AX58400 Motor Control User Guide

Revision 2.20
December 20th, 2024

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Revision History

Revision	Date	Description
1.00	2021/06/28	Initial release
1.10	2021/08/17	Modify AX58400-TSB-1 and encoder power source
1.20	2023/10/17	1. Added Evaluation Board License Agreement. 2. Added another power USB connectivity and update related figures and tables.
2.00	2024/04/10	1. Modified the object 0x6098 and 0x609a access from “RO” to “RW” in table 9-3. 2. Removed the object 0xF030 and 0xF050 in table 9-3. 3. Moved chapters 6 and 7 to the AX58x00_TwinCAT_UserGuide document. 4. Added a new chapter that describes how to set the external clock.
2.10	2024/11/28	1. Added MotorControl Workbench & STM32CubeMX required in Chapter 2. 2. Added PID gain objects in 0x2000~0x3FFF area in Section 9-1. 3. Added AX58400 pinout patch description in Section 4-2. 4. Added AX58400 MC Workbench patch description in Section 4-3. 5. Added Section 12 - Firmware Architecture. 6. Added Section 13 - Tuning the PID gain values with TwinCAT. 7. Added Section 14 - Building the Reference Design Begin with MotorControl Workbench.
2.20	2024/12/20	1. Adjust Reference Design building flow in Chapter 5. 2. Adjust Chapter 4 & 6 description. 3. Added disabling “**Debug Description**” in Section 6-2.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

CONTENT

1. Introduction 8

2. Requirements 9

3. Board Description 14

3-1 AX58400-TSB 14
3-2 X-NUCLEO-IHM08M1 15

4. Environment Setup 16

4-1 Hardware Connectivity 16
4-2 AX58400 Pinout Patch for STM32CubeMX 17
4-3 STM MotorControl Workbench Patch 18

5. Building Reference Design Begin with MC Workbench 19

5-1 Performs the MC Workbench Patch for AX58400 19
5-2 Create New Project and Generate Source Code in MC Workbench 19
5-3 Modify Peripheral Settings in STM32CubeMX 21
5-4 Generate Source Code in STM32CubeMX 26
5-5 Source Code Adjustment 28
5-6 Generate EtherCAT Slave Stack Code 33
5-7 IDE Setting Adjustment 34

6. Build Up and Download Firmware 40

6-1 Build up Firmware Binary 40
6-2 Download Bootloader Firmware 41
6-3 Download Application Firmware 42

7. Basic Operation of TwinCAT 44

8. PLC Application 45

9. Flash Memory Allocation and FoE Upgrading 46

9-1 FoE Firmware Upgrade 47

10. Object Dictionary 48

10-1 Objects Description 48

11. Performance Evaluation 51

11-1 Minimum DC Cycle Time 51

12. Using the High Speed Clock (HSE) 53

12-1 Modify RCC Setting 53

13. Firmware Architecture 54

13-1 Dual Core Shared Modules 54
13-2 CM7 Core Modules 56
13-3 CM4 Core Modules 56

14. Tuning the PID Loop Gains with TwinCAT 57

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

14-1 Add New Objects into Dictionary in CM4 57
14-2 Link New Objects to Data Structure in CM4 57
14-3 Add Handling Functions According to New Object Indexes in CM7 58
14-4 Create Scope Project in TwinCAT 59
14-5 Position Loop Gain Value Manual Tuning in TwinCAT 62
14-6 Speed Loop Gain Value Manual Tuning in TwinCAT 63

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

List of Figure

Figure 1-1 8
Figure 2-1 9
Figure 2-2 9
Figure 2-3 9
Figure 2-4 10
Figure 2-5 10
Figure 2-6 10
Figure 2-7 10
Figure 2-8 11
Figure 2-9 11
Figure 2-10 13
Figure 3-1 14
Figure 3-2 15
Figure 3-3 15
Figure 4-1 16
Figure 4-2 17
Figure 4-3 18
Figure 4-4 18
Figure 5-1 19
Figure 5-2 19
Figure 5-3 20
Figure 5-4 20
Figure 5-5 21
Figure 5-6 21
Figure 5-7 21
Figure 5-8 22
Figure 5-9 22
Figure 5-10 23
Figure 5-11 23
Figure 5-12 23
Figure 5-13 24
Figure 5-14 24
Figure 5-15 25
Figure 5-16 25
Figure 5-17 26
Figure 5-18 26
Figure 5-19 27
Figure 5-20 27
Figure 5-21 28
Figure 5-22 28
Figure 5-23 29
Figure 5-24 29
Figure 5-25 30
Figure 5-26 30
Figure 5-27 31
Figure 5-28 31
Figure 5-29 32
Figure 5-30 32
Figure 5-31 34
Figure 5-32 34
Figure 5-33 34
Figure 5-34 35
Figure 5-35 35
Figure 5-36 36
Figure 5-37 37
Figure 5-38 37

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Figure 5-39 38
Figure 5-40 38
Figure 5-41 38
Figure 5-42 39
Figure 6-1 40
Figure 6-2 40
Figure 6-3 41
Figure 6-4 42
Figure 6-5 42
Figure 6-6 42
Figure 6-7 43
Figure 9-1 46
Figure 9-2 47
Figure 9-3 47
Figure 9-4 47
Figure 11-1 51
Figure 11-2 51
Figure 11-3 52
Figure 12-1 53
Figure 12-2 53
Figure 13-1 54
Figure 14-1 57
Figure 14-2 57
Figure 14-3 58
Figure 14-4 58
Figure 14-5 58
Figure 14-6 59
Figure 14-7 59
Figure 14-8 60
Figure 14-9 60
Figure 14-10 60
Figure 14-11 61
Figure 14-12 61
Figure 14-13 62
Figure 14-14 62
Figure 14-15 63

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

List of Table

Table 2-1 12
Table 2-2 12
Table 2-3 13
Table 3-1 14
Table 3-2 15
Table 4-1 16
Table 9-1 46
Table 10-1 48
Table 10-2 49
Table 10-3 49
Table 10-4 50
Table 13-1 55
Table 13-2 55

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

1. Introduction

This document introduces how to setup entire environment for motor control application on AX58400-TSB-1 with ST X-NUCLEO-IHM08M1. The demo application of EtherCAT master side develops using PLC (Programmable Logic Controller) language through BECKHOFF TwinCAT engineering (XAE). The IDE relies on Microsoft Visual Studio 2010 / 2013. We can execute the application through the IDE or pure BECKHOFF runtime package (XAR). The firmware part of EtherCAT slave side develops using C language through ARM KEIL MDK (Microcontroller Development Kit).

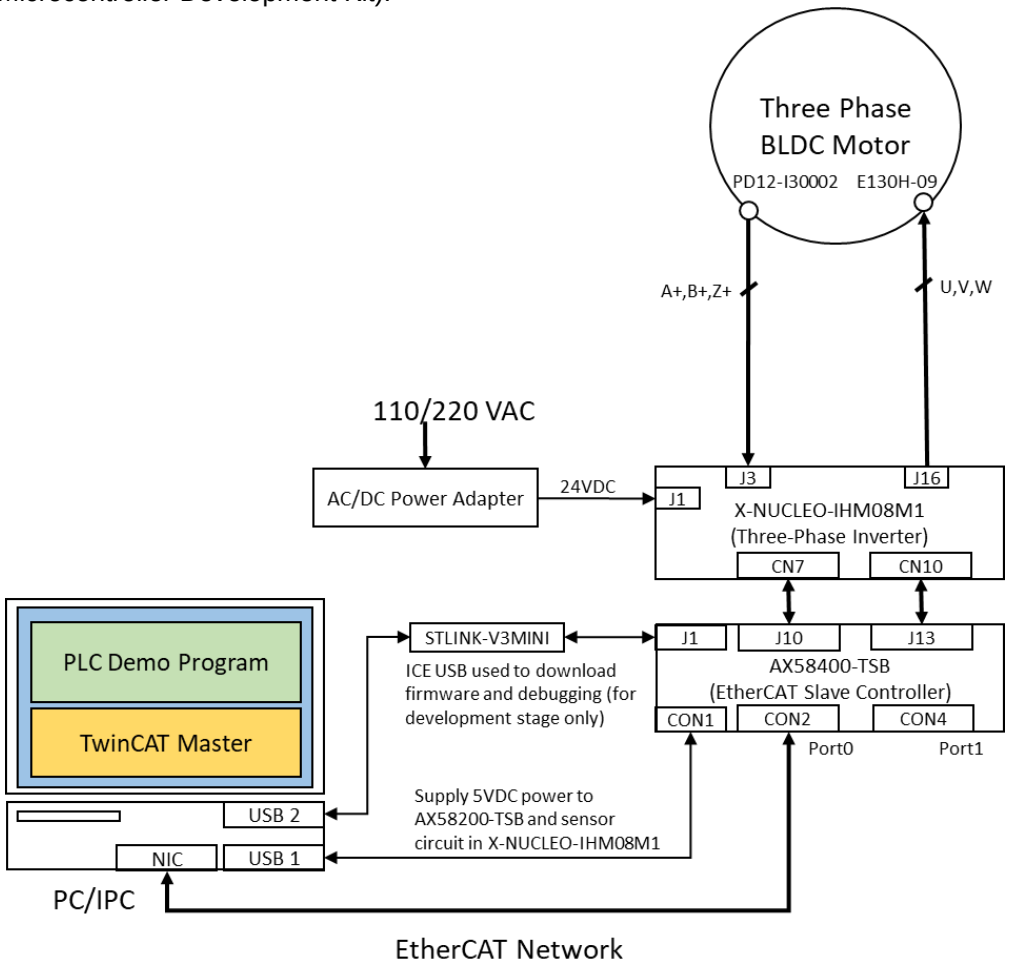


Figure 1-1

The AX58400 firmware mainly handles below affairs:

- BECKHOFF SSC (Slave Stack Code)
- Hardware access of SSC through SPI interface at MCU side
- CiA402 state machine and CSP, CSV and Homing mode operating.
- Control/status word, target position/velocity, actual position/velocity and mode of operation are accessed by EtherCAT PD (Process Data)
- Motor control state machine handling, current reading, incremental encoder sensing and FOC/SVPWM driving for 3-phase BLDC/PMSM motors.

The PLC Demo Program handles below tasks:

- PD mapping / handling by PLC for CSP/CSV/Homing combination mode
- A demo GUI

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

2. Requirements

Before starting to setup the environment, we should prepare some software and hardware. Of course, they are all running based on general desktop PC with windows 10 / 11 operation systems. We list software / hardware requirements as following:

[Software at EtherCAT master side]

- Windows 10 / 11 operation system
- Microsoft Visual Studio 2010 / 2013
- Beckhoff TwinCAT XAE v3.1.4022.28 or later
  You can download the BECKHOFF TwinCAT here

Figure 2-1

[Software at EtherCAT slave side]

- ST MotorControl Workbench6 version 6.2.1 (for development only)
  You can download the MotorControl Workbench here.

Figure 2-2

- STM32CubeMX version 6.12.1 (for development only)
  You can download the STM32CubeMX here.

Figure 2-3

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- In this reference design, two types of Integration Development Environment (IDE) could be supported, you can choose one for the development.

  - ARM KEIL MDK micro-Vision 5 or later (for development only)
    Notes: If you need to download firmware or enter dual-cores debug mode on KEIL MDK IDE, please ensure the version is v5.25 or later.

Figure 2-4

  - STM32 CubeIDE version 1.15.0 or later (for development only)
    You can download the STM32CubeIDE here.

Figure 2-5

- STM32CubeProgrammer
  You can download the STM32CubeProgrammer here.

Figure 2-6

- SSC Tool V5.13 (for development only)
  You can download the SSC Tool from EtherCAT Technology Group (ETG) here

Figure 2-7

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

[Hardware]

- AX58400 Test Board (AX58400-TSB) x 1
  For more details, please refer to the link below and the following chapter.
  ASIX official link: https://www.asix.com.tw/

- STLINK-V3MINI x 1
  You can download the driver here

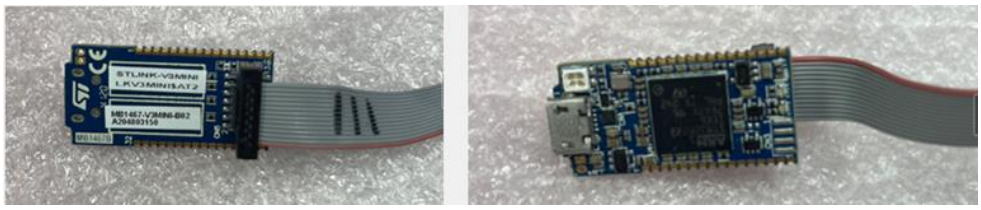


Figure 2-8

- X-NUCLEO-IHM08M1 three-phase brushless DC motor driver expansion board x 1
  For more details, please refer to the link below and the following chapter.
  STMicroelectronics official link: https://www.st.com/en/ecosystems/x-nucleo-ihm08m1.html

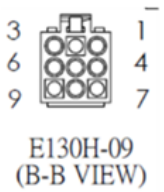
- MOTION TECH BL60M24D8E00430080 Motor x 1
  For more details, please refer to the link below.
  MTM official link: http://www.motiontech.com.tw/

Figure 2-9

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

BL60M24D8E00430080 Performance Spec.	
Rated Voltage	24VDC
Direction of Rotation	C.W.
Rated Power	80W
Rated Torque	2.6 Kg-cm
Rated Input Current	4.0±5%A
Rated Speed	3000±5%RPM
L-L Resistance	486±5%mΩ
Hi-Pot	1000VAC/1mA/1sec.
Insulation Test	500VDC/50MΩ min.

Table 2-1

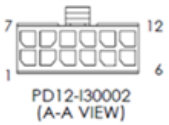


E130H-09
(B-B VIEW)

E130H-09 Connector Wire Definition		
PIN	Definition	Wire color
1	U	Blue
2	V	Purple
3	W	Gray
4	Hall Vcc	Yellow
5	Hall Gnd	Green
6	Null	X
7	S1	Brown
8	S2	Red
9	S3	Orange

Table 2-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.



PD12-I30002 Connector Wire Definition		
PIN	Definition	Wire color
1	Null	X
2	Null	X
3	Z+	Yellow
4	B+	Red
5	A+	Black
6	GND	White
7	Null	X
8	Null	X
9	Z-	Brown
10	B-	Orange
11	A-	Green
12	+5V	Blue

Table 2-3

- 24VDC/8A Power Supply x 1

Figure 2-10

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

3. Board Description

3-1 AX58400-TSB

The AX58400-TSB is a development board for AX58400 IC and has below components:

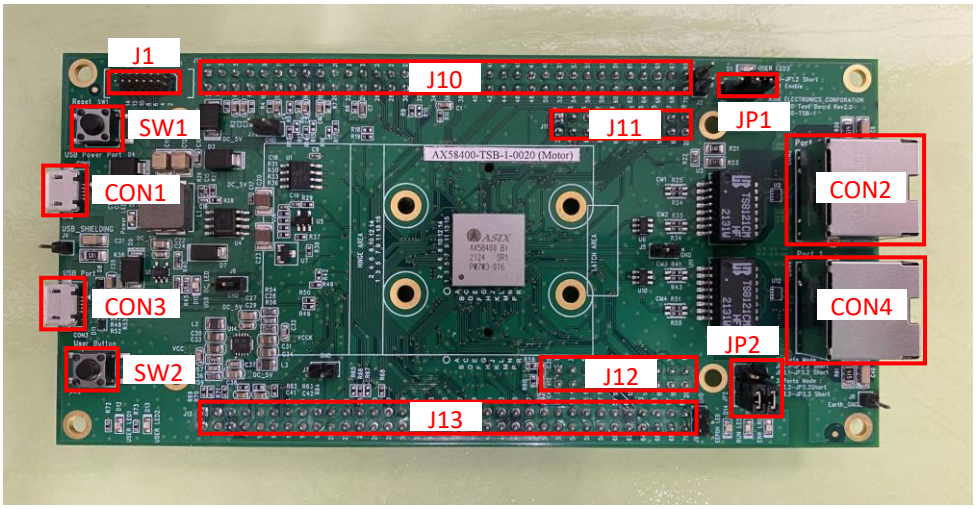


Figure 3-1

- J1: ICE header connector, used to connect STLINK-V3MINI.
- SW1: Reset button, press this button to reset AX58400.
- CON1: USB 5V auxiliary power port.
- CON3: USB OTG FS port.
- SW2: User button for application.
- J10 & J13: Standard connectors that are used to mount with X-NUCLEO-IHM08M1 driver board.
- J11 & J12: ESC I/O extension connectors
- JP1: Boot mode selector.
  Short JP1.1-JP1.2: Boot up from STM32 bootloader.
  Open JP1.1-JP1.2: Boot up from user firmware.
- JP2&JP3: Ports mode selector.
  Short JP2.1-JP2.2 and JP3.1-JP3.2: 2 ports mode.
  Short JP2.2-JP2.3 and JP3.2-JP3.3: 3 ports mode.
- CON2: EtherCAT port 0.
- CON4: EtherCAT port 1.

Attention! because AX58400-TSB is a multi-purpose module, for the Motor Control Reference Design, please ensure that the on-board components are meet the BOM change table as below.

Components	Status
R153, R101, R200, R160, R66, R196, R211, R9	Mounted
R158, R102, R201, R159, R65, R67, R195, R197, R210, R212, R10	Unmounted

Table 3-1

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

3-2 X-NUCLEO-IHM08M1

The X-NUCLEO-IHM08M1 is a three-phase brushless DC motor driver expansion board.

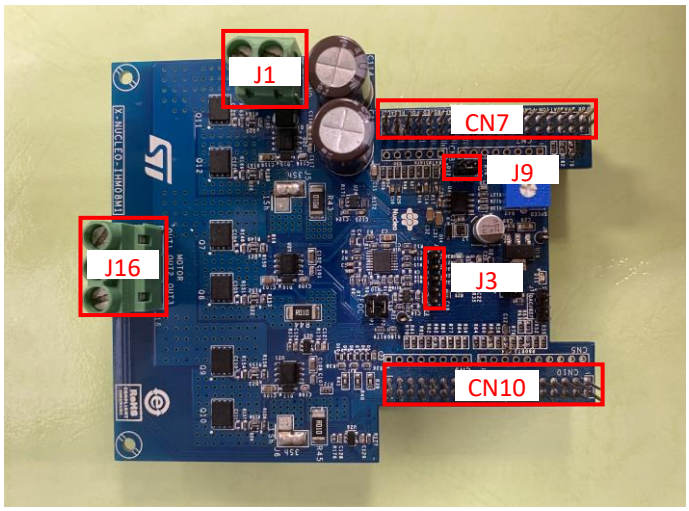


Figure 3-2

- J16: U/V/W three-phase output connector.
- J1: 24V DC link power source connector.
- CN7/CN10: Standard connectors, mounting with J10/J13 on AX58400-TSB-1 board.
- J9: External power selector.
- J3: Incremental encoder interface.

Attention! Before starting to build-up the system, please make sure the X-NUCLEO-IHM08M1 board is ready by check these jumps and components status as below table.

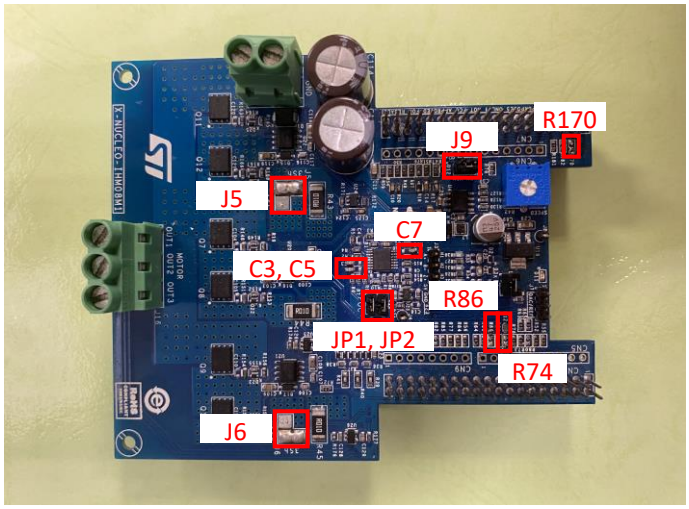


Figure 3-3

Components	Status
JP1, JP2, J9	Closed
J5, J6	Short 3sh side
R74, R86, R170, C3, C5, C7	Unmounted

Table 3-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

4. Environment Setup

4-1 Hardware Connectivity

Please based on below table to connect all necessary equipment.

AX58400 Motor Control Reference Design Connectivity						
Connectivity Item	24VDC	PC/IPC	Axis			
			STLINK-V3MINI	AX58400-TSB-1	X-NUCLEO-IHM08M1	MTM Motor
EtherCAT network		Ethernet port		CON2 (port0)		
				CON4 (port1)		
Three phase inverter control bus				J10	CN7	
				J13	CN10	
Three phase inverter power	+24VDC				J1.1	
	GND				J1.2	
Motor U phase					J16.3 (OUT1)	E130H-09.1 (Blue)
Motor V phase					J16.2 (OUT2)	E130H-09.2 (Purple)
Motor W phase					J16.1 (OUT3)	E130H-09.3 (Gray)
Encoder A					J3.1	PD12-I30002.5
Encoder B					J3.2	PD12-I30002.4
Encoder Z					J3.3	PD12-I30002.3
Encoder +5V					J3.4	PD12-I30002.12
Encoder GND					J3.5	PD12-I30002.6
Power USB		USB Port 1		CON1		
ICE USB		USB Port 2	CN1			
(for development stage only)			CN5	J1		

Table 4-1

If your environment is set up ready, it should look like this.

