)

4.6.2 control word of position profile

15 ~ 14	13	12	11	10	9 ~ 0
*	Following error	Set_point acknowledge	*	Target reached	*

*: please refer to previous chapters

Name	Value	Description
Target reached	0	Halt = 0: Target position not reached Halt = 1: Axle decelerates
	1	Halt = 0: Target position reached Halt = 1: Velocity of axle is 0
Set-point acknowledge	0	Trajectory generator has not assumed the positioning values (yet)
	1	Trajectory generator has assumed the positioning values
Following error	0	No following error
	1	Following error

4.6.3 Parameters related to position control

Index	Name	Type	Attr.	PDO Mapping	M/O
6040_h	Control word	UINT16	RW	YES	M
6041_h	Status word	UINT16	RO	YES	
607A $_h$	target_position	INT32	RW	YES	M
$607B_h$	Positin_range_limit	INT32	RW	NO	O
6081 $_h$	profile_velocity	UINT32	RW	YES	M
6082 $_h$	end_velocity	UINT32	RW	YES	O
6083 $_h$	profile_acceleration	UINT32	RW	YES	O
6084 $_h$	profile_deceleration	UINT32	RW	YES	O
6085 $_h$	quick_stop_deceleration	UINT32	RW	YES	O
6086 $_h$	motion_profile_type	INT16	RW	YES	M

target_position

Target_position is reference position and this position could be an incremental value or a absolute value. It is up to bit6 of control word.

Index	607A $_h$
Name	target_ position
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	0

profile_velocity

Profile_velocity is the speed that the servo motor could finally reach after acceleration.

Index	6081 $_h$
Name	profile_velocity
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

end_velocity

End_velocity is the speed when servo motor reaches the target_position. Normally we set this value as 0 in order to stop the servo motor when the servo motor reaches the requested position. But in continuous multiple position, this value could be set as a non-zero value.

Index	6082 h
Name	end_velocity
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

profile_acceleration

profile_acceleration is the acceleration speed before reaching the target position.

Index	6083 h
Name	profile_acceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	100000 R/10min/s

profile_deceleration

profile_deceleration is the deceleration speed before reaching the target position.

Index	6084 h
Name	profile_deceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	100000 R/10min/s

quick_stop_deceleration

quick_stop_deceleration is the deceleration speed in Quick Stop.

Index	6085 h
Name	quick_stop_deceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	200000 R/10min/s

motion_profile_type

Motion_profile_type is used to select the motion curve. Now we only support trapezoid speed curve.

Index	6086 h
Name	motion_profile_type
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0
Default Value	0

4.6.4 Function description.

There are two methods to allocate a reference position.

Single step setting:

After the servo motor reaches the target position, servo drive will inform the master of reference position reached. And the servo drive will start new motion after getting new target position. Before getting the new reference position, the velocity of the servo motor is zero.

Continuous setting:

After reaching the target position, the servo motor will keep moving toward next target position which is set in advance. In this way, the servo motor could move continuously without pause. Between two reference position, the servo motor doesn't need to decelerate to zero.

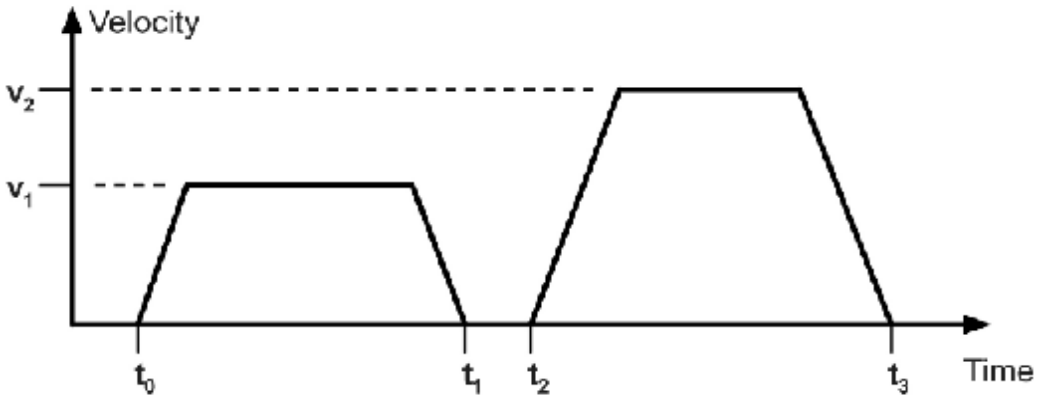
Two methods above could

Above two methods could be switched to each other by control word bit 4, bit 5 and statues word bit 12(set_point_acknowledge) in real time. Through hand shaking mechanism , we could pause the position control in the process and use these bits

above to reset the target position and then re-active and operate.

