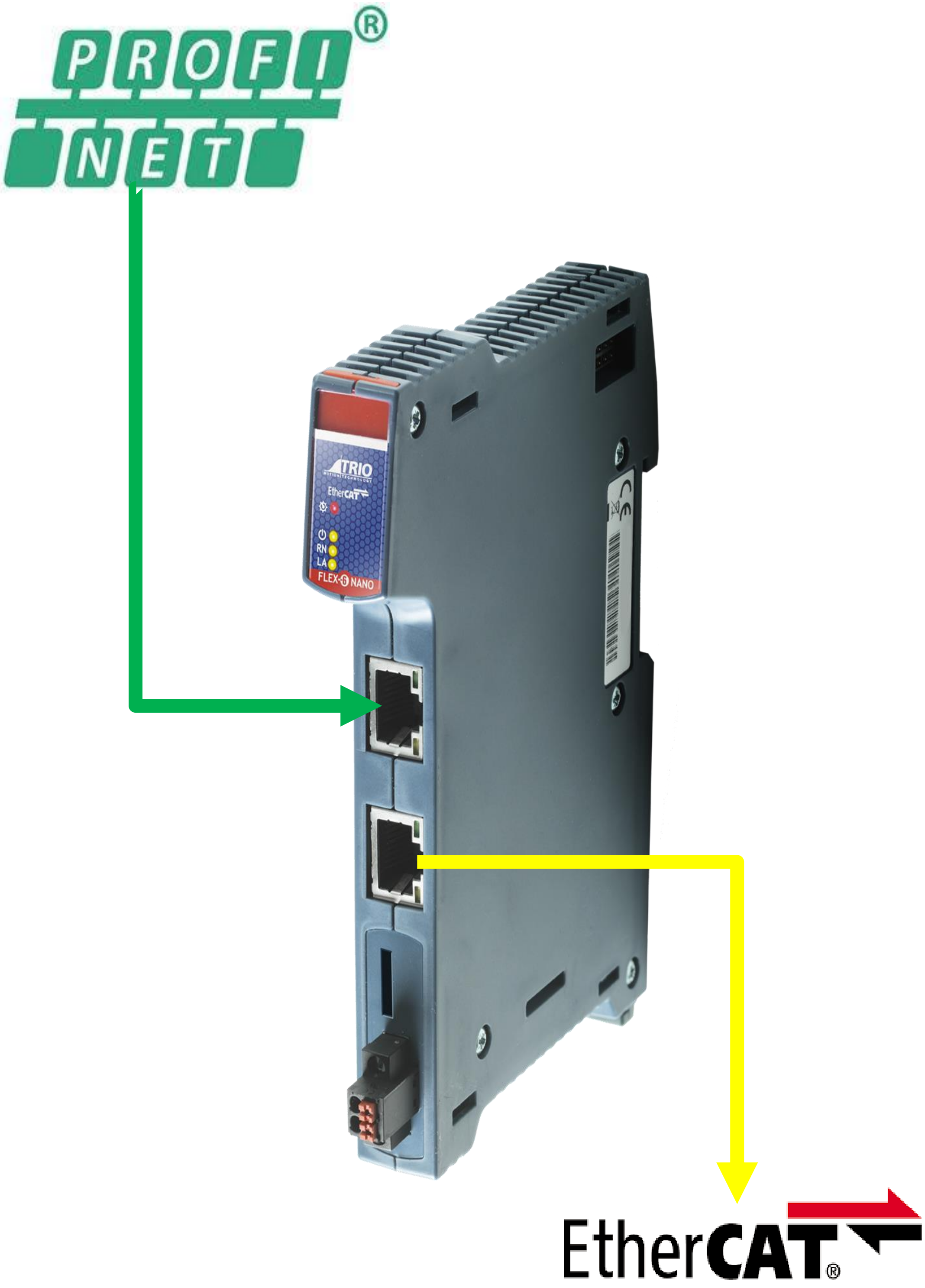
Technical Instruction

Profinet-Gateway



General:
The purpose of this instruction is to get started quickly with the Profinet-Gateway. This instruction describes how to control an Ethercat drive from a Siemens PLC via the Gateway.

The gateway itself can be seen as a black box and is therefore not described in this instruction, only the hardware connections will be described.

Explanations of Ethercat drives are beyond the scope of this instruction and will therefore not described. For this reference is made to the relevant manuals from Estun and Leadshine.

Principle and operation:
The principle of controlling an Ethercat drive is based on "Drive-based" control. This means that various Ethercat objects are provided with data. This data can include position, speed, acceleration, deceleration, mode or operation, etc.

These objects are described via the gateway from a Profinet master.