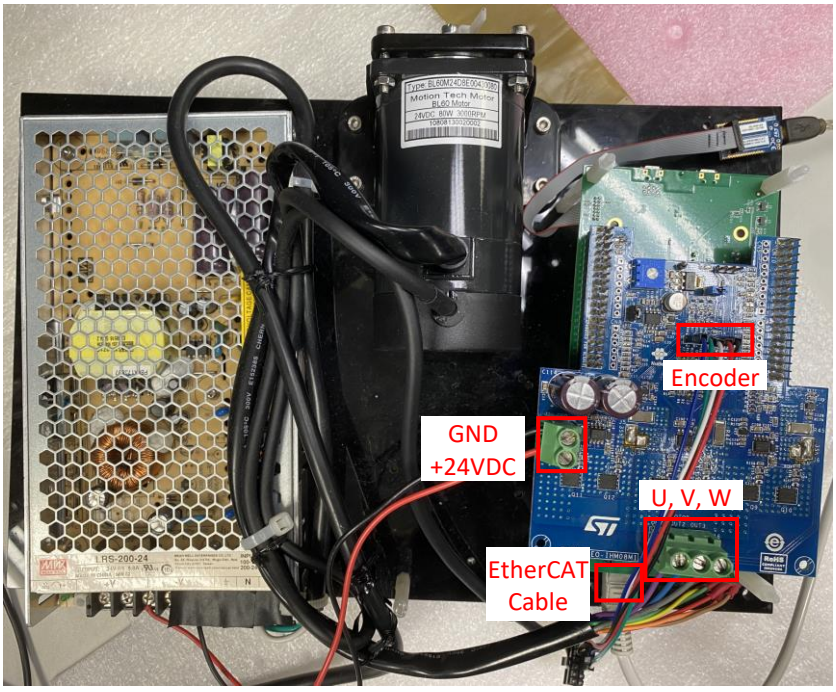


Figure 4-1

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

4-2 AX58400 Pinout Patch for STM32CubeMX

Assume the STM32CubeMX has been installed successfully, and if the developer would like to get a more suitable pinout view of AX58400 on STM32CubeMX, please follow the steps below to achieve this.

Step 1: Open the below files with text editor and modify the "**STMMX_ROOT**" as your **STM32CubeMX** installation path.
AX58400_Pinout_Patch_vXXX\AX58400_Pinout_Patch.cmd
AX58400_Pinout_Patch_vXXX\AX58400_Pinout_UnPatch.cmd

```
rem - STM32CubeMX installation root path configuration
set "STMMX_ROOT=%programfiles%\STMicroelectronics\STM32Cube\STM32CubeMX\"
set "PATCH_SRC_PATH=%~dp0STMMX\"
```

Step 2: Performs the file as below with the administrator permission to apply the patch.
AX58400_Pinout_Patch_vXXX\AX58400_Pinout_Patch.cmd

Step 3: Open the STM32CubeMX and check result. If the patch is successful, you will see the pinout view is as below figure. Depending on the STM32CubeMX version, the MCU name differs:
- Before STM32CubeMX v6.4.0: MCU name is "**STM32_AX58400BIx**".
- STM32CubeMX v6.4.0 and later: MCU name is "**STM32H755ZITx**".

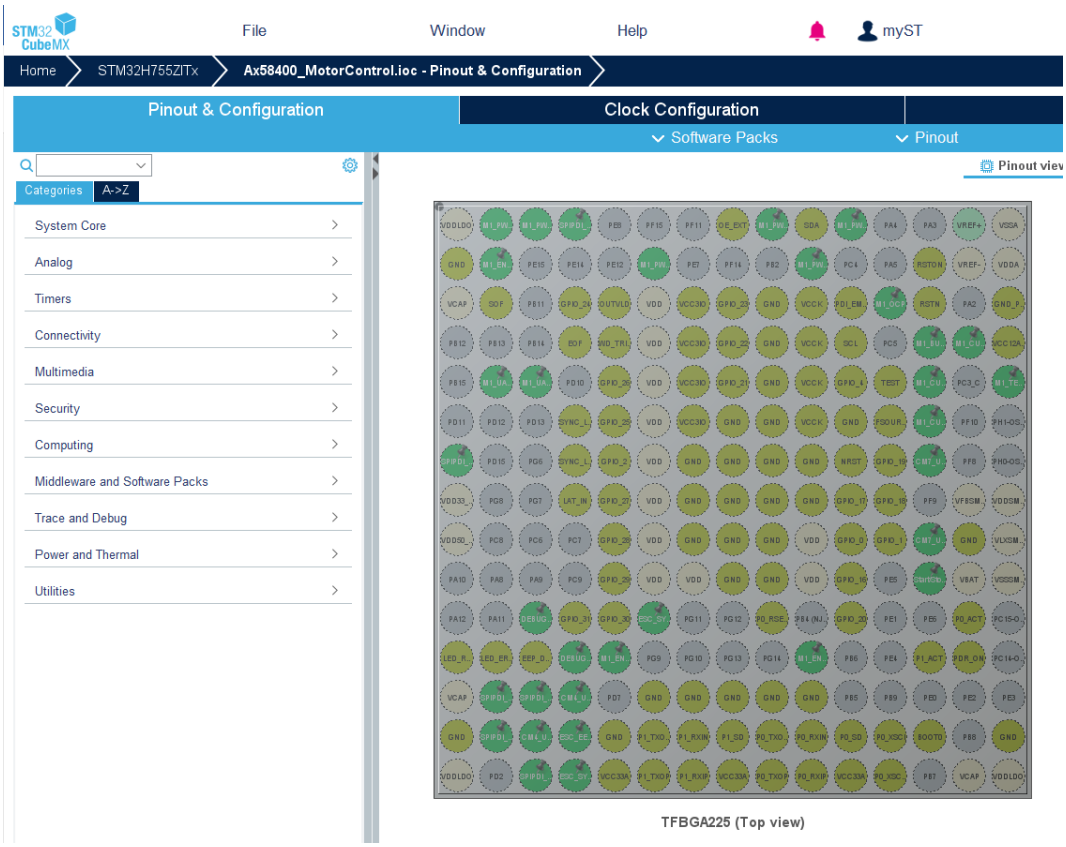


Figure 4-2

Also, if you want go back to original state, please performing the un-patch file below.
AX58400_Pinout_Patch_vXXX\AX58400_Pinout_UnPatch.cmd

Notice: This step requires the additional installation of the "**AX58400 Pinout Patch**" package. Please contact ASIX's Technical Support (support@asix.com.tw) for further support.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

4-3 STM MotorControl Workbench Patch

Due to version 6.2.1 of MotorControl Workbench (MCWB) without support the code generation for STM32H755 MCU and control board, here provides a simple patch to let the STM32H755 is selectable. By the way, the original Workbench always generate MCSDK on CM4 core, the patch also adjusts the MCSDK code generation onto CM7 core.

Please ensure the STM MotorControl Workbench has been installed, then you can follow the steps below to achieve this.

Step 1: Open the below files with text editor and modify the "**MCWB_ROOT**" as your MCWB installation path.
AX58400_MCWB_Patch_vXXX\AX58400_MCWB_Patch.cmd
AX58400_MCWB_Patch_vXXX\AX58400_MCWB_UnPatch.cmd

```
rem - MotorControl Workbench installation root path configuration
set "MCWB_ROOT=%programfiles(x86)%\STMicroelectronics\MC_SDK_6.2.1\Utilities\PC_Software\STMCWB\"
set "PATCH_SRC_PATH=%~dp0STMCWB\"
```

Figure 4-3

Step 2: Performs the file as below with the administrator permission to apply the patch.
AX58400_MCWB_Patch_vXXX\AX58400_MCWB_Patch.cmd

Step 3: Check the patch result.
Executes MotorControl Workbench, if the patch is successful, you will found the NUCLEO-H755ZI is selectable at "**New Project>Control board**" tab, like this.

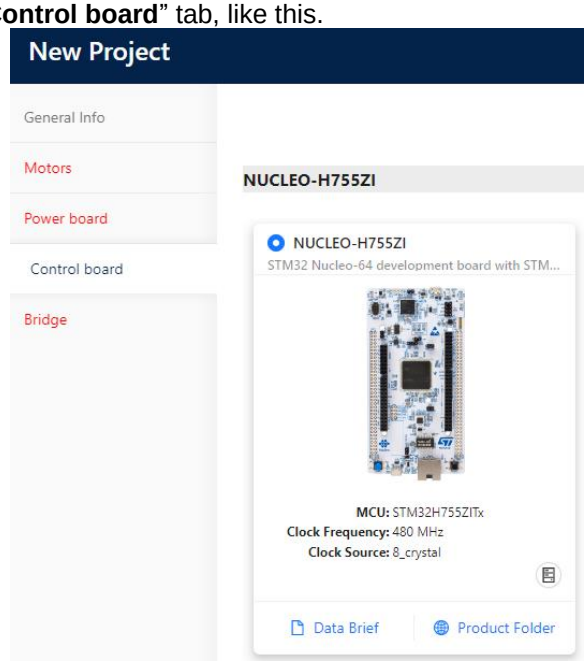


Figure 4-4

Also, if you want go back to original state, please performing the un-patch file below.
AX58400_MCWB_Patch_vXXX\AX58400_MCWB_UnPatch.cmd

Notice: This step requires the additional installation of the "**AX58400 MCWB Patch**" package. Please contact ASIX's Technical Support (support@asix.com.tw) for further support.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

5. Building Reference Design Begin with MC Workbench

This chapter is specifically for designers who need to generate the source code package on their own. Generally, designers can refer to Sections 5-6 to generate the SSC source code, and then Section 6 to build the firmware code directly.

Here will introduce the procedure to build up the complete source code package on MC Workbench v6.2.1, please follow the steps below to do this.

5-1 Performs the MC Workbench Patch for AX58400

This patch will enable the STM32H755 MCU & Control board to be selectable on MC Workbench, please see the Section 4-3 to do this.

5-2 Create New Project and Generate Source Code in MC Workbench

Step 1: Open the MC Workbench and create a new project.
Notice: In order to keep the corrected reference path in BSP, please save the project folder in the path below.
BSP_ROOT\SampleCode

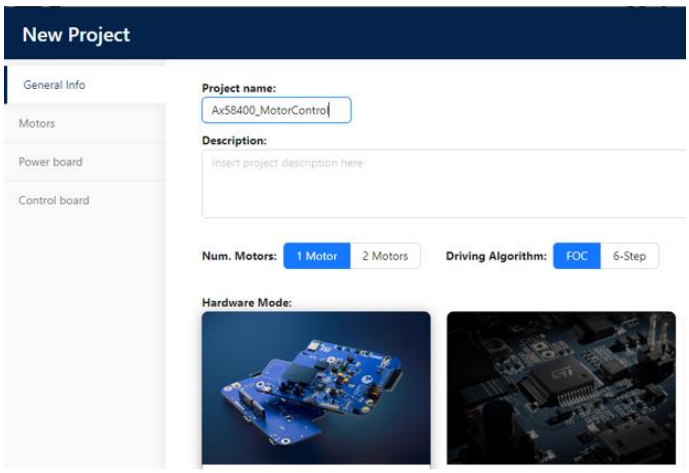


Figure 5-1

Step 2: Select SM-PMSM 24V motor, X-NUCLEO-IHM08M1 power board and NUCLEO-H755ZI control board.

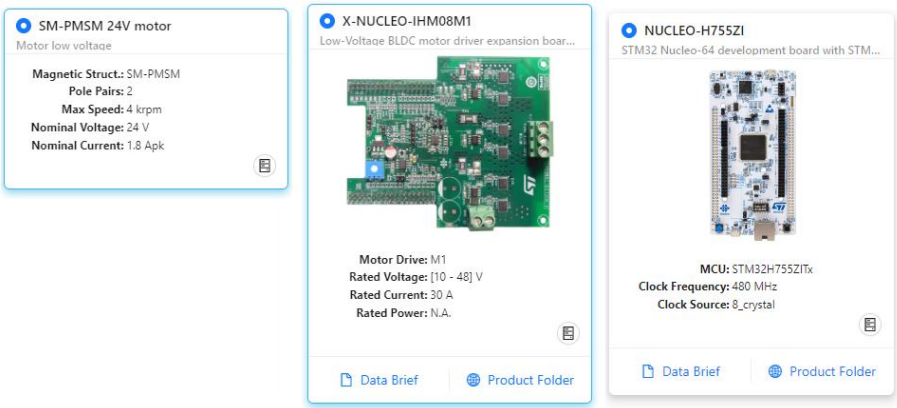


Figure 5-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 3: Setup Motor Properties tab

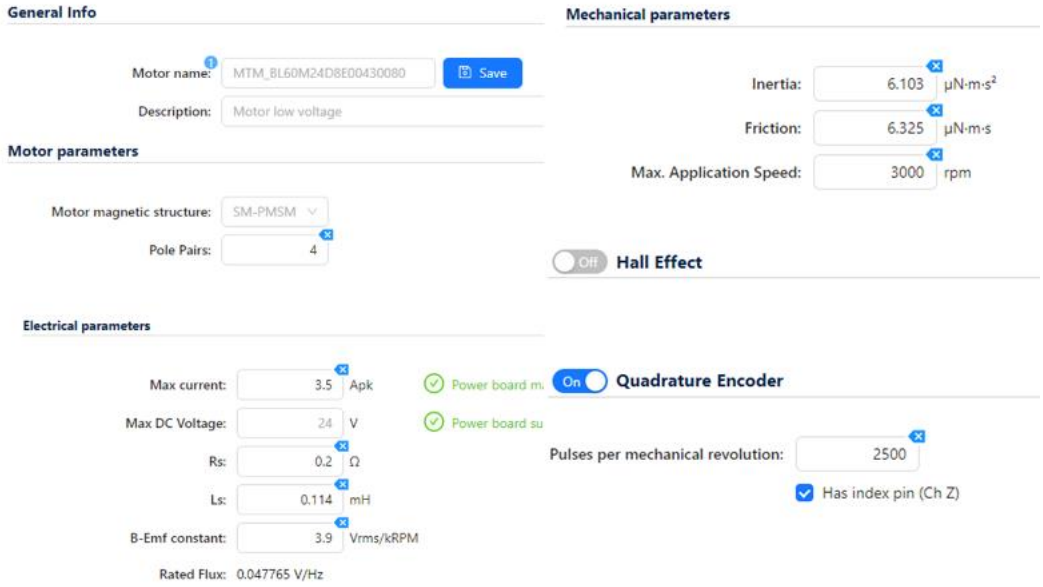


Figure 5-3

Step 4: Setup remanning property tabs as the setting in MC Workbench project file below.
BSP_ROOT\SampleCode\Ax58400_MotorControl\Ax58400_MotorControl.stwb6

Step 5: Generates the source code with the CubeMX v6.12.1 and select Firmware Package Version v1.11.1.

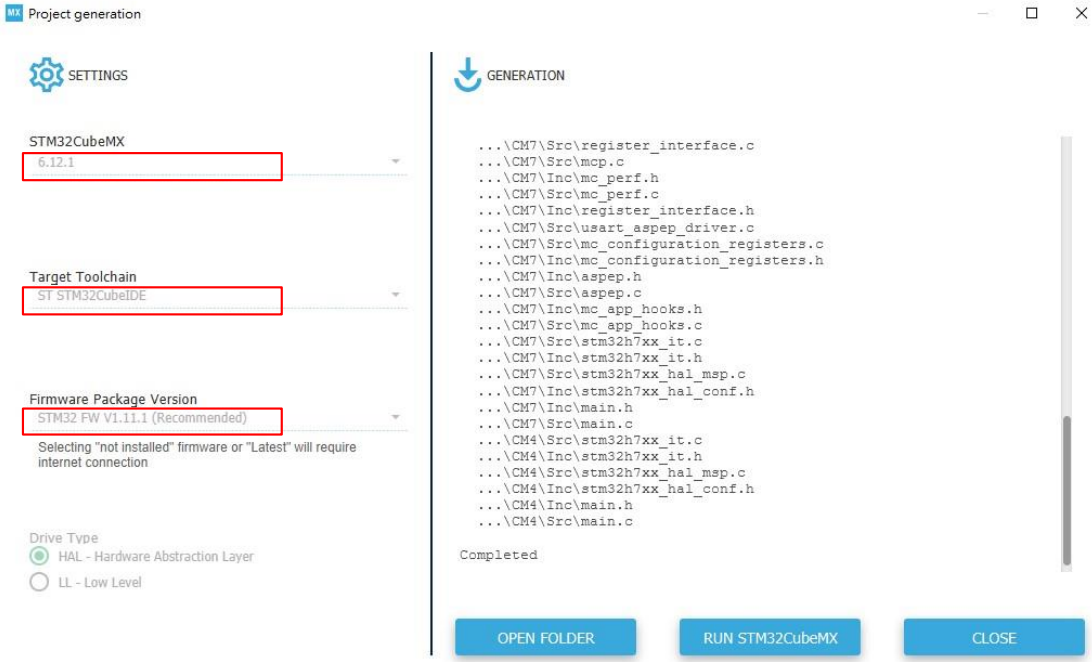


Figure 5-4

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

5-3 Modify Peripheral Settings in STM32CubeMX

Step 1: Modify Mcu.CPN property in CubeMX project file.
Open the Ax58400_MotorControl.ioc file with text editor, then modify the **Mcu.CPN** property value from **STM32H755ZIT3** to be **STM32H755ZIT6**. This can remove the 400MHz limitation in clock configuration. Finally, double click Ax58400_MotorControl.ioc to open it on STM32CubeMX.

Step 2: Disable the "**Pinout & Configuration > System Core > RCC > High Speed Clock (HSE)**" setting and "**Pinout & Configuration > Analog > DAC1**" setting as the figure below.

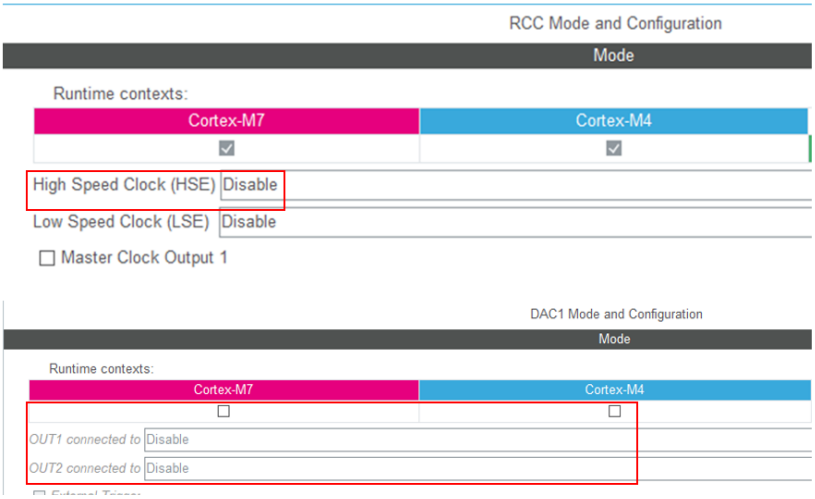
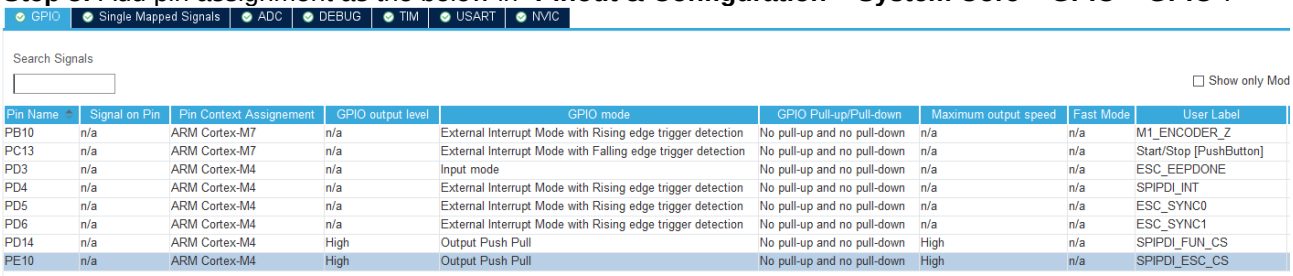


Figure 5-5

Step 3: Add pin assignment as the below in "**Pinout & Configuration > System Core > GPIO > GPIO**".



