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
E-mail: 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

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
E - MAIL: 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

Revision	Data	Comment
Version 0.1	2018.12.06	Preliminary Version

Contents

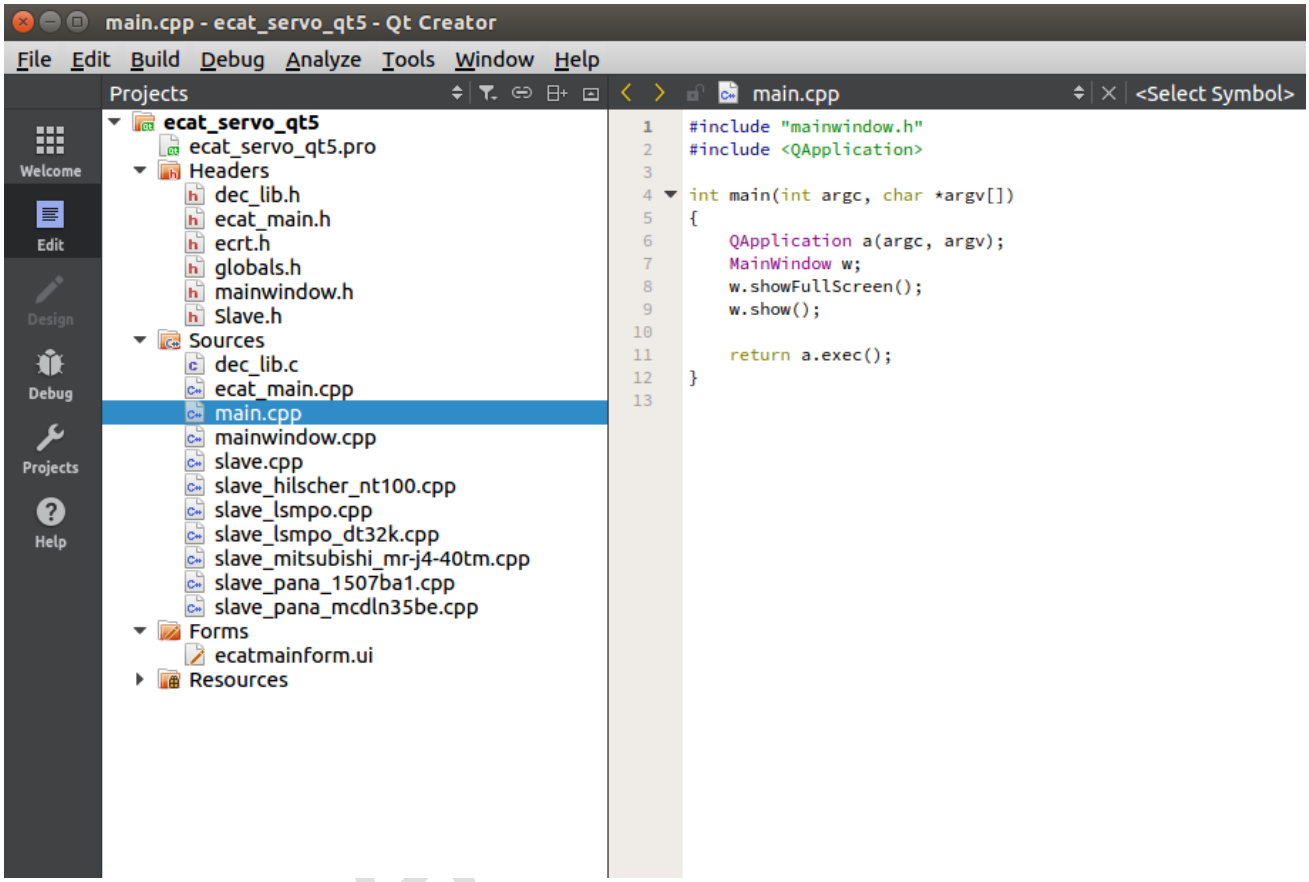
1. Introduction

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

2. Method of EtherCAT Slave porting

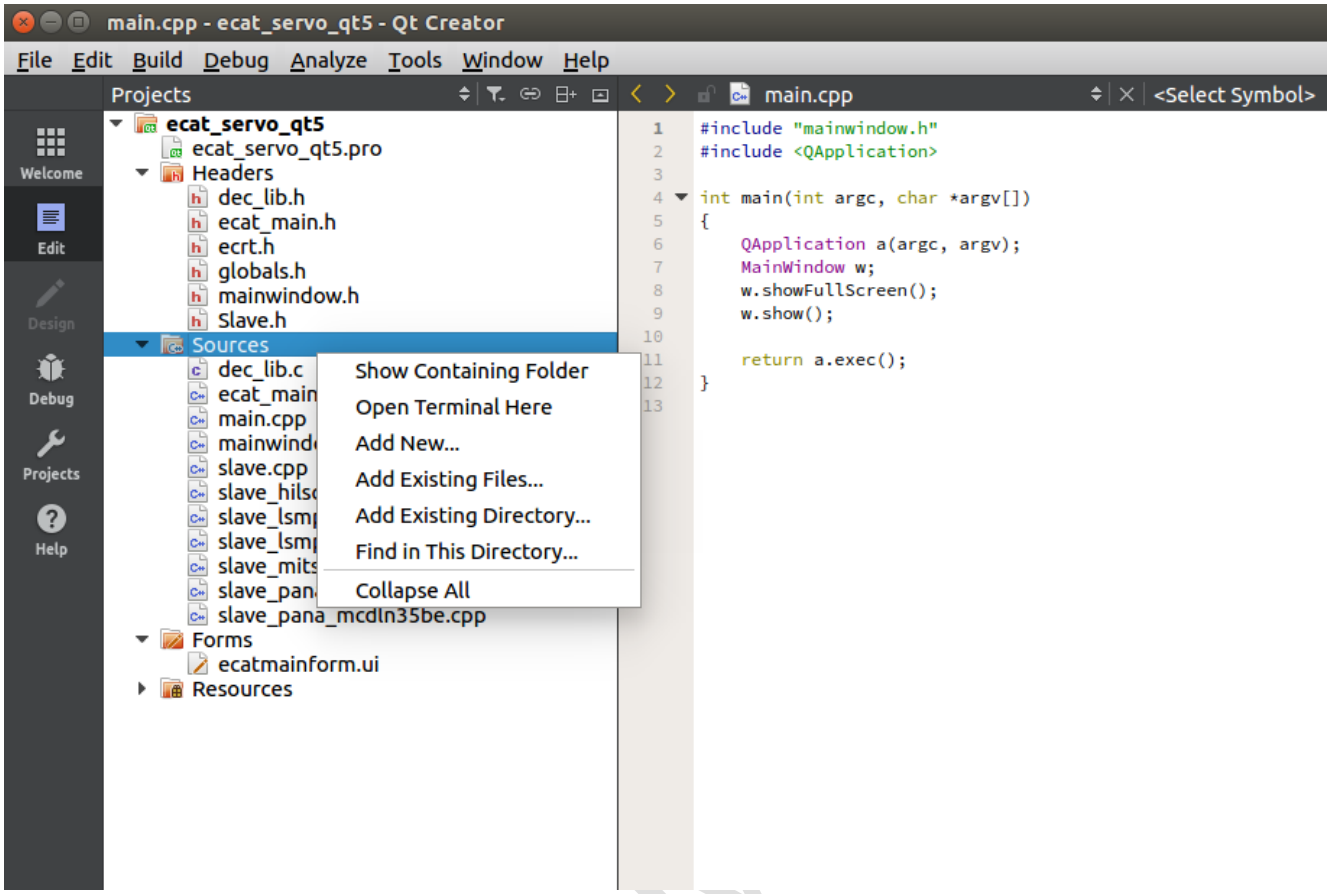
2.1. EtherCAT sample code open

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.