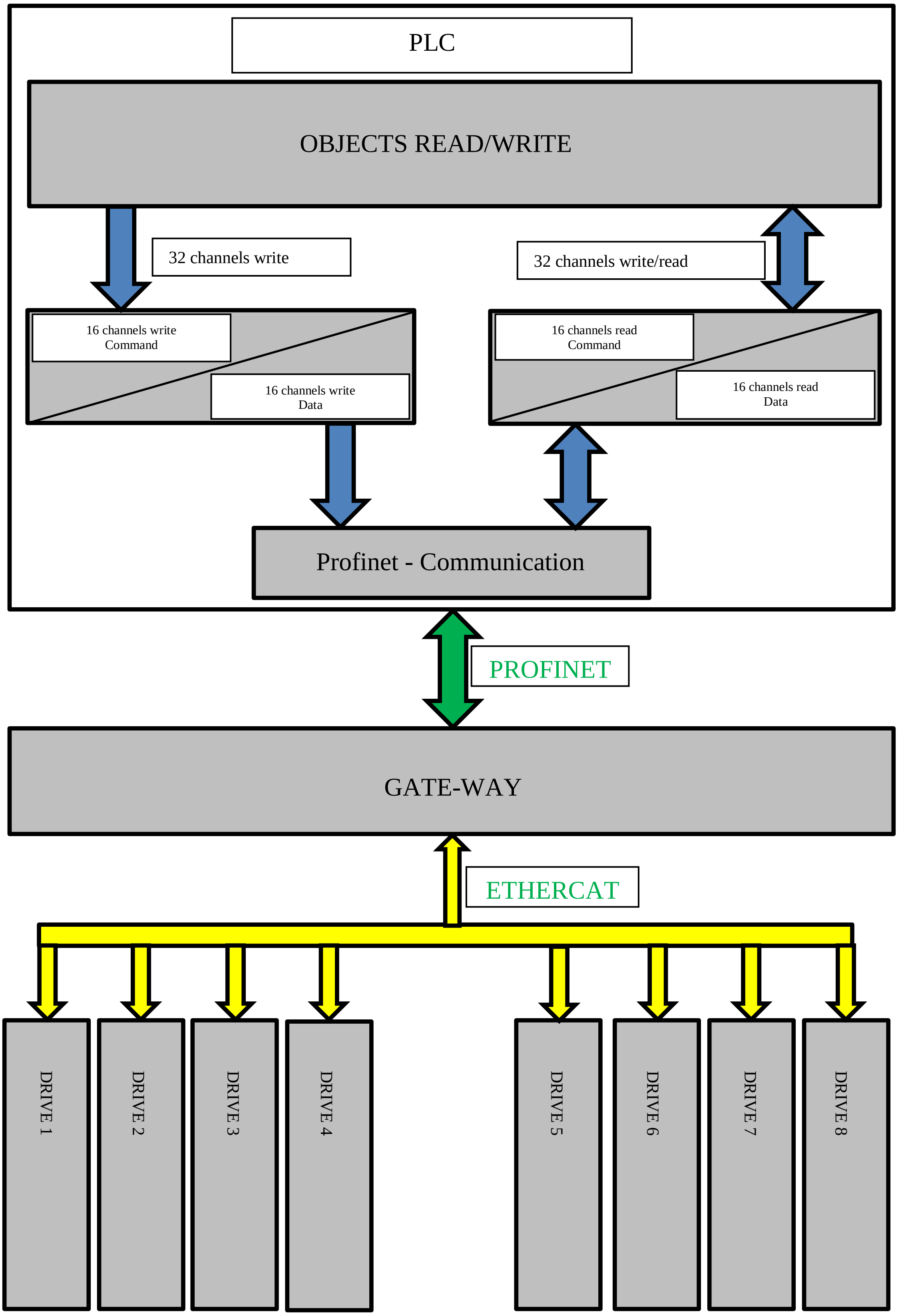
The gateway consists of 64 channels, each channel can communicate with the Profinet master, these 64 channels are split into 32 channels for writing/reading and 32 channels for writing.

These 32 write/read and 32 write channels are then divided into Object channels and data channels.

This means that 1 channel is used to read a particular Ethercat object and another channel is used for the associated data.

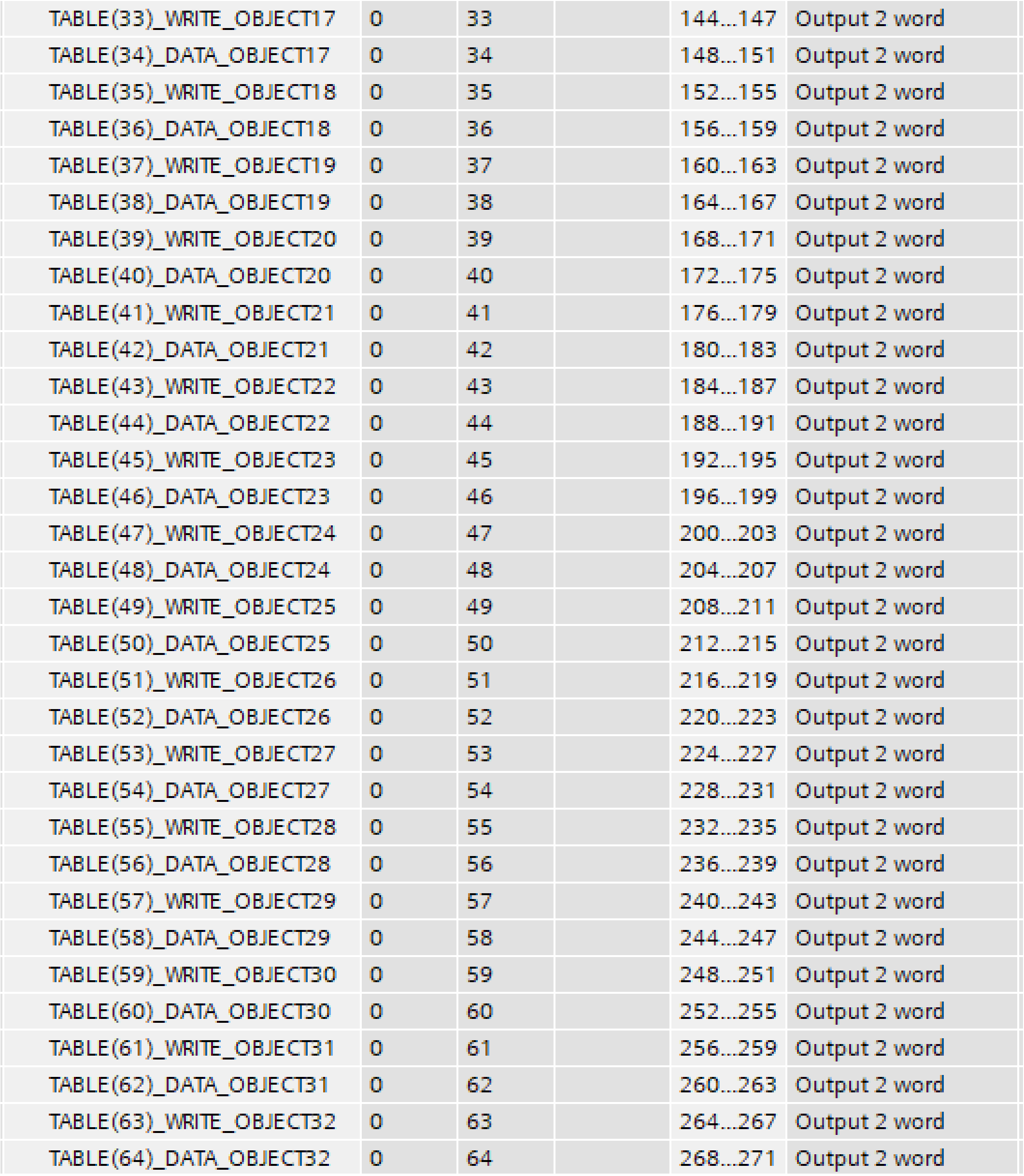
The same applies to writing, 1 channel for the Object to be written and the other channel the data associated with the object to be written.