Pin Name	Signal on Pin	Pin Context Assignement	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output speed	Fast Mode	User Label
PB10	n/a	ARM Cortex-M7	n/a	External Interrupt Mode with Rising edge trigger detection	No pull-up and no pull-down	n/a	n/a	M1_ENCODER_Z
PC13	n/a	ARM Cortex-M7	n/a	External Interrupt Mode with Falling edge trigger detection	No pull-up and no pull-down	n/a	n/a	Start/Stop [PushButton]
PD3	n/a	ARM Cortex-M4	n/a	Input mode	No pull-up and no pull-down	n/a	n/a	ESC_EEPDONE
PD4	n/a	ARM Cortex-M4	n/a	External Interrupt Mode with Rising edge trigger detection	No pull-up and no pull-down	n/a	n/a	SPIPDI_INT
PD5	n/a	ARM Cortex-M4	n/a	External Interrupt Mode with Rising edge trigger detection	No pull-up and no pull-down	n/a	n/a	ESC_SYNC0
PD6	n/a	ARM Cortex-M4	n/a	External Interrupt Mode with Rising edge trigger detection	No pull-up and no pull-down	n/a	n/a	ESC_SYNC1
PD14	n/a	ARM Cortex-M4	High	Output Push Pull	No pull-up and no pull-down	High	n/a	SPIPDI_FUN_CS
PE10	n/a	ARM Cortex-M4	High	Output Push Pull	No pull-up and no pull-down	High	n/a	SPIPDI_ESC_CS

Figure 5-6

Step 4: Modify TIM1 pin assignment in "**Pinout & Configuration > System Core > GPIO > TIM**".
Change TIM1 pins, from PA8, PA9 and PA10 to be PE9, PE11 and PE13. (To select the PA8 pin, hold down Ctrl + left mouse button, then drag it to the PE9 pin.)

Pin Name	Signal on Pin	Pin Context Assignement	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output speed	Fast Mode	User Label
PA6	TIM1_BKIN	ARM Cortex-M7	n/a	Alternate Function Open Drain	Pull-up	n/a	n/a	M1_OCP
PA7	TIM1_CH1N	ARM Cortex-M7	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	M1_PWM_UL
PA15 (JTDI)	TIM2_CH1	ARM Cortex-M7	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	M1_ENCODER_A
PB0	TIM1_CH2N	ARM Cortex-M7	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	M1_PWM_VL
PB1	TIM1_CH3N	ARM Cortex-M7	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	M1_PWM_WL
PB3 (JTDO/TRACESWO)	TIM2_CH2	ARM Cortex-M7	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	M1_ENCODER_B
PE9	TIM1_CH1	ARM Cortex-M7	n/a	Alternate Function Push Pull	Pull-down	Very High	n/a	M1_PWM_UH
PE11	TIM1_CH2	ARM Cortex-M7	n/a	Alternate Function Push Pull	Pull-down	Very High	n/a	M1_PWM_VH
PE13	TIM1_CH3	ARM Cortex-M7	n/a	Alternate Function Push Pull	Pull-down	Very High	n/a	M1_PWM_WH

Figure 5-7

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 5: Add pin assignment for SPI3, UART4 and UART7 on CM4 core.
Firstly, set PH0 & PH1 pins to “Reset State”, then add the pins as below.

GPIO | Single Mapped Signals | ADC | DEBUG | TIM | USART | NVIC

Search Signals

Search (Ctrl+F)

Pin Name	Signal on Pin	Pin Context Assignement	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output speed	Fast Mode	User Label
PC10	SPI3_SCK	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	High	n/a	SPIPDI_CLK
PC11	SPI3_MISO	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	High	n/a	SPIPDI_MISO
PC12	SPI3_MOSI	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	High	n/a	SPIPDI_MOSI
PD0	UART4_RX	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	CM4_UART_RX
PD1	UART4_TX	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	CM4_UART_TX
PF6	UART7_RX	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	CM7_UART_RX
PF7	UART7_TX	n/a	n/a	Alternate Function Push Pull	No pull-up and no pull-down	Low	n/a	CM7_UART_TX

Figure 5-8

Step 6: Modify “**Pinout & Configuration > Timer > TIM1**” settings as highlighted in red.

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Break And Dead Time management - Output Configuration	
Automatic Output State	Disable
Off State Selection for Run Mode (OSSR)	Enable
Off State Selection for Idle Mode (OSSI)	Enable
Lock Configuration	Lock Level 1
Dead Time	((DEAD_TIME_COUNTS) / 2)
Clear Input	
Clear Input Source	Disable
PWM Generation Channel 1 and 1N	
Mode	PWM mode 1
Pulse (16 bits value)	((PWM_PERIOD_CYCLES) / 4)
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High
CHN Polarity	Low
CH Idle State	Reset
CHN Idle State	Set
PWM Generation Channel 2 and 2N	
Mode	PWM mode 1
Pulse (16 bits value)	((PWM_PERIOD_CYCLES) / 4)
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High
CHN Polarity	Low
CH Idle State	Reset
CHN Idle State	Set
PWM Generation Channel 3 and 3N	
Mode	PWM mode 1
Pulse (16 bits value)	((PWM_PERIOD_CYCLES) / 4)
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High
CHN Polarity	Low
CH Idle State	Reset
CHN Idle State	Set

Figure 5-9

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 7: Activated "**Pinout & Configuration > Timer > TIM6**" and "**TIM7**" on CM4 core.

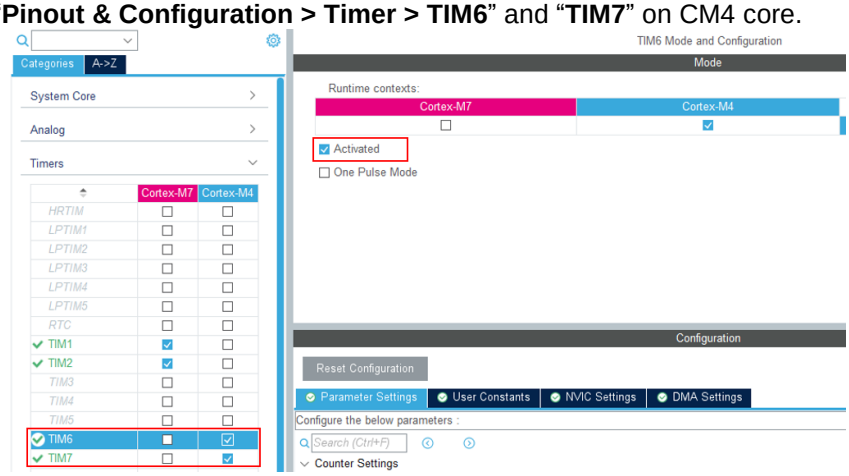


Figure 5-10

Step 8: Enable "**Pinout & Configuration > Connectivity > SPI3**" on CM4 with the setting as highlighted in red.

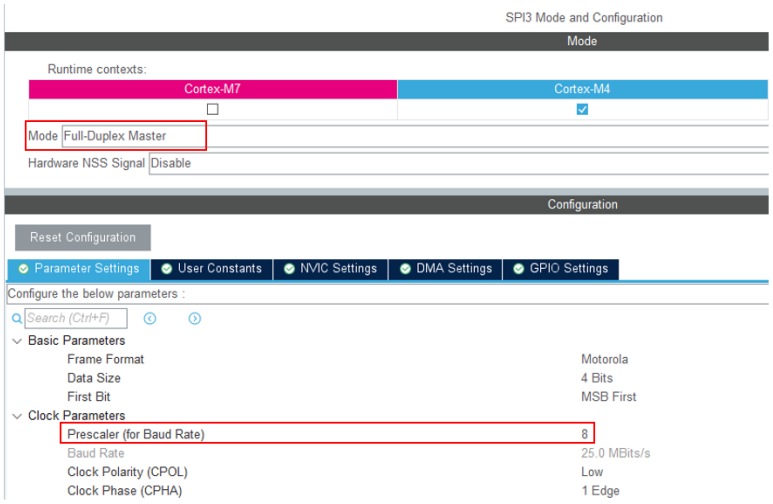


Figure 5-11

Step 9: Enable "**Pinout & Configuration > Connectivity > UART4**" on CM4 and "**UART7**" on CM7.

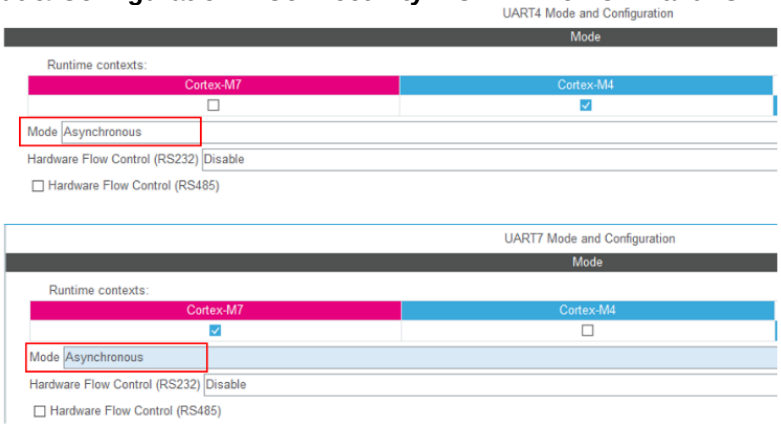


Figure 5-12

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 10: Modify the NVIC1 setting as below.

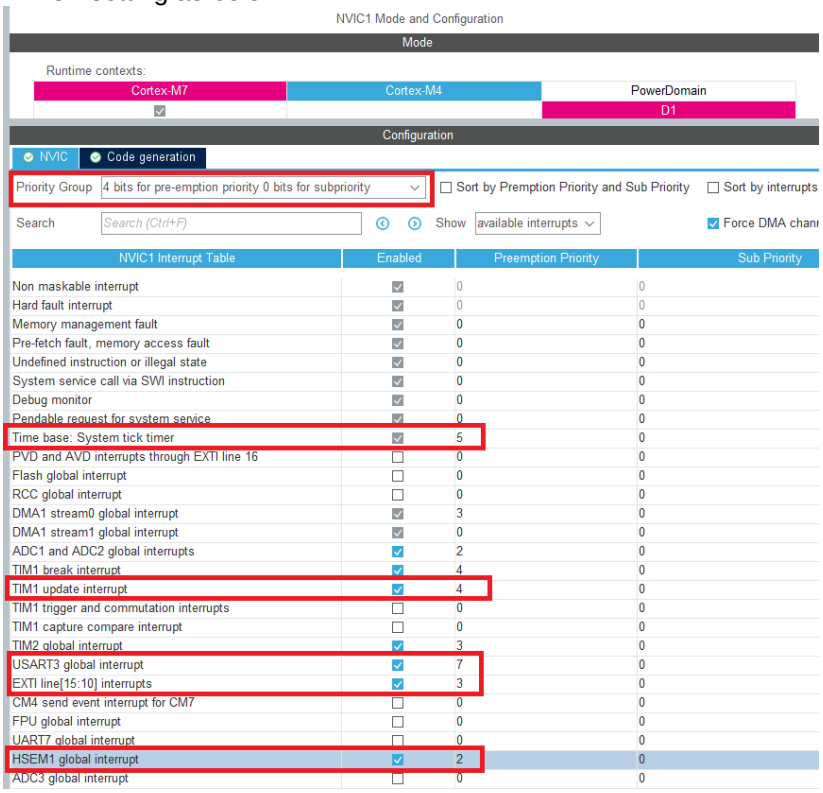


Figure 5-13

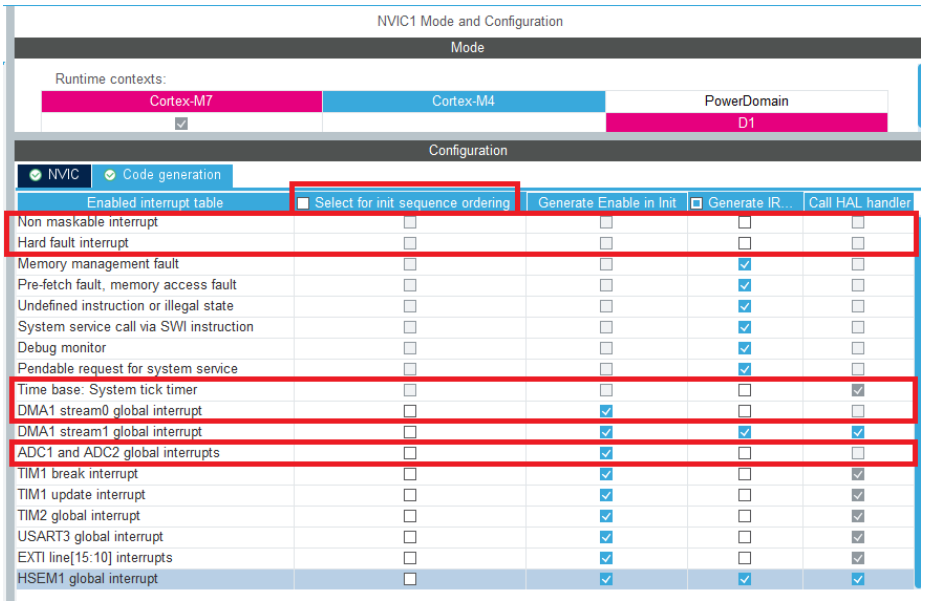


Figure 5-14

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 11: Modify the NVIC2 setting as below.

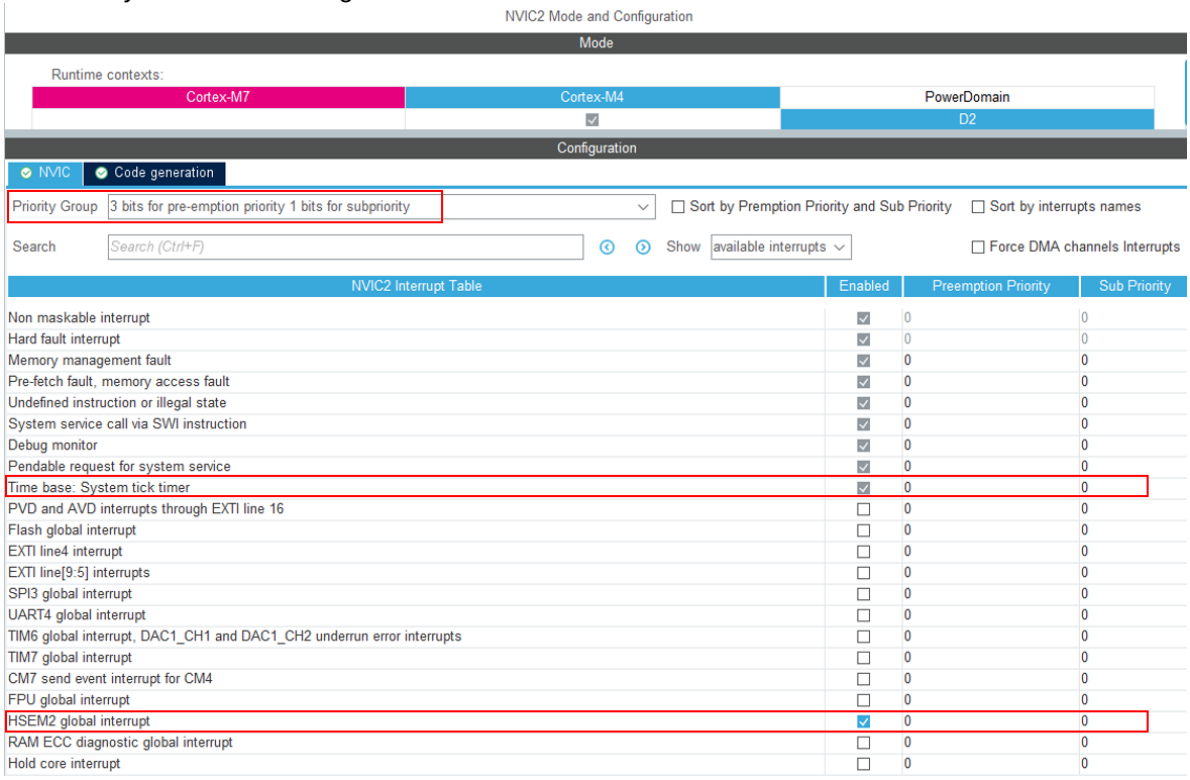


Figure 5-15

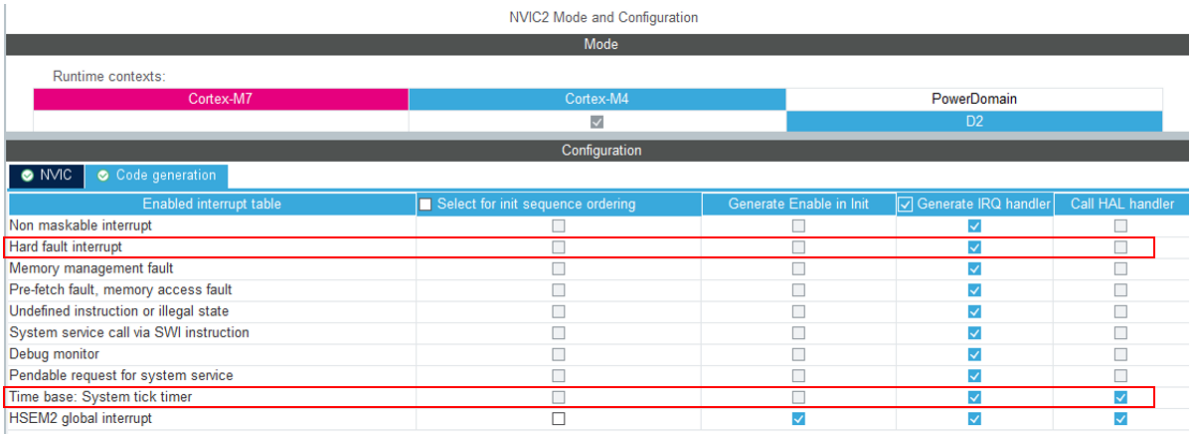


Figure 5-16

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 12: Adjusting the clock tree in "**Clock Configuration**" page.
Press "**NO**" to ignore auto solver process, then modify the settings as the red highlighted as below.

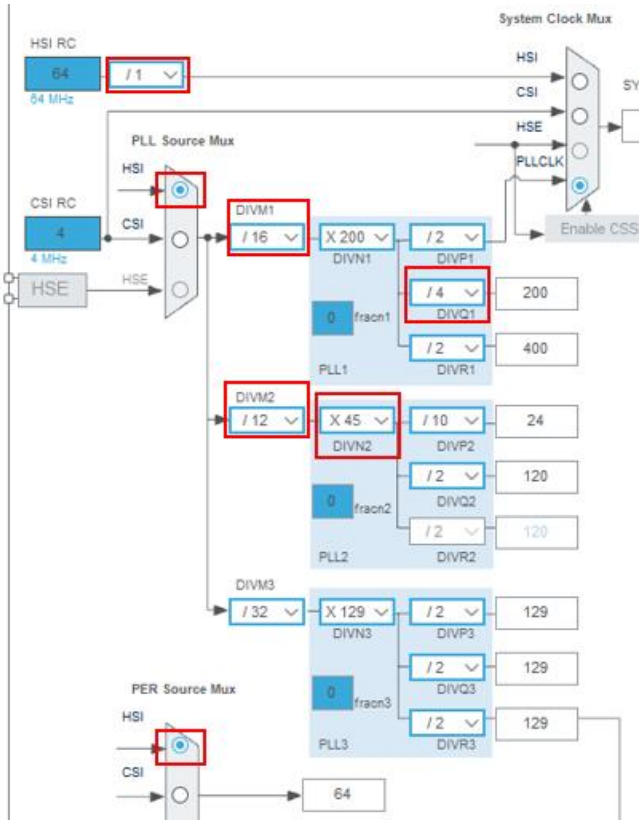


Figure 5-17

5-4 Generate Source Code in STM32CubeMX

Step 1: In "**Project Manager > Project**" page, extended the minimum heap size to **0x1000** on CM4, and select Firmware Package as "**STM32Cube FW_H7 V1.11.1**".

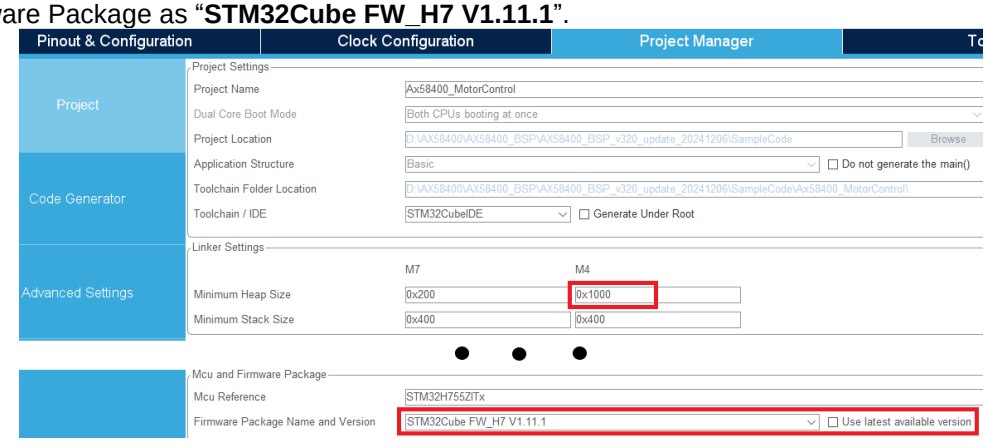


Figure 5-18

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 2: In "**Project Manager > Code Generator**" page, select "**Copy all used libraries into the project folder**".

