

# EtherCAT slave porting user manual

## (Preliminary) Version

**DAINCUBE Corp.**

EtherCAT slave porting user manual



**EtherCAT slave porting user manual  
Form 180612A — November, 2018**

DAINCUBE Corp.

Web: [www.daincube.com](http://www.daincube.com)

E-mail: [sales@daincube.com](mailto:sales@daincube.com)

Tel: 82-32-329-9783~4

Fax: 82-32-329-9785

#401-701, Bucheon TechnoPark 4-Danji,  
655 Pyeongcheon-ro, Wonmi-gu, Bucheon-Si,  
Gyeonggi-Do, Republic of Korea

# Preface

## Copyright notice

Copyright © 2005–2018 Daincube. All rights reserved.

Copying of this document, and giving it to others and the use or communication of the Contents thereof, is forbidden without express authority. Offenders are liable to the payment of damages.

All rights are reserved in the event of the grant of a patent or the registration of a utility model or design.

## Important information

This documentation is intended for qualified audience only. The product described herein is not an end user product. It was developed and manufactured for further processing by trained personnel.

## Disclaimer

Although this document has been generated with the utmost care no warranty or liability for correctness or suitability for any particular purpose is implied. The information in this document is provided “as is” and is subject to change without notice.

## Trademarks

All used product names, logos or trademarks are property of their respective owners.

## Product support

DAINCUBE Corp.

Web: [www.daincube.com](http://www.daincube.com)

E - MAIL: [sales@daincube.com](mailto:sales@daincube.com)

## Safety precautions

Be sure to observe all of the following safety precautions.

Strict observance of these warning and caution indications are a MUST for preventing accidents, which could result in bodily injury and substantial property damage. Make sure you fully understand all definitions of these terms and related symbols given below, before you proceed to the manual.

## Safety precautions

The following symbols may be used in this specification:



### **Warning:**

Warnings indicate conditions that, if not observed, can cause personal injury.



### **Caution :**

Cautions warn the user about how to prevent damage to hardware or loss of data.



### **Note:**

Notes call attention to important information that should be observed.

## Revision History

[illegible]

Contents

- 1. Introduction ..... 5
- 2. Method of EtherCAT Slave porting ..... 5
  - 2.1. EtherCAT sample code open ..... 5
  - 2.2. Slave file creation..... 6
  - 2.3. esi file open..... 8
  - 2.4. Create slave file contents ..... 8
- 3. Q&A ..... 12

User's Manual

# 1. Introduction

This document describes how to port a Slave via the EtherCAT sample code.