Single step setting procedure:

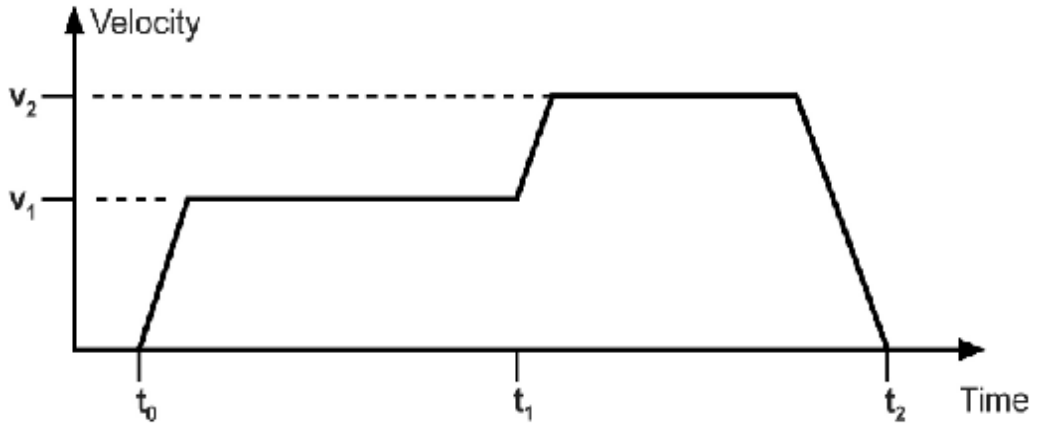
1) At first, set the NMT status into Operational and set the control mode parameter （6060 $_h$）as 1.
2) According to the actual demand, we could set the target position(target_positon: 607A $_h$）and so on.
3) We need set bit4 （new_set_point）of the control word as "1, bit 5 （change_set_immediately）as "0", bit 6(absolue/incremental) should be determined by whether the reference target position is an absolute value or an incremental value.
4) We use Bit12 (set_point_acknowledge) of the status word to configure the servo drive acknowledge mechanism. And then we start to operate position control.
5) After reaching the target position, servo drive will need to respond through bit 10(target_reached) of the status word. And then servo drive will follow the program to keep moving or accept new target position.



Continuous step setting procedure:

1) At first, we need to set NMT status into operational and set control mode（6060 $_h$）as 1. According to actual demand, we need to set the first target position(target_position: 607A $_h$), target speed, acceleration/deceleration and other relevant parameters.
2) Set bit 4 （new_set_point）of control word as 1. Set bit 5 （change_set_immediately）as 0. Set Bit6(absolute/incremental) according to the type of object position.
3) Set bit 12(set_point_acknowledge) of the status word and then start to operate position control.
4) Set the second target position(target_position:607A $_h$), target speed, acceleration/deceleration speed.
5) Set bit4(new_set_point) as 1, bit 5(change_set_immediately) as 0. Set Bit6(absoloue/relevant) according to the target position type.
6) After reaching the first target position, the servo drive will not stop and keep moving toward the second target position. After reaching the second target

position, the servo drive will respond through status word bit 10(target_reached). And then the servo motor will follow the program to keep moving or accept new target position.



4.7 CYCLIC SYNCHRONOUS POSITION MODE

Cyclic Synchronous position mode is similar to position interpolation mode. In this control mode, the master could offer extra speed and torque to achieve speed and torque feedforward control. The interpolation cycle time defines the time for target position updating. In this case, interpolation cycle time is the same as sync time.

4.7.1 parameters related to CYCLIC SYNCHRONOUS POSITION MODE

Index	Name	Type	Attr.	PDO Mapping	M/O
6040_h	Controlword	UINT16	RW	YES	M
6041_h	Statusword	UINT16	RO	YES	M
6041_h	Position_actual_value	INT32	RO	YES	M
$607A_h$	target_position	INT32	RW	YES	M
$607B_h$	Positin_range_limit	INT32	RW	NO	O
6081_h	profile_velocity	UINT32	RW	YES	M
6082_h	end_velocity	UINT32	RW	YES	O
6083_h	profile_acceleration	UINT32	RW	YES	O
6084_h	profile_deceleration	UINT32	RW	YES	O
6085_h	quick_stop_deceleration	UINT32	RW	YES	O

5 EtherCAT Communcation example

In this example, we use Beckhoff TwinCat software as the real time master. Please prepare as below before the test :

1) Identify the network interface model number and install the network interface correctly.