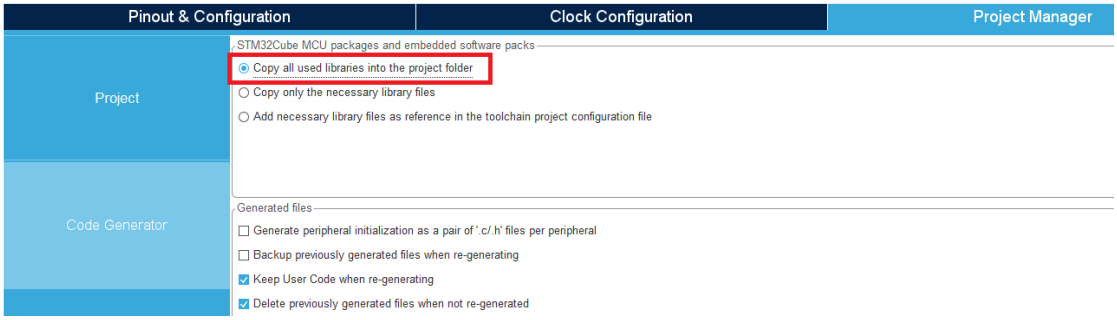


Figure 5-19

Step 3: In "**Project Manager > Advanced Settings**" page, disable initial code generation of SPI3, TIM6 and TIM7 on CM4 core.

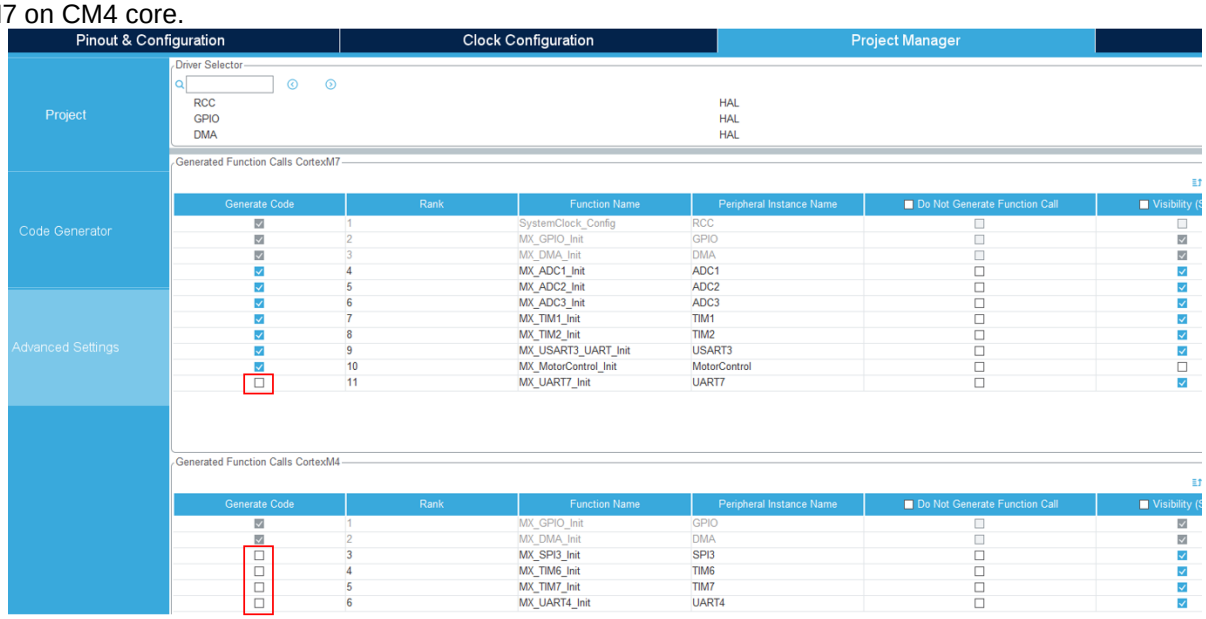


Figure 5-20

Step 4: Generates the source code.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

5-5 Source Code Adjustment

Step 1: Add the AX58400 source package.
Copy the folders as below to your STM32CubeMX project folder.
BSP_ROOT\SampleCode\Ax58400_MotorControl\AX58400
BSP_ROOT\SampleCode\Ax58400_MotorControl\Binary
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_SSC_Tool
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_TwinCAT

Step 2: Adjust the Vector Table Address.
As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\Common\Src\system_stm32h7xx_dualcore_boot_cm4_cm7.c

```
85 /*!< Uncomment the following line if you need to relocate the vector table
86      anywhere in FLASH BANK1 or AXI SRAM, else the vector table is kept at the automatic
87      remap of boot address selected */
88 #define USER_VECT_TAB_ADDRESS
89
90 #if defined(USER_VECT_TAB_ADDRESS)
      ● ● ●
106 #elif defined(CORE_CM7)
107 /*!< Uncomment the following line if you need to relocate your vector Table
108      in D1 AXI SRAM else user remap will be done in FLASH BANK1. */
109 /* #define VECT_TAB_SRAM */
110 #if defined(VECT_TAB_SRAM)
111 #define VECT_TAB_BASE_ADDRESS   D1_AXISRAM_BASE   /*!< Vector Table base address field.
112                                                        This value must be a multiple of 0x400. */
113 #define VECT_TAB_OFFSET         0x000020000U      /*!< Vector Table base offset field.
114                                                        This value must be a multiple of 0x400. */
115 #else
116 #define VECT_TAB_BASE_ADDRESS   FLASH_BANK1_BASE  /*!< Vector Table base address field.
117                                                        This value must be a multiple of 0x400. */
118 #define VECT_TAB_OFFSET         0x000020000U      /*!< Vector Table base offset field.
119                                                        This value must be a multiple of 0x400. */
```

Figure 5-21

Step 3: Add the Share Memory declaration for CM4.

- **For Keil MDK IDE**

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\stm32h755xx_flash_CM4.sct

```
1 ; *************************************************************
2 ; *** Scatter-Loading Description File generated by uVision ***
3 ; *************************************************************
4
5 LR_IROM1 0x08100000 0x00100000  {    ; load region size_region
6   ER_IROM1 0x08100000 0x00100000  {  ; load address = execution address
7    *.o (RESET, +First)
8    *(InRoot$$Sections)
9    .ANY (+RO)
10  }
11  RW_IRAM1 0x10000000 0x00048000{  ; RW data
12   .ANY (+RW +ZI)
13  }
14  RW_IRAM3 0x38000000 0x00010000{
15   *(.RAM_D3)
16  }
17 }
```

Figure 5-22

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- **For STM32CubeIDE**

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\CM4\STM32H755ZITX_FLASH.ld

```
43 /* Specify the memory areas */
44 MEMORY
45 {
46 FLASH (rx)      : ORIGIN = 0x08100000, LENGTH = 1024K
47 RAM (xrw)       : ORIGIN = 0x10000000, LENGTH = 288K
48 RAM_D3 (xrw)    : ORIGIN = 0x38000000, LENGTH = 64K
49 }
     ● ● ●
88   .ARM.extab : /* The READONLY keyword is only supported in GCC
89   {
     ● ● ●
91   } >FLASH
92   .ARM : /* The READONLY keyword is only supported in GCC11 and
93   {
     ● ● ●
97   } >FLASH
98
99   .preinit_array : /* The READONLY keyword is only supported in
100  {
     ● ● ●
104  } >FLASH
105  .init_array : /* The READONLY keyword is only supported in GC
106  {
     ● ● ●
111  } >FLASH
112  .fini_array : /* The READONLY keyword is only supported in GC
113  {
     ● ● ●
163  } >RAM
164
165  .RAM_D3 :
166  {
167  } >RAM_D3 AT>FLASH
168
169  /* Remove information from the standard libraries */
```

Figure 5-23

Step 4: Add the Share Memory declaration for CM7.

- **For Keil MDK IDE**

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\stm32h755xx_flash_CM7.sct

```
1 ; *************************************************************
2 ; *** Scatter-Loading Description File generated by uVision ***
3 ; *************************************************************
4
5 LR_IROM1 0x08020000 0x000E0000  {    ; load region size_region
6   ER_IROM1 0x08020000 0x000E0000  {  ; load address = execution address
7    *.o (RESET, +First)
8    *(InRoot$$Sections)
9    .ANY (+RO)
10  }
11  RW_IRAM1 0x20000000 0x00020000{  ; RW data
12   .ANY (+RW +ZI)
13  }
14  RW_IRAM3 0x38000000 0x00010000{
15   *(.RAM_D3)
16  }
17 }
```

Figure 5-24

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- **For STM32CubeIDE**

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\CM7\STM32H755ZITX_FLASH.ld

```
/* Memories definition */
MEMORY
{
  RAM_D1 (xrw)    : ORIGIN = 0x20000000, LENGTH = 128K
  FLASH   (rx)    : ORIGIN = 0x08020000, LENGTH = 896K     /* Memo
  RAM_D2 (xrw)    : ORIGIN = 0x30000000, LENGTH = 288K
  RAM_D3 (xrw)    : ORIGIN = 0x38000000, LENGTH = 64K
  ITCMRAM (xrw)   : ORIGIN = 0x00000000, LENGTH = 64K
}
    ● ● ●
  .ARM.extab : /* The READONLY keyword is only supported in GCC1
  {
    ● ● ●
  } >FLASH
  .ARM : /* The READONLY keyword is only supported in GCC11 and
  {
    ● ● ●
  } >FLASH

  .preinit_array : /* The READONLY keyword is only supported in
  {
    ● ● ●
  } >FLASH

  .init_array : /* The READONLY keyword is only supported in GCC
  {
    ● ● ●
  } >FLASH

  .fini_array : /* The READONLY keyword is only supported in GCC
  {
    ● ● ●
  } >RAM_D1

  .RAM_D3 :
  {
  } >RAM_D3 AT>FLASH

  /* Remove information from the compiler libraries */
```

Figure 5-25

Step 5: Modify source code on CM4 core.
As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\CM4\Src\main.c

```
/* Private includes ----------------------------------------------
/* USER CODE BEGIN Includes */
#include "AX58400_Hw.h"
#include "applInterface.h"
#include "ax58400.h"
#include "AX58400_MotorControl.h"
#include "bootmode.h"
#include "ax58400_FoeAppl.h"
/* USER CODE END Includes */
    ● ● ●
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  /* USER CODE BEGIN 2 */
#if (AX58400_DEBUG_ENABLE)
  AX_UART_Init();
  CONSOLE_Init();
  DBG_PRINT("%s M4 Firmware v%s\r\n", DEVICE_NAME, DEVICE_SW_VERSION);
#endif //#if (AX58400_DEBUG_ENABLE)
  CC_Init();
#if (SSC_STACK_ENABLE)
  if (HW_Init())
  {
    HW_Release();
  }
  else
  {
    MainInit();
    CiA402_Init();
    AX58400_FoeInit();
  }
#endif
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
#if (AX58400_DEBUG_ENABLE)
        CONSOLE_Task();
#endif //#if (AX58400_DEBUG_ENABLE)

#if (SSC_STACK_ENABLE)
        MainLoop();
        BL_Reboot();

#if (ESC_PHY_ENHANCEMENT)
        HW_PhyEnhancement();
#endif //#if (ESC_PHY_ENHANCEMENT)

#endif //#if (SSC_STACK_ENABLE)
```

Figure 5-26

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\CM4\Src\stm32h7xx_it.c

```
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32h7xx_it.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "ax58400.h"
/* USER CODE END Includes */
      ● ● ●
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  /* USER CODE BEGIN SysTick_IRQn 1 */
#if (AX58400_DEBUG_ENABLE)
  CONSOLE_TimeTick();
#endif //#if (AX58400_DEBUG_ENABLE)
  /* USER CODE END SysTick_IRQn 1 */
}
```

Figure 5-27

Step 6: Modify source code on CM7 core.
As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\CM7\Src\main.c

```
/* USER CODE BEGIN Includes */
#include "ecat_def.h"
#include "ax58400.h"
#include "AX58400_Hw.h"
/* USER CODE END Includes */
      ● ● ●
  MX_USART3_UART_Init();
  MX_MotorControl_Init();
  /* USER CODE BEGIN 2 */
  HAL_SYSTICK_Config(SystemCoreClock / SYS_TICK_FREQUENCY);//Correct the tick time under CPU1 under 400Mhz
#if (AX58400_DEBUG_ENABLE)
  AX_UART_Init();
  CONSOLE_Init();
  DBG_PRINT("%s M7 Firmware v%s\r\n", DEVICE_NAME, DEVICE_SW_VERSION);
#endif //#if (AX58400_DEBUG_ENABLE)
  CC_Init();
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
#if (AX58400_DEBUG_ENABLE)
    CONSOLE_Task();
#endif //#if (AX58400_DEBUG_ENABLE)
    CC_McStatusChangeReq();
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

Figure 5-28

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\CM7\Src\stm32h7xx_mc_it.c

```
/* USER CODE BEGIN Includes */
#include "ax58400.h"
#include "AX58400_Hw.h"
/* USER CODE END Includes */
      ● ● ●
void SysTick_Handler(void)
{
#ifdef MC_HAL_IS_USED
static uint8_t SystickDividerCounter = SYSTICK_DIVIDER;
  /* USER CODE BEGIN SysTick_IRQn 0 */
#if (AX58400_DEBUG_ENABLE)
  if (SystickDividerCounter == SYSTICK_DIVIDER)
  {
      CONSOLE_TimeTick();
  }
#endif //#if (AX58400_DEBUG_ENABLE)

  /* USER CODE END SysTick_IRQn 0 */
  if (SystickDividerCounter == SYSTICK_DIVIDER)
  {
    HAL_IncTick();
    HAL_SYSTICK_IRQHandler();
    SystickDividerCounter = 0;
  }
  SystickDividerCounter ++;
      ● ● ●
void EXTI15_10_IRQHandler (void)
{
/* USER CODE BEGIN START_STOP_BTN */
    MCI_Handle_t *pMC = pMCI[M1];

  if (MCU_GPIO_EXTI_GET_IT(LL_EXTI_LINE_13))
  {
      MCU_GPIO_EXTI_CLEAR_FLAG(LL_EXTI_LINE_13);
      UI_HandleStartStopButton_cb ();
  }

  // Hardcoded to PB10
  if (MCU_GPIO_EXTI_GET_IT(LL_EXTI_LINE_10))
  {
    MCU_GPIO_EXTI_CLEAR_FLAG(LL_EXTI_LINE_10);
    TC_EncoderReset(pMC->pPosCtrl);
  }
/* USER CODE END START_STOP_BTN */
}
```

Figure 5-29

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58400_MotorControl\CM7\Src\mc_task.c

```
#include "mc_app_hooks.h"