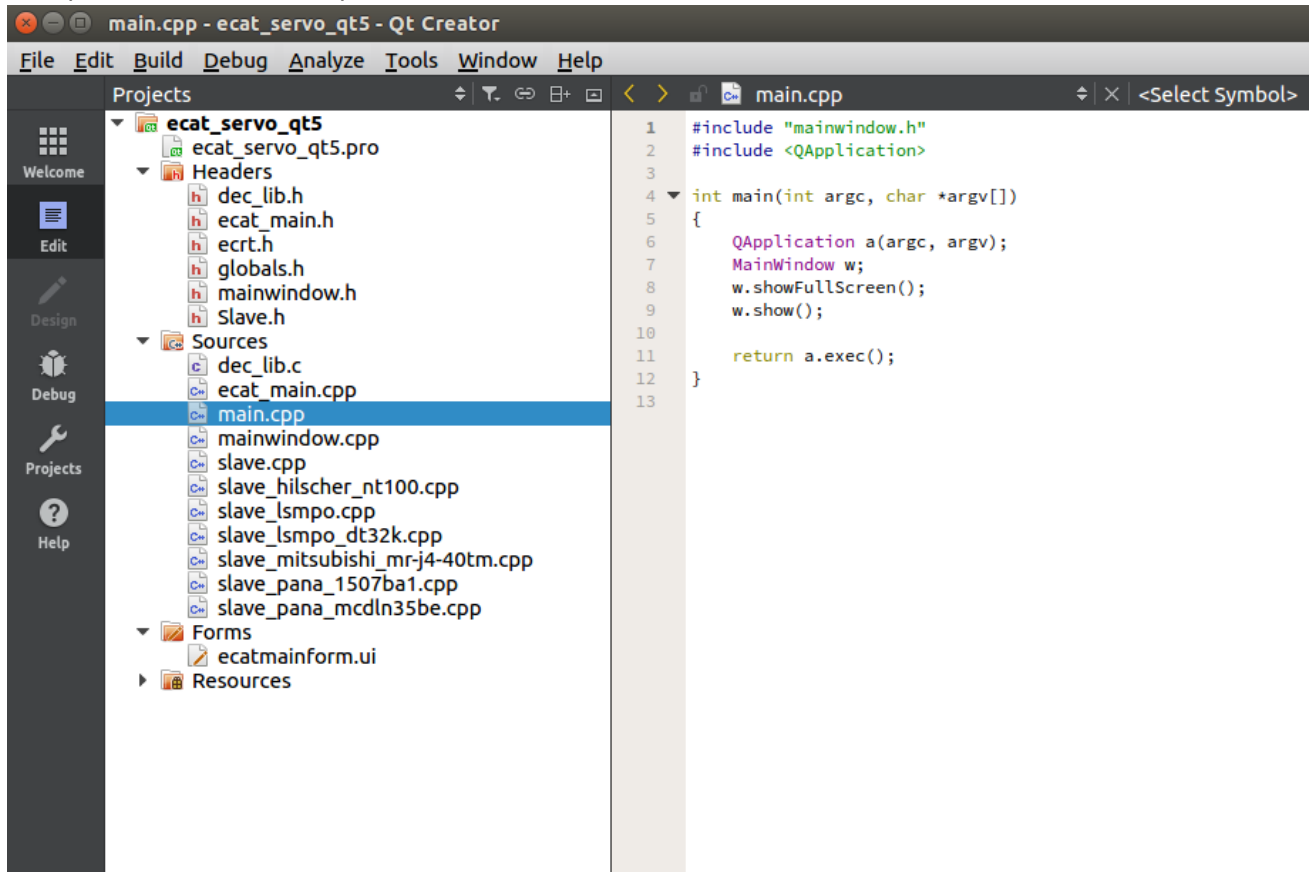
## 2. Method of EtherCAT Slave porting

### 2.1. EtherCAT sample code open

After completing the development environment settings, open the EtherCAT sample code in Qt.