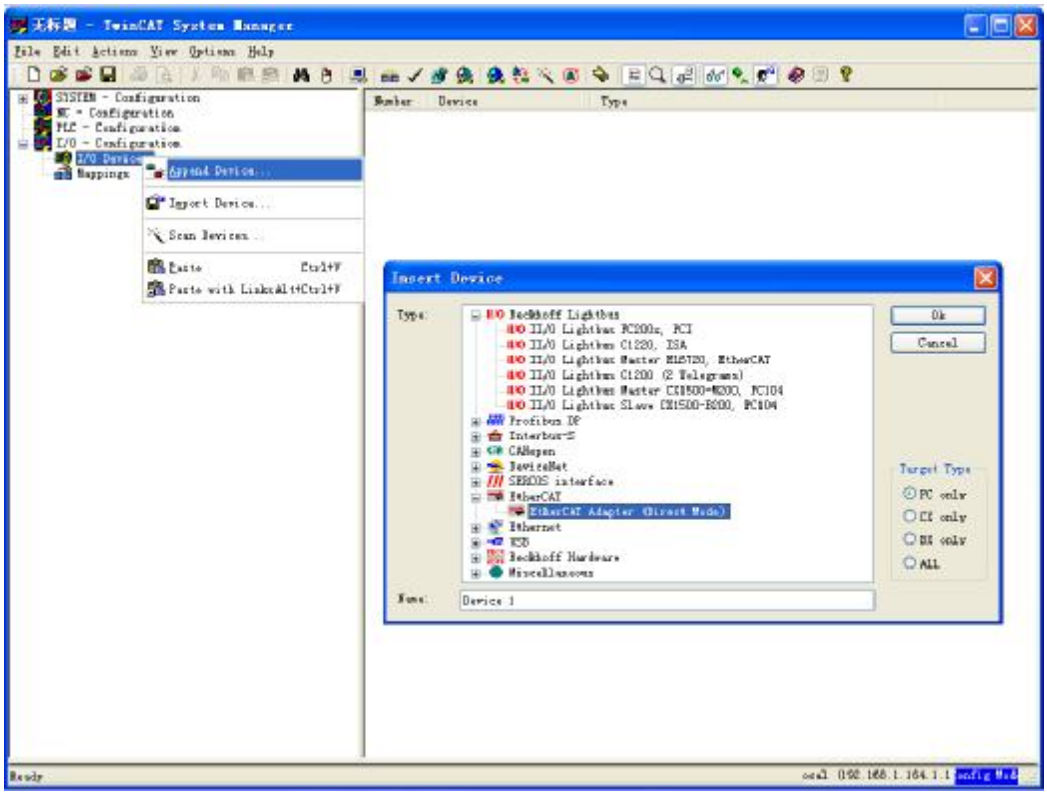
2) Install Beckhoff TwinCat software.

3) Copy the device description document (.XML document) to the directory c:\TwinCAT\IO\EtherCAT. (You could contact Estun to have this XML document)