/* USER CODE BEGIN Includes */
#include "cmdcode.h"
/* USER CODE END Includes */

/* USER CODE BEGIN Private define */
      ● ● ●
__weak void TSK_MediumFrequencyTaskM1(void)
{
  /* USER CODE BEGIN MediumFrequencyTask M1 0 */
      ● ● ●
      case RUN:
      {
        if (MCI_STOP == Mci[M1].DirectCommand)
        {
          TSK_MF_StopProcessing(M1);
        }
        else
        {
          /* USER CODE BEGIN MediumFrequencyTask M1 2 */
          //#if (MC_STACK_ENABLE)
          if (McInfo.Status.LoopCtrlMode == LC_SPD_CTRL)
          {
              MCI_ExecBufferedCommands(&Mci[M1]);
              FOC_CalcCurrRef(M1);
          }
          else
          {
              TC_PositionRegulation(pPosCtrl[M1]);
              MCI_ExecBufferedCommands(&Mci[M1]);
              FOC_CalcCurrRef(M1);
          }
          break;
          //#endif (MC_STACK_ENABLE)
          /* USER CODE END MediumFrequencyTask M1 2 */
```

Figure 5-30

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

5-6 Generate EtherCAT Slave Stack Code

Due to the ETG Slave Stack Code (SSC) license limitation, the user must re-generate the source code by the SSC tool locally, about how to get the SSC tool and how to generate targeted source code, please refer to the user guide as below.

BSP_ROOT\Document\AX58x00_SSC_Tool_Configuration_Import_UserGuide_vXXX.pdf

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

5-7 IDE Setting Adjustment

- **For Keil MDK IDE:**

Double clicks the file as below to open project.
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\Ax58400_MotorControl.uvproj

Step 1: Add source code both on dual cores.
Clock on "**Project > Manage > Project Items…**" to open "**Manage Project Items**" dialog, then add groups and files as the figure below.

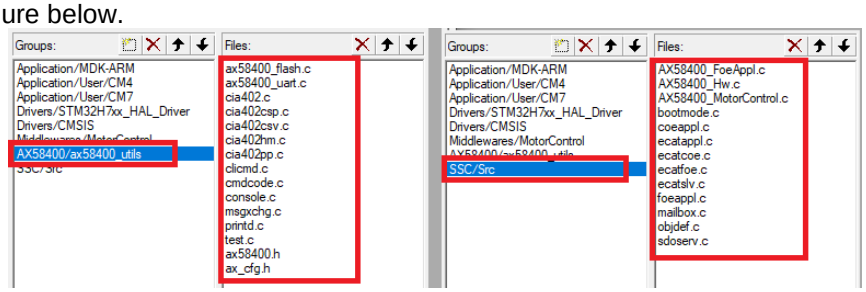


Figure 5-31

Step 2: Add include path both on dual cores.
Click on "**Project > Options for Target 'xxxx'… > C/C++ > Include Path**" to open "**Folder Setup**" dialog, then add paths as the figure below.

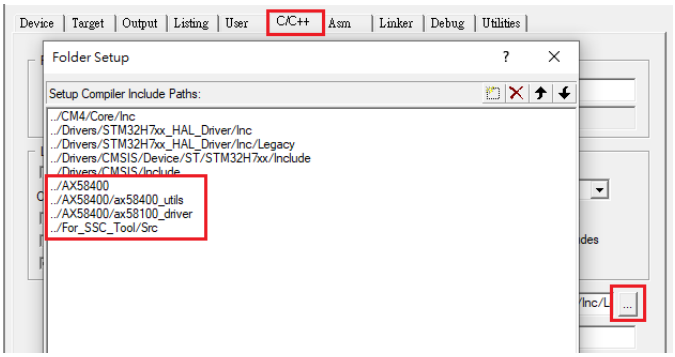


Figure 5-32

Step 3: Setup Utilities.
Click on "**Project > Options for Target 'xxxx'… > Utilities > Settings**", then select "**Flash Download**" page to add "**Programming Algorithm**" as the value below.

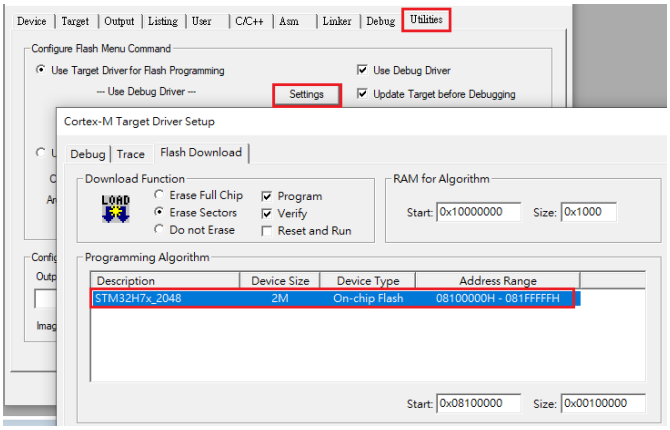


Figure 5-33

For CM4 core: Start = 0x08100000, Size = 0x00100000
For CM7 core: Start = 0x08020000, Size = 0x000E0000

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 4: Add post build actions.
Find the setting of "**Project > Options for Target 'xxxx'… > User > After Build/Rebuild**", then add actions as the value below.

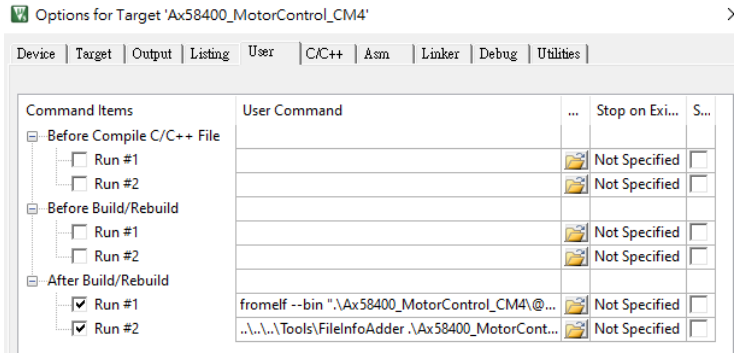


Figure 5-34

For CM4 core:
Run #1: **fromelf --bin ".\Ax58400_MotorControl_CM4\@L.axf" --output ".\Ax58400_MotorControl_CM4\@L.bin"**
Run #2:
..\..\..\Tools\FileInfoAdder .\Ax58400_MotorControl_CM4\Ax58400_MotorControl_CM4.bin ..\For_TwinCAT\FoE\AX58400_MotorControl_CM4.efw -ft 2

For CM7 core:
Run #1: **fromelf --bin ".\Ax58400_MotorControl_CM7\@L.axf" --output ".\Ax58400_MotorControl_CM7\@L.bin"**
Run #2:
..\..\..\Tools\FileInfoAdder .\Ax58400_MotorControl_CM7\Ax58400_MotorControl_CM7.bin ..\For_TwinCAT\FoE\AX58400_MotorControl_CM7.efw -ft 1

Step 5: Modify "**Target Options**".
Find the setting of "**Project > Options for Target 'xxxx'… > Target > Read/Write Memory Areas**", then modify it as the value below.

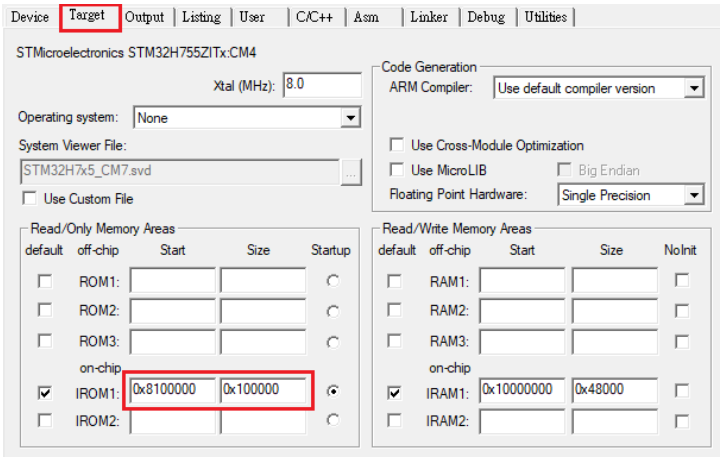


Figure 5-35

For CM4 core:
IROM Start = 0x8100000, Size = 0x100000
IRAM1 Start = 0x10000000, Size = 0x48000

For CM7 core:
IROM Start = 0x8020000, Size = 0xE0000
IRAM1 Start = 0x20000000, Size = 0x20000
IRAM2 Start = 0x0, Size = 0x10000

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 6: Exclude source build on CM7.
Switch to Ax58400_MotorControl_CM7 project, then exclude the source from target build as the figure below.

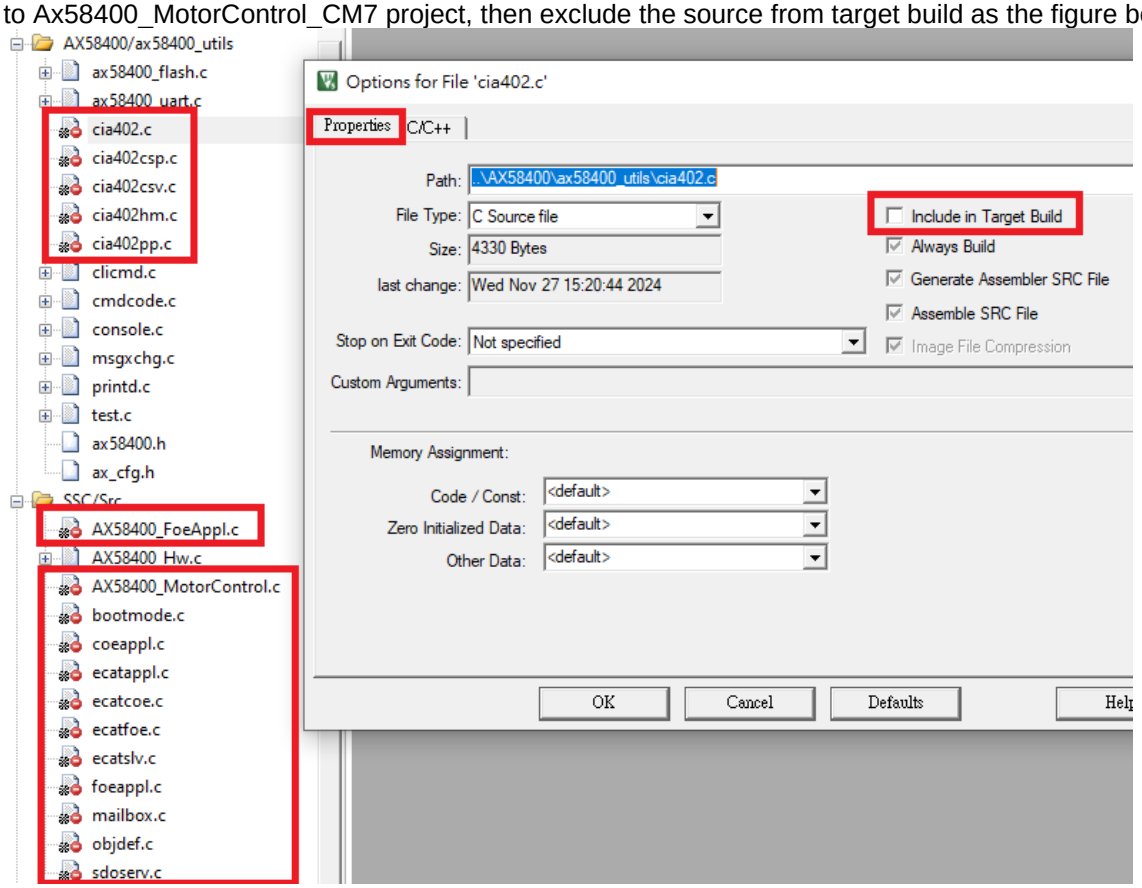


Figure 5-36

Now, entire building procedure has done for Keil MDK IDE.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- **For STM32CubeIDE:**

Double clicks the file as below to open project.
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\.project

Step 1: Add source code.
On "**Project Explorer**", add virtual folders and linking the source files for CM4 and CM7 respectively as below.

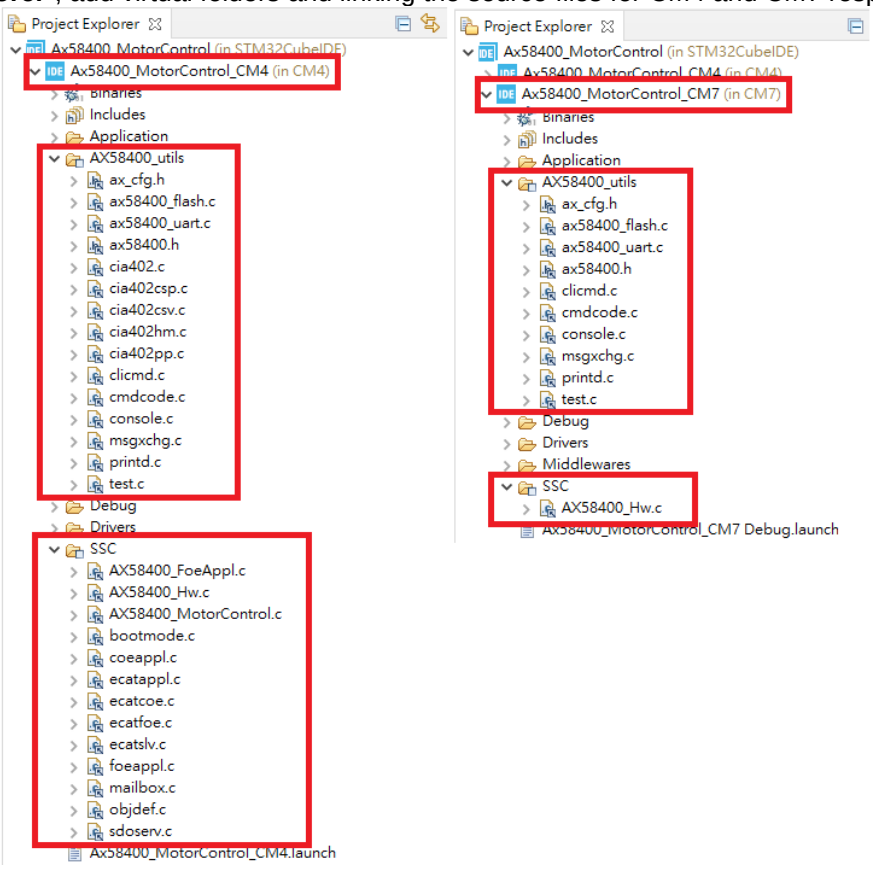


Figure 5-37

- **Add Virtual Folder:** Right click on target project and select "**New > Folder**", then enter new folder name and select "**Folder is not located in the file system (Virtual Folder)**" to create a folder.

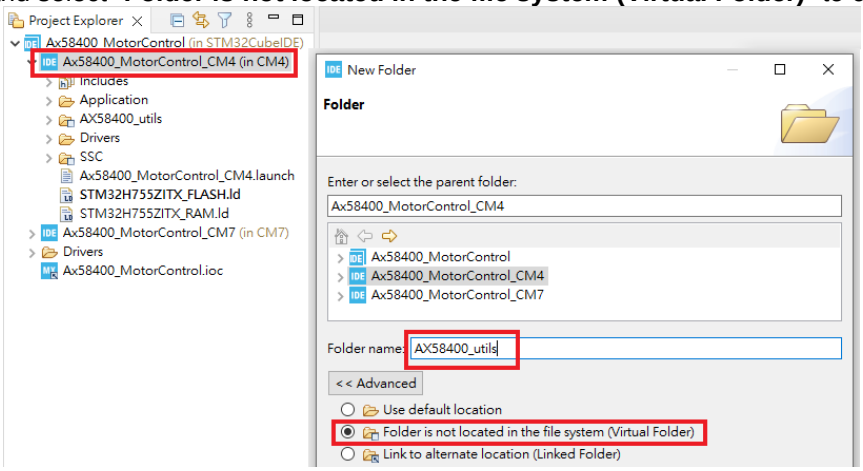


Figure 5-38

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- **Add File Link:** Select and drag the files onto the created virtual folder.

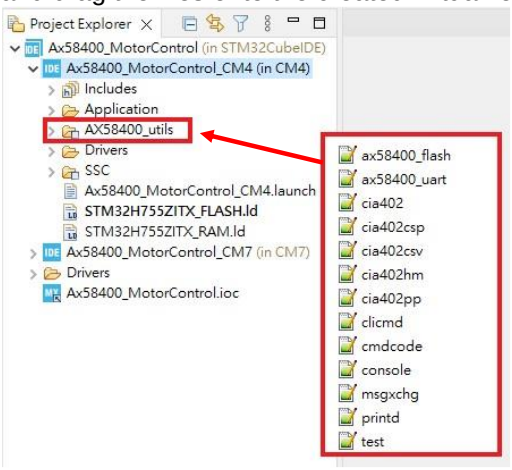


Figure 5-39

Step 2: Enable binary & Hex output files conversion on dual cores.
Select target project, then find setting on "**Project > Properties > C/C++ Build > Settings > Tool Settings > MCU Post build outputs**", then give the setting as highlighted as below.

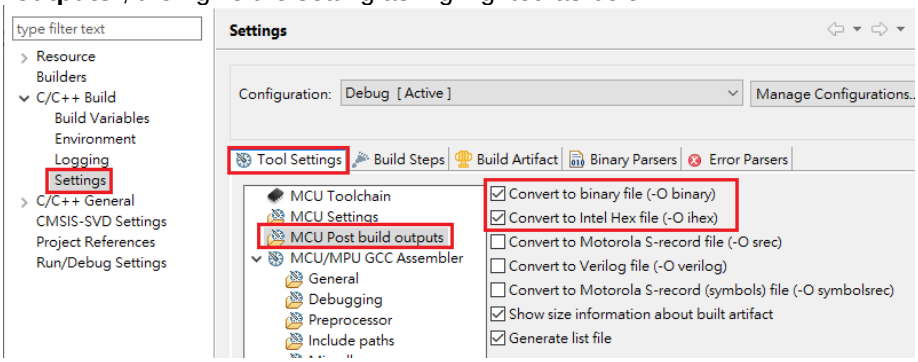


Figure 5-40

Step 2: Add include path both on dual cores.
Find setting on "**Project > Properties > C/C++ Build > Settings > Tool Settings > MCU/MCP GCC Compiler > Include paths**", then add paths as the figure below.

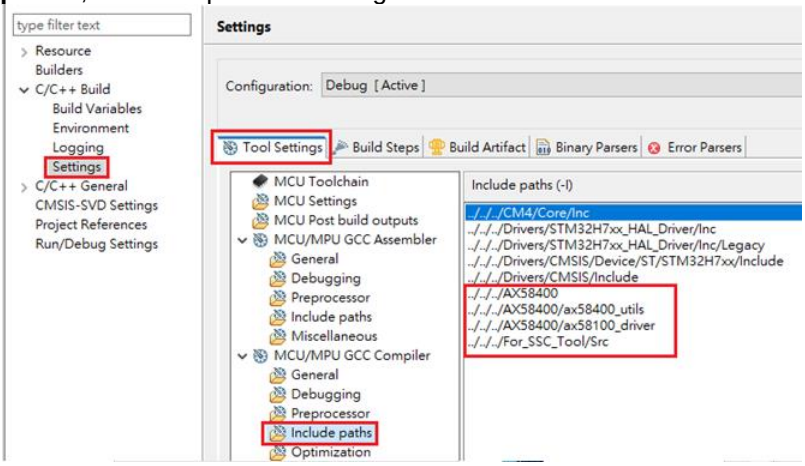


Figure 5-41

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 3: Add post build actions on dual cores.
Find the setting of "**Project > Properties > C/C++ Build > Settings > Build Steps > Post-build steps**", then add actions as the value below.

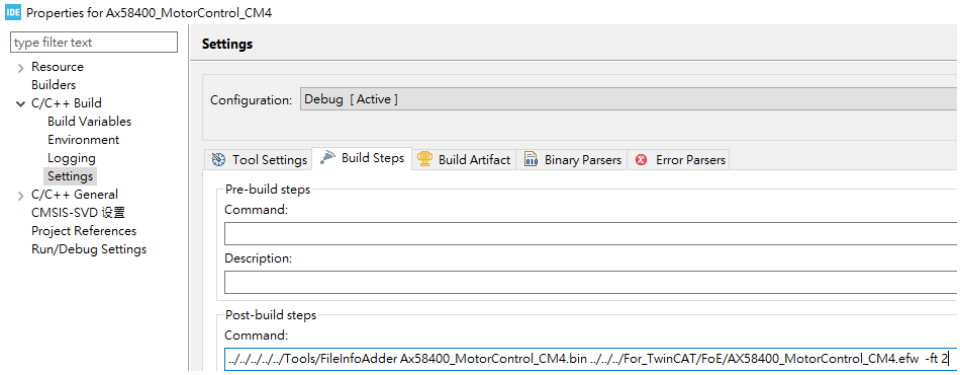


Figure 5-42

For CM4 core:
../../../../../Tools/FileInfoAdder
Ax58400_MotorControl_CM4.bin ../../../For_TwinCAT/FoE/AX58400_MotorControl_CM4.efw -ft 2

For CM7 core:
../../../../../Tools/FileInfoAdder
Ax58400_MotorControl_CM7.bin ../../../For_TwinCAT/FoE/AX58400_MotorControl_CM7.efw -ft 1

Now, entire building procedure has done for STM32CubeIDE.

So far, the entire project source has been ready for building firmware binary.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

6. Build Up and Download Firmware

Here will introduce how to build up firmware binary in your IDE, and how to download firmware using IDE or other utilities

6-1 Build up Firmware Binary

- **For Keil MDK IDE:**

Step 1: Double clicks on project file in the folder below to open project.
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\Ax58400_MotorControl.uvprojx.

Step 2: Execute "**Rebuild**" under "**Menu Bar > Project > Batch Build…**" to build CM4 & CM7 firmware binary files.

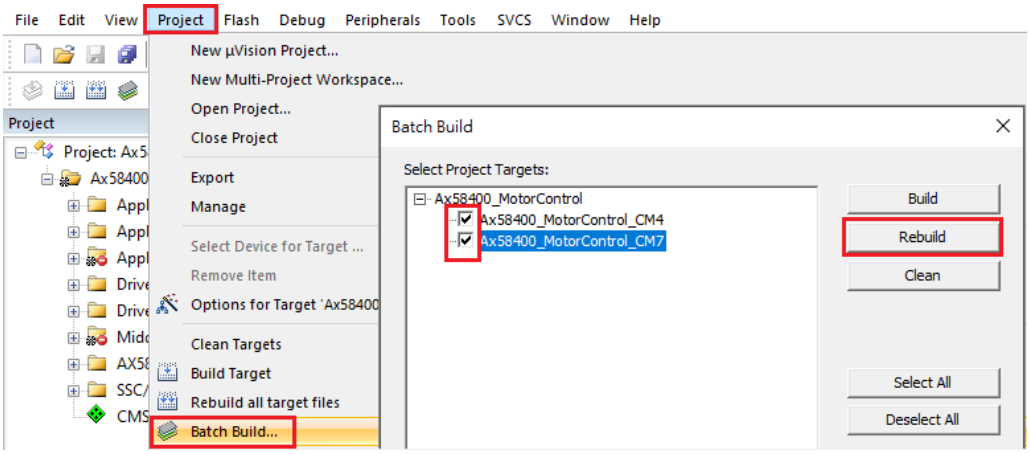


Figure 6-1

- **For STM32CubeIDE:**

Step 1: Double clicks the project file as below path to open project.
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\.project

Step 2: Execute "**Build All**" under "**Menu Bar > Project**" to build CM4 & CM7 firmware binary files.

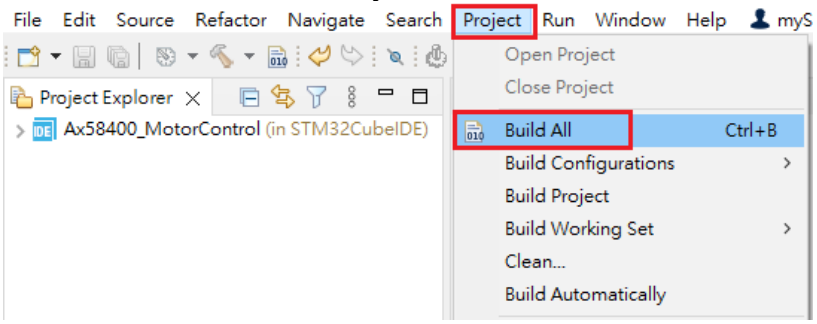


Figure 6-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

6-2 Download Bootloader Firmware

Step 1: Open STM32CubeProgrammer utility and connect to the AX58400-TSB board via STLINK-V3MINI programmer.

Step 2: Select the "**Erasing & Programming**" page, then click "**Browse**" button to select bootloader firmware binary file in the path below.
BSP_ROOT\SampleCode\Ax58400_Bldr\Binary\Ax58400_Bldr_CM7_vX.X.X

Please ensure the "**Start address**" is 0x08000000.

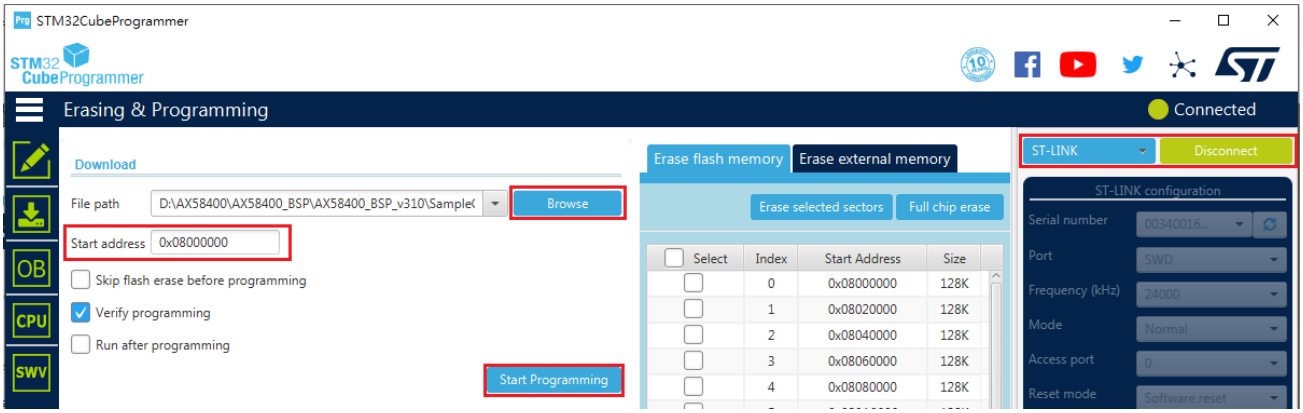


Figure 6-3

Step 3: Press "**Start Programming**" button to download bootloader firmware.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

6-3 Download Application Firmware

- **For Keil MDK IDE:**

Step 1: Select CM4 & CM7 projects respectively, then find the "**Menu bar > Project > Options for Target 'xxxx' > Debug**" tab, select "**ST-Link Debugger**" and click "**Settings**" button, select "**Pack**" tab and uncheck "**Enable**" item.

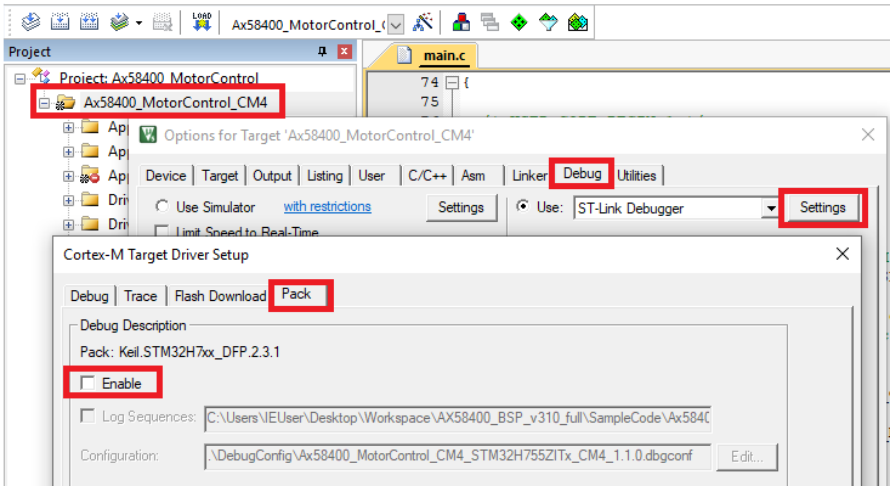


Figure 6-4

Step 2: Select target project then click "**Menu bar > Flash > Download**" to download CM4 & CM7 application firmware respectively.

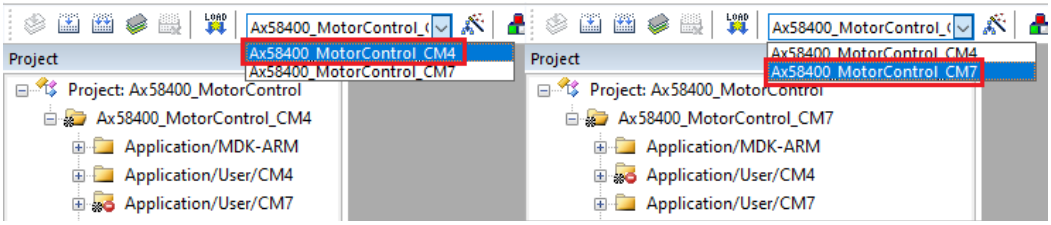


Figure 6-5

- **For STM32CubeIDE:**

Step 1: Select target project and click "**Menu bar > Run > Run As > STM32 C/C++ Application**" to download CM4 & CM7 application firmware respectively.

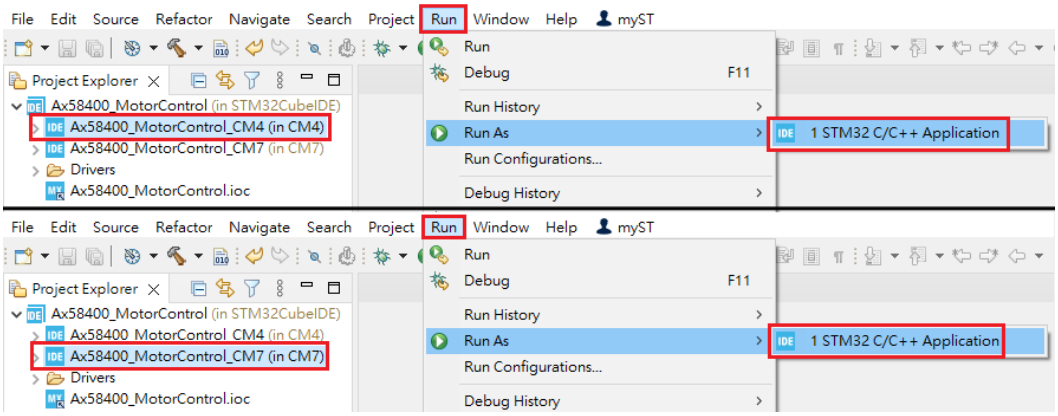


Figure 6-6

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

- **Using STM32CubeProgrammer Utility:**

Step 1: Similarly, please ensure the utility has connected to target AX58400-TSB board.

Step 2: Select the "**Erasing & Programming**" page, then click "**Browse**" button to select CM7 application firmware file in the path below.

- For Keil MDK IDE:
  BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\Ax58400_MotorControl_CM7\Ax58400_MotorControl_CM7.bin

- For STM32CubeIDE:
  BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\Debug\Ax58400_MotorControl_CM7.bin

Please ensure the "**Start address**" is 0x08020000.

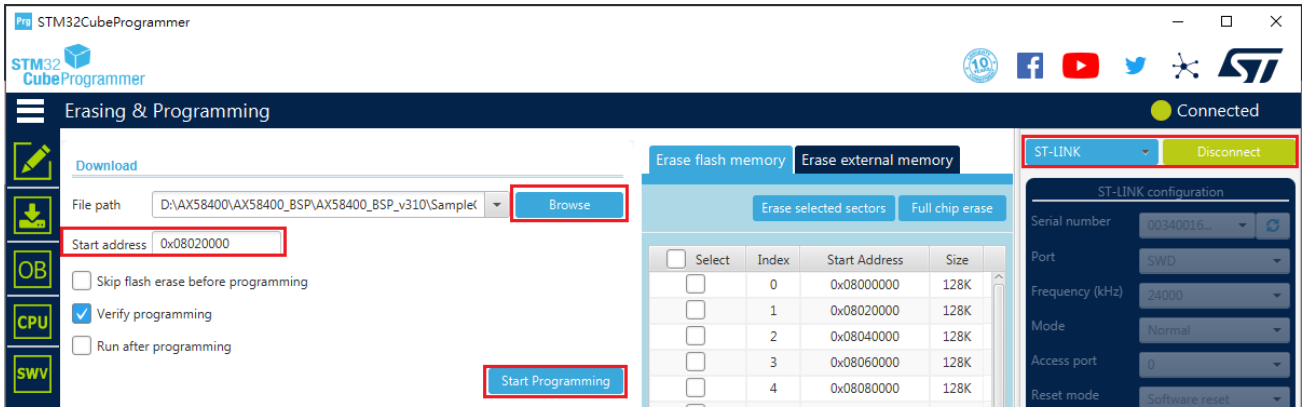


Figure 6-7

Step 3: Press "**Start Programming**" button to download CM7 application firmware.

Step 4: Similarly, for CM4 application firmware download, the firmware path and flash address are as below.

- For Keil MDK IDE:
  BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\Ax58400_MotorControl_CM4\Ax58400_MotorControl_CM4.bin

- For STM32CubeIDE:
  BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\Debug\Ax58400_MotorControl_CM4.bin

Please ensure the "**Start address**" is 0x08100000.

Step 5: Press "**Start Programming**" button to download CM4 application firmware.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

7. Basic Operation of TwinCAT

Please refer to Section 3 of "**AX58x00_TwinCAT_UserGuide**" for more details.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

8. PLC Application

Please refer to Section 5 of “**AX58x00_TwinCAT_UserGuide**” for more details.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

9. Flash Memory Allocation and FoE Upgrading

In order to support “File Access over EtherCAT” (FoE) feature, this reference design has the flash memory allocation on the AX58400 as below.

Flash Area	Flash Memory Addresses	Size (bytes)
Bootloader	0x08000000 ~ 0x0801FFFF	128K
CM7 Application	0x08020000 ~ 0x0809FFFF	512K
Reserved	0x080A0000 ~ 0x080FFFFF	384K
CM4 Application	0x08100000 ~ 0x0817FFFF	512K
Flash Buffer	0x08180000 ~ 0x081FFFFF	512K

Table 9-1

- **Bootloader Area** (Address=0x08000000, Size=128Kbytes):
  The Bootloader firmware must be pre-burn in this area, it’ll responsible for coping the received new firmware from “Flash buffer area” into “Runtime image area”, then executes the new firmware.

- **CM7 Application Area** (Address=0x08020000, Size=512Kbytes):
  Used to run Motor Control Stack firmware.

- **CM4 Application Area** (Address=0x08100000, Size=512Kbytes):
  Used to run EtherCAT Slave Code (SSC) firmware, it is also responsible for store the received .efw file into “Flash buffer area” via FoE data transfer.

- **Flash Buffer Area** (Address=0x08180000, Size=512Kbytes):
  Used to store entire .efw file.

The .efw file is dedicate use for FoE transfer, it’s a firmware binary file with 32-bytes header pre-fix.
Every time the project build done, the .efw file will auto be generated in the path below by “**FileInfoAdder.exe**” utility.
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_TwinCAT\FoE\AX58400_MotorControl_CM4.efw
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_TwinCAT\FoE\AX58400_MotorControl_CM7.efw

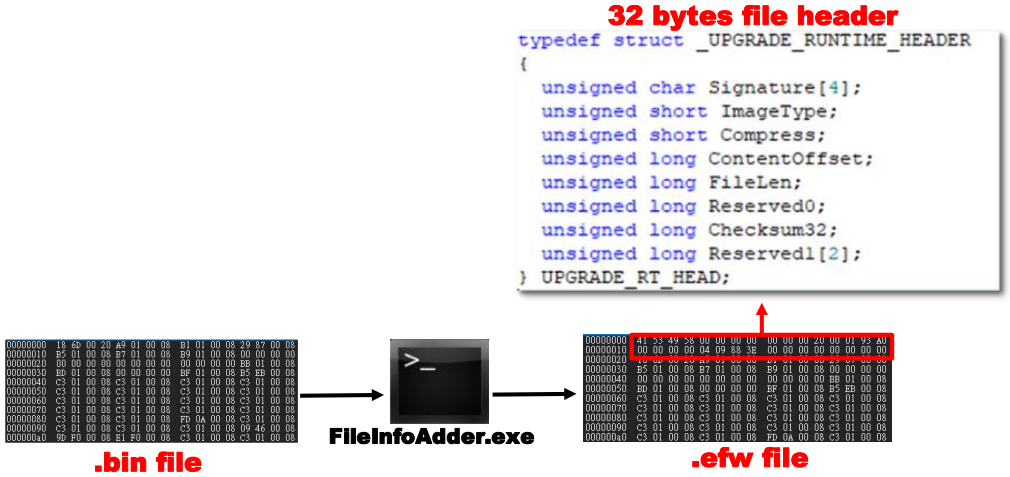


Figure 9-1

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

9-1 FoE Firmware Upgrade

This section shows the steps of firmware update with EtherCAT FoE.

Step 1: Select "**Solution Explorer > AX58400_MotorControl > I/O > Devices > XXX > Online**", then right click on target device and select "**Firmware Update**".

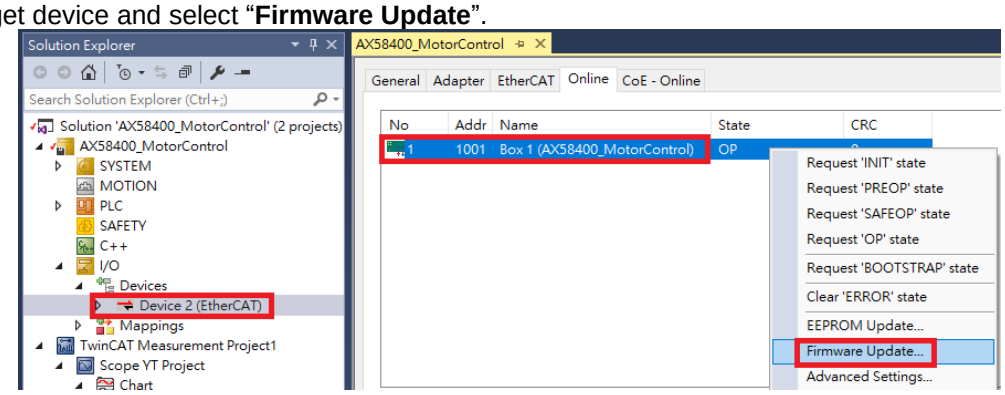


Figure 9-2

Step 2: Select the file as path below, and click "**OK**" to start CM4 application firmware upgrading.
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_TwinCAT\FoE\AX58400_MotorControl_CM4.efw

Step 3: Check the progress bar status and waiting for the processing done.

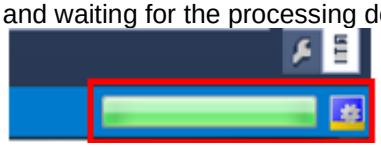


Figure 9-3

Step 4: If the firmware upgrading is complete successfully, the pop up dialog will appear as the figure below.

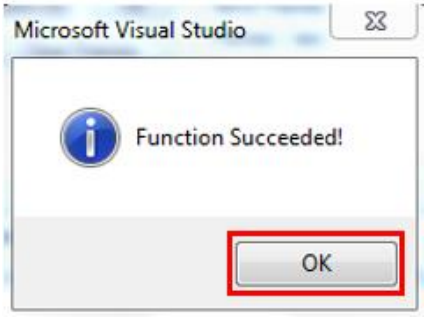


Figure 9-4

Step 5: Similarly, upgrading the CM7 application firmware by select the file as path below.
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_TwinCAT\FoE\AX58400_MotorControl_CM7.efw

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

10. Object Dictionary

These reference design follows CiA402 device profile and used as an interface to integrate Motor Control Stack and EtherCAT Slave Stack, the CiA402 protocol implement a state machine and object dictionary for external EtherCAT master access, Master side can control motor, switch operation mode and monitor motor status by manipulation of object dictionary.

10-1 Objects Description

This section lists out all supported objects in the reference design, please note that some objects may not implemented actual functions with Motor Control Stack side, but they are accessible in EtherCAT Slave side, so developer can still add application functions on them.

Object type definition is as below

Object Type	Description
VAR	A single value
ARRAY	A multiple data field object where each data field is a simple variable of same basic data type.
RECORD	A multiple data field object where the data fields may be any combination of simple variables.

Table 10-1

0x1000~0x1FFF area object definition

Index	Object Type	Name	Data Type	Access	Remark
0x1000	VAR	Device type	UNSIGNED32	RO	
0x1001	VAR	Error register	UNSIGNED8	RO	
