

AX58100 GPIO / AIO User Guide

Revision 1.00
January 21th, 2025

Revision History

Revision	Date	Description
1.00	2025/01/21	Initial release

CONTENT

1. Introduction	6
2. Requirements	7
3. Board Description	10
3-1 AX58100-EVB Evaluation Board	10
3-2 NUCLEO-F303RE	11
4. Environment Setup.....	12
4-1 Hardware Connectivity	12
5. Building Reference Design Begin with STM32CubeMX	13
5-1 Create New Project in STM32CubeMX	13
5-2 Modify Peripheral Settings in STM32CubeMX	14
5-3 Generate Source Code in STM32CubeMX and MDK-ARM	18
5-4 Source Code Adjustment	20
5-5 Generate EtherCAT Slave Stack Code	25
5-6 IDE Setting Adjustment	26
6. Build Up and Download the Firmware	32
6-1 Build up Firmware Binary	32
6-2 Download Bootloader Firmware	33
6-3 Download GPIO / AIO Firmware	33
6-4 Check Firmware Status	35
7. Basic Operation of TwinCAT	36
8. PLC Application.....	36
9. Flash Memory Allocation and FoE Upgrading	37
9-1 Firmware Upgrade.....	38
10. Virtual ADIO Mode	39
10-1 Console Command Usage	39
11. Switch ID Description	40
12. Shortest EtherCAT Cycle Evaluation	46

List of Figure

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

Figure 6-7	35
Figure 6-8	35
Figure 9-1	37
Figure 9-2	37
Figure 9-3	38
Figure 9-4	38
Figure 9-5	38
Figure 11-1	40
Figure 11-2	40
Figure 11-3	41
Figure 11-4	41
Figure 11-5	42
Figure 11-6	42
Figure 11-7	42
Figure 11-8	43
Figure 11-9	43
Figure 11-10	43
Figure 11-11	44
Figure 11-12	44
Figure 11-13	44
Figure 11-14	45
Figure 11-15	45
Figure 12-1	46
Figure 12-2	46
Figure 12-3	47

List of Table

Table 10-1.....	39
-----------------	----

1. Introduction

This document introduces how to setup entire environment for GPIO / AIO application on AX58100 evaluation platform (NUCLEO-F303RE + AX58100-EVB-SSPDI boards). The demo application of EtherCAT master side is developed using PLC (Programmable Logic Controller) language running on 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 is developed using C language through ARM KEIL MDK (Microcontroller Development Kit).

The firmware mainly handles the tasks below:

- BECKHOFF SSC (Slave Stack Code)
- Hardware access of SSC through SPI interface at MCU side
- Simple object form for CIA-401 digital/analog IO
- Peripherals (GPIO / ADC / DAC) accessed by EtherCAT PD (Process Data)

The PLC at EtherCAT master side mainly handles the tasks below:

- PD mapping / handling by PLC for GPIO / AIO
- A demo GUI

2. Requirements

Before starting to setup the environment, the user should prepare some software and hardware. Of course, they are all running based on general desktop PC with windows 10 / 11 operation systems. The software and hardware requirements are listed 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]

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



Figure 2-2

- 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)



Figure 2-3

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



Figure 2-4

- STM32CubeProgrammer
You can download the STM32CubeProgrammer [here](#).



Figure 2-5

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



Figure 2-6

[Hardware]

- ST NUCLEO-F303RE Control Board x 1 PCS

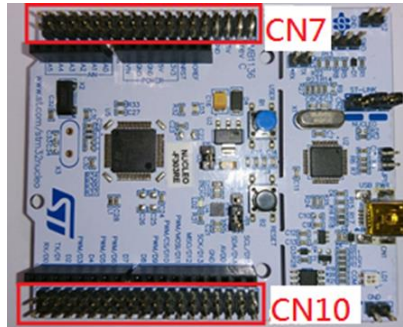


Figure 2-7

- AX58100-EVB-SSPDI-1 V1.0 EVB x 1 PCS
For more details, please refer to the link below and the following chapter.
ASIX official link: <https://www.asix.com.tw/>

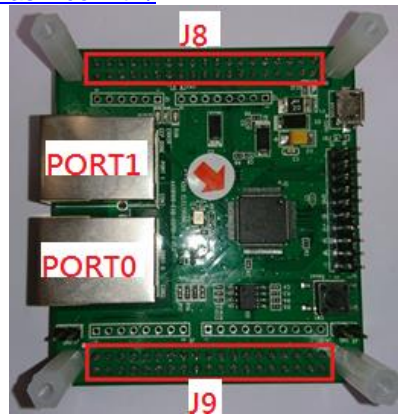


Figure 2-8

3. Board Description

3-1 AX58100-EVB Evaluation Board

AX58100-EVB is an EtherCAT Slave Controller (ESC) Evaluation Board with AX58100 IC, please refer to ASIX official website as below for more detail.

<https://www.asix.com.tw/en/product/IndustrialEthernet/EtherCAT/AX58100>

The AX58100-EVB-SSPDI is an evaluation board designed for the AX58100 IC and includes the following components:

- J8 – Pins that provides GPIO or additional connections, typically interfacing with other devices or expansion boards.
- J9 – Pins that provides GPIO or additional connections, like J8.
- ESC Port0 – Port0 of the ESC
- ESC Port1 – Port1 of the ESC
- Reset – A reset button of AX58100

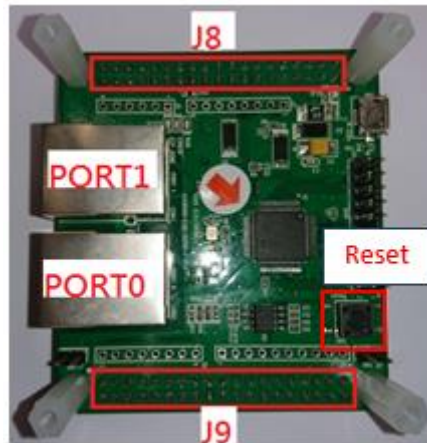


Figure 3-1

3-2 NUCLEO-F303RE

The STM32 Nucleo-64 board offers an affordable platform for prototyping with STM32 microcontrollers, combining performance and low power. It supports ARDUINO® Uno V3 and ST morpho headers for expansion and includes an integrated ST-LINK debugger/programmer. Free software libraries and examples are available via the STM32Cube MCU Package. For more details, visit the official product page: <https://www.st.com/en/evaluation-tools/nucleo-f303re.html>

The NUCLEO-F303RE functions as the control board for the AX58100 GPIO / AIO application and provides the following components:

- USB – ST-LINK USB mini-B connector
- CN7 – ST morpho connector
- CN10 – ST morpho connector
- USER – User button
- Reset – Reset button