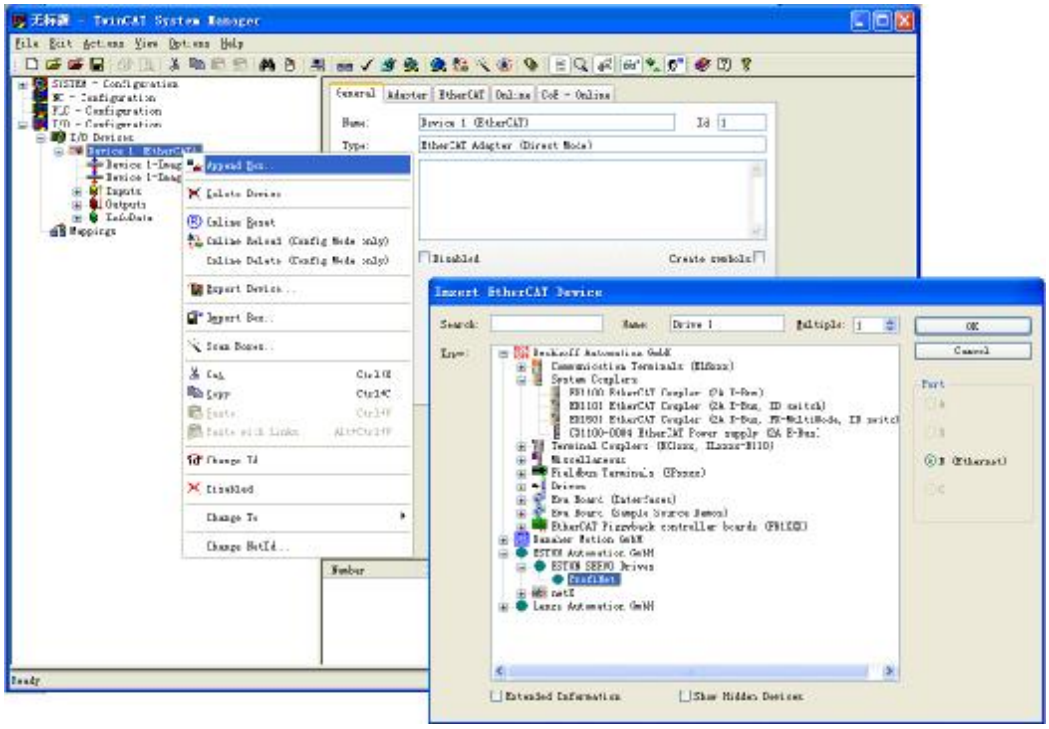
4) After finishing copying, reactivate TwinCAT software. Then TwinCAT will list an ESTUN Pronet servo drive EtherCAT bus option.

And then please follow steps as below ....

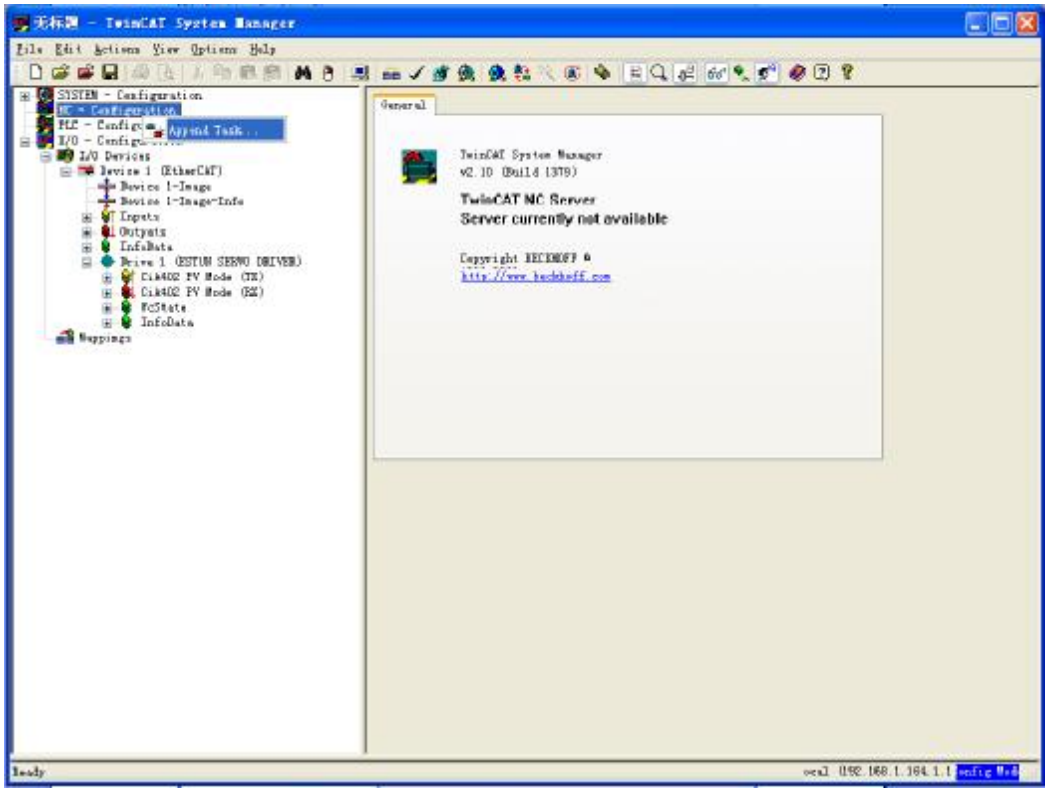
1. Use the right button of the mouse to single click I/O Device and choose EtherCAT network adapter. Name it as "Device 1".