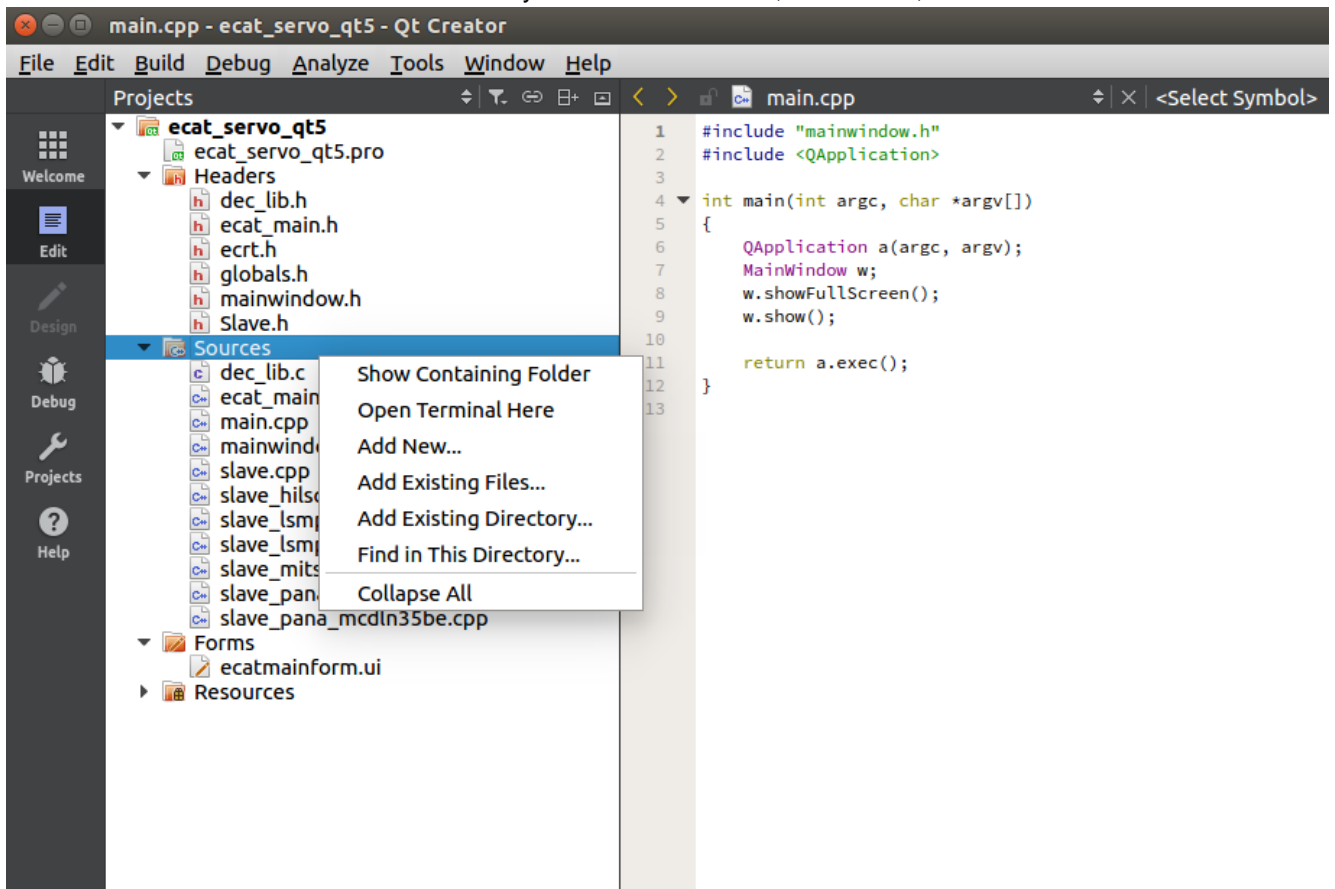
This document describes the UI version, `ecat_servo_qt5_181207.tgz`, and the console version, `ecat_servo_motion_qt_181207.tgz`, has almost the same modification point.