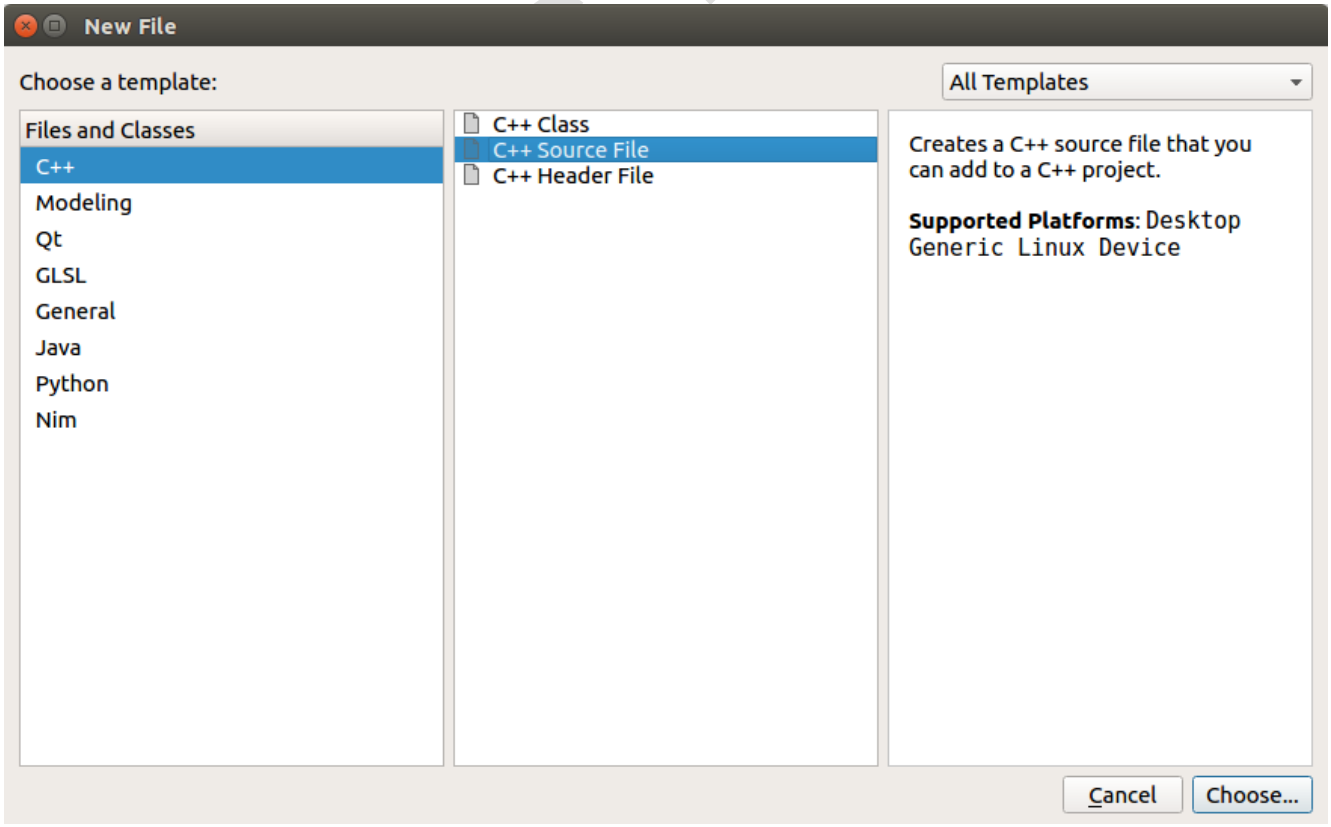


2.2. Slave file creation

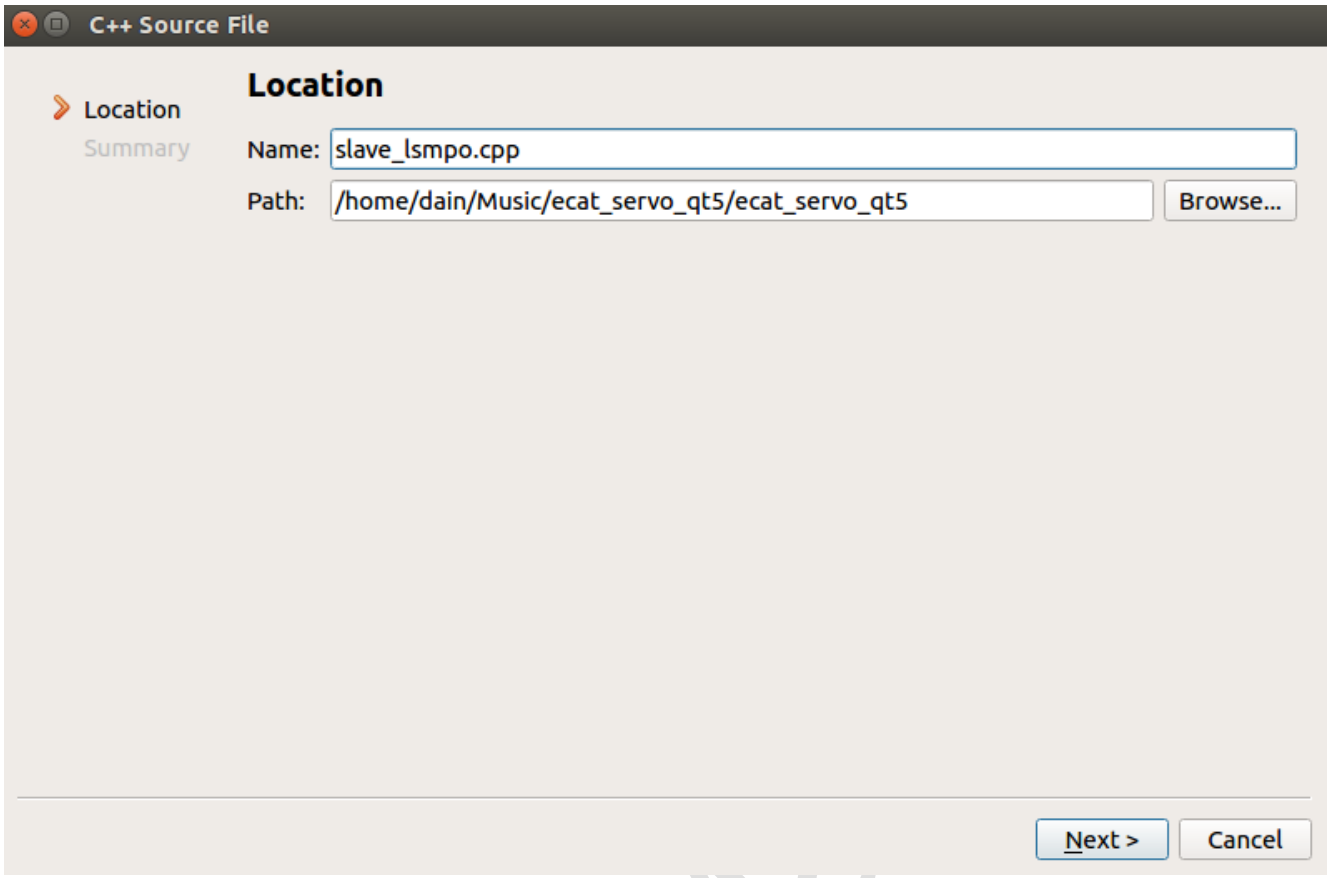
Enter the information of the servo driver you want to use. First, in Sources, click "Add New".