This is shown in a block diagram below:



Block diagram read/write Ethercat objects

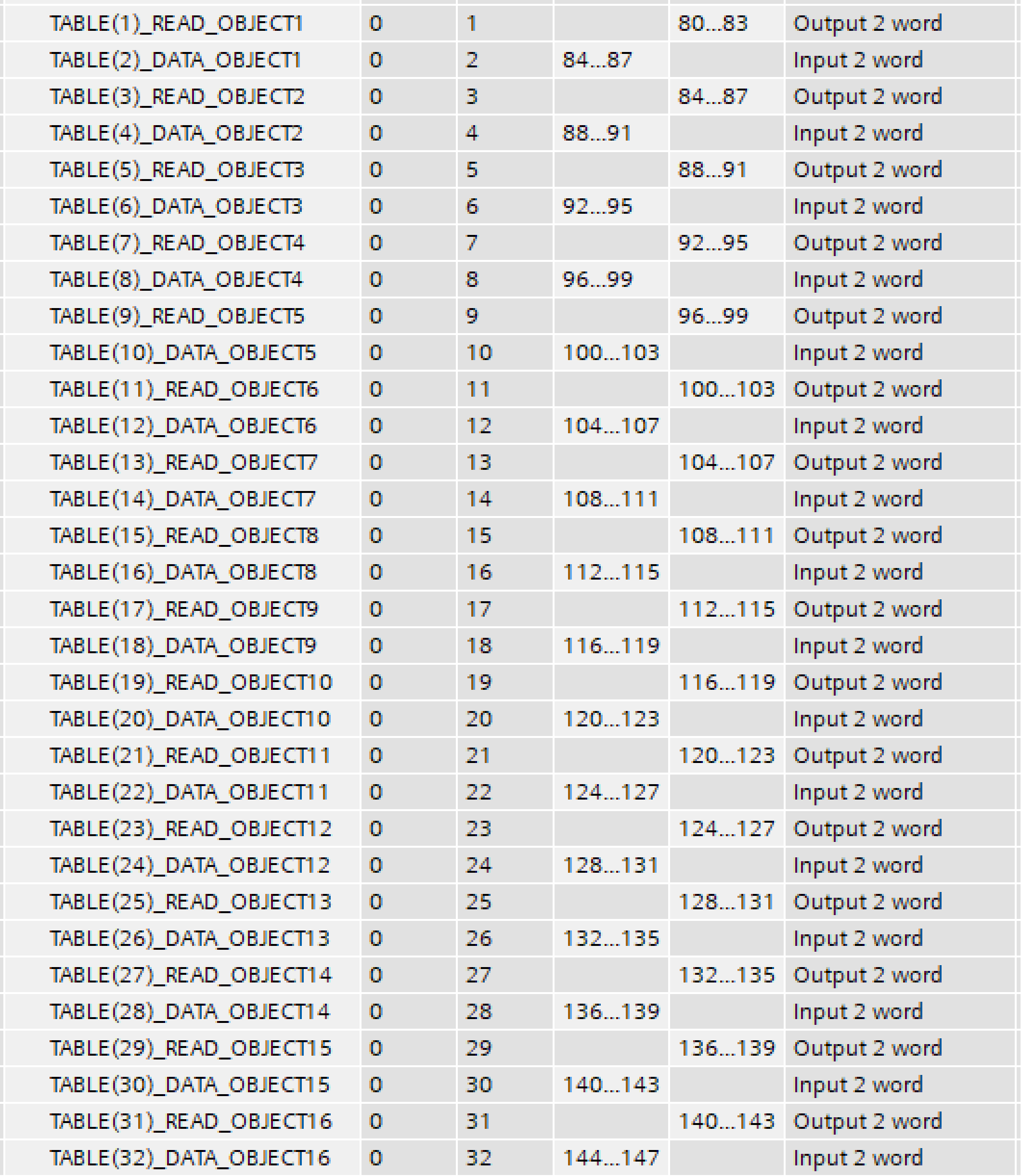
For writing it will look like the picture below:



TABLE(33)_WRITE_OBJECT17	0	33		144...147	Output 2 word
TABLE(34)_DATA_OBJECT17	0	34		148...151	Output 2 word
TABLE(35)_WRITE_OBJECT18	0	35		152...155	Output 2 word
TABLE(36)_DATA_OBJECT18	0	36		156...159	Output 2 word
TABLE(37)_WRITE_OBJECT19	0	37		160...163	Output 2 word
TABLE(38)_DATA_OBJECT19	0	38		164...167	Output 2 word
TABLE(39)_WRITE_OBJECT20	0	39		168...171	Output 2 word
TABLE(40)_DATA_OBJECT20	0	40		172...175	Output 2 word
TABLE(41)_WRITE_OBJECT21	0	41		176...179	Output 2 word
TABLE(42)_DATA_OBJECT21	0	42		180...183	Output 2 word
TABLE(43)_WRITE_OBJECT22	0	43		184...187	Output 2 word
TABLE(44)_DATA_OBJECT22	0	44		188...191	Output 2 word
TABLE(45)_WRITE_OBJECT23	0	45		192...195	Output 2 word
TABLE(46)_DATA_OBJECT23	0	46		196...199	Output 2 word
TABLE(47)_WRITE_OBJECT24	0	47		200...203	Output 2 word
TABLE(48)_DATA_OBJECT24	0	48		204...207	Output 2 word
TABLE(49)_WRITE_OBJECT25	0	49		208...211	Output 2 word
TABLE(50)_DATA_OBJECT25	0	50		212...215	Output 2 word
TABLE(51)_WRITE_OBJECT26	0	51		216...219	Output 2 word
TABLE(52)_DATA_OBJECT26	0	52		220...223	Output 2 word
TABLE(53)_WRITE_OBJECT27	0	53		224...227	Output 2 word
TABLE(54)_DATA_OBJECT27	0	54		228...231	Output 2 word
TABLE(55)_WRITE_OBJECT28	0	55		232...235	Output 2 word
TABLE(56)_DATA_OBJECT28	0	56		236...239	Output 2 word
TABLE(57)_WRITE_OBJECT29	0	57		240...243	Output 2 word
TABLE(58)_DATA_OBJECT29	0	58		244...247	Output 2 word
TABLE(59)_WRITE_OBJECT30	0	59		248...251	Output 2 word
TABLE(60)_DATA_OBJECT30	0	60		252...255	Output 2 word
TABLE(61)_WRITE_OBJECT31	0	61		256...259	Output 2 word
TABLE(62)_DATA_OBJECT31	0	62		260...263	Output 2 word
TABLE(63)_WRITE_OBJECT32	0	63		264...267	Output 2 word
TABLE(64)_DATA_OBJECT32	0	64		268...271	Output 2 word