Open the EtherCAT sample code in Qt.

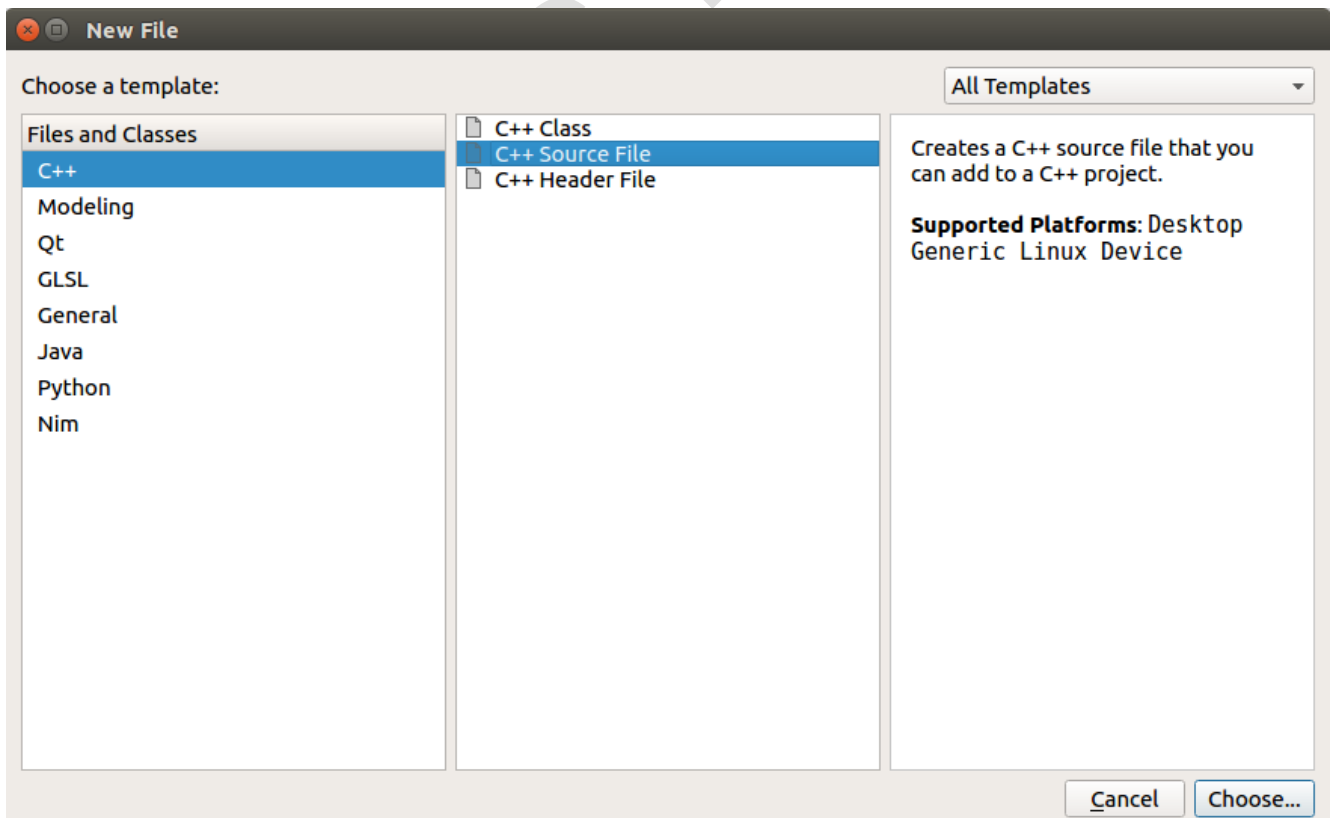


## 2.2. Slave file creation

Enter the information of the servo driver you want to use. First, in Sources, click "Add New".



Select C ++ Source File and click Choose.



Usually, make slave\_manufacturer\_slave name .cpp form. In the example below, the slave\_lsmpp.cpp file is an example.

C++ Source File

Location

Summary

### Location

Name:

Path:

Next >

Cancel

Click Finish to create the file.

C++ Source File

Location

Summary

### Project Management

Add to project:

Add to version control:

Files to be added in

/home/dain/Music/ecat\_servo\_qt5/ecat\_servo\_qt5:

slave\_lsmpo.cpp

< Back

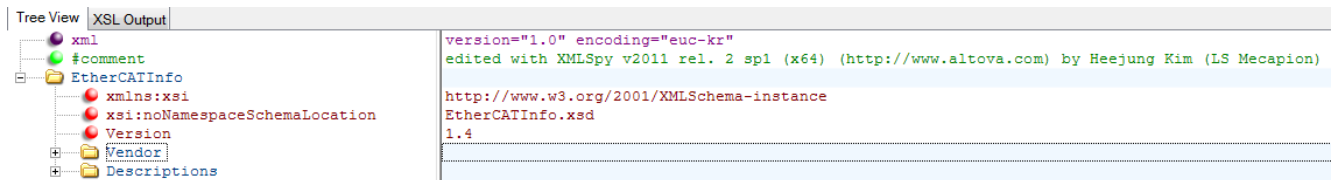
Finish

Cancel

## 2.3. esi file open

Each servo drive has an esi file (EtherCAT Slave Information). The esi file consists of .xml. Therefore, you need to obtain the servo driver's esi file to use in your company. Download the esi file and download and install XML Notepad. This document does not cover how to install XML Notepad.

If you open the servo driver's esi file as XML Notepad, you will see the following screen. The information may differ for each servo driver.



## 2.4. Create slave file contents

Refer to the servo driver's esi file and add the contents to the generated slave source file. This document explains how the contents of the esi file are matched by exemplifying the LS Mecapion's L7NA004B servo driver. Find the Vendor ID and Product ID in the esi file and write it in source code in hexadecimal format.



```
1 #include "ecat_main.h"
2 #include "Slave.h"
3
4 /*
5  * *****
6  * Lsmo setup
7  * *****
8  */
9 #define LSMPO_VENDOR_ID 0x00007595
10 #define LSMPO_PRODUCT_ID 0x00000000
```



In Descriptions -> Devices, find the part where sm is among RxPdo and TxPdo.

+	Sm	
+	Sm	
+	RxPdo	
-	RxPdo	
	Fixed	0
	Sm	2
	Index	#x1601
	Name	RxPdo mapping 2
	Exclude	#x1600
	Exclude	#x1602
	Exclude	#x1603
	Entry	
	Entry	
+	RxPdo	
+	RxPdo	
+	TxPdo	
-	TxPdo	
	Fixed	0
	Sm	3
	Index	#x1A01
	Name	TxPdo mapping 2
	Exclude	#x1A00
	Exclude	#x1A02
	Exclude	#x1A03
	Entry	
	Entry	

When you expand the Entry of RxPdo with sm, there are Control word and Target position. If you expand TxPdo's entry with sm in the same way for TxPdo, there is Status word and Position actual value. Write this part in your source code. At this time, it reflects RxPdo index value, sm value, Index value of entry, Sub index value, BitLen etc.