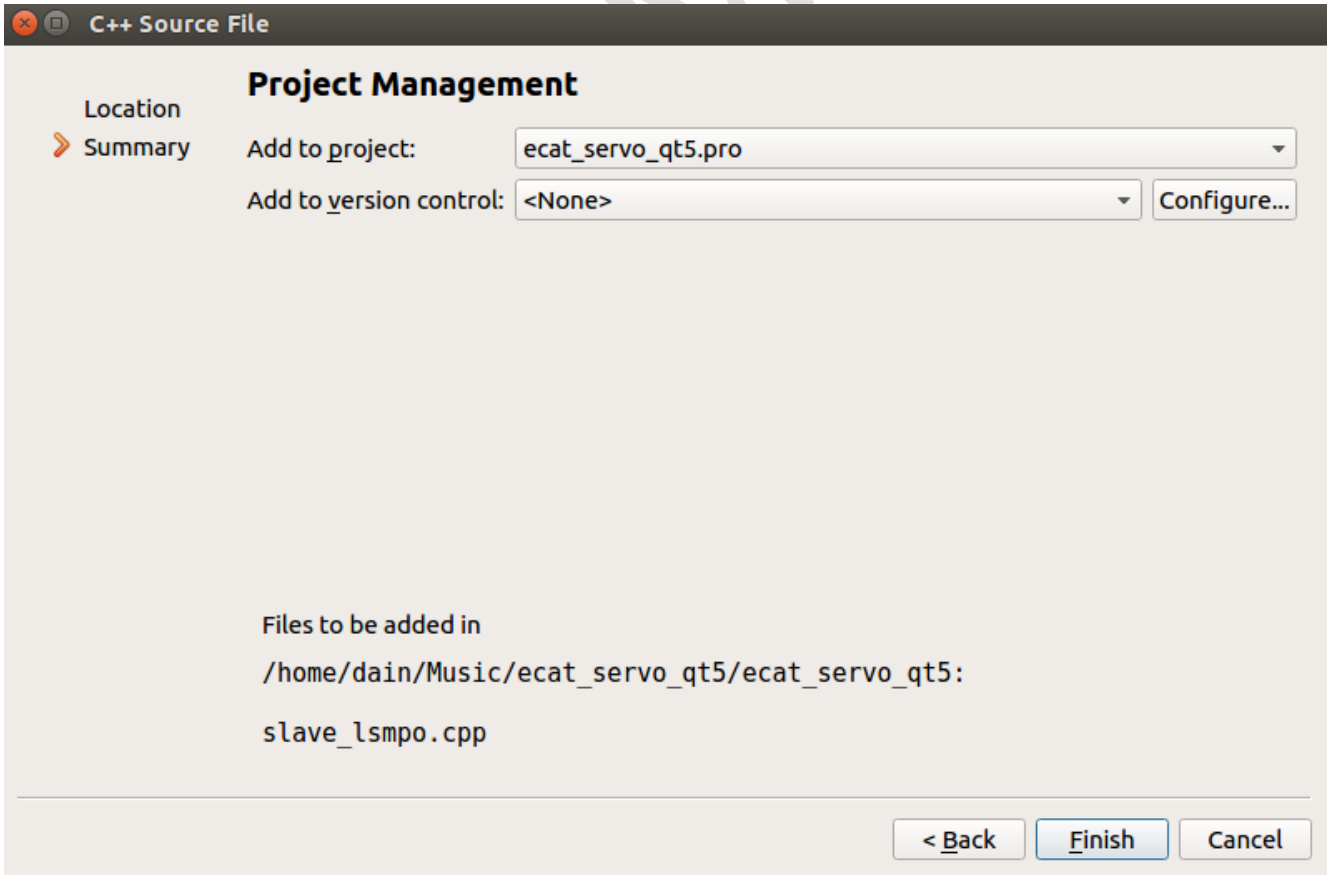
Select C ++ Source File and click Choose.



Usually, make slave_manufacturer_slave name .cpp form. In the example below, the slave_lsmpo.cpp file is an example.