0x1008	VAR	Device name	Visible String	RO	
0x1009	VAR	Hardware version	Visible String	RO	
0x100A	VAR	Software version	Visible String	RO	
0x1018	RECORD	Identity	Identity	RO	
0x10F1	RECORD	Error settings	-	RO	
0x10F8	VAR	Time stamp	UNSIGNED64	RO	
0x1C00	ARRAY	Sync manager type	UNSIGNED8	RO	
0x1C32	RECORD	Sync manager output parameters	SM Parameter	RO	
0x1C33	RECORD	Sync manager input parameters	SM Parameter	RO	
0x1600	RECORD	Rx pdo mapping of csp/csv	PDO Mapping	RO	Mapped objects: Controlword Target position Target velocity Modes of operation
0x1601	RECORD	Rx pdo mapping of csp	PDO Mapping	RO	Mapped objects: Controlword Target position
0x1602	RECORD	Rx pdo mapping of csv	PDO Mapping	RO	Mapped objects: Controlword Target velocity
0x1A00	RECORD	Tx pdo mapping of csp/csv	PDO Mapping	RO	Mapped objects Statusword Position actual value Velocity actual value Modes of operation display
0x1A01	RECORD	Tx pdo mapping of csp	PDO Mapping	RO	Mapped objects: Statusword Position actual value

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

0x1A02	RECORD	Tx pdo mapping of csv	PDO Mapping	RO	Mapped objects: Statusword Position actual value
0x1C12	VAR	SM2 assignment	UNSIGNED16	RW	Assign by default the csv process data mapping
0x1C13	VAR	SM3 assignment	UNSIGNED16	RW	Assign by default the csv process data mapping

Table 10-2

0x2000~0x3FFF object definition

Index	Object Type	Name	Data Type	Access	Mappable	Remark
0x3000	VAR	PosKpGain	INTEGER16	RW	N	
0x3001	VAR	PosKpDivisor	INTEGER16	RO	N	
0x3002	VAR	PosKiGain	INTEGER16	RW	N	
0x3003	VAR	PosKiDivisor	INTEGER16	RO	N	
0x3004	VAR	PosKdGain	INTEGER16	RW	N	
0x3005	VAR	PosKdDivisor	INTEGER16	RO	N	
0x3010	VAR	SpdKpGain	INTEGER16	RW	N	
0x3011	VAR	SpdKpDivisor	INTEGER16	RO	N	
0x3012	VAR	SpdKiGain	INTEGER16	RW	N	
0x3013	VAR	SpdKiDivisor	INTEGER16	RO	N	
0x3014	VAR	SpdKdGain	INTEGER16	RW	N	
0x3015	VAR	SpdKdDivisor	INTEGER16	RO	N	
0x3020	VAR	TrqKpGain	INTEGER16	RW	N	
0x3021	VAR	TrqKpDivisor	INTEGER16	RO	N	
0x3022	VAR	TrqKiGain	INTEGER16	RW	N	
0x3023	VAR	TrqKiDivisor	INTEGER16	RO	N	
0x3024	VAR	TrqKdGain	INTEGER16	RW	N	
0x3025	VAR	TrqKdDivisor	INTEGER16	RO	N	

Table 10-3

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

0x6000~0xFFFF object definition

Index	Object Type	Name	Data Type	Access	Mappable	Remark
0x603F	VAR	Error code	UNSIGNED16	RO	N	
0x6040	VAR	Controlword	UNSIGNED16	RW	Y	
0x6041	VAR	Statusword	UNSIGNED16	RO	Y	
0x604F	VAR	VI Ramp Function Time	UNSIGNED 32	RW	N	
0x605A	VAR	Quick stop option code	INTEGER16	RW	N	
0x605B	VAR	Shutdown option code	INTEGER16	RW	N	
0x605C	VAR	Disable operation option code	INTEGER16	RW	N	
0x605E	VAR	Fault reaction option code	INTEGER16	RW	N	
0x6060	VAR	Modes of operation	INTEGER8	RW	Y	
0x6061	VAR	Modes of operation display	INTEGER8	RO	Y	
0x6064	VAR	Position actual value	INTEGER32	RO	Y	
0x606C	VAR	Velocity actual value	INTEGER32	RO	Y	
0x6077	VAR	Torque actual value	INTEGER16	RO	N	
0x607A	VAR	Target position	INTEGER32	RW	Y	
0x607C	VAR	Home offset	INTEGER32	RW	N	
0x607D	ARRAY	Software position limit	-	RW	N	
0x6085	VAR	Quick stop deceleration	UNSIGNED32	RW	N	
0x6098	VAR	Homing method	INTEGER8	RW	N	
0x6099	RECORD	Homing speed	-	RW	N	
0x609A	VAR	Homing acceleration	UNSIGNED32	RW	N	
0x60C2	RECORD	Interpolation time period	-	RW	N	
0x60E3	RECORD	Supported homing methods	-	RO	N	
0x60FD	VAR	Digital Inputs	UNSIGNED32	RO	N	
0x60FF	VAR	Target velocity	INTEGER32	RW	Y	
0x6502	VAR	Supported drive modes	UNSIGNED32	RO	N	
0xF000	RECORD	Modular device profile	-	RO	N	
0xF010	VAR	Module profile list	UNSIGNED32	RO	N	

Table 10-4

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

11. Performance Evaluation

11-1 Minimum DC Cycle Time

With TwinCAT Master, the DC and PD conditions are shown as below. Master use LRW command to do PD transfer with 14bytes input and 14bytes output, under this setting, the minimum DC cycle time can down to 100us.

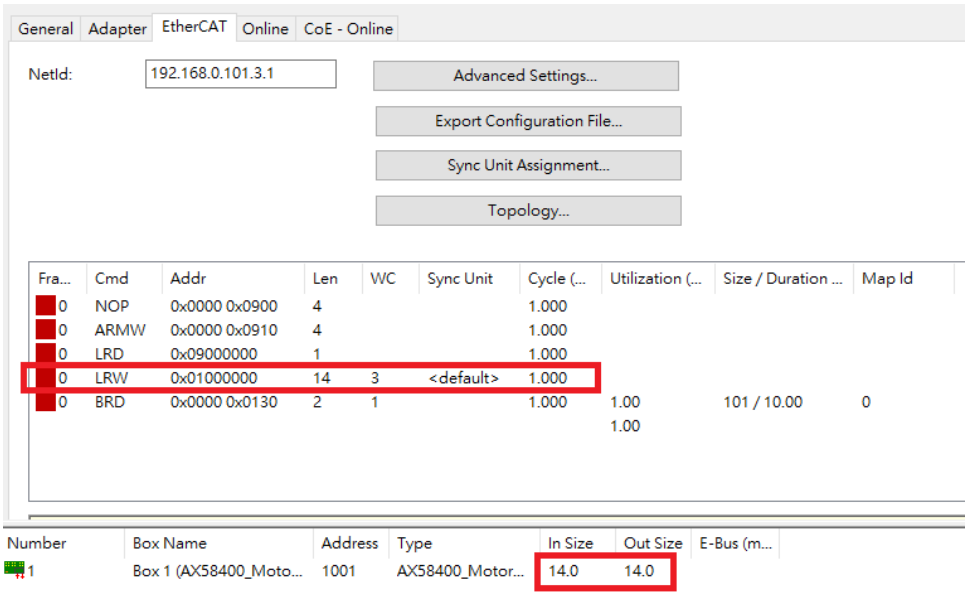


Figure 11-1

Without frame lost and Tx/Rx error at Master side.

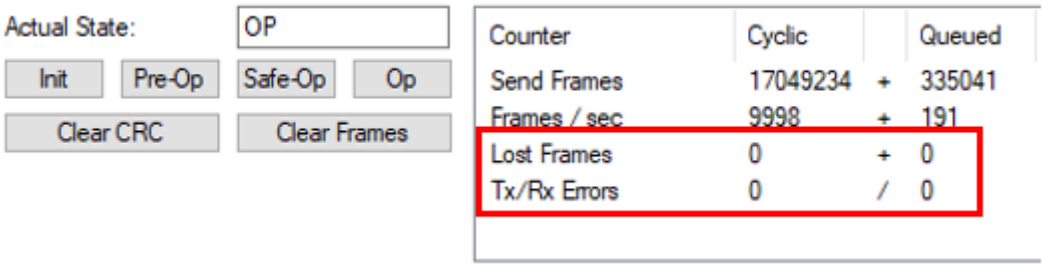


Figure 11-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Also, without detected SM-Event Missed, Cycle Time Too Small and Sync Error at Slave side.

Index	Name	Flags	Value	Unit
1C13:0	SM3 assignment	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C32:01	Synchronization Type	RW	0x0002 (2)	
1C32:02	Cycle Time	RO	0x000186A0 (100000)	
1C32:04	Synchronization Types supported	RO	0x401F (16415)	
1C32:05	Minimum Cycle Time	RO	0x000186A0 (100000)	
1C32:06	Calc and Copy Time	RO	0x00000000 (0)	
1C32:08	Get Cycle Time	RW	0x0000 (0)	
1C32:09	Delay Time	RO	0x00000000 (0)	
1C32:0A	Sync0 Cycle Time	RW	0x000186A0 (100000)	
1C32:0B	SM-Event Missed	RO	0x0000 (0)	
1C32:0C	Cycle Time Too Small	RO	0x0000 (0)	
1C32:20	Sync Error	RO	FALSE	
1C33:0	SM input parameter	RO	> 32 <	
1C33:01	Synchronization Type	RW	0x0003 (3)	
1C33:02	Cycle Time	RO	0x000186A0 (100000)	
1C33:04	Synchronization Types supported	RO	0x401F (16415)	
1C33:05	Minimum Cycle Time	RO	0x000186A0 (100000)	
1C33:06	Calc and Copy Time	RO	0x00000000 (0)	
1C33:08	Get Cycle Time	RW	0x0000 (0)	
1C33:09	Delay Time	RO	0x00000000 (0)	
1C33:0A	Sync0 Cycle Time	RW	0x000186A0 (100000)	
1C33:0B	SM-Event Missed	RO	0x0000 (0)	
1C33:0C	Cycle Time Too Small	RO	0x0000 (0)	
1C33:20	Sync Error	RO	FALSE	

Figure 11-3

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

12. Using the High Speed Clock (HSE)

12-1 Modify RCC Setting

For this reference design, the HSI is used for default clock source, if higher precision clock is needed, please follow the steps to change the clock source to an external crystal (HSE).

Step 1: Double clicks on the file below to open STM32CubeMX.
BSP_ROOT\SampleCode\Ax58400_MotorControl\Ax58400_MotorControl.ioc

Step 2: Select "**Pinout & Configuration > RCC**" and enable "**High Speed Clock (HSE)**" as "**Crystal/Ceramic Resonator**" setting.

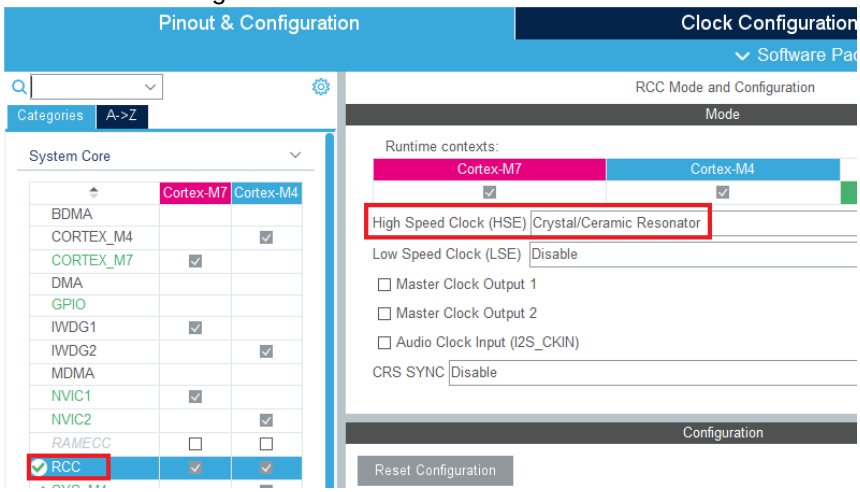


Figure 12-1

Step 3: Select "**Clock Configuration**" page, then modify the setting as highlighted in red as below.

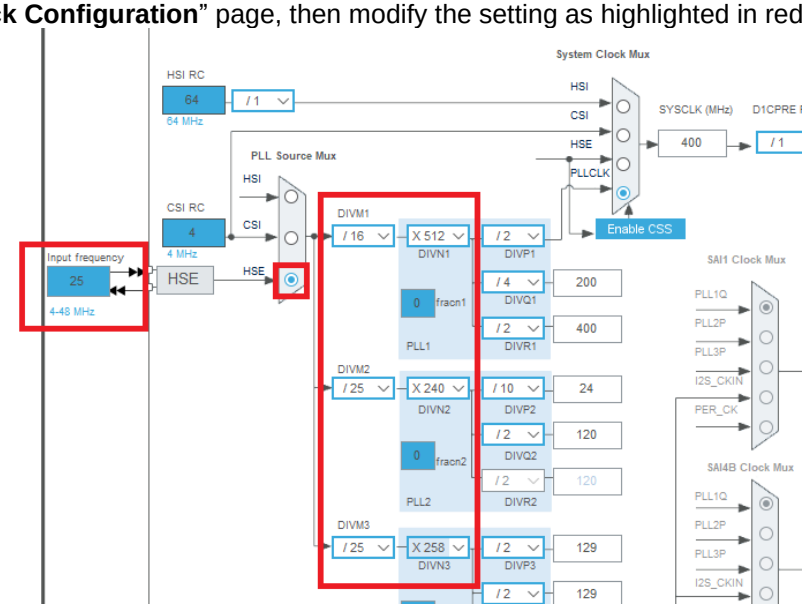


Figure 12-2

Step 4: Re-generate the source code, then re-build the target and upgrade the firmware as mentioned previously.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

13. Firmware Architecture

Below figure shows the simple software architecture of AX58400 Motor Control Reference Design.
All the firmware modules could be classified three parts as below.

- Dual Core shared modules
- CM7 Core modules
- CM4 Core modules

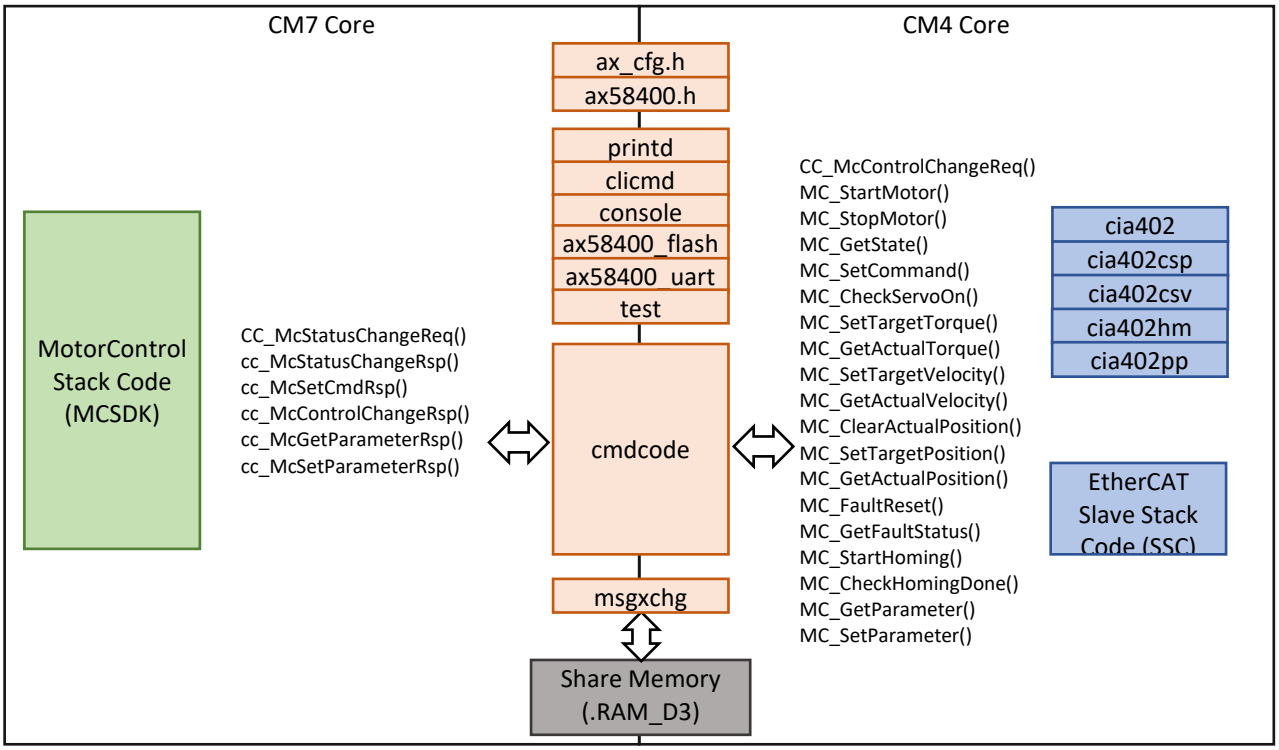


Figure 13-1

13-1 Dual Core Shared Modules

These modules are shared by CM4 and CM7 cores, some of shared modules allow be configured as suitable functions to run on CM4 or CM7 core.

- “ax_cfg.h” and “ax58400.h”
  These modules provide some configurations and allow developer to allocate suitable shared modules or functions for CM4 or CM7 core.

- “printd”, “clicmd”, “console”, “ax58400_uart” and “test”
  These modules provide a serial port (RS232 or USB VCP) for debugging or functional test during the development stage, it’s also useful for troubleshooting & diagnostics.

- “cmdcode” and “msgxchg”
  These modules are important for handling the dual core data exchange, the “msaxchg” module provides a channel configure structure named “MSGXCHG_ChannelConfig_T”, the developer should depend on the application to allocate channel number and TX/RX buffer size,

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

For example, in this reference design, there are three channels will be allocated as below.

- BufOffset field: Indicates the memory offset in Share Memory (.RAM_D3).
- Buf1Size field: Indicates the buffer size in bytes for CM4 to CM7 transmit direction.
- Buf2Size field: Indicates the buffer size in bytes for CM7 to CM4 transmit direction.
- Ctrl field: Specifies that whether or not to enable data integrity check.

Channel Index	Channel Name	BufOffset	Buf1Size	Buf2Size	Ctrl
0	MSGXCHG_MC_PD_CH	0x0000	0x0080	0x0080	0
1	MSGXCHG_MC_CMD_CH	0x0100	0x0080	0x0080	MSGXCHG_CTL_CHKSUM
2	MSGXCHG_MC_CFG_CH	0x0200	0x0080	0x0080	MSGXCHG_CTL_CHKSUM

Table 13-1

- Channel 0: This channel is used to exchange process data between dual cores and without response need from receiver. In this reference design, if there are control data changes in CM4 core, the whole control data will be transmitted to CM7 core, similarly, if there are status data changes in CM7 core, the status data will be transmitted to CM4 core.

- Channel 1: This channel is used to transmit command data from CM4 to CM7 core and with response need from receiver. The CM4 to CM7 buffer address is 0x38000100 with 0x80 byte size and CM7 to CM4 buffer address is 0x38000180, also with 0x80 byte size.

- Channel 2: This channel is used to transmit configure data from CM4 to CM7 core and with response need from receiver. Similarly, the CM4 to Cm7 buffer address is 0x38000200, CM7 to CM4 buffer address is 0x38000280, and two buffers have the same size as 0x80 bytes.

Based on these channels, the “cmdcode” module is responsible for providing upper-layer API functions for application calling. In this reference design, “cmdcode” provided the API functions and dependent channel index are as below.

CM4 to CM7 API Function	CM7 to CM4 API Function	Used Channel Name
CC_McControlChangeReq() cc_McStatusChangeRsp()	cc_McControlChangeRsp() CC_McStatusChangeReq()	MSGXCHG_MC_PD_CH
MC_StartMotor() MC_StopMotor() MC_GetState() MC_SetCommand() MC_CheckServoOn() MC_SetTargetTorque() MC_GetActualTorque() MC_SetTargetVelocity() MC_GetActualVelocity() MC_ClearActualPosition() MC_SetTargetPosition() MC_GetActualPosition() MC_FaultReset() MC_GetFaultStatus() MC_StartHoming() MC_CheckHomingDone()	cc_McSetCmdRsp()	MSGXCHG_MC_CMD_CH
MC_GetParameter() MC_SetParameter()	cc_McGetParameterRsp() cc_McSetParameterRsp()	MSGXCHG_MC_CFG_CH

Table 13-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

By the way, all the allocated channel buffers are mapped to the Share Memory (.RAM_D3), which be declared in the files as below, the developer can adjust the Share Memory size properly according to the application.
For Keil_MDK IDE:
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\stm32h755xx_flash_CM4.sct
BSP_ROOT\SampleCode\Ax58400_MotorControl\MDK-ARM\stm32h755xx_flash_CM7.sct

For STM32CubeMX IDE:
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\CM4\ STM32H755ZITX_FLASH.sct
BSP_ROOT\SampleCode\Ax58400_MotorControl\STM32CubeIDE\CM7\ STM32H755ZITX_FLASH.sct

13-2 CM7 Core Modules

The CM7 core is responsible for executing motor control stack and receiving process/control/command data from CM4 core. The related modules are listed in below.

- Motor Control Stack (MCSDK)
  This stack is generated by ST MotorControl workbench + STM32CubeMX.

13-3 CM4 Core Modules

The CM4 core is mainly executing SSC stack for handling EtherCAT communication and CiA402 device profiler. It includes the modules as below.

- cia402, cia402csp, cia402csv, cia402hm and cia402pp
  Every supported mode in CiA402 has been implemented as a single module, common functions and definitions are implemented in cia402 module.

- EtherCAT Slave Stack Code (SSC)
  This stack is generated by ETG's SSC tool.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

14. Tuning the PID Loop Gains with TwinCAT

Here will show you how to add PID gain objects into object dictionary and tuning PID gain over EtherCAT. For most of applications, you may need to extend your own objects, please follow the procedure below to do this.

14-1 Add New Objects into Dictionary in CM4

Step 1: Open the "**AX58400_MotorControl.xlsx**" file in below path.
BSP_ROOT\SampleCode\Ax58400_MotorControl\For_SSC_Tool\Import\Configuration\files

Step 2: In Vendor specify area (0x2000~0x3FFF), add PID gain objects for position, speed and torque loops as below.

//0x2000 - 0x3FFF	Axis objects (0x2000 - 0x3FFF)							
0x3000			INT16	PosKpGain	rw	CoeRead	CoeWrite	Position loop proportional gain (Unit: digit)
0x3001			INT16	PosKpDivisor	ro	CoeRead	CoeWrite	Position loop proportional divisor (Unit: digit)
0x3002			INT16	PosKiGain	rw	CoeRead	CoeWrite	Position loop integral gain (Unit: digit)
0x3003			INT16	PosKiDivisor	ro	CoeRead	CoeWrite	Position loop integral divisor (Unit: digit)
0x3004			INT16	PosKdGain	rw	CoeRead	CoeWrite	Position loop derivative gain (Unit: digit)
0x3005			INT16	PosKdDivisor	ro	CoeRead	CoeWrite	Position loop derivative divisor (Unit: digit)
0x3010			INT16	SpdKpGain	rw	CoeRead	CoeWrite	Speed loop proportional gain (Unit: digit)
0x3011			INT16	SpdKpDivisor	ro	CoeRead	CoeWrite	Speed loop proportional divisor (Unit: digit)
0x3012			INT16	SpdKiGain	rw	CoeRead	CoeWrite	Speed loop integral gain (Unit: digit)
0x3013			INT16	SpdKiDivisor	ro	CoeRead	CoeWrite	Speed loop integral divisor (Unit: digit)
0x3014			INT16	SpdKdGain	rw	CoeRead	CoeWrite	Speed loop derivative gain (Unit: digit)
0x3015			INT16	SpdKdDivisor	ro	CoeRead	CoeWrite	Speed loop derivative divisor (Unit: digit)
0x3020			INT16	TrqKpGain	rw	CoeRead	CoeWrite	Torque loop proportional gain (Unit: digit)
0x3021			INT16	TrqKpDivisor	ro	CoeRead	CoeWrite	Torque loop proportional divisor (Unit: digit)
0x3022			INT16	TrqKiGain	rw	CoeRead	CoeWrite	Torque loop integral gain (Unit: digit)
0x3023			INT16	TrqKiDivisor	ro	CoeRead	CoeWrite	Torque loop integral divisor (Unit: digit)
0x3024			INT16	TrqKdGain	rw	CoeRead	CoeWrite	Torque loop derivative gain (Unit: digit)
0x3025			INT16	TrqKdDivisor	ro	CoeRead	CoeWrite	Torque loop derivative divisor (Unit: digit)

Figure 14-1

Step 3: Re-generate the SSC source code, for more detail, please refer to Chapter 5-1.

Step 4: Update the new ESI file into TwinCAT data base.
Normally, the ESI data base path is "**C:\TwinCAT\3.1\Config\Io\EtherCAT**".

14-2 Link New Objects to Data Structure in CM4

Step 1: Add new members in CiA402_OBJ_T data structure, notes each data type must be same as the new objects on "**AX58400_MotorControl.xlsx**", like this.

AX58400_MotorControl.h