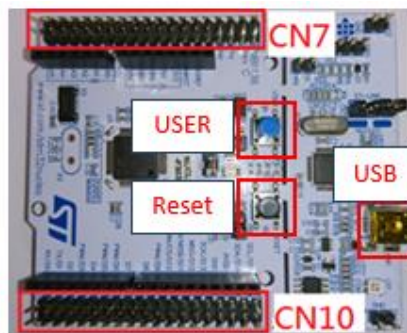
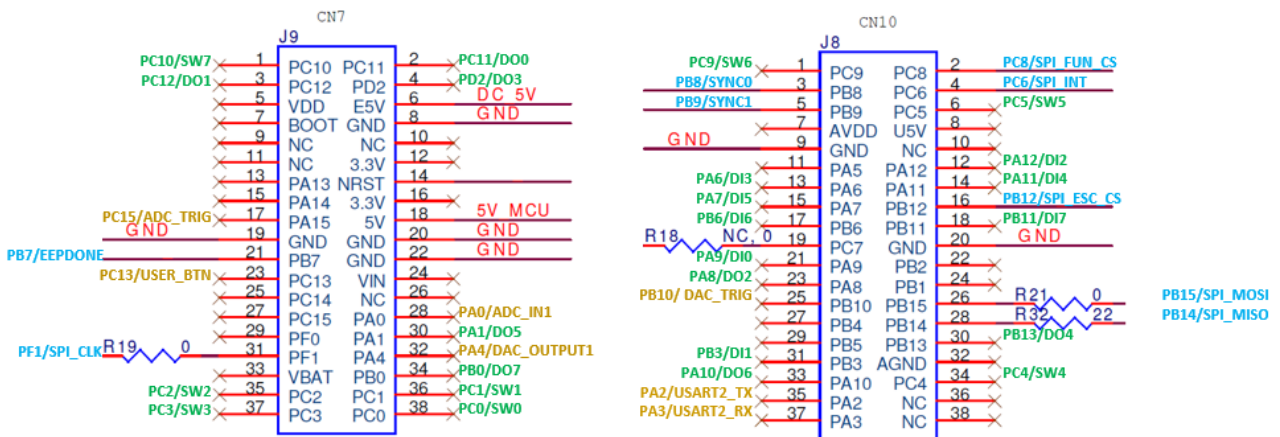


Figure 3-2

Pinout



4. Environment Setup

4-1 Hardware Connectivity

Please refer to the diagram below to see the combination of the NUCLEO-F303RE and AX58100-EVB-SSPDI.

J8 (AX58100-EVB-SSPDI) connects to CN10 (NUCLEO-F303RE), and J9 (AX58100-EVB-SSPDI) connects to CN7 (NUCLEO-F303RE).

Connect the “EtherCAT Port 0” and “USB Port” to your host:

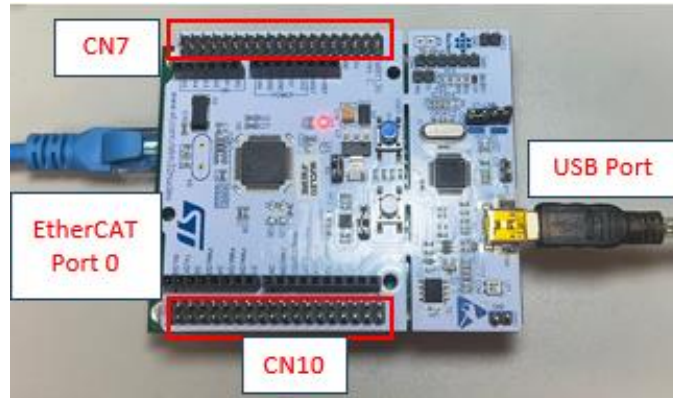


Figure 4-1

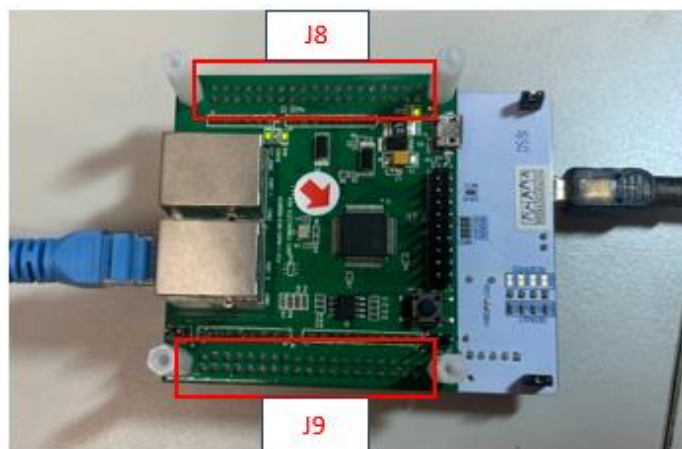


Figure 4-2

5. Building Reference Design Begin with STM32CubeMX

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

Notice: If you just need to generate the SSC code only, please refer to section 5-5 directly.

5-1 Create New Project in STM32CubeMX

Step 1: Execute the STM32CubeMX, then click “ACCESS TO MCU SELECTOR” on main page.

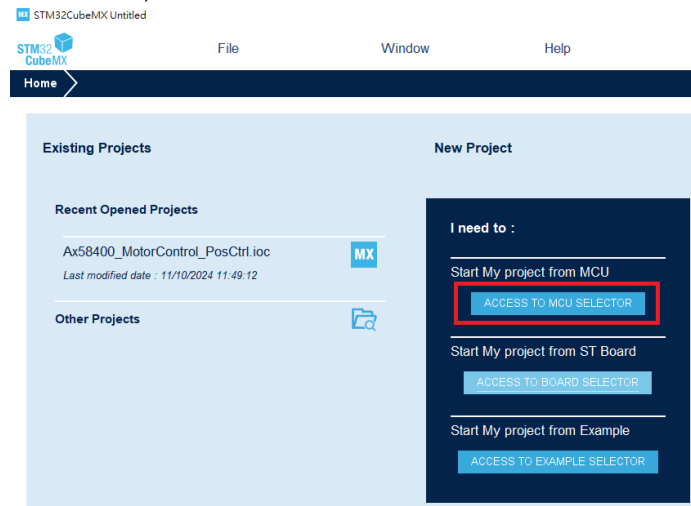


Figure 5-1

Step 2: Searching part number “STM32F303RET6”, then select it on “MCUs/MPU List” and press “Start Project” to create project.



Figure 5-2

Notice: In order to keep the corrected reference path in BSP, please save the project folder in the path below.
BSP_ROOT\SampleCode

5-2 Modify Peripheral Settings in STM32CubeMX

Step 1: Disable the “Pinout & Configuration > System Core > RCC > High Speed Clock (HSE)” setting as the figure below.