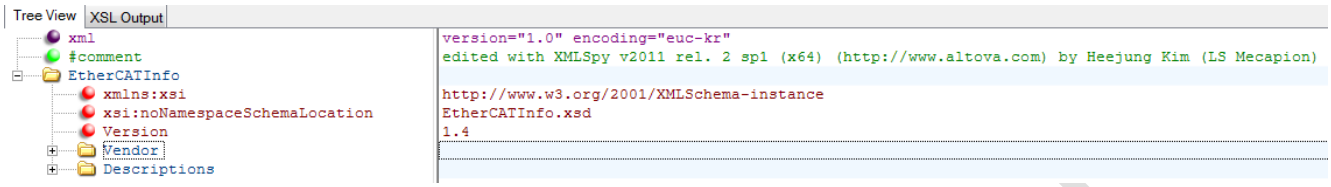
Click Finish to create the file.



2.3. esi file open

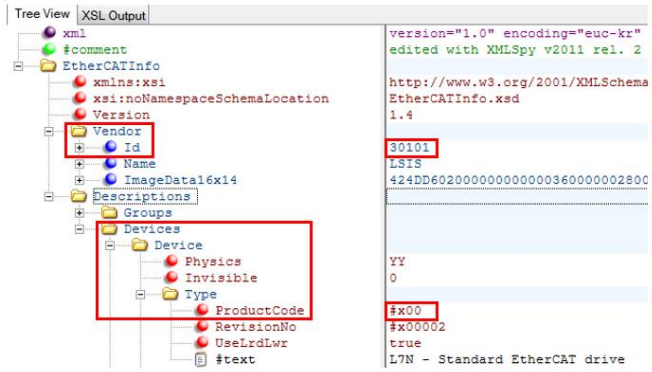
Each servo drive has an esi file (EtherCAT Slave Information). The esi file consists of .xml. Therefore, you need to obtain the servo driver's esi file to use in your company. Download the esi file and download and install XML Notepad. This document does not cover how to install XML Notepad.

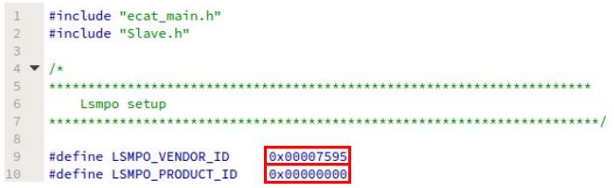
If you open the servo driver's esi file as XML Notepad, you will see the following screen. The information may differ for each servo driver.



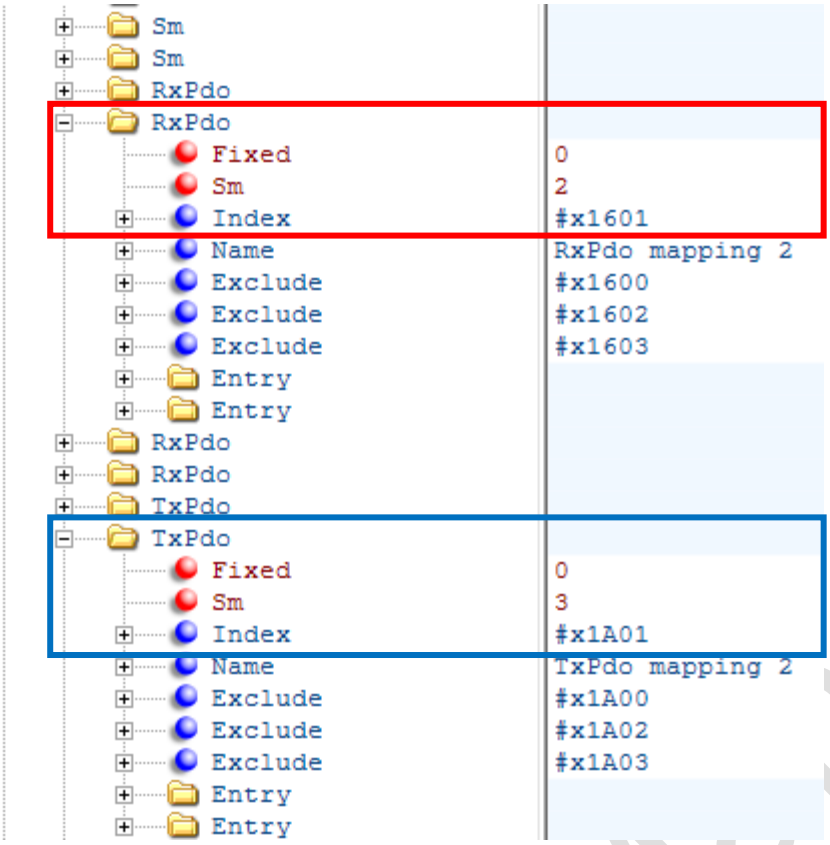
2.4. Create slave file contents

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.