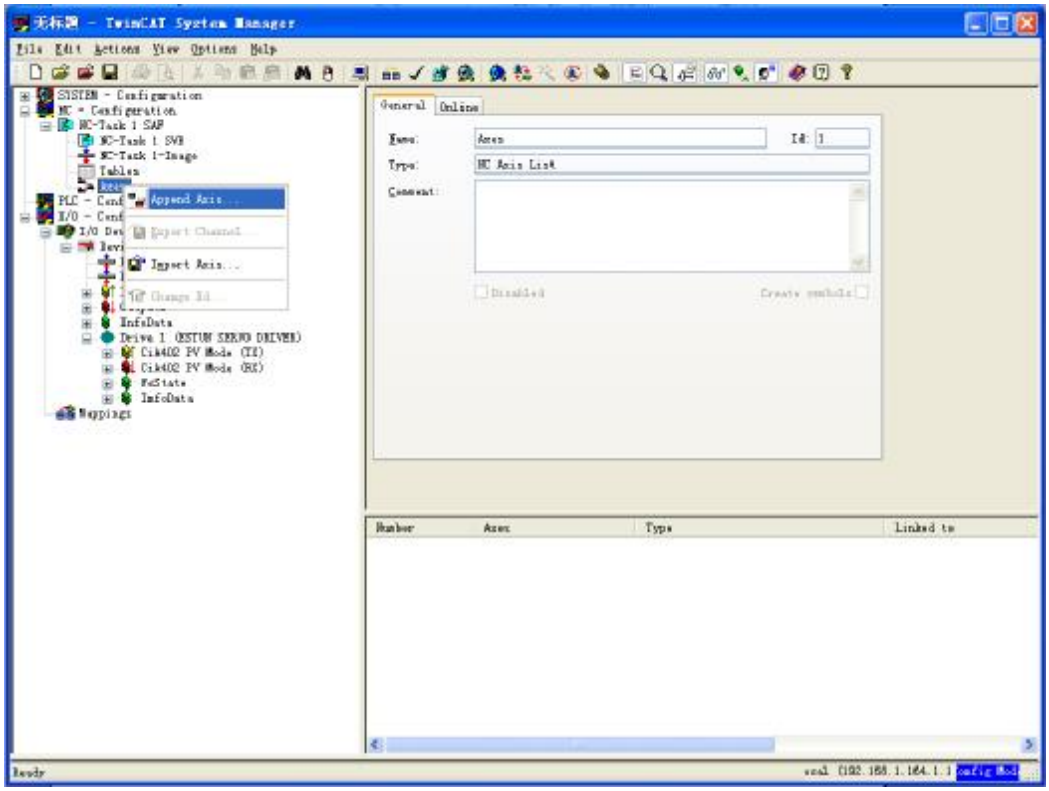
2. Use the right button of the mouse to single click Device 1 and add a slave Pronet device.



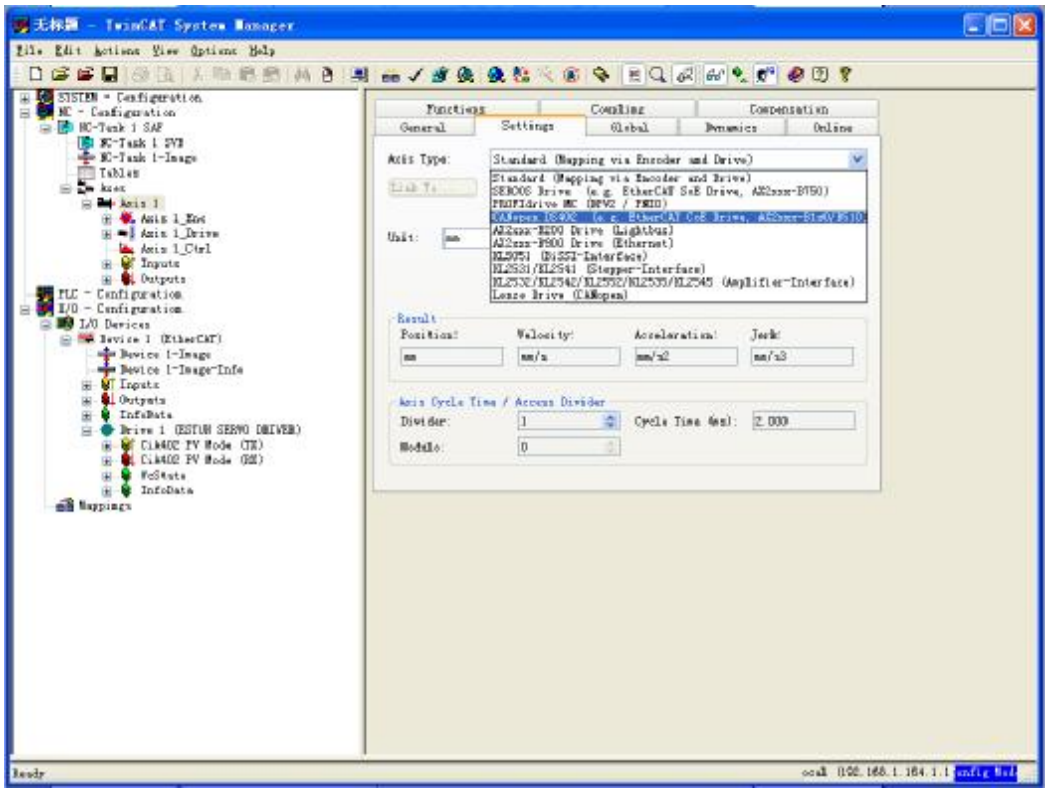
3. Add one NC task and name it as "Task 1".