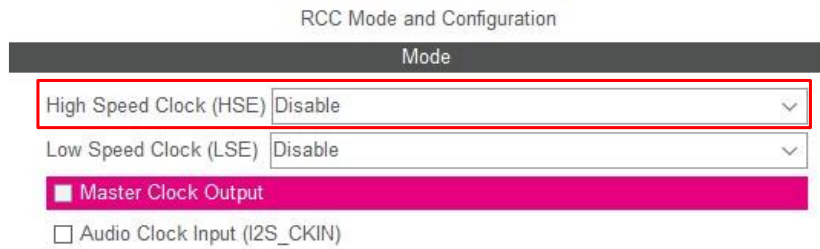
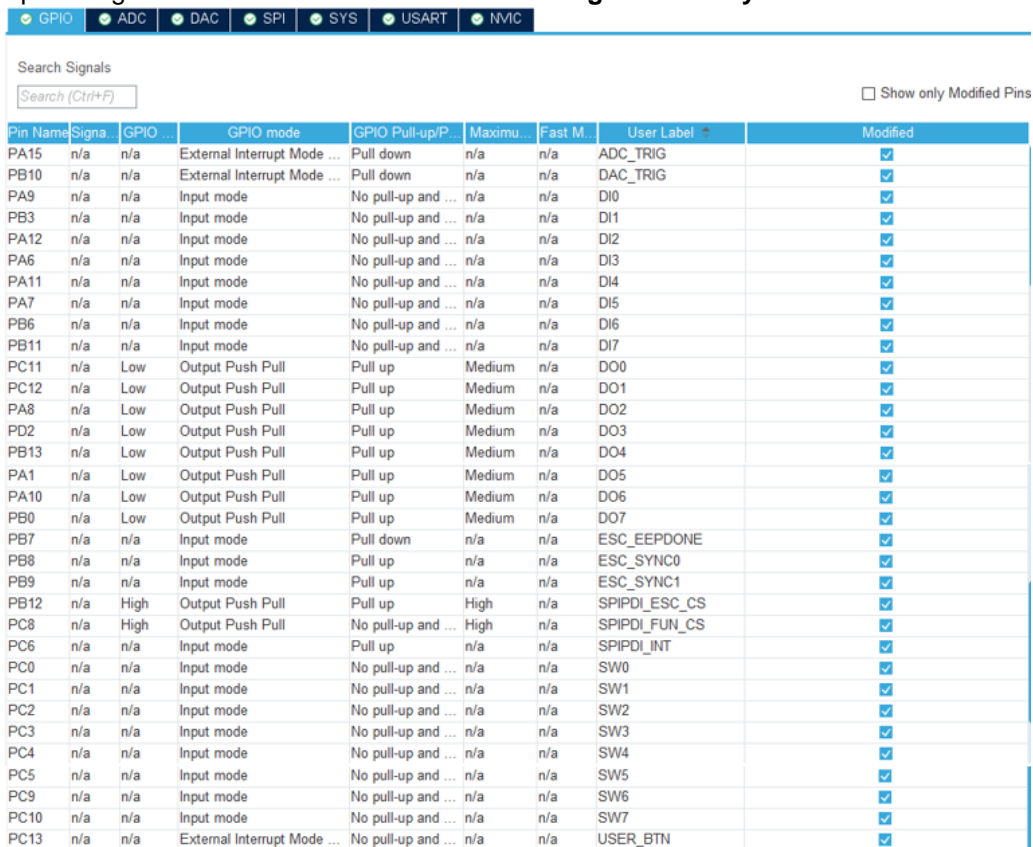


Figure 5-3

Step 2: Add pin assignment as the below in “Pinout & Configuration > System Core > GPIO > GPIO”.



Pin Name	Signal	GPIO	GPIO mode	GPIO Pull-up/P	Maximum...	Fast M.	User Label	Modified
PA15	n/a	n/a	External Interrupt Mode ...	Pull down	n/a	n/a	ADC_TRIG	<input checked="" type="checkbox"/>
PB10	n/a	n/a	External Interrupt Mode ...	Pull down	n/a	n/a	DAC_TRIG	<input checked="" type="checkbox"/>
PA9	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI0	<input checked="" type="checkbox"/>
PB3	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI1	<input checked="" type="checkbox"/>
PA12	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI2	<input checked="" type="checkbox"/>
PA6	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI3	<input checked="" type="checkbox"/>
PA11	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI4	<input checked="" type="checkbox"/>
PA7	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI5	<input checked="" type="checkbox"/>
PB6	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI6	<input checked="" type="checkbox"/>
PB11	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	DI7	<input checked="" type="checkbox"/>
PC11	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO0	<input checked="" type="checkbox"/>
PC12	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO1	<input checked="" type="checkbox"/>
PA8	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO2	<input checked="" type="checkbox"/>
PD2	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO3	<input checked="" type="checkbox"/>
PB13	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO4	<input checked="" type="checkbox"/>
PA1	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO5	<input checked="" type="checkbox"/>
PA10	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO6	<input checked="" type="checkbox"/>
PB0	n/a	Low	Output Push Pull	Pull up	Medium	n/a	DO7	<input checked="" type="checkbox"/>
PB7	n/a	n/a	Input mode	Pull down	n/a	n/a	ESC_EEPCONE	<input checked="" type="checkbox"/>
PB8	n/a	n/a	Input mode	Pull up	n/a	n/a	ESC_SYNC0	<input checked="" type="checkbox"/>
PB9	n/a	n/a	Input mode	Pull up	n/a	n/a	ESC_SYNC1	<input checked="" type="checkbox"/>
PB12	n/a	High	Output Push Pull	Pull up	High	n/a	SPIPDI_ESC_CS	<input checked="" type="checkbox"/>
PC8	n/a	High	Output Push Pull	No pull-up and ...	High	n/a	SPIPDI_FUN_CS	<input checked="" type="checkbox"/>
PC6	n/a	n/a	Input mode	Pull up	n/a	n/a	SPIPDI_INT	<input checked="" type="checkbox"/>
PC0	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW0	<input checked="" type="checkbox"/>
PC1	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW1	<input checked="" type="checkbox"/>
PC2	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW2	<input checked="" type="checkbox"/>
PC3	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW3	<input checked="" type="checkbox"/>
PC4	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW4	<input checked="" type="checkbox"/>
PC5	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW5	<input checked="" type="checkbox"/>
PC9	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW6	<input checked="" type="checkbox"/>
PC10	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	SW7	<input checked="" type="checkbox"/>
PC13	n/a	n/a	External Interrupt Mode ...	No pull-up and ...	n/a	n/a	USER_BTN	<input checked="" type="checkbox"/>

Figure 5-4

Step 3: Add pin assignment for ADC, DAC, SPI2, USART2, SYS as below.

Pin Na...	Signal on Pin	GPI...	GPIO mode	GPIO Pull...	Maxi...	Fast ...	User Label	Modified
PA0	ADC1_IN1	n/a	Analog m...	No pull-up ...	n/a	n/a		<input type="checkbox"/>
PA2	USART2_TX	n/a	Alternate ...	No pull-up ...	High	n/a		<input type="checkbox"/>
PA3	USART2_RX	n/a	Alternate ...	No pull-up ...	High	n/a		<input type="checkbox"/>
PA4	DAC1_OUT1	n/a	Analog m...	No pull-up ...	n/a	n/a		<input type="checkbox"/>
PA13	SYS_JTMS-SWDIO	n/a	n/a	n/a	n/a	n/a		<input type="checkbox"/>
PA14	SYS_JTCK-SWCLK	n/a	n/a	n/a	n/a	n/a		<input type="checkbox"/>
PB14	SPI2_MISO	n/a	Alternate ...	No pull-up ...	High	n/a	SPIPDI_MISO	<input checked="" type="checkbox"/>
PB15	SPI2_MOSI	n/a	Alternate ...	Pull up	High	n/a	SPIPDI_MOSI	<input checked="" type="checkbox"/>
PF1-OSC...	SPI2_SCK	n/a	Alternate ...	Pull up	High	n/a	SPIPDI_CLK	<input checked="" type="checkbox"/>

Figure 5-5

Step 4: Enable ADC1 as the settings highlighted in red.