+	RxPdo	
	Fixed	0
	Sm	2
	Index	#x1601
	Name	RxPdo mapping 2
	Exclude	#x1600
	Exclude	#x1602
	Exclude	#x1603
	Entry	
	Index	#x6040
	SubIndex	0
	BitLen	16
	Name	Controlword
	DataType	UINT
	Entry	
	Index	#x607A
	SubIndex	0
	BitLen	32
	Name	Target Position
	DataType	DINT
+	TxPdo	
	Fixed	0
	Sm	3
	Index	#x1A01
	Name	TxPdo mapping 2
	Exclude	#x1A00
	Exclude	#x1A02
	Exclude	#x1A03
	Entry	
	Index	#x6041
	SubIndex	0
	BitLen	16
	Name	Statusword
	DataType	UINT
	Entry	
	Index	#x6064
	SubIndex	0
	BitLen	32
	Name	Position actual value
	DataType	DINT

```

1 #include "ecat_main.h"
2 #include "Slave.h"
3
4 /*
5 .....
6 Lsmo setup
7 .....
8
9 #define LSMPO_VENDOR_ID    0x00007595
10 #define LSMPO_PRODUCT_ID   0x00000000
11
12 #define lsmo_domain_out_reg_x(x)
13 {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6040, 0, &off_out_control_word[x]},
14 {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x607A, 0, &off_out_target_position[x]}
15
16 #define lsmo_domain_in_reg_x(x)
17 {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6041, 0, &off_in_status_word[x]},
18 {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6064, 0, &off_in_position_act_val[x]}
19
20
21 ec_pdo_entry_reg_t lsmo_out_reg[]={
22     lsmo_domain_out_reg_x(0),
23     lsmo_domain_out_reg_x(1),
24     lsmo_domain_out_reg_x(2),
25     lsmo_domain_out_reg_x(3),
26     lsmo_domain_out_reg_x(4),
27     lsmo_domain_out_reg_x(5)
28 };
29
30 ec_pdo_entry_reg_t lsmo_in_reg[]={
31     lsmo_domain_in_reg_x(0),
32     lsmo_domain_in_reg_x(1),
33     lsmo_domain_in_reg_x(2),
34     lsmo_domain_in_reg_x(3),
35     lsmo_domain_in_reg_x(4),
36     lsmo_domain_in_reg_x(5)
37 };
38
39 ec_pdo_entry_info_t lsmo_pdo_entries[]={
40     {0x6040, 0, 16}, // Control word
41     {0x607A, 0, 32}, // Target position
42     {0x6041, 0, 16}, // Status word
43     {0x6064, 0, 32} // Position actual value
44 };
45
46 ec_pdo_info_t lsmo_pdos[]={
47     {0x1601, 2, &lsmo_pdo_entries[0]},
48     {0x1A01, 2, &lsmo_pdo_entries[2]}
49 };
50

```

When the #define and structure creation are completed, the init function registers it in the g\_vendor\_config structure.

```

108 void lsmo_init (int vendor_inx)
109 {
110
111     g_vendor_config[vendor_inx].slave_domain_out_regs = (ec_pdo_entry_reg_t *)&lsmo_out_reg;
112     g_vendor_config[vendor_inx].slave_domain_in_regs = (ec_pdo_entry_reg_t *)&lsmo_in_reg;
113     g_vendor_config[vendor_inx].slave_pdo_entries = (ec_pdo_entry_info_t *)&lsmo_pdo_entries;
114     g_vendor_config[vendor_inx].slave_pdos = (ec_pdo_info_t *)&lsmo_pdos;
115     g_vendor_config[vendor_inx].slave_syncs = (ec_sync_info_t *)&lsmo_syncs;
116
117     g_vendor_config[vendor_inx].vendor_id = LSMPO_VENDOR_ID;
118     g_vendor_config[vendor_inx].product_id = LSMPO_PRODUCT_ID;
119
120     g_vendor_config[vendor_inx].slave_models = lsmo_slave_models;
121     g_vendor_config[vendor_inx].vendor_name = lsmo_vendor_name;
122
123     g_vendor_config[vendor_inx].is_servo_drive = true;
124
125     g_vendor_config[vendor_inx].Domain_value_init= lsmo_domain_value_init;
126     g_vendor_config[vendor_inx].Slave_send = lsmo_slave_send;
127     g_vendor_config[vendor_inx].Slave_receive = lsmo_slave_rcv;
128
129 }