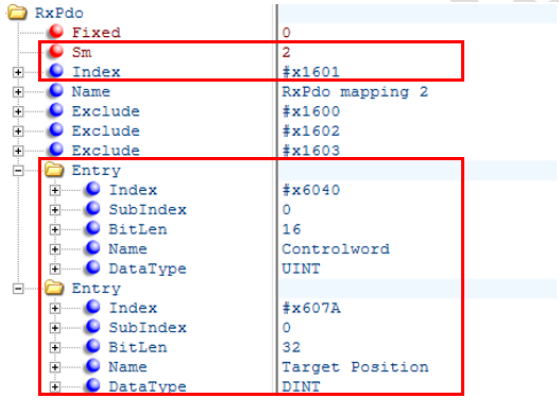


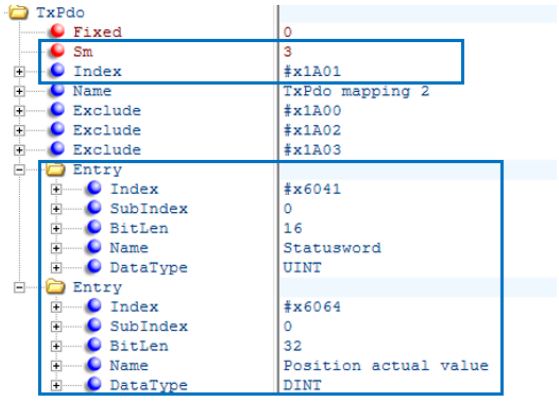


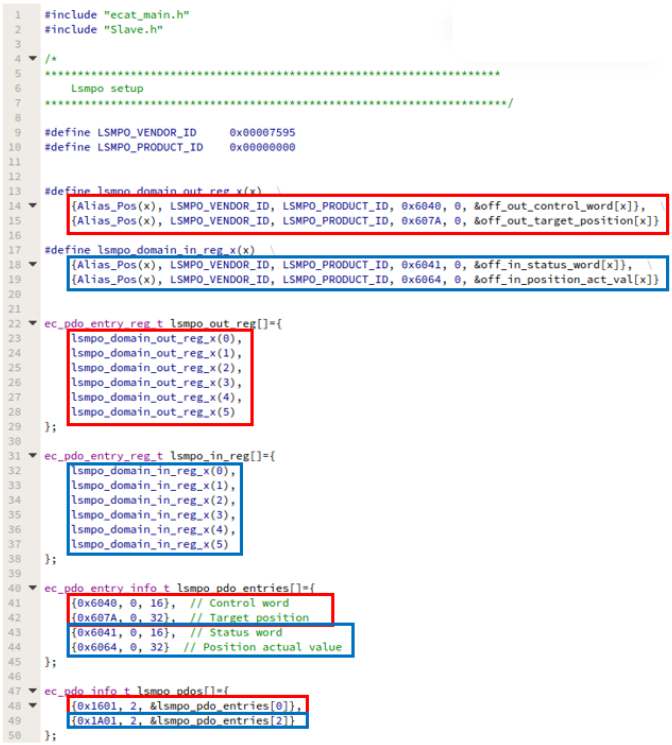
In Descriptions -> Devices, find the part where sm is among RxPdo and TxPdo.



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.







```
#include "ecat_main.h"
#include "Slave.h"

/*
*****************************************************************************
    Lsmpo setup
*****************************************************************************/

#define LSMPO_VENDOR_ID     0x00007595
#define LSMPO_PRODUCT_ID    0x00000000


#define lsmpo_domain_out_reg_x(x) \
    {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6040, 0, &off_out_control_word[x]}, \
    {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x607A, 0, &off_out_target_position[x]}

#define lsmpo_domain_in_reg_x(x) \
    {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6041, 0, &off_in_status_word[x]}, \
    {Alias_Pos(x), LSMPO_VENDOR_ID, LSMPO_PRODUCT_ID, 0x6064, 0, &off_in_position_act_val[x]}


ec_pdo_entry_reg_t lsmpo_out_reg[]={
    lsmpo_domain_out_reg_x(0),
    lsmpo_domain_out_reg_x(1),
    lsmpo_domain_out_reg_x(2),
    lsmpo_domain_out_reg_x(3),
    lsmpo_domain_out_reg_x(4),
    lsmpo_domain_out_reg_x(5)
};

ec_pdo_entry_reg_t lsmpo_in_reg[]={
    lsmpo_domain_in_reg_x(0),
    lsmpo_domain_in_reg_x(1),
    lsmpo_domain_in_reg_x(2),
    lsmpo_domain_in_reg_x(3),
    lsmpo_domain_in_reg_x(4),
    lsmpo_domain_in_reg_x(5)
};

ec_pdo_entry_info_t lsmpo_pdo_entries[]={
    {0x6040, 0, 16},  // Control word
    {0x607A, 0, 32},  // Target position
    {0x6041, 0, 16},  // Status word
    {0x6064, 0, 32}  // Position actual value
};

ec_pdo_info_t lsmpo_pdos[]={
    {0x1601, 2, &lsmpo_pdo_entries[0]},
    {0x1A01, 2, &lsmpo_pdo_entries[2]}
};
```