Picture 1: channels for writing objects

For reading it will look like the picture below:



TABLE(1)_READ_OBJECT1	0	1		80...83	Output 2 word
TABLE(2)_DATA_OBJECT1	0	2	84...87		Input 2 word
TABLE(3)_READ_OBJECT2	0	3		84...87	Output 2 word
TABLE(4)_DATA_OBJECT2	0	4	88...91		Input 2 word
TABLE(5)_READ_OBJECT3	0	5		88...91	Output 2 word
TABLE(6)_DATA_OBJECT3	0	6	92...95		Input 2 word
TABLE(7)_READ_OBJECT4	0	7		92...95	Output 2 word
TABLE(8)_DATA_OBJECT4	0	8	96...99		Input 2 word
TABLE(9)_READ_OBJECT5	0	9		96...99	Output 2 word
TABLE(10)_DATA_OBJECT5	0	10	100...103		Input 2 word
TABLE(11)_READ_OBJECT6	0	11		100...103	Output 2 word
TABLE(12)_DATA_OBJECT6	0	12	104...107		Input 2 word
TABLE(13)_READ_OBJECT7	0	13		104...107	Output 2 word
TABLE(14)_DATA_OBJECT7	0	14	108...111		Input 2 word
TABLE(15)_READ_OBJECT8	0	15		108...111	Output 2 word
TABLE(16)_DATA_OBJECT8	0	16	112...115		Input 2 word
TABLE(17)_READ_OBJECT9	0	17		112...115	Output 2 word
TABLE(18)_DATA_OBJECT9	0	18	116...119		Input 2 word
TABLE(19)_READ_OBJECT10	0	19		116...119	Output 2 word
TABLE(20)_DATA_OBJECT10	0	20	120...123		Input 2 word
TABLE(21)_READ_OBJECT11	0	21		120...123	Output 2 word
TABLE(22)_DATA_OBJECT11	0	22	124...127		Input 2 word
TABLE(23)_READ_OBJECT12	0	23		124...127	Output 2 word
TABLE(24)_DATA_OBJECT12	0	24	128...131		Input 2 word
TABLE(25)_READ_OBJECT13	0	25		128...131	Output 2 word
TABLE(26)_DATA_OBJECT13	0	26	132...135		Input 2 word
TABLE(27)_READ_OBJECT14	0	27		132...135	Output 2 word
TABLE(28)_DATA_OBJECT14	0	28	136...139		Input 2 word
TABLE(29)_READ_OBJECT15	0	29		136...139	Output 2 word
TABLE(30)_DATA_OBJECT15	0	30	140...143		Input 2 word
TABLE(31)_READ_OBJECT16	0	31		140...143	Output 2 word
TABLE(32)_DATA_OBJECT16	0	32	144...147		Input 2 word

Picture 2: channels for reading objects

Taking into account the minimum objects required for controlling a drive, a maximum of 8 axes can be controlled with 1 gateway. It must be taken into account that these axes cannot be started simultaneously. All channels are handled sequentially.

First the channels are handled for writing and then only if there is new data for that channel to write, then the channels for reading. Even then only if there is new data available.

It should be taken into account a delay time of about 10ms between the reading / writing of two successive channels.

As mentioned earlier, controlling a drive requires at least a few objects for reading and writing, these are the following objects:

For writing:

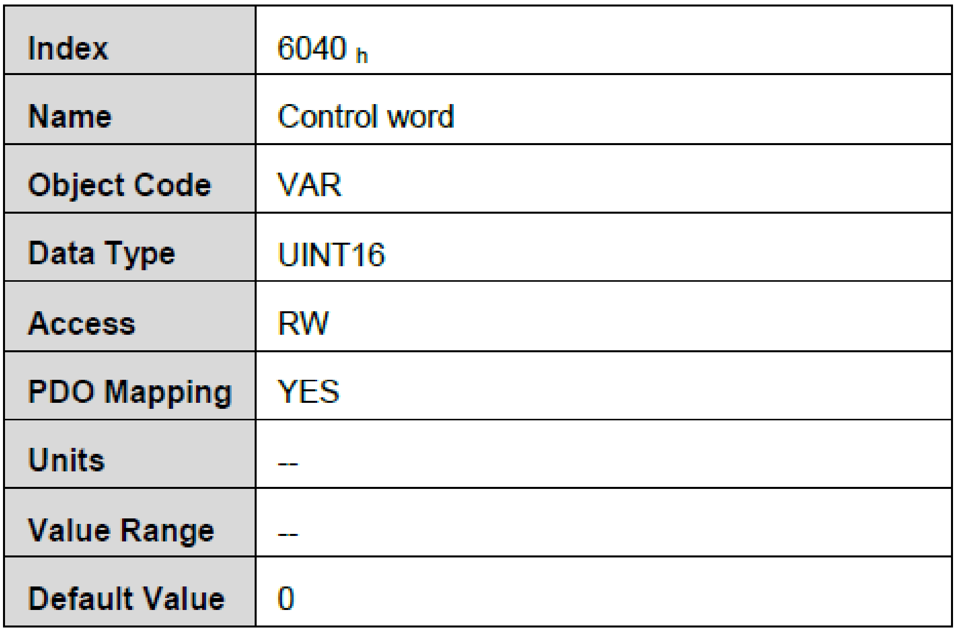
- Control Word
- Target Position

For reading:

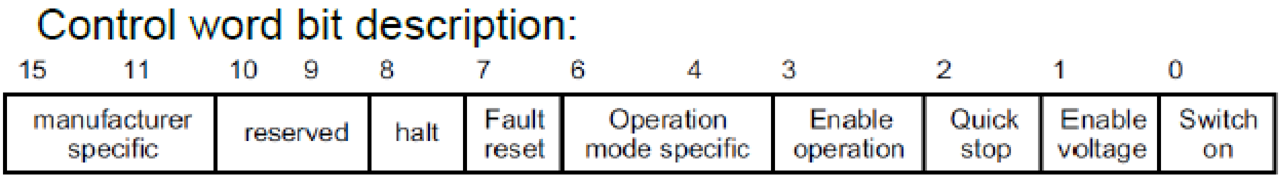
- Status Word
- Current Positon

In view of the above, it can be said that four read and four write channels are at least required to be able to control a Ethercat controller. This means that a maximum of 8 axes can be controlled per gateway. In order not to switch too much between different objects, it is preferable to control 4 axes per gateway.