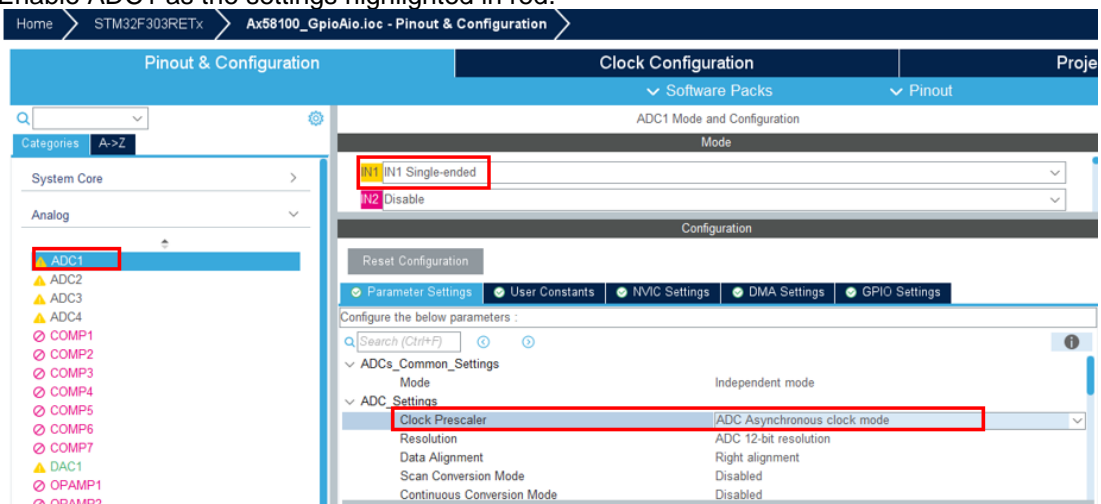


Figure 5-6

Step 5: Enable DAC with the setting highlighted in red.

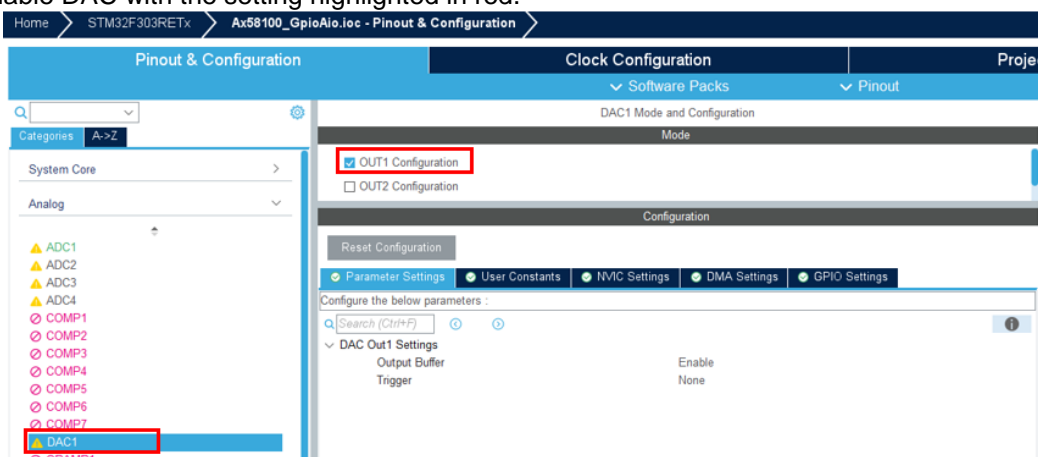


Figure 5-7

Step 6: Activated TIM6 & TIM7 as the setting below.

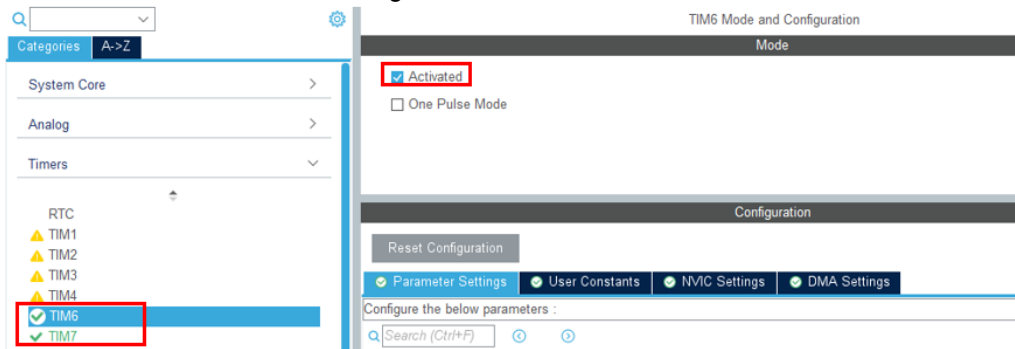


Figure 5-8

Step 7: Enable SPI2 with the setting below.

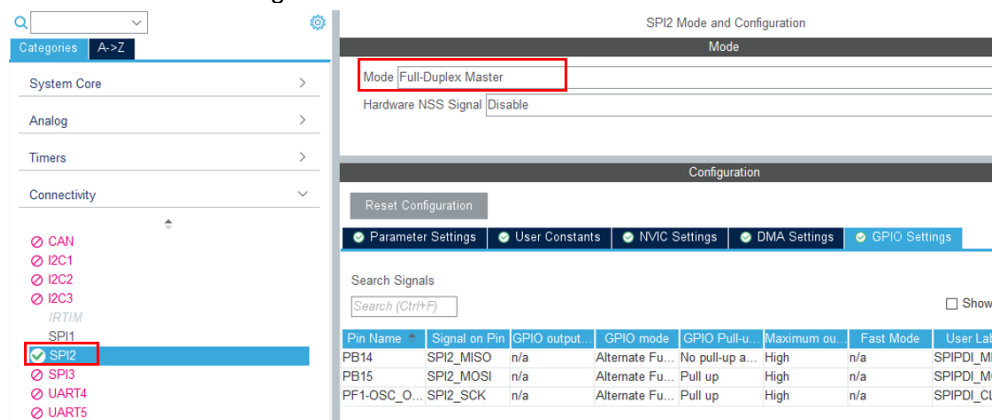


Figure 5-9

Step 8: Enable USART2 for debug console as the setting below.

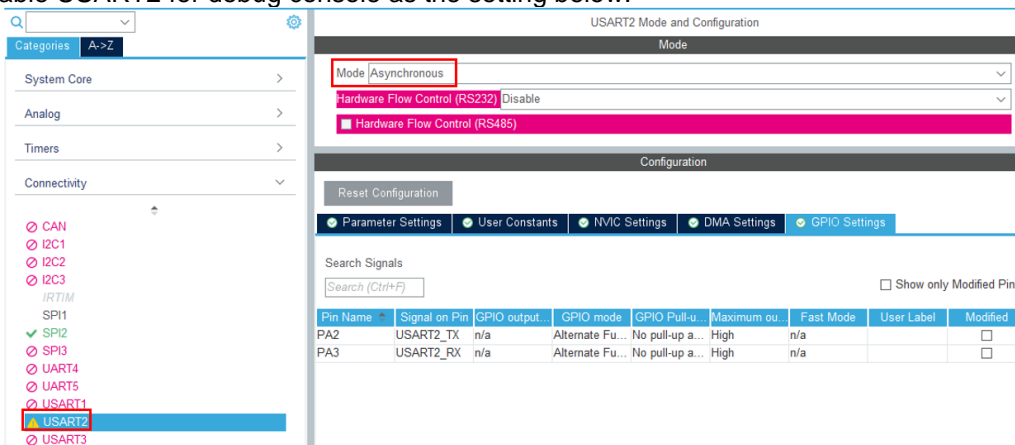


Figure 5-10

Step 9: Modify the NVIC setting as below.