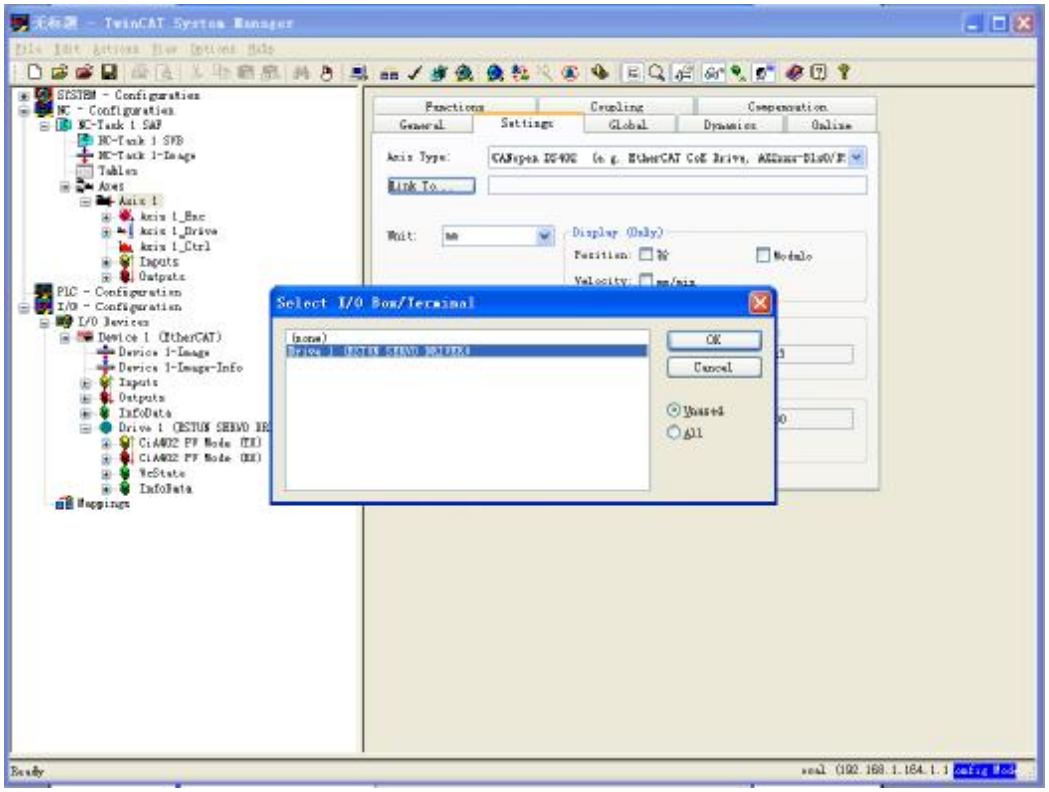
4. Add Axis 1 under NC task.