```
typedef struct STRUCT_PACKED_START
{
    ...
    /* Vendor Specified Objects */
    INT16    PosKpGain0x3000;
    INT16    PosKpDivisor0x3001;
    INT16    PosKiGain0x3002;
    INT16    PosKiDivisor0x3003;
    INT16    PosKdGain0x3004;
    INT16    PosKdDivisor0x3005;
    INT16    SpdKpGain0x3010;
    INT16    SpdKpDivisor0x3011;
    INT16    SpdKiGain0x3012;
    INT16    SpdKiDivisor0x3013;
    INT16    SpdKdGain0x3014;
    INT16    SpdKdDivisor0x3015;
    INT16    TrqKpGain0x3020;
    INT16    TrqKpDivisor0x3021;
    INT16    TrqKiGain0x3022;
    INT16    TrqKiDivisor0x3023;
    INT16    TrqKdGain0x3024;
    INT16    TrqKdDivisor0x3025;
}STRUCT_PACKED_END
CiA402_OBJ_T;
```

Figure 14-2

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Step 2: Links new object dictionary (OD) to new members of CiA402_OBJ_T structure in CiA402_Init () function, like this.

AX58400_MotorControl.c

```
321     /* Vendor Specified Objects */
322     case 0x3000:
323       HMEMCPY((UINT8*)&(pCiA402->PosKpGain0x3000), (UINT8*)&(PosKpGain0x3000), SIZEOF(PosKpGain0x3000));
324       pCiA402->PosKpGain0x3000 = PosKpGain0x3000;
325       pDiCEntry->pVarPtr = &(pCiA402->PosKpGain0x3000);
326       break;
327
328     case 0x3001:
329       HMEMCPY((UINT8*)&(pCiA402->PosKpDivisor0x3001), (UINT8*)&(PosKpDivisor0x3001), SIZEOF(PosKpDivisor0x3001));
330       pCiA402->PosKpDivisor0x3001 = PosKpDivisor0x3001;
331       pDiCEntry->pVarPtr = &(pCiA402->PosKpDivisor0x3001);
332       break;
333
334     case 0x3002:
335       HMEMCPY((UINT8*)&(pCiA402->PosKiGain0x3002), (UINT8*)&(PosKiGain0x3002), SIZEOF(PosKiGain0x3002));
336       pCiA402->PosKiGain0x3002 = PosKiGain0x3002;
337       pDiCEntry->pVarPtr = &(pCiA402->PosKiGain0x3002);
338       break;
339
340     case 0x3003:
341       HMEMCPY((UINT8*)&(pCiA402->PosKiDivisor0x3003), (UINT8*)&(PosKiDivisor0x3003), SIZEOF(PosKiDivisor0x3003));
342       pCiA402->PosKiDivisor0x3003 = PosKiDivisor0x3003;
343       pDiCEntry->pVarPtr = &(pCiA402->PosKiDivisor0x3003);
344       break;
```

Figure 14-3

14-3 Add Handling Functions According to New Object Indexes in CM7

Step 1: Add new objects index for write operations as below.

cmdcode.c

```
433 static s32 cc_McSetParameterRsp(MSGXCHG_Channel_T *pCh, u8 *pMsg, u32 msgLen)
434 {
      ● ● ●
448
449   switch (pParm->Index)
450   {
451   /* Set position PID */
452   case 0x3000:
453     memcpy((u8*)&(pPidPos->hKpGain), pParm->Data, sizeof(pPidPos->hKpGain));
454     break;
455   case 0x3001:
456     memcpy((u8*)&(pPidPos->hKpDivisor), pParm->Data, sizeof(pPidPos->hKpDivisor));
457     break;
458   case 0x3002:
459     memcpy((u8*)&(pPidPos->hKiGain), pParm->Data, sizeof(pPidPos->hKiGain));
460     break;
```

Figure 14-4

Step 2: Add new objects index for read operations as below.

cmdcode.c