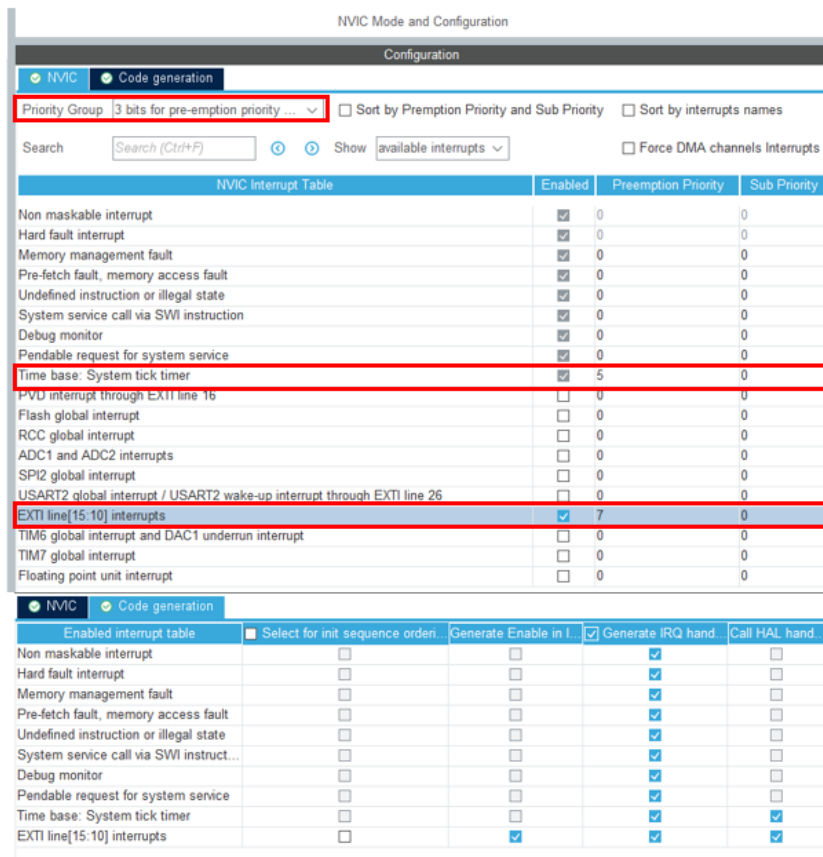


Figure 5-11

Step 10: Adjusting the clock tree in “Clock Configuration” page. Press “NO” to ignore auto solver process, then modify the settings as the red highlighted below.

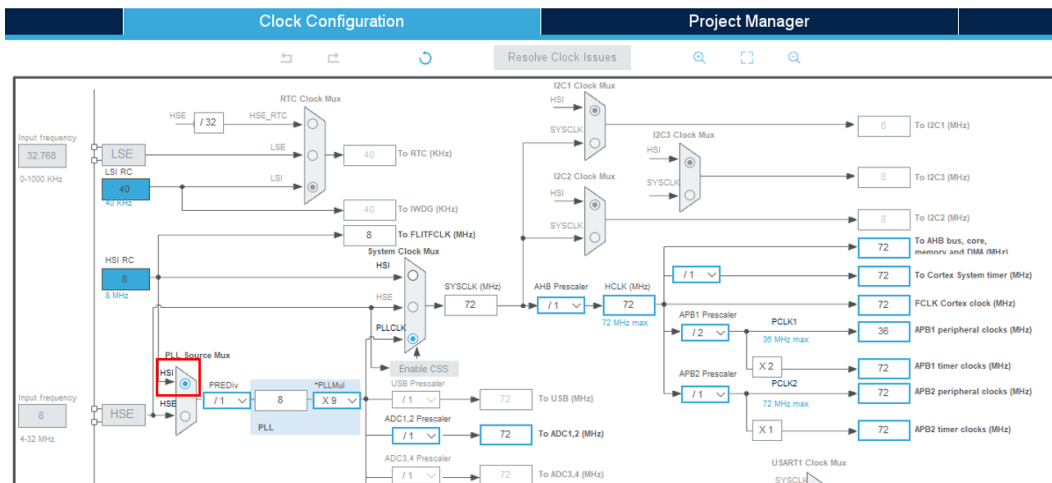
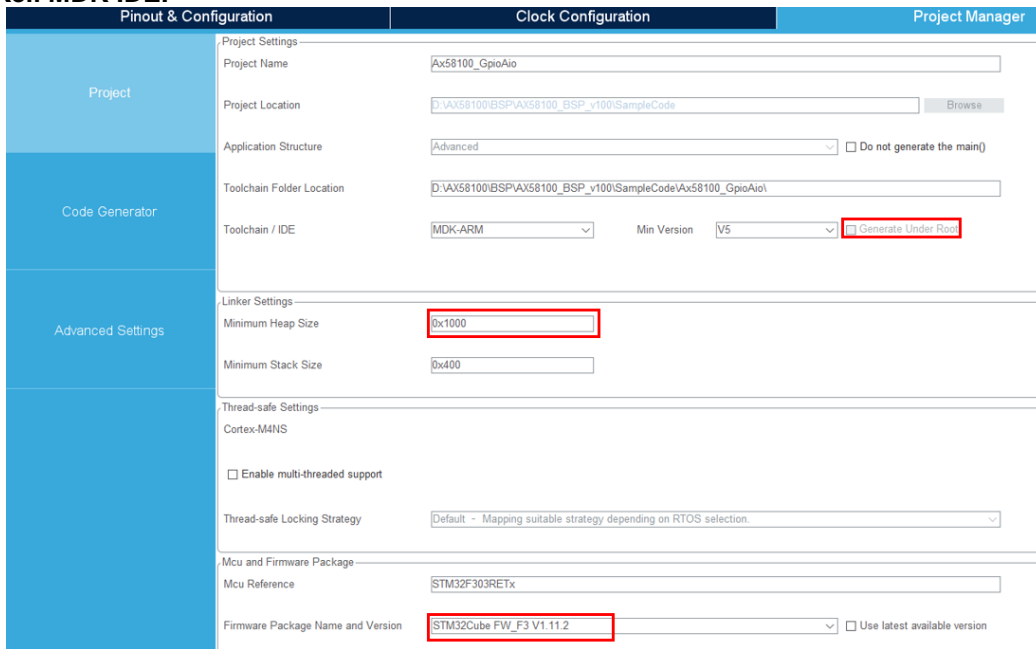


Figure 5-12

5-3 Generate Source Code in STM32CubeMX and MDK-ARM

Step 1: In “Project Manager > Project” page, extended the minimum heap size to **0x1000**, and select Firmware Package as “**STM32Cube FW_F3 V1.11.2**”.