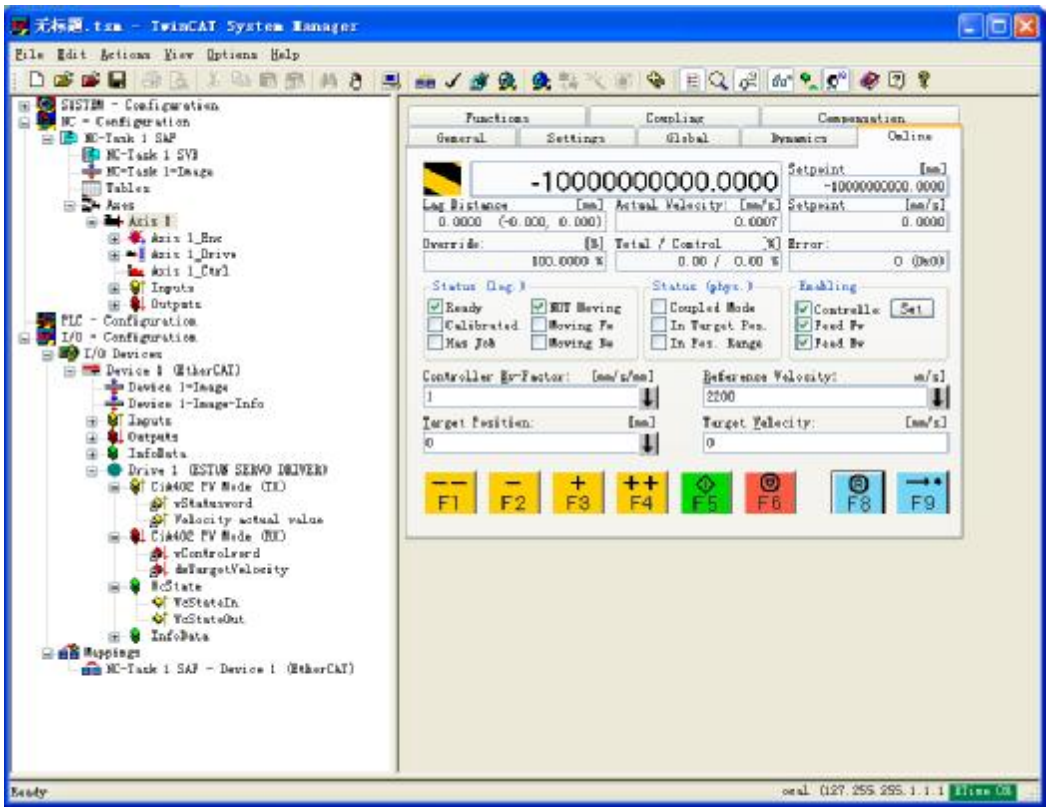
5. Choose application layer protocol "CoE"



6. Click "Link to" button and map servo drive axis to the device.



7. Click "activate configuration" button on the toolbar and activate configuration. Click "online" label and start to operate on servo axis.



Appendix: Object dictionary

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
1000	0	Device type	UINT32	RO	NO	●						
1001	0	Error register	UINT8	RO	NO	●						
1008	0	Manufacturer device name	STR	RO	NO	●						
1009	0	Manufacturer hardware version	STR	RO	NO	●						
100A	0	Manufacturer software version	STR	RO	NO	●						
1018	Identity Object											
	0	Number of entries	UINT8	RO	NO	●						
	1	Vender ID	UINT32	RO	NO	●						
	2	Product code	UINT32	RO	NO	●						
	3	Revision number	UINT32	RO	NO	●						
	4	Serial number	UINT32	RO	NO	●						
1600	1st Receive PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
1601	2nd Receive PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1602	3rd Receive PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1603	4^{th} Receive PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
1603	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1A00	1st Receive PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1A01	2nd Transmit PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						

Index	Sub index	Name	Type	Access.	PDO mapping	support						Unit
						All	PP	PV	HM	IP	CSP	
1A02	3rd Transmit PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1A03	4thTransmit PDO Mapping											
	0	Number of entries	UINT8	RW	NO	●						
	1	Mapping entry 1	UINT32	RW	NO	●						
	2	Mapping entry 2	UINT32	RW	NO	●						
	3	Mapping entry 3	UINT32	RW	NO	●						
	4	Mapping entry 4	UINT32	RW	NO	●						
	5	Mapping entry 5	UINT32	RW	NO	●						
	6	Mapping entry 6	UINT32	RW	NO	●						
	7	Mapping entry7	UINT32	RW	NO	●						
	8	Mapping entry 8	UINT32	RW	NO	●						
1C00	Sync Manager Communication Type											
	0	Number of used Sync Manager channels	UINT8	RW	NO	●						
	1	Communication type sync manager 0	UINT32	RW	NO	●						
	2	Communication type sync manager 1	UINT32	RW	NO	●						
	3	Communication type sync manager 2	UINT32	RW	NO	●						