```
319 static s32 cc_McGetParameterRsp(MSGXCHG_Channel_T *pCh, u8 *pMsg, u32 msgLen)
320 {
      ● ● ●
334
335   switch (pParm->Index)
336   {
337   /* Response position PID */
338   case 0x3000:
339     memcpy(pFrmRsp->Data, (u8*)&(pPidPos->hKpGain), sizeof(pPidPos->hKpGain));
340     len += sizeof(pPidPos->hKpGain);
341     break;
342   case 0x3001:
343     memcpy(pFrmRsp->Data, (u8*)&(pPidPos->hKpDivisor), sizeof(pPidPos->hKpDivisor));
344     len += sizeof(pPidPos->hKpDivisor);
345     break;
346   case 0x3002:
347     memcpy(pFrmRsp->Data, (u8*)&(pPidPos->hKiGain), sizeof(pPidPos->hKiGain));
348     len += sizeof(pPidPos->hKiGain);
349     break;
```

Figure 14-5

Step 3: Re-build the project and update new firmware into AX58400.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

14-4 Create Scope Project in TwinCAT

In order to do manual tuning of the control loop gain, we need to observe the position and speed dynamic response, so here must create a scope project to plot some important curves.
Please follows the steps below to do this.

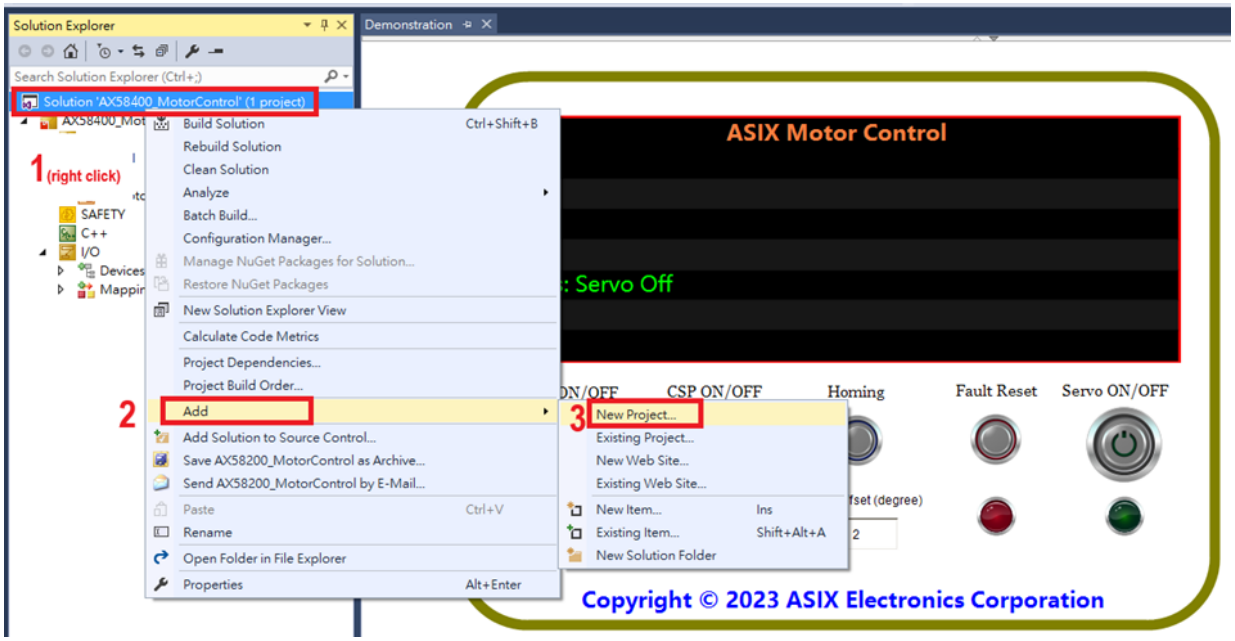


Figure 14-6

Here selects the “Scope YT Project” then click “OK” button to complete the scope creation.

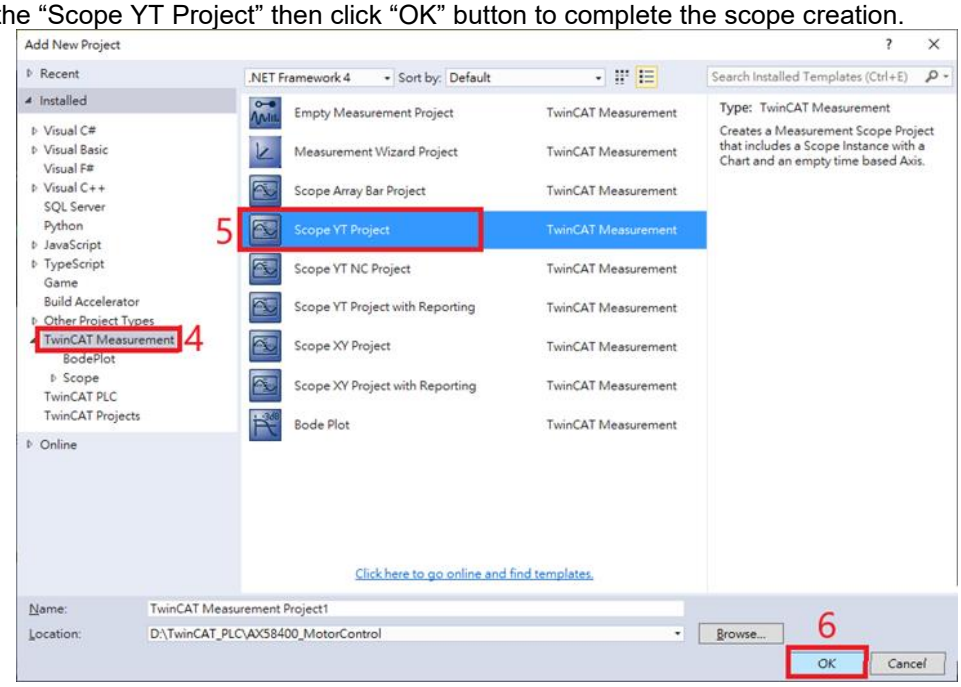


Figure 14-7

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Immediately, need to add some process data that we want observing, that includes poTargetPosition, piPositionActualValue, poTargetSpeed and piSpeedActualValue.
Please follows the steps below to do this.

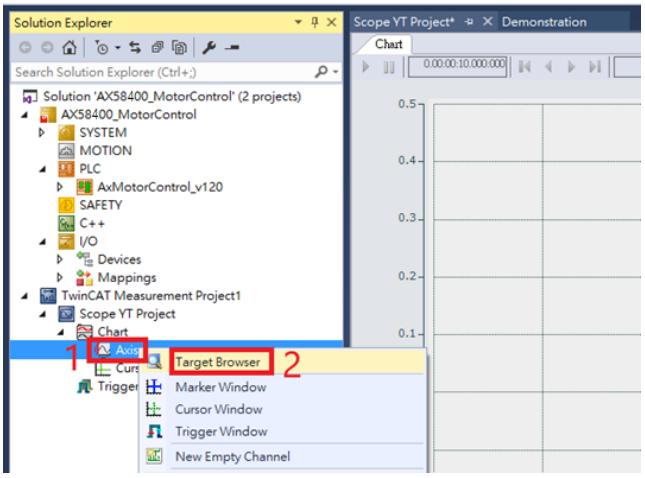


Figure 14-8

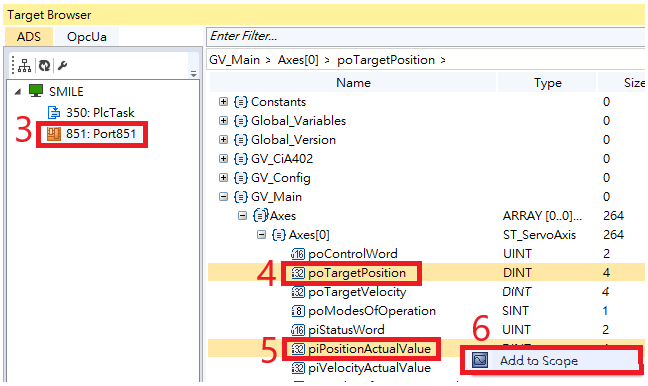


Figure 14-9

Here we create a new plot axis to display speed response curve.

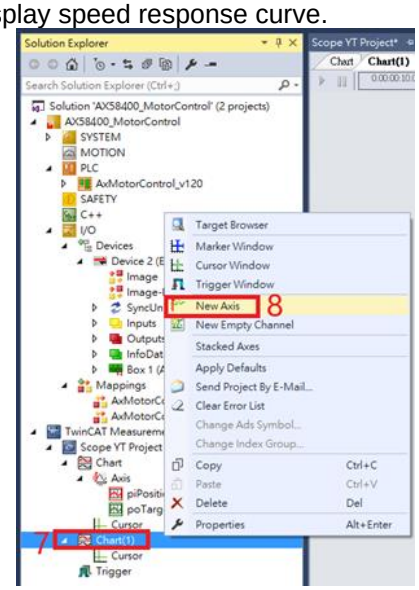


Figure 14-10

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

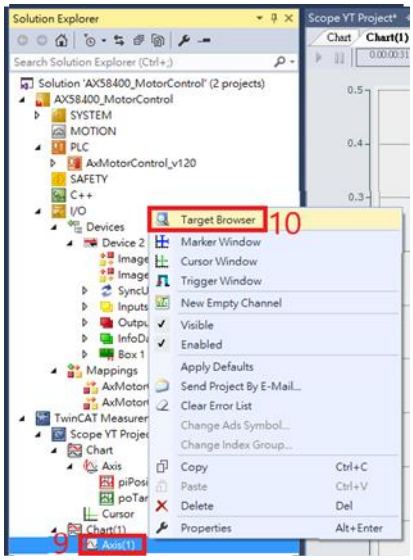


Figure 14-11

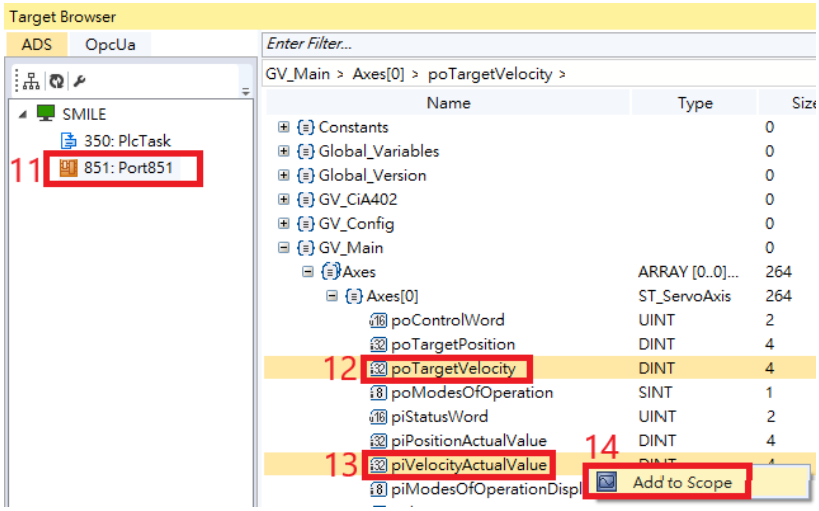


Figure 14-12

So far, the scope is ready to record the process data, we can start to tuning the position and speed loop gains.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

14-5 Position Loop Gain Value Manual Tuning in TwinCAT

Let TwinCAT entered running mode, then press “Servo ON/OFF” button to enter CiA402 operation state. Turn on “CSP ON/OFF” switch to do position loop gain tuning.

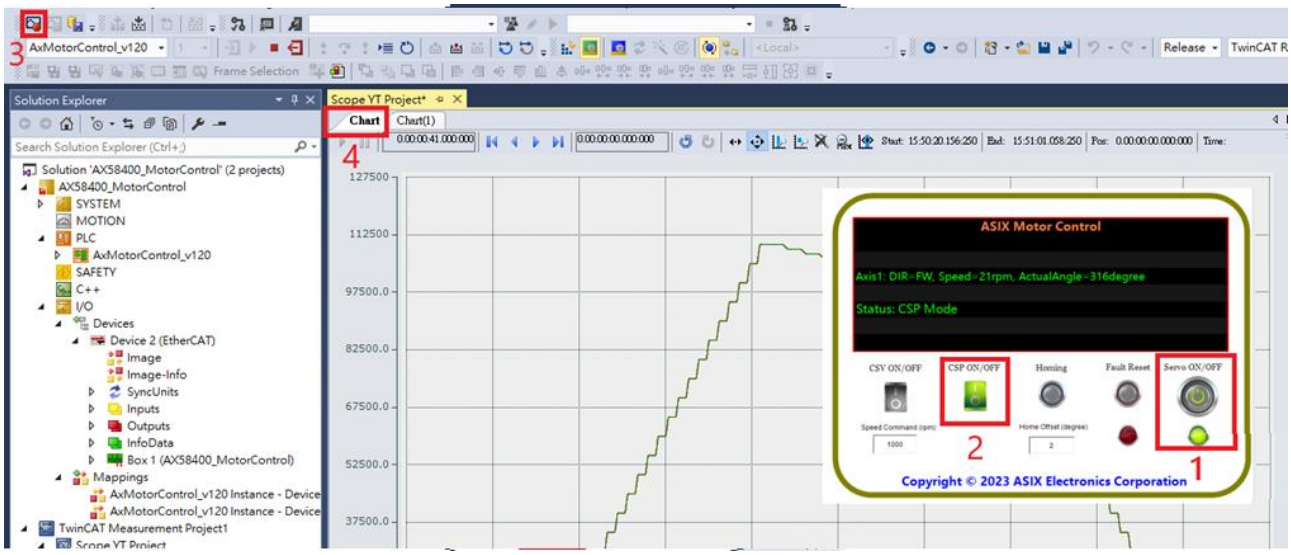


Figure 14-13

If you want to adjust the gain values, please switch back to “CoE-Online” function tab as below and write PosKpGain, PosKiGain and PosKdGain via SDO transfer directly.

Please note that, the gain value modification over here is not permanent, if the gain tuning is finalized, please update these gain values in “drive_parameters.h” file, then re-build and upgrade the firmware.

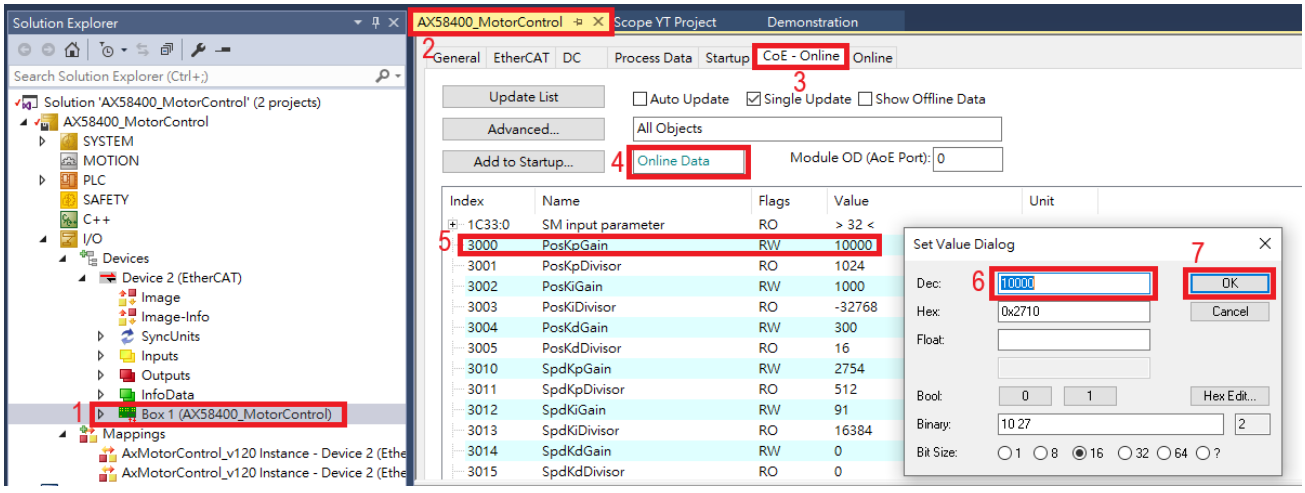


Figure 14-14

If the tuning is complete, please turn off the “CSP ON/OFF” switch and wait for the motor stop.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

14-6 Speed Loop Gain Value Manual Tuning in TwinCAT

Want to do speed loop gain tuning, please turn on “CSV ON/OFF” switch and change the speed command, then record the curves to observe its overshot, settling time or following error.
Similarly, switch back to “CoE-Online” function tab to adjust SpdKpGain and SpdKiGain until satisfying the application requirement.

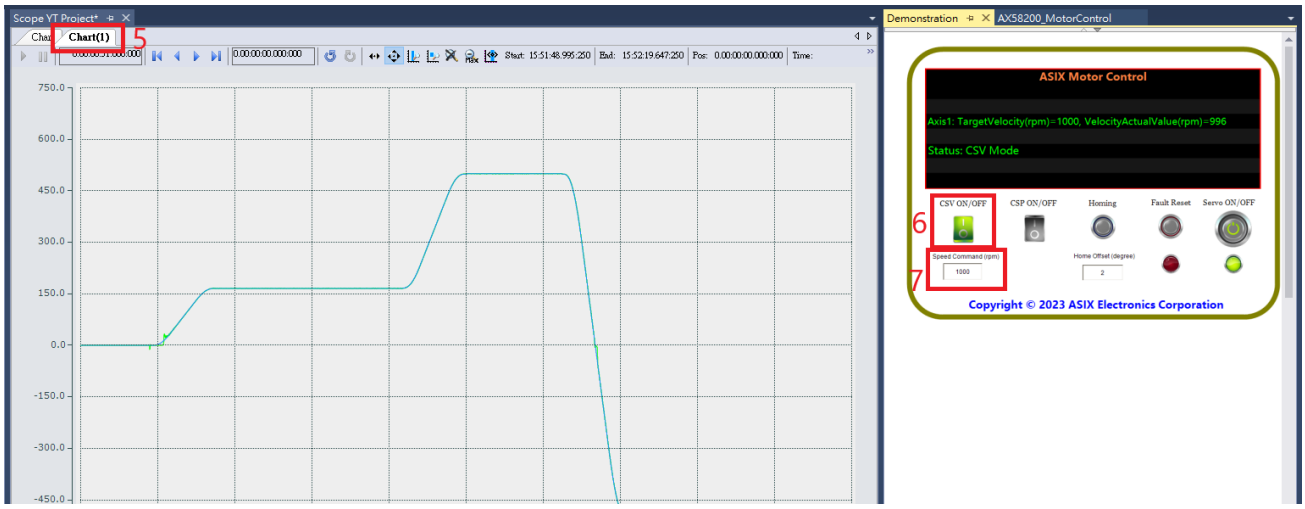


Figure 14-15

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Evaluation Board License Agreement

By using this evaluation board or kit (together with all associated software, hardware and documentation provided by ASIX, collectively, the "Evaluation Board"), you ("You") agree to be bound by the terms and conditions of this Evaluation Board License Agreement ("Agreement "). Do not use the evaluation board until you have read and agreed to this agreement. Your use of the Evaluation Board constitutes your acceptance of this Agreement.

License:
ASIX Electronics (“ASIX”) grants you the right to use the accompanying evaluation board, which provides limited functionality, for the sole purpose of evaluating and testing ASIX products for your evaluation and testing purposes in a research and development environment. Under no circumstances should the evaluation board be assembled, directly or indirectly, as part of any of your products, as it has been developed for evaluation purposes only and has no direct functionality and is not a finished product.

Notice:
(1) The description of hardware circuit, software tools and other related information in this document are only used to illustrate the operation of application examples for ASIX Electronic products. You are solely responsible for the use of such circuits, software and information in the design of your equipment. ASIX Electronics is not responsible for any losses arising out of the use of the circuit, software or information by you or a third party.
(2) ASIX Electronics has taken reasonable care in describing the information in this document, but ASIX Electronics does not warrant that such information is error-free. ASIX Electronics disclaims all liability for any damages you may suffer as a result of errors or omissions in the information contained herein.
(3) ASIX Electronics shall not be liable for infringement of patents, copyrights or other intellectual property rights of third parties due to the use of ASIX Electronics products or the technical information described in this document. No license, express or implied, or otherwise, is granted under any patent, copyright or other intellectual property rights of ASIX Electronics or others.
(4) You may not alter, modify, copy or otherwise misappropriate any ASIX electronic product, whether in whole or in part. ASIX Electronics shall not be liable for any loss incurred by you or a third party as a result of altering, modifying, copying or otherwise misappropriating ASIX Electronics products.
(5) You should use the ASIX Electronic products described in this document within the range specified by ASIX Electronic, especially the maximum rating, operating power supply voltage range, mobile power supply voltage range, heat radiation characteristics, installation and other product characteristics. ASIX Electronics accepts no liability for malfunctions or damages resulting from use of ASIX Electronics products outside of such designations.
(6) When using ASIX electronic products, please comply with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including but not limited to the EU RoHS Directive. ASIX Electronics disclaims all liability for damage or loss resulting from your failure to comply with applicable laws and regulations.
(7) The buyer or distributor of ASIX electronic products is responsible for distributing, disposing of or otherwise placing the product to a third party, and notifying the third party of the contents and conditions specified in this document in advance, ASIX Electronics shall not be liable for any loss suffered as a result of authorized use of ASIX Electronics products.
(8) This document may not be reproduced or reproduced in any form, in whole or in part, without the prior written consent of ASIX Electronics.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

Copyright © 2021-2024 ASIX Electronics Corporation. All rights reserved.

DISCLAIMER

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of ASIX. ASIX may make changes to the product specifications and descriptions in this document at any time, without notice.

ASIX provides this document “as is” without warranty of any kind, either expressed or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.

Designers must not rely on the absence or characteristics of any features or registers marked “reserved”, “undefined” or “NC”. ASIX reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Always contact ASIX to get the latest document before starting a design of ASIX products.

TRADEMARKS

ASIX, the ASIX logo are registered trademarks of ASIX Electronics Corporation. All other trademarks are the property of their respective owners.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.

ASIX Electronics Corporation

4F, No.8, Hsin Ann RD., Hsinchu Science Park,
Hsinchu, Taiwan, R.O.C.

TEL: +886-3-5799500
FAX: +886-3-5799558

Email: support@asix.com.tw
Web: https://www.asix.com.tw

Copyright (C) 2021-2024 Reserved by ASIX Electronics Corporation, All Rights Reserved.