An object for writing is constructed as follows, the Object of the ControlWord 6040hex is taken as an example.



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:

15 11	10 9	8	7	6 4	3	2	1	0
manufacturer specific	reserved	halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch on

The object consists of:

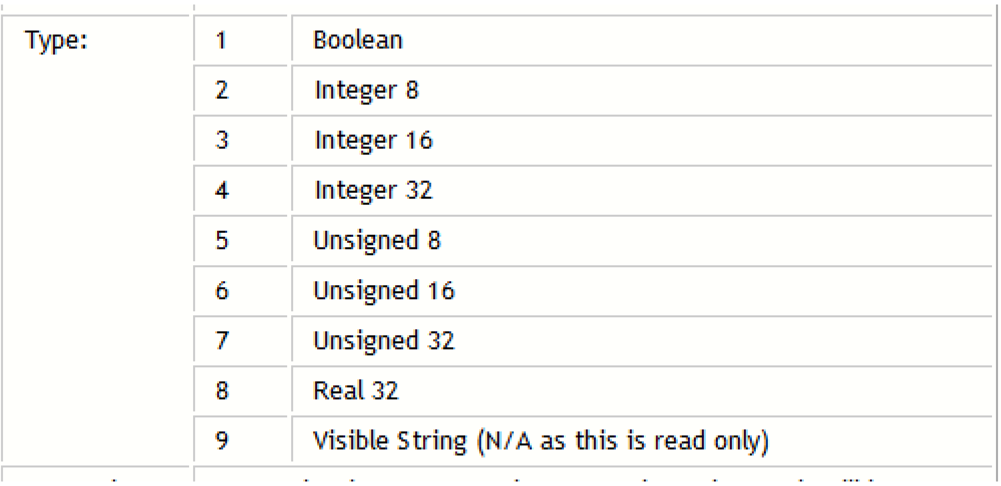
- Node
- Index
- Sub_index
- Type
- Reserved
- Data

Node is the drive number: 1 t/m maximum 8

Index is het Ethercat object address, in this example for the ControlWord 6040h.

Sub_index is ‘0’ in this example.

Type, is the data type of the object, in this example it is UINT16



Type:	1	Boolean
	2	Integer 8
	3	Integer 16
	4	Integer 32
	5	Unsigned 8
	6	Unsigned 16
	7	Unsigned 32
	8	Real 32
	9	Visible String (N/A as this is read only)

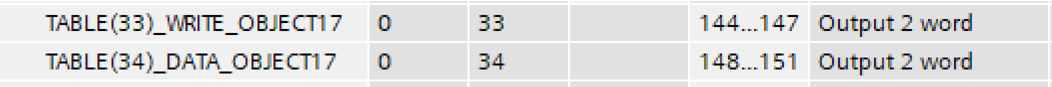
Reserved is a spare place and is not used.

Data is the value that is written in the relevant object.

For example, if you want to write the value 0 in the Object 6040h (Controlword for the first drive, you must send the following from the Profinet master:

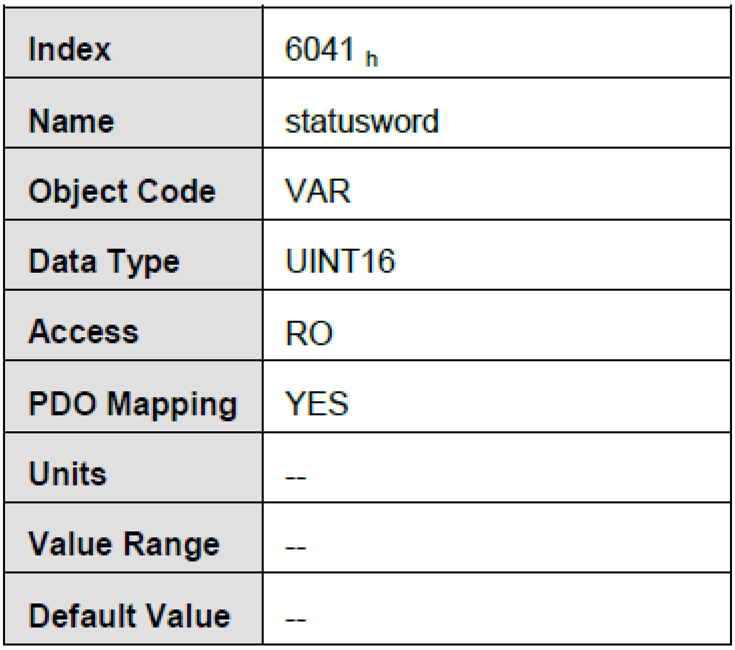
```
#Object.Object_Address[#uiObjectAddressWrite].Node := #iNode; (= 1)
#Object.Object_Address[#uiObjectAddressWrite].Object_address := 16#6040;
#Object.Object_Address[#uiObjectAddressWrite].Reserved := 0;
#Object.Object_Address[#uiObjectAddressWrite].Sub := 0;
#Object.Object_Address[#uiObjectAddressWrite].Type := 6;
#Object.Object_Data[#uiObjectAddressWrite] := 2#0000000000000000;
```

Aangezien voor de Node, Object_address, Reserved, Sub en type een kanaal gebruikt wordt en voor de data een kanaal, schrijft men in TABLE(33) het node nummer, Object_address, Reserved, Sub en type en in TABLE(34) de data met als waarde 0.



TABLE(33)_WRITE_OBJECT17	0	33		144...147	Output 2 word
TABLE(34)_DATA_OBJECT17	0	34		148...151	Output 2 word

Voor het lezen van bijvoorbeeld het StatusWord gaat dit op dezelfde manier, echter daar worden de kanalen TABLE(3) en TABLE(4) voor gebruikt.