```

The method of writing / reading Pdo data is as follows. In case of RxPdo data, initialize to 0. In the Send / Receive function, the desired Pdo data is written / read as shown in the example below.

```

58 char lsmo_slave_models[] = "LSMPO Servo";
59 char lsmo_vendor_name[] = "LSMPO";
60
61 void lsmo_domain_value_init(int slv_inx)
62 {
63     EC_WRITE_S16(domain1_pd + off_out_control_word[slv_inx], 0);
64     EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], 0);
65 }
66
67 void lsmo_slave_send(int slv_inx)
68 {
69     long pos_step_val;
70
71     if (master_state.al_states < EC_AL_STATES_OP || domain_state.state.wc_state != EC_WC_COMPLETE)
72     {
73         return ;
74     }
75
76     pos_step_val = run_speed * 200;
77
78     if (req_servo_state[slv_inx] == REQ_SERVO_RUN)
79     {
80         if (cur_step[slv_inx] == step_target_pos_fw)
81         {
82             set_target_val[slv_inx] += pos_step_val;
83             EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], set_target_val[slv_inx]);
84         }
85         else if (cur_step[slv_inx] == step_target_pos_rev)
86         {
87             set_target_val[slv_inx] -= pos_step_val;
88             EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], set_target_val[slv_inx]);
89         }
90     }
91 }
92
93
94 void lsmo_slave_rcv(int slv_inx)
95 {
96     long val;
97
98     val = EC_READ_U16(domain1_pd + off_in_status_word[slv_inx]);
99     if (val != cur_status_word[slv_inx])
100         cur_status_word[slv_inx] = val;
101
102     val = EC_READ_S32(domain1_pd + off_in_position_act_val[slv_inx]);
103     if (is_new_pos(val, cur_position[slv_inx]))
104         cur_position[slv_inx] = val;
105 }
106

```

When setting the value for the slave is completed, add the init function to Slave.h and add it to the void slave\_vendor\_init (void) function of slave.cpp..

```
173 void sanyo_init (int vendor_inx);
174 void omron_init (int vendor_inx);
175 void lsmpo_init (int vendor_inx);
176 void elmo_init (int vendor_inx);
177 void maxon_init (int vendor_inx);
178 void sunin_init (int vendor_inx);
179 void comi_init (int vendor_inx);
180 void higen_init (int vendor_inx);
181 void pana_1505ba1_init (int vendor_inx);
182 void pana_1507ba1_init (int vendor_inx);
183 void pana_2510ba1_init (int vendor_inx);
184 void beckhoff_ek1100_init (int vendor_inx);
185 void beckhoff_el1008_init (int vendor_inx);
186 void beckhoff_el2008_init (int vendor_inx);
187 void beckhoff_el1088_init (int vendor_inx);
188 void beckhoff_el2088_init (int vendor_inx);
189 void beckhoff_el5101_init (int vendor_inx);
190 void rs_csd7_01bn1_init (int vendor_inx);
191 void rs_csd7_04bn1_init (int vendor_inx);
192 void crevis_init (int vendor_inx);
193 void wago_750_354_init (int vendor_inx);
194 void servotronix_init (int vendor_inx);
195 void eraetech_init (int vendor_inx);
196 void mitsubishi_init (int vendor_inx);
197 void lsmpo_lisa_init (int vendor_inx);
198 void daincube_safeio_init (int vendor_inx);
199 void hans_st5s4_init (int vendor_inx);
200 void kollmorgen_akd_p00306_init (int vendor_inx);
201 void omron_gx_id1611_init (int vendor_inx);
202 void omron_gx_od1611_init (int vendor_inx);
203 void hyrobotics_io_init (int vendor_inx);
204 void lsmpo_dt32k_init (int vendor_inx);
205 void hilscher_nt100_init (int vendor_inx);
206 void pana_mcdln35be_init (int vendor_inx);
```