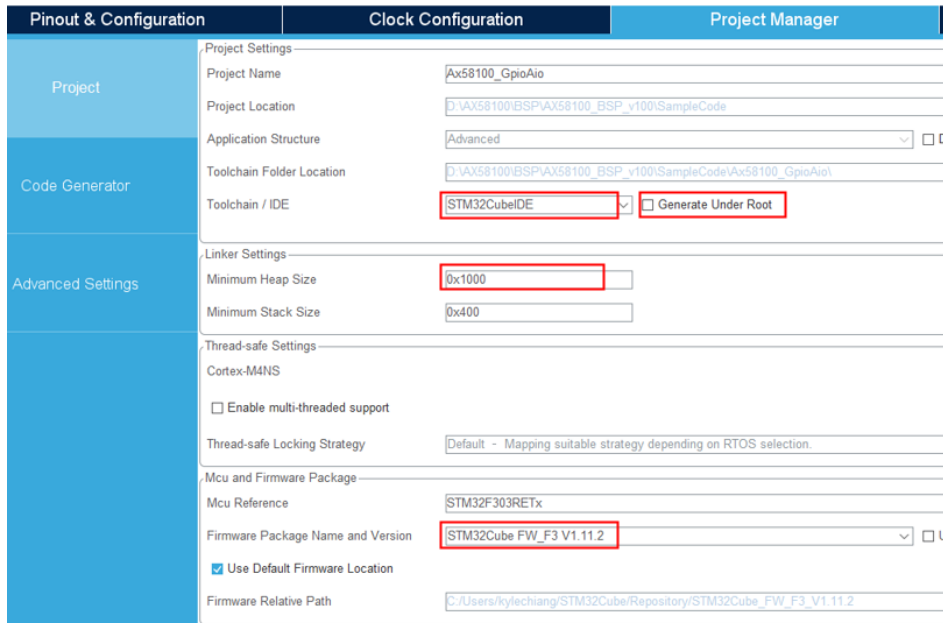
- **For Keil MDK IDE:**



Pinout & Configuration	Clock Configuration	Project Manager
Project Settings		
Project	Project Name	Ax58100_GpioAio
	Project Location	D:\AX58100\BSP\AX58100_BSP_v100\SampleCode
	Application Structure	Advanced <input type="checkbox"/> Do not generate the main()
Code Generator		
	Toolchain Folder Location	D:\AX58100\BSP\AX58100_BSP_v100\SampleCode\Ax58100_GpioAio\
	Toolchain / IDE	MDK-ARM Min Version V5 <input type="checkbox"/> Generate Under Root
Linker Settings		
Advanced Settings	Minimum Heap Size	0x1000
	Minimum Stack Size	0x400
Thread-safe Settings		
Cortex-M4NS		
<input type="checkbox"/> Enable multi-threaded support		
Thread-safe Locking Strategy: Default - Mapping suitable strategy depending on RTOS selection.		
Mcu and Firmware Package		
	Mcu Reference	STM32F303RETx
	Firmware Package Name and Version	STM32Cube FW_F3 V1.11.2 <input type="checkbox"/> Use latest available version

Figure 5-13

- **For STM32CubeIDE:**



Pinout & Configuration	Clock Configuration	Project Manager
Project Settings		
Project	Project Name	Ax58100_GpioAio
	Project Location	D:\AX58100\BSP\AX58100_BSP_v100\SampleCode
	Application Structure	Advanced <input type="checkbox"/> Do not generate the main()
Code Generator		
	Toolchain Folder Location	D:\AX58100\BSP\AX58100_BSP_v100\SampleCode\Ax58100_GpioAio\
	Toolchain / IDE	STM32CubeIDE <input type="checkbox"/> Generate Under Root
Linker Settings		
Advanced Settings	Minimum Heap Size	0x1000
	Minimum Stack Size	0x400
Thread-safe Settings		
Cortex-M4NS		
<input type="checkbox"/> Enable multi-threaded support		
Thread-safe Locking Strategy: Default - Mapping suitable strategy depending on RTOS selection.		
Mcu and Firmware Package		
	Mcu Reference	STM32F303RETx
	Firmware Package Name and Version	STM32Cube FW_F3 V1.11.2 <input type="checkbox"/> Use latest available version
	<input checked="" type="checkbox"/> Use Default Firmware Location	
	Firmware Relative Path	C:\Users\kylechiang\STM32Cube\Repository\STM32Cube_FW_F3_V1.11.2

Figure 5-14

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

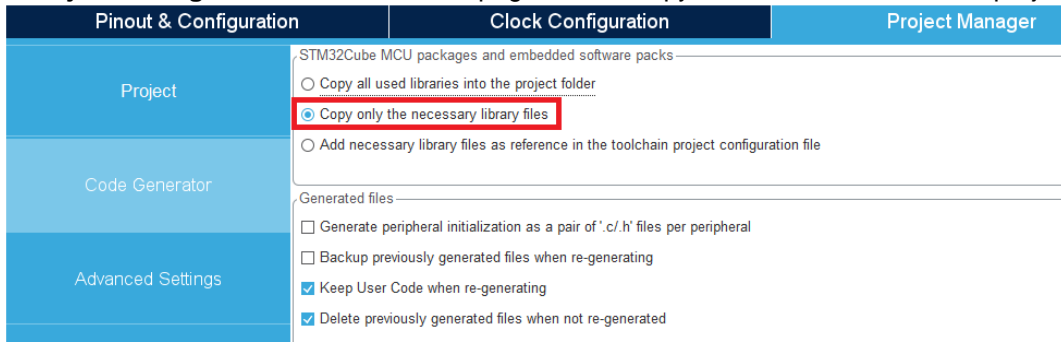


Figure 5-15

Step 3: In “Project Manager > Advanced Settings” page, disable initial code generation as below.

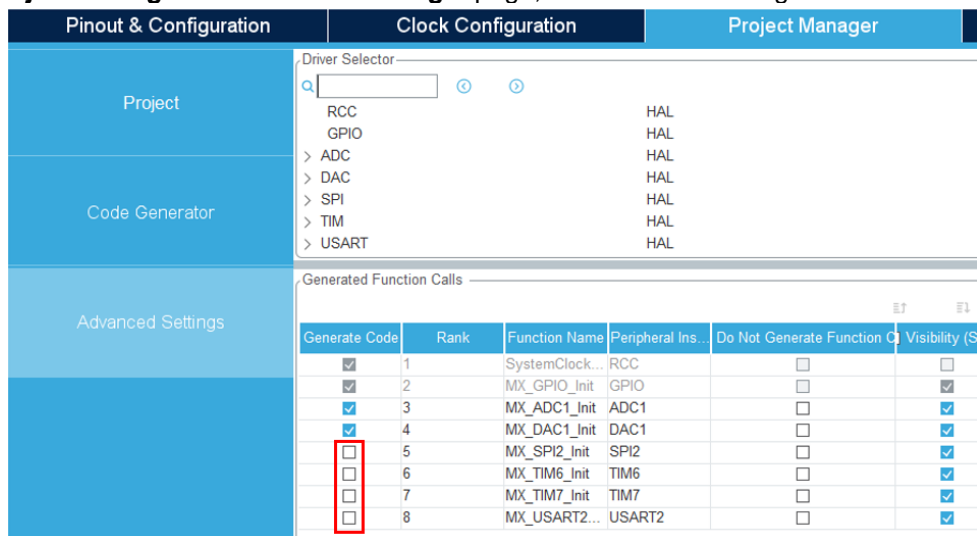


Figure 5-16

Step 4: Generates the source code.

5-4 Source Code Adjustment

Step 1: Add the AX58100 source package.
Copy the folders as below to your STM32CubeMX project folder.
BSP_ROOT\SampleCode\Ax58100_GpioAio\AX58100
BSP_ROOT\SampleCode\Ax58100_GpioAio\Binary
BSP_ROOT\SampleCode\Ax58100_GpioAio\For_SSC_Tool
BSP_ROOT\SampleCode\Ax58100_GpioAio\For_TwinCAT
 Copy the folders as below to your root folder.