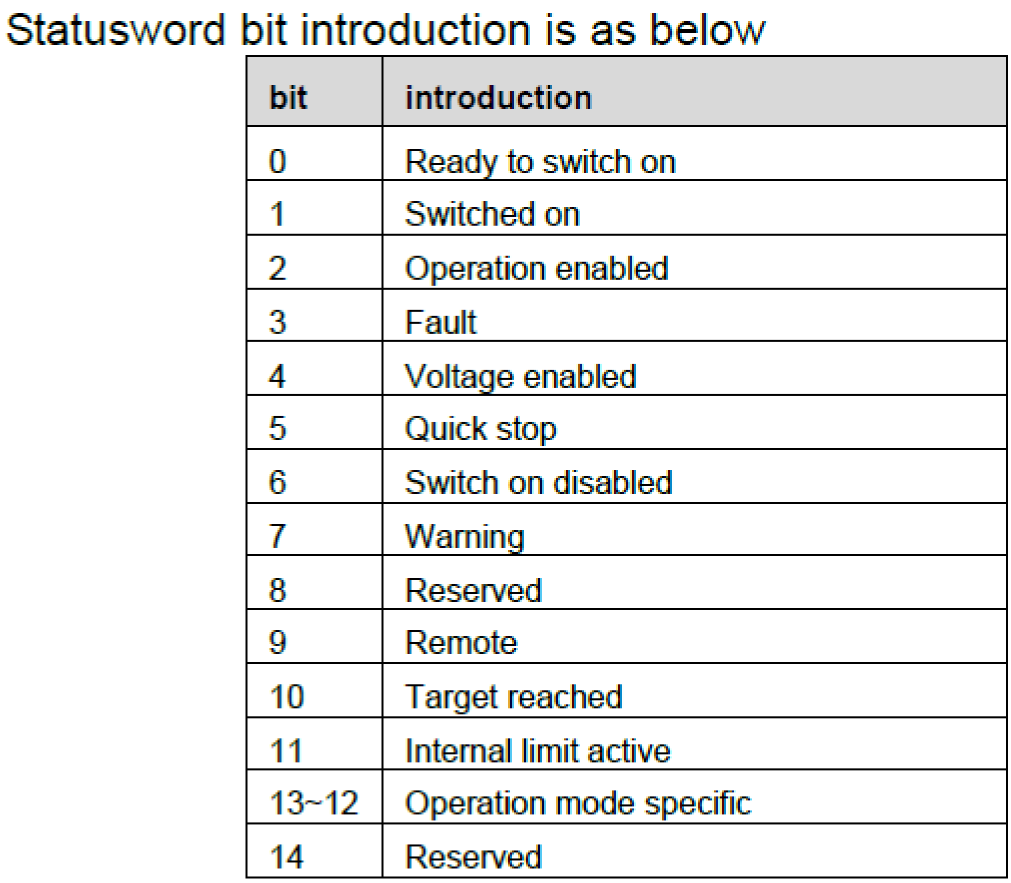


Index	6041 $_h$
Name	statusword
Object Code	VAR
Data Type	UINT16
Access	RO
PDO Mapping	YES
Units	--
Value Range	--
Default Value	--

If we want to read the value of the Status Word via Profibus, the following must be sent to the gateway:

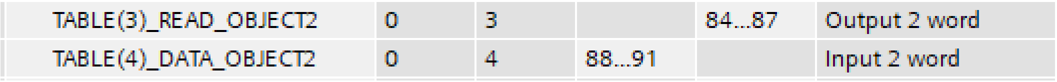
```
// Read Status
#Object.Object_Address[#uiObjectAddressRead+1].Node := #iNode; (= 1)
#Object.Object_Address[#uiObjectAddressRead+1].Object_address := 16#6041;
#Object.Object_Address[#uiObjectAddressRead+1].Sub := 0;
#Object.Object_Address[#uiObjectAddressRead+1].Type := 6;

#uiStatusWord := DINT_TO_UINT(#Object.Object_Data[#uiObjectAddressRead+1]);
```



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
8	Reserved
9	Remote
10	Target reached
11	Internal limit active
13~12	Operation mode specific
14	Reserved

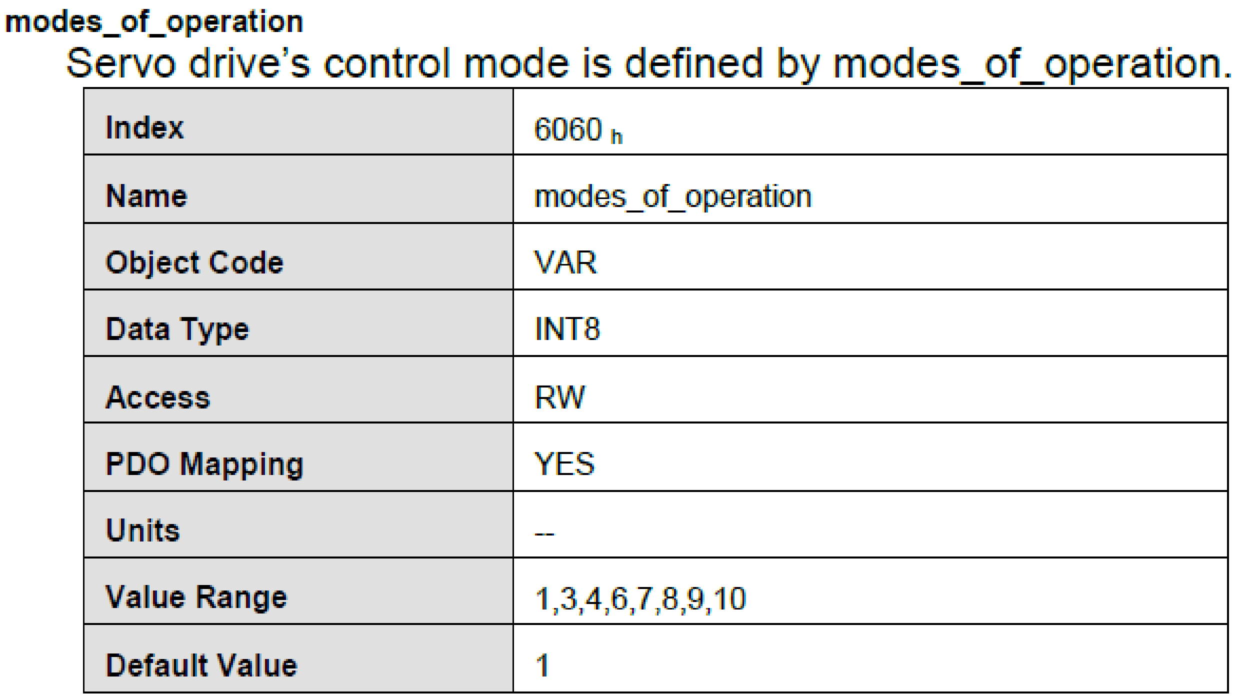


TABLE(3)_READ_OBJECT2	0	3		84...87	Output 2 word
TABLE(4)_DATA_OBJECT2	0	4	88...91		Input 2 word

Example to set "Mode of Operation" to homing mode for drive 1:

To set "Mode of Operation" in Homing mode, proceed as follows.