[Slave.h] File

```
67 void Slave_vendor_init (void)
68 {
69     int vendor_inx;
70
71     g_vendor_config[LSMPO_INDEX].Slave_init = lsmpo_init;
72     g_vendor_config[PANA_1507BA1_INDEX].Slave_init = pana_1507ba1_init;
73     //g_vendor_config[LSMPO_DT32K_INDEX].Slave_init = lsmpo_dt32k_init;
74     //g_vendor_config[MITSUBISHI_MRJ440TM_INDEX].Slave_init = mitsubishi_init;
75     //g_vendor_config[HILSCHER_NT100_INDEX].Slave_init = hilscher_nt100_init;
76     //g_vendor_config[PANA_MCDLN35BE_INDEX].Slave_init = pana_mcdln35be_init;
77
78
79     //g_vendor_config[LSMPO_INDEX].Slave_init(LSMPO_INDEX);
80     //g_vendor_config[PANA_1507BA1_INDEX].Slave_init(PANA_1507BA1_INDEX);
81     //g_vendor_config[LSMPO_DT32K_INDEX].Slave_init(LSMPO_DT32K_INDEX);
82
83     for(vendor_inx=0; vendor_inx<MAX_VENDOR; vendor_inx++)
84         g_vendor_config[vendor_inx].Slave_init(vendor_inx);
85
86 }
```

[slave.cpp] File



### 3. Q&A

#### ➤ Question )

What should I do if my esi file RxPdo, TxPdo has a different entry besides Control word, Target position, Status word, Position Actual value?