When the #define and structure creation are completed, the init function registers it in the g_vendor_config structure.

```
void lsmpo_init (int vendor_inx)
{

    g_vendor_config[vendor_inx].slave_domain_out_regs = (ec_pdo_entry_reg_t *)&lsmpo_out_reg;
    g_vendor_config[vendor_inx].slave_domain_in_regs = (ec_pdo_entry_reg_t *)&lsmpo_in_reg;
    g_vendor_config[vendor_inx].slave_pdo_entries = (ec_pdo_entry_info_t *)&lsmpo_pdo_entries;
    g_vendor_config[vendor_inx].slave_pdos = (ec_pdo_info_t *)&lsmpo_pdos;
    g_vendor_config[vendor_inx].slave_syncs = (ec_sync_info_t *)&lsmpo_syncs;

    g_vendor_config[vendor_inx].vendor_id = LSMPO_VENDOR_ID;
    g_vendor_config[vendor_inx].product_id = LSMPO_PRODUCT_ID;

    g_vendor_config[vendor_inx].slave_models = lsmpo_slave_models;
    g_vendor_config[vendor_inx].vendor_name = lsmpo_vendor_name;

    g_vendor_config[vendor_inx].is_servo_drive = true;

    g_vendor_config[vendor_inx].Domain_value_init= lsmpo_domain_value_init;
    g_vendor_config[vendor_inx].Slave_send = lsmpo_slave_send;
    g_vendor_config[vendor_inx].Slave_receive = lsmpo_slave_recv;

}
```

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.

```
char lsmpo_slave_models[] = "LSMPO Servo";
char lsmpo_vendor_name[] = "LSMPO";

void lsmpo_domain_value_init(int slv_inx)
{
    EC_WRITE_S16(domain1_pd + off_out_control_word[slv_inx], 0);
    EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], 0);
}

void  lsmpo_slave_send(int slv_inx)
{
    long pos_step_val;

    if (master_state.al_states < EC_AL_STATES_OP || domain_state.state.wc_state != EC_WC_COMPLETE)
    {
        return ;
    }

    pos_step_val = run_speed * 200;

    if (req_servo_state[slv_inx] == REQ_SERVO_RUN)
    {
        if (cur_step[slv_inx]  == step_target_pos_fw)
        {
            set_target_val[slv_inx] += pos_step_val;
            EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], set_target_val[slv_inx]);
        }
        else if (cur_step[slv_inx]  == step_target_pos_rev)
        {
            set_target_val[slv_inx] -= pos_step_val;
            EC_WRITE_S32(domain1_pd + off_out_target_position[slv_inx], set_target_val[slv_inx]);
        }
    }
}


void lsmpo_slave_recv(int slv_inx)
{
    long val;

    val = EC_READ_U16(domain1_pd + off_in_status_word[slv_inx]);
    if (val != cur_status_word[slv_inx])
        cur_status_word[slv_inx] = val;

    val = EC_READ_S32(domain1_pd + off_in_position_act_val[slv_inx]);
    if (is_new_pos(val, cur_position[slv_inx]))
        cur_position[slv_inx] = val;
 }