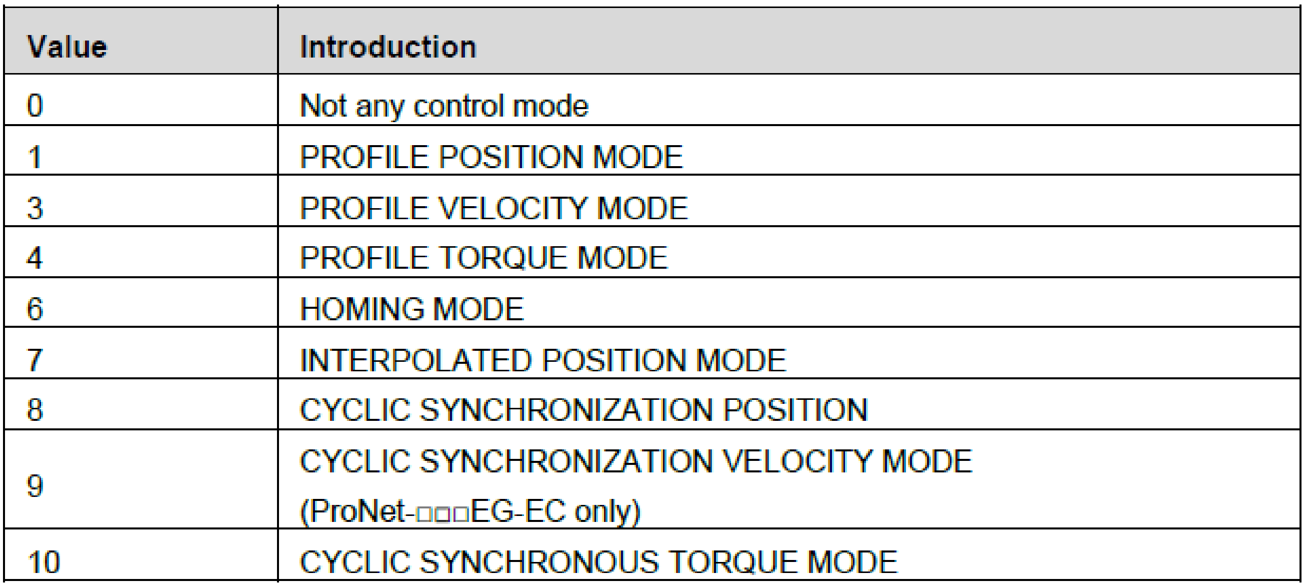
The Ethercat Object for setting mode of operation is 6060h, see below:



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	1,3,4,6,7,8,9,10
Default Value	1



Value	Introduction
0	Not any control mode
1	PROFILE POSITION MODE
3	PROFILE VELOCITY MODE
4	PROFILE TORQUE MODE
6	HOMING MODE
7	INTERPOLATED POSITION MODE
8	CYCLIC SYNCHRONIZATION POSITION
9	CYCLIC SYNCHRONIZATION VELOCITY MODE (ProNet-□□□EG-EC only)
10	CYCLIC SYNCHRONOUS TORQUE MODE

The following must be sent to the gateway from the Profinet-Master:

// Set mode of operation to Homing mode

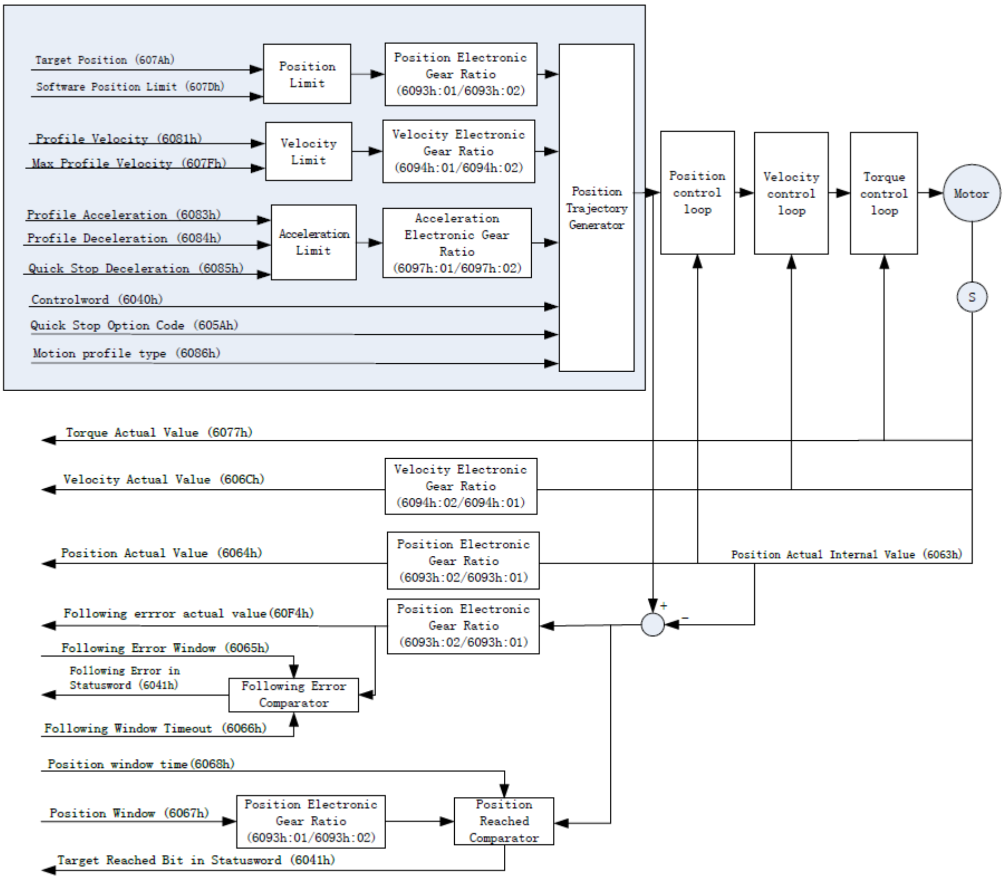
```
#Object.Object_Address[#uiObjectAddressWrite + 1].Node := #iNode; (=1)
#Object.Object_Address[#uiObjectAddressWrite + 1].Object_address := 16#6060;
#Object.Object_Address[#uiObjectAddressWrite + 1].Reserved := 0;
#Object.Object_Address[#uiObjectAddressWrite + 1].Sub := 0;
#Object.Object_Address[#uiObjectAddressWrite + 1].Type := 2;

#Object.Object_Data[#uiObjectAddressWrite + 1] := 6;
```