#### ➤ Answer )

Add a function to slave.cpp, and extern it to Slave.h.

```
10 unsigned int off_out_control_word[MAX_SLAVES];
11 unsigned int off_out_control_DI1[MAX_SLAVES];
12 unsigned int off_out_control_DI2[MAX_SLAVES];
13 unsigned int off_out_control_DI3[MAX_SLAVES];
14 unsigned int off_out_modes_of_operation_word[MAX_SLAVES];
15 unsigned int off_out_target_position[MAX_SLAVES];
16 unsigned int off_out_position_offset[MAX_SLAVES];
17 unsigned int off_out_profile_velocity[MAX_SLAVES];
18 unsigned int off_out_velocity_offset[MAX_SLAVES];
19 unsigned int off_out_profile_acceleration[MAX_SLAVES];
20 unsigned int off_out_profile_deceleration[MAX_SLAVES];
21 unsigned int off_out_target_velocity[MAX_SLAVES];
22 unsigned int off_out_velocity_limit_value[MAX_SLAVES];
23 unsigned int off_out_target_torque[MAX_SLAVES];
24 unsigned int off_out_max_profile_velocity[MAX_SLAVES];
25 unsigned int off_out_positive_torque_limit_value[MAX_SLAVES];
26 unsigned int off_out_negative_torque_limit_value[MAX_SLAVES];
27 unsigned int off_out_torque_offset[MAX_SLAVES];
28 unsigned int off_out_touch_probe_func[MAX_SLAVES];
29 unsigned int off_out_digital_output[MAX_SLAVES];
30 unsigned int off_out_led_control[MAX_SLAVES];
31 unsigned int off_out_padding[MAX_SLAVES][8];
32 unsigned int off_out_bit_position[MAX_SLAVES][20]={0,};
33 unsigned int off_out_control[MAX_SLAVES];
34 unsigned int off_out_value[MAX_SLAVES][20]={0,};
35
36 unsigned int off_in_status_word[MAX_SLAVES];
37 unsigned int off_in_status_D01[MAX_SLAVES];
38 unsigned int off_in_status_D02[MAX_SLAVES];
39 unsigned int off_in_status_D03[MAX_SLAVES];
40 unsigned int off_in_position_act_val[MAX_SLAVES];
41 unsigned int off_in_error_act_val[MAX_SLAVES];
42 unsigned int off_in_velocity_act_val[MAX_SLAVES];
43 unsigned int off_in_operation_mode[MAX_SLAVES];
44 unsigned int off_in_error_code[MAX_SLAVES];
45 unsigned int off_in_torque_act_val[MAX_SLAVES];
46 unsigned int off_in_touch_probe_status[MAX_SLAVES];
47 unsigned int off_in_touch_probe_pos1_pval[MAX_SLAVES];
48 unsigned int off_in_touch_probe_pos1_nval[MAX_SLAVES];
49 unsigned int off_in_touch_probe_pos2_pval[MAX_SLAVES];
50 unsigned int off_in_touch_probe_pos2_nval[MAX_SLAVES];
51 unsigned int off_in_digital_input[MAX_SLAVES];
52 unsigned int off_in_torque_value1[MAX_SLAVES];
53 unsigned int off_in_torque_value2[MAX_SLAVES];
54 unsigned int off_in_padding[MAX_SLAVES][8];
55 unsigned int off_in_bit_position[MAX_SLAVES][20]={0,};
56 unsigned int off_in_status[MAX_SLAVES];
57 unsigned int off_in_value[MAX_SLAVES][20]={0,};
58 unsigned int off_in_latch[MAX_SLAVES];
```

```
89 extern unsigned int off_out_control_word[MAX_SLAVES];
90 extern unsigned int off_out_control_DI1[MAX_SLAVES];
91 extern unsigned int off_out_control_DI2[MAX_SLAVES];
92 extern unsigned int off_out_control_DI3[MAX_SLAVES];
93 extern unsigned int off_out_modes_of_operation_word[MAX_SLAVES];
94 extern unsigned int off_out_target_position[MAX_SLAVES];
95 extern unsigned int off_out_position_offset[MAX_SLAVES];
96 extern unsigned int off_out_profile_velocity[MAX_SLAVES];
97 extern unsigned int off_out_velocity_offset[MAX_SLAVES];
98 extern unsigned int off_out_profile_acceleration[MAX_SLAVES];
99 extern unsigned int off_out_profile_deceleration[MAX_SLAVES];
100 extern unsigned int off_out_target_velocity[MAX_SLAVES];
101 extern unsigned int off_out_velocity_limit_value[MAX_SLAVES];
102 extern unsigned int off_out_target_torque[MAX_SLAVES];
103 extern unsigned int off_out_max_profile_velocity[MAX_SLAVES];
104 extern unsigned int off_out_positive_torque_limit_value[MAX_SLAVES];
105 extern unsigned int off_out_negative_torque_limit_value[MAX_SLAVES];
106 extern unsigned int off_out_torque_offset[MAX_SLAVES];
107 extern unsigned int off_out_touch_probe_func[MAX_SLAVES];
108 extern unsigned int off_out_digital_output[MAX_SLAVES];
109 extern unsigned int off_out_led_control[MAX_SLAVES];
110 extern unsigned int off_out_padding[MAX_SLAVES][8];
111 extern unsigned int off_out_bit_position[MAX_SLAVES][20];
112 extern unsigned int off_out_control[MAX_SLAVES];
113 extern unsigned int off_out_value[MAX_SLAVES][20];
114
115 extern unsigned int off_in_status_word[MAX_SLAVES];
116 extern unsigned int off_in_status_DO1[MAX_SLAVES];
117 extern unsigned int off_in_status_DO2[MAX_SLAVES];
118 extern unsigned int off_in_status_DO3[MAX_SLAVES];
119 extern unsigned int off_in_position_act_val[MAX_SLAVES];
120 extern unsigned int off_in_error_act_val[MAX_SLAVES];
121 extern unsigned int off_in_velocity_act_val[MAX_SLAVES];
122 extern unsigned int off_in_operation_mode[MAX_SLAVES];
123 extern unsigned int off_in_error_code[MAX_SLAVES];
124 extern unsigned int off_in_torque_act_val[MAX_SLAVES];
125 extern unsigned int off_in_touch_probe_status[MAX_SLAVES];
126 extern unsigned int off_in_touch_probe_pos1_pval[MAX_SLAVES];
127 extern unsigned int off_in_touch_probe_pos1_nval[MAX_SLAVES];
128 extern unsigned int off_in_touch_probe_pos2_pval[MAX_SLAVES];
129 extern unsigned int off_in_touch_probe_pos2_nval[MAX_SLAVES];
130 extern unsigned int off_in_digital_input[MAX_SLAVES];
131 extern unsigned int off_in_torque_value1[MAX_SLAVES];
132 extern unsigned int off_in_torque_value2[MAX_SLAVES];
133 extern unsigned int off_in_padding[MAX_SLAVES][8];
134 extern unsigned int off_in_bit_position[MAX_SLAVES][20];
135 extern unsigned int off_in_status[MAX_SLAVES];
136 extern unsigned int off_in_value[MAX_SLAVES][20];
137 extern unsigned int off_in_latch[MAX_SLAVES];
```