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
1C00	4	Communication type sync manager 3	UINT32	RW	NO	●						
1C10	0	Sync Manager PDO assignment 0	UINT8	RO	NO	●						
1C10	0	Sync Manager PDO assignment 1	UINT8	RO	NO	●						
1C12	Sync Manager PDO assignment 2											
	0	Number of assigned PDOs	UINT8	RW	NO	●						
	1	Index of assigned RxPDO 1	UINT16	RW	NO	●						
	2	Index of assigned RxPDO 2	UINT16	RW	NO	●						
1C13	Sync Manager PDO assignment 3											
	0	Number of assigned PDOs	UINT8	RW	NO	●						
	1	Index of assigned TxPDO 1	UINT16	RW	NO	●						
	2	Index of assigned TxPDO 2	UINT16	RW	NO	●						
603F	0	Error code	UINT16	RW	YES	●						
6040	0	Control word	UINT16	RW	YES	●						
6041	0	Status word	UINT16	RO	YES	●						
605A	0	Quick stop option code	INT16	RW	NO	●						
605B	0	Shutdown option code	INT16	RW	NO	●						
605C	0	Disable operation option code	INT16	RW	NO	●						
605D	0	Stop option code	INT16	RW	NO	●						
605E	0	Fault reaction option code	UINT16	RW	NO	●						
6060	0	Modes of operation	INT8	RW	YES	●						
6061	0	Modes of operation display	INT8	RO	YES	●						
6062	0	Position demand value	INT32	RO	YES		●			●		position units
6063	0	Position actual value*	INT32	RO	YES		●			●		inc
6064	0	Position actual value	INT32	RO	YES		●			●		position units
6065	0	Following error window	UINT32	RW	YES		●					position units
6066	0	Following error time out	UINT16	RW	YES		●					ms
6067	0	Position window	UINT32	RW	YES		●					position units
6068	0	Position window time	UINT16	RW	YES		●					ms

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
6069	0	Velocity sensor actual value	UINT16	RW	YES			●				speed units
606B	0	Velocity demand value	INT32	RO	YES			●				speed units
606C	0	Velocity actual value	INT32	RO	YES			●				speed units
606D	0	Velocity window	UINT16	RW	YES			●				speed units
606E	0	Velocity window time	UINT16	RW	YES			●				ms
606F	0	Velocity threshold	UINT16	RW	YES			●				speed units
6070	0	Velocity threshold time	UINT16	RW	YES			●				ms
607A	0	Target position	INT32	RW	YES		●					position units
607B	Position range limit											
	0	Number of entries	UINT8	RW	NO		●			●		
	1	Min position range limit	INT32	RW	NO		●			●		position units
	2	Max position range limit	INT32	RW	NO		●			●		position units
607C	0	Home offset	INT32	RW	YES		●		●	●		position units
6081	0	Profile velocity	UINT32	RW	YES		●					speed units
6082	0	End velocity	UINT32	RW	YES		●					speed units
6083	0	Profile acceleration	UINT32	RW	YES		●	●				acceleration units
6084	0	Profile deceleration	UINT32	RW	YES		●	●		●		acceleration units
6085	0	Quick stop deceleration	UINT32	RW	YES		●	●		●		acceleration units
6086	0	Motion profile type	INT16	RO	YES		●	●		●		
6093	Position factor											
	0	Number of entries	UINT32	RW	NO		●		●	●		
	1	numerator	UINT32	RW	NO		●		●	●		
	2	divisor	UINT32	RW	NO		●		●	●		

Index	Sub index	Name	Type	Access.	PDO mapping	support						unit
						All	PP	PV	HM	IP	CSP	
6094	0	Velocity encoder factor	--	--	--	●						
	0	Number of entries	UINT32	RW	NO	●						
	1	numerator	UINT32	RW	NO	●						
	2	divisor	UINT32	RW	NO	●						
6097	0	Acceleration factor	--	--	--	●						
	0	Number of entries	UINT32	RW	NO	●						
	1	numerator	UINT32	RW	NO	●						
6098	0	Homing method	INT8	RW	YES				●			
6099	--											
	0	Number of entries	UINT8	RW	YES				●			
	1	Speed during search for switch	UINT32	RW	YES				●			speed units
	2	Speed during search for zero	UINT32	RW	YES				●			speed units
609A	0	Homing acceleration	UINT32	RW	YES				●			acceleration units
60C0	0	Interpolation sub mode select	INT16	RW	NO					●		
60C1	0	Interpolation data record	--	--	--					●		
	0	number of entries	UINT8	RO	NO					●		
	1	the first parameter of ip function	INT32	RW	YES					●		
	2	the second parameter of ip function	INT32	RW	YES					●		
60FA	0	Control effort	INT32	RO	YES		●			●		