```

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.

```
unsigned int off_out_control_word[MAX_SLAVES];
unsigned int off_out_control_DI1[MAX_SLAVES];
unsigned int off_out_control_DI2[MAX_SLAVES];
unsigned int off_out_control_DI3[MAX_SLAVES];
unsigned int off_out_modes_of_operation_word[MAX_SLAVES];
unsigned int off_out_target_position[MAX_SLAVES];
unsigned int off_out_position_offset[MAX_SLAVES];
unsigned int off_out_profile_velocity[MAX_SLAVES];
unsigned int off_out_velocity_offset[MAX_SLAVES];
unsigned int off_out_profile_acceleration[MAX_SLAVES];
unsigned int off_out_profile_deceleration[MAX_SLAVES];
unsigned int off_out_target_velocity[MAX_SLAVES];
unsigned int off_out_velocity_limit_value[MAX_SLAVES];
unsigned int off_out_target_torque[MAX_SLAVES];
unsigned int off_out_max_profile_velocity[MAX_SLAVES];
unsigned int off_out_positive_torque_limit_value[MAX_SLAVES];
unsigned int off_out_negative_torque_limit_value[MAX_SLAVES];
unsigned int off_out_torque_offset[MAX_SLAVES];
unsigned int off_out_touch_probe_func[MAX_SLAVES];
unsigned int off_out_digital_output[MAX_SLAVES];
unsigned int off_out_led_control[MAX_SLAVES];
unsigned int off_out_padding[MAX_SLAVES][8];
unsigned int off_out_bit_position[MAX_SLAVES][20]={0,};
unsigned int off_out_control[MAX_SLAVES];
unsigned int off_out_value[MAX_SLAVES][20]={0,};

unsigned int off_in_status_word[MAX_SLAVES];
unsigned int off_in_status_DO1[MAX_SLAVES];
unsigned int off_in_status_DO2[MAX_SLAVES];
unsigned int off_in_status_DO3[MAX_SLAVES];
unsigned int off_in_position_act_val[MAX_SLAVES];
unsigned int off_in_error_act_val[MAX_SLAVES];
unsigned int off_in_velocity_act_val[MAX_SLAVES];
unsigned int off_in_operation_mode[MAX_SLAVES];
unsigned int off_in_error_code[MAX_SLAVES];
unsigned int off_in_torque_act_val[MAX_SLAVES];
unsigned int off_in_touch_probe_status[MAX_SLAVES];
unsigned int off_in_touch_probe_pos1_pval[MAX_SLAVES];
unsigned int off_in_touch_probe_pos1_nval[MAX_SLAVES];
unsigned int off_in_touch_probe_pos2_pval[MAX_SLAVES];
unsigned int off_in_touch_probe_pos2_nval[MAX_SLAVES];
unsigned int off_in_digital_input[MAX_SLAVES];
unsigned int off_in_torque_value1[MAX_SLAVES];
unsigned int off_in_torque_value2[MAX_SLAVES];
unsigned int off_in_padding[MAX_SLAVES][8];
unsigned int off_in_bit_position[MAX_SLAVES][20]={0,};
unsigned int off_in_status[MAX_SLAVES];
unsigned int off_in_value[MAX_SLAVES][20]={0,};
unsigned int off_in_latch[MAX_SLAVES];
```

```
extern unsigned int off_out_control_word[MAX_SLAVES];
extern unsigned int off_out_control_DI1[MAX_SLAVES];
extern unsigned int off_out_control_DI2[MAX_SLAVES];
extern unsigned int off_out_control_DI3[MAX_SLAVES];
extern unsigned int off_out_modes_of_operation_word[MAX_SLAVES];
extern unsigned int off_out_target_position[MAX_SLAVES];
extern unsigned int off_out_position_offset[MAX_SLAVES];
extern unsigned int off_out_profile_velocity[MAX_SLAVES];
extern unsigned int off_out_velocity_offset[MAX_SLAVES];
extern unsigned int off_out_profile_acceleration[MAX_SLAVES];
extern unsigned int off_out_profile_deceleration[MAX_SLAVES];
extern unsigned int off_out_target_velocity[MAX_SLAVES];
extern unsigned int off_out_velocity_limit_value[MAX_SLAVES];
extern unsigned int off_out_target_torque[MAX_SLAVES];
extern unsigned int off_out_max_profile_velocity[MAX_SLAVES];
extern unsigned int off_out_positive_torque_limit_value[MAX_SLAVES];
extern unsigned int off_out_negative_torque_limit_value[MAX_SLAVES];
extern unsigned int off_out_torque_offset[MAX_SLAVES];
extern unsigned int off_out_touch_probe_func[MAX_SLAVES];
extern unsigned int off_out_digital_output[MAX_SLAVES];
extern unsigned int off_out_led_control[MAX_SLAVES];
extern unsigned int off_out_padding[MAX_SLAVES][8];
extern unsigned int off_out_bit_position[MAX_SLAVES][20];
extern unsigned int off_out_control[MAX_SLAVES];
extern unsigned int off_out_value[MAX_SLAVES][20];

extern unsigned int off_in_status_word[MAX_SLAVES];
extern unsigned int off_in_status_DO1[MAX_SLAVES];
extern unsigned int off_in_status_DO2[MAX_SLAVES];
extern unsigned int off_in_status_DO3[MAX_SLAVES];
extern unsigned int off_in_position_act_val[MAX_SLAVES];
extern unsigned int off_in_error_act_val[MAX_SLAVES];
extern unsigned int off_in_velocity_act_val[MAX_SLAVES];
extern unsigned int off_in_operation_mode[MAX_SLAVES];
extern unsigned int off_in_error_code[MAX_SLAVES];
extern unsigned int off_in_torque_act_val[MAX_SLAVES];
extern unsigned int off_in_touch_probe_status[MAX_SLAVES];
extern unsigned int off_in_touch_probe_pos1_pval[MAX_SLAVES];
extern unsigned int off_in_touch_probe_pos1_nval[MAX_SLAVES];
extern unsigned int off_in_touch_probe_pos2_pval[MAX_SLAVES];
extern unsigned int off_in_touch_probe_pos2_nval[MAX_SLAVES];
extern unsigned int off_in_digital_input[MAX_SLAVES];
extern unsigned int off_in_torque_value1[MAX_SLAVES];
extern unsigned int off_in_torque_value2[MAX_SLAVES];
extern unsigned int off_in_padding[MAX_SLAVES][8];
extern unsigned int off_in_bit_position[MAX_SLAVES][20];
extern unsigned int off_in_status[MAX_SLAVES];
extern unsigned int off_in_value[MAX_SLAVES][20];
extern unsigned int off_in_latch[MAX_SLAVES];
```