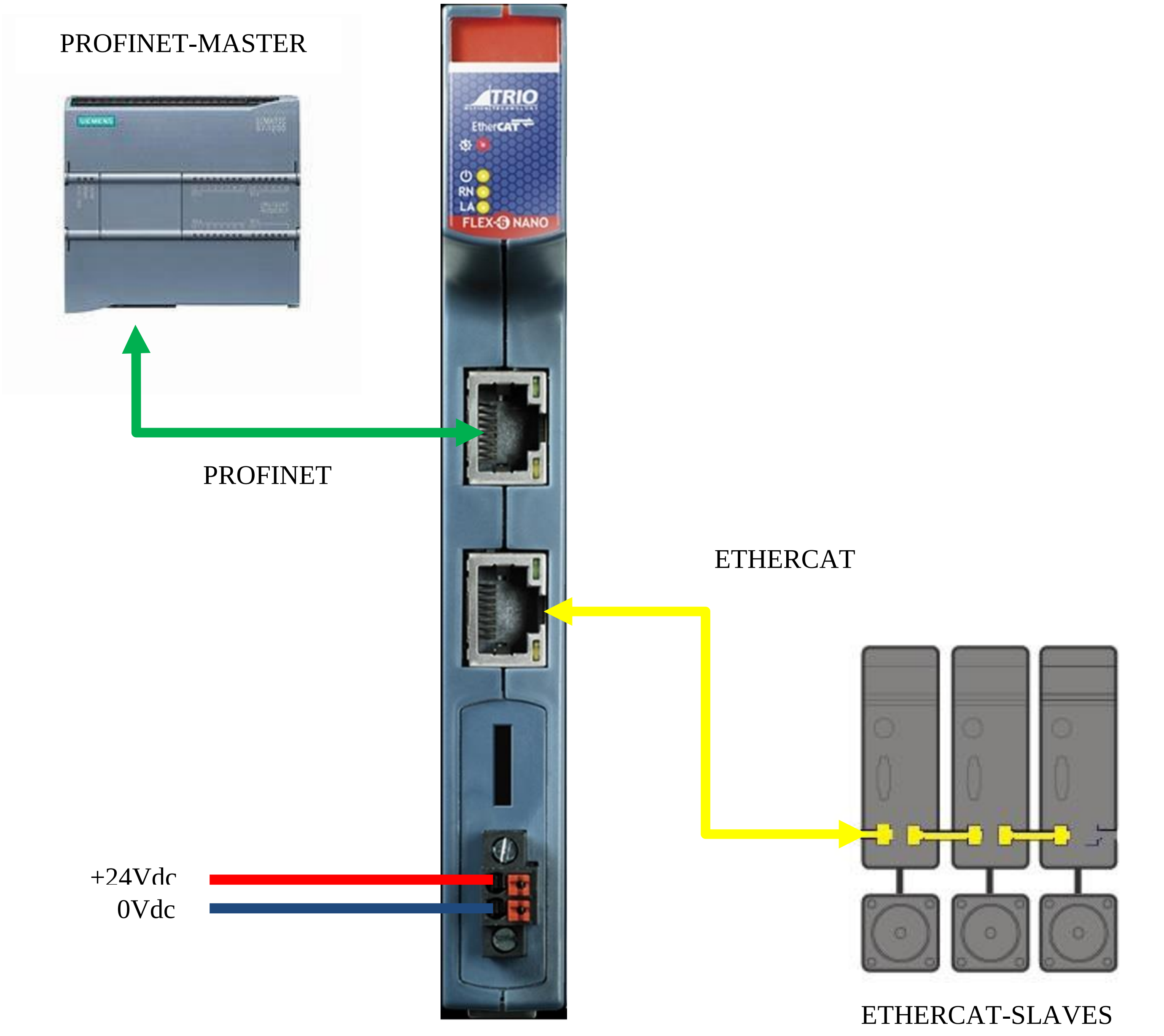
This can be controlled via the channels TABLE (35) and TABLE (36).

TABLE (35) for Node, Object_address, Reserved, Sub and type, TABLE (36) for the value, in this case 6.

The following block diagram shows motor control in Profile Position mode. (In this example this concerns an Estun Pronet-EC drive, for a different brand of drive some objects may be different)


Target Position (607Ah)
Software Position Limit (607Dh)
Position Limit
Position Electronic Gear Ratio (6093h:01/6093h:02)
Profile Velocity (6081h)
Max Profile Velocity (607Fh)
Velocity Limit
Velocity Electronic Gear Ratio (6094h:01/6094h:02)
Profile Acceleration (6083h)
Profile Deceleration (6084h)
Quick Stop Deceleration (6085h)
Acceleration Limit
Acceleration Electronic Gear Ratio (6097h:01/6097h:02)
Controlword (6040h)
Quick Stop Option Code (605Ah)
Motion profile type (6086h)
Position Trajectory Generator
Position control loop
Velocity control loop
Torque control loop
Motor
S
Torque Actual Value (6077h)
Velocity Actual Value (606Ch)
Velocity Electronic Gear Ratio (6094h:02/6094h:01)
Position Actual Value (6064h)
Position Electronic Gear Ratio (6093h:02/6093h:01)
Position Actual Internal Value (6063h)
Following errror actual value(60F4h)
Position Electronic Gear Ratio (6093h:02/6093h:01)
+
-
Following Error Window (6065h)
Following Error in Statusword (6041h)
Following Error Comparator
Following Window Timeout (6066h)
Position window time(6068h)
Position Window (6067h)
Position Electronic Gear Ratio (6093h:01/6093h:02)
Position Reached Comparator
Target Reached Bit in Statusword (6041h)

Power supply, Profinet and Ethercat connection of the Gateway:



Disclaimer
All information provided by or on behalf of ATB Automation BV with regard to its products and services, whether in the form of data, recommendations or otherwise, is assumed to be reliable, but ATB Automation BV does not accept any liability with regard to the application, processing or using such information, products or services, or any consequence thereof.