Step 2: Adjust the Vector Table Address.
As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58100_GpioAio\Core\Src\system_stm32f3xx.c

```

91
92 #if !defined (HSI_VALUE)
93 #define HSI_VALUE ((uint32_t)8000000) /*!< Default value of the Internal os
94                                     This value can be provided and a
95 #endif /* HSI_VALUE */
96
97 /*!< Uncomment the following line if you need to relocate your vector Table in
98     Internal SRAM. */
99 /* #define VECT_TAB_SRAM */
100 #define VECT_TAB_OFFSET 0x4000 /*!< Vector Table base offset field.
101                                This value must be a multiple of 0x200. */
102 /**
103  * @)
104  */
105
106
107
108
109
110
111
112 void SystemInit(void)
113 {
114 /* FPU settings -----*/
115 #if (__FPU_PRESENT == 1) && (__FPU_USED == 1)
116 SCB->CPACR |= ((3UL << 10*2)|(3UL << 11*2)); /* set CP10 and CP11 Full Access */
117 #endif
118
119 /* Configure the Vector Table location -----*/
120 #if defined(USER_VECT_TAB_ADDRESS)
121 SCB->VTOR = VECT_TAB_BASE_ADDRESS | VECT_TAB_OFFSET; /* Vector Table Relocation in
    Internal SRAM */
122 #else
123 SCB->VTOR = FLASH_BASE | VECT_TAB_OFFSET; /* Vector Table Relocation in Internal FLASH */
124 #endif /* USER_VECT_TAB_ADDRESS */
125 }
126 ...
  
```

Figure 5-17

Step 3: Adjust the Linker setting.

- **For Keil MDK IDE:**

Skip this step.

- **For STM32CubeIDE:**

BSP_ROOT\SampleCode\Ax58100_GpioAio\STM32CubeIDE\STM32F303RETX_FLASH.ld

```

45 /* Memories definition */
46 MEMORY
47 {
48 CCMRAM (xrw) : ORIGIN = 0x10000000, LENGTH = 16K
49 RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 64K
50 FLASH (rx) : ORIGIN = 0x8004000, LENGTH = 246K
51 }
52
  
```

Figure 5-18

Step 4: Modify source code.

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58100_GpioAio\Core\Src\main.c

```

22 /* Private includes -----
23 /* USER CODE BEGIN Includes */
24 #include "AX58100_Hw.h"
25 #include "applInterface.h"
26 #include "ax58100.h"
27 #include "AX58100_GPIO_8Bit_AIO_16Bit.h"
28 #include "bootmode.h"
29 #include "ax58100_FoeAppl.h"
30
31 #include "ecat_def.h"
32 #include "ax58100_utils.h"
33 /* USER CODE END Includes */
34
35
36
37
38
39
40
41
42
43
44
45 /* Private macro -----
46 /* USER CODE BEGIN PM */
47 #define DO_LEVEL_REVERSE 0
48
49 #define ADC_STATE_INIT 0
50 #define ADC_STATE_WAIT_CONVERSION 1
51 #define ADC_STATE_CONVERSION_FINISH 2
52 #define ADC_STATE_WAIT_NEXT 3
53
54 #define ADC_NEXT_TIMEOUT 1//ms
55 /* USER CODE END PM */
56
57
58
59
60
61
62 /* USER CODE BEGIN PV */
63 #8 virtualAioEnable;
64 /* USER CODE END PV */
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180

```

Figure 5-19

```

154 /* Initialize all configured peripherals */
155 MX_GPIO_Init();
156 MX_ADC1_Init();
157 MX_DAC1_Init();
158 /* USER CODE BEGIN 2 */
159 #if (AX58100_DEBUG_ENABLE)
160 AX_UART_Init();
161 CONSOLE_Init();
162 DBG_PRINT("%s Firmware v%s\r\n", DEVICE_NAME, DEVICE_SW_VERSION);
163 #endif //if (AX58100_DEBUG_ENABLE)
164
165 #if (SSC_STACK_ENABLE)
166 if (HW_Init())
167 {
168     HW_Release();
169 }
170 else
171 {
172     MainInit();
173     GpioAio_Init();
174     AX58100_FoeInit();
175 }
176 #endif //if (SSC_STACK_ENABLE)
177
178 ADIO_Init();
179 /* USER CODE END 2 */
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure 5-20

```

509 /* USER CODE BEGIN 4 */
510 /*
511 * -----
512 * Function Name: void GetDigitalInput(void)
513 * Purpose:
514 * Params: None
515 * Returns: None
516 * Note:
517 * -----
518 */
519 u8 GetDigitalInput(void)
520 {
521     u8 val = 0;
522
523     /* Handle input */
524     val |= HAL_GPIO_ReadPin(DI0_GPIO_Port, DI0_Pin) ? 0x01 : 0x00;
525     val |= HAL_GPIO_ReadPin(DI1_GPIO_Port, DI1_Pin) ? 0x02 : 0x00;
526     val |= HAL_GPIO_ReadPin(DI2_GPIO_Port, DI2_Pin) ? 0x04 : 0x00;
527     val |= HAL_GPIO_ReadPin(DI3_GPIO_Port, DI3_Pin) ? 0x08 : 0x00;
528     val |= HAL_GPIO_ReadPin(DI4_GPIO_Port, DI4_Pin) ? 0x10 : 0x00;
529     val |= HAL_GPIO_ReadPin(DI5_GPIO_Port, DI5_Pin) ? 0x20 : 0x00;
530     val |= HAL_GPIO_ReadPin(DI6_GPIO_Port, DI6_Pin) ? 0x40 : 0x00;
531     val |= HAL_GPIO_ReadPin(DI7_GPIO_Port, DI7_Pin) ? 0x80 : 0x00;
532     return val;
533 }
534 /* End of GetDigitalInput() */
535
536 /*
537 * -----
538 * Function Name: void DigitalOutputHandler(void)
539 * Purpose:
540 * Params: None
541 * Returns: None
542 * Note:
543 * -----
544 */
545 void SetDigitalOutput(u8 val)
546 {
547
548 #if (DO_LEVEL_REVERSE)
549     /* Reverse the output level */
550     val = ~val;
551 #endif // #if (DO_LEVEL_REVERSE)
552
553     /* Handle output */
554     HAL_GPIO_WritePin(DO0_GPIO_Port, DO0_Pin, ((val & BIT_0) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
555     HAL_GPIO_WritePin(DO1_GPIO_Port, DO1_Pin, ((val & BIT_1) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
556     HAL_GPIO_WritePin(DO2_GPIO_Port, DO2_Pin, ((val & BIT_2) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
557     HAL_GPIO_WritePin(DO3_GPIO_Port, DO3_Pin, ((val & BIT_3) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
558     HAL_GPIO_WritePin(DO4_GPIO_Port, DO4_Pin, ((val & BIT_4) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
559     HAL_GPIO_WritePin(DO5_GPIO_Port, DO5_Pin, ((val & BIT_5) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
560     HAL_GPIO_WritePin(DO6_GPIO_Port, DO6_Pin, ((val & BIT_6) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
561     HAL_GPIO_WritePin(DO7_GPIO_Port, DO7_Pin, ((val & BIT_7) ? GPIO_PIN_SET :
GPIO_PIN_RESET));
562 }
563 /* End of SetDigitalOutput() */
564

```

Figure 5-21

```

565 /*
566 * -----
567 * Function Name: ul6 GetAnalogInput(void)
568 * Purpose:
569 * Params: None
570 * Returns: None
571 * Note:
572 * -----
573 */
574 ul6 GetAnalogInput(void)
575 {
576     static u32 adcState = ADC_STATE_INIT, tickstart=0;
577     u32 tmp16;
578     static ul6 val=0;
579
580     if (adcState == ADC_STATE_INIT)
581     {
582         /* Start conversion */
583         if (HAL_ADC_Start(&hadcl) != HAL_OK)
584         {
585             HAL_ADC_Stop(&hadcl);
586             return val;
587         }
588         adcState = ADC_STATE_WAIT_CONVERSION;
589     }
590     else if (adcState == ADC_STATE_WAIT_CONVERSION)
591     {
592         /* Ignore handle, just wait conversion complete */
593         if (hadcl.Instance->ISR & ADC_FLAG_EOC)
594         {
595             adcState = ADC_STATE_CONVERSION_FINISH;
596         }
597     }
598     else if (adcState == ADC_STATE_CONVERSION_FINISH)
599     {
600         /* Get the conversion result of the sample module 0 */
601         val = (ul6)HAL_ADC_GetValue(&hadcl);
602         tickstart = HAL_GetTick();
603         adcState = ADC_STATE_WAIT_NEXT;
604     }
605     else if (adcState == ADC_STATE_WAIT_NEXT)
606     {
607         /* Wait a while for next conversion */
608         tmp16 = HAL_GetTick();
609         if (tmp16 >= tickstart)
610         {
611             if ((tmp16 - tickstart) >= ADC_NEXT_TIMEOUT)
612             {
613                 adcState = ADC_STATE_INIT;
614             }
615         }
616         else
617         {
618             tickstart = tmp16;
619         }
620     }
621
622     return val;
623 } /* End of GetAnalogInput() */
624

```

Figure 5-22

```

625 /*
626 * -----
627 * Function Name: void SetAnalogOutput(void)
628 * Purpose:
629 * Params: None
630 * Returns: None
631 * Note:
632 * -----
633 */
634 void SetAnalogOutput(ul6 val)
635 {
636     u32 tmp16, tickstart=0;
637
638     tmp16 = val & 0x0FFF;
639
640     /* Start A/D conversion */
641     HAL_DAC_SetValue(&hadcl, DAC_CHANNEL_1, DAC_ALIGN_12B_R, tmp16);
642     HAL_DAC_Start(&hadcl, DAC_CHANNEL_1);
643
644     /* Wait conversion done */
645     tickstart = HAL_GetTick();
646     while ((HAL_GetTick() - tickstart) < 3)
647     {
648         if (0 == READ_BIT(hadcl.Instance->SWTRIGR, DAC_SWTRIGR_SWTRIG1))
649         {
650             break;
651         }
652     }
653 } /* End of SetAnalogOutput() */
654
655 /*
656 * -----
657 * Function Name: u8 GetSwitchID(void)
658 * Purpose: Get current digit of "Rotary Dip Switch" of AX58400-EXB-AD10
659 * Params: None
660 * Returns: Current digit
661 * Note:
662 * -----
663 */
664 u8 GetSwitchID(void)
665 {
666     u8 currentSwitchID = 0;
667
668     currentSwitchID |= HAL_GPIO_ReadPin(SW0_GPIO_Port, SW0_Pin) ? BIT_0 : 0;
669     currentSwitchID |= HAL_GPIO_ReadPin(SW1_GPIO_Port, SW1_Pin) ? BIT_1 : 0;
670     currentSwitchID |= HAL_GPIO_ReadPin(SW2_GPIO_Port, SW2_Pin) ? BIT_2 : 0;
671     currentSwitchID |= HAL_GPIO_ReadPin(SW3_GPIO_Port, SW3_Pin) ? BIT_3 : 0;
672     currentSwitchID |= HAL_GPIO_ReadPin(SW4_GPIO_Port, SW4_Pin) ? BIT_4 : 0;
673     currentSwitchID |= HAL_GPIO_ReadPin(SW5_GPIO_Port, SW5_Pin) ? BIT_5 : 0;
674     currentSwitchID |= HAL_GPIO_ReadPin(SW6_GPIO_Port, SW6_Pin) ? BIT_6 : 0;
675     currentSwitchID |= HAL_GPIO_ReadPin(SW7_GPIO_Port, SW7_Pin) ? BIT_7 : 0;
676
677     return (~currentSwitchID);
678 } /* End of GetSwitchID() */
679

```

Figure 5-23

```

700 /*
701 * -----
702 * Function Name: HAL_GPIO_EXTI_Callback
703 * Purpose:
704 * Params:
705 * Returns:
706 * Note:
707 * -----
708 */
709 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
710 {
711     if (GPIO_Pin == DAC_TRIG_Pin)
712     {
713         /* Detected DAC trigger signal */
714         MCU_GPIO_EXTI_CLEAR_FLAG(DAC_TRIG_Pin);
715     }
716     if (GPIO_Pin == ADC_TRIG_Pin)
717     {
718         /* Detected ADC trigger signal */
719         MCU_GPIO_EXTI_CLEAR_FLAG(ADC_TRIG_Pin);
720     }
721     if (GPIO_Pin == USER_BTN_Pin)
722     {
723         /* Detected user button signal */
724         MCU_GPIO_EXTI_CLEAR_FLAG(USER_BTN_Pin);
725     }
726 } /* End of HAL_GPIO_EXTI_Callback() */
727
728 /* USER CODE END 4 */
729
730 ...
731 ...

```

Figure 5-24

As highlighted in the figure below, modify file content in the path:
BSP_ROOT\SampleCode\Ax58100_GpioAio\Core\Src\stm32f3xx_it.c

```

20 /* Includes -----
21 #include "main.h"
22 #include "stm32f3xx_it.h"
23 /* Private includes -----
24 /* USER CODE BEGIN Includes */
25 #include "ax58100_utils.h"
26 /* USER CODE END Includes */
27
28 /* Private typedef -----
29 /* USER CODE BEGIN TD */
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182 /* @brief This function handles System tick timer.
183 */
184 void SysTick_Handler(void)
185 {
186     /* USER CODE BEGIN SysTick_IRQn 0 */
187
188     /* USER CODE END SysTick_IRQn 0 */
189     HAL_IncTick();
190     /* USER CODE BEGIN SysTick_IRQn 1 */
191     #if (AX58100_DEBUG_ENABLE)
192     console_TimeTick();
193     #endif // #if (AX58100_DEBUG_ENABLE)
194     /* USER CODE END SysTick_IRQn 1 */
195 }
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure 5-25

5-5 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

Assume the SSC tool has been installed, the related files needed are as below.

SSC tool project file:

BSP_ROOT\SampleCode\Ax58100_GpioAio\For_SSC_Tool\AX58100_GPIO_8Bit_AIO_12Bit.esp

Configuration file for custom import:

BSP_ROOT\SampleCode\Ax58100_GpioAio\For_SSC_Tool\Import\Configuration\AsixAx58100GpioAioConfigurations.xml

Excel file for application import:

BSP_ROOT\SampleCode\Ax58100_GpioAio\For_SSC_Tool\Import\Configuration\files\AX58100_GPIO_8Bit_AIO_12Bit.xlsx

5-6 IDE Setting Adjustment

- **For Keil MDK IDE:**

Double clicks the file as below to open project.

BSP_ROOT\SampleCode\Ax58100_GpioAio\MDK-ARM\Ax58100_GpioAio.uvproj

Step 1: Add source code both on dual cores.

Click on **"Project > Manage > Project Items..."** to open **"Manage Project Items"** dialog, then add groups and files as the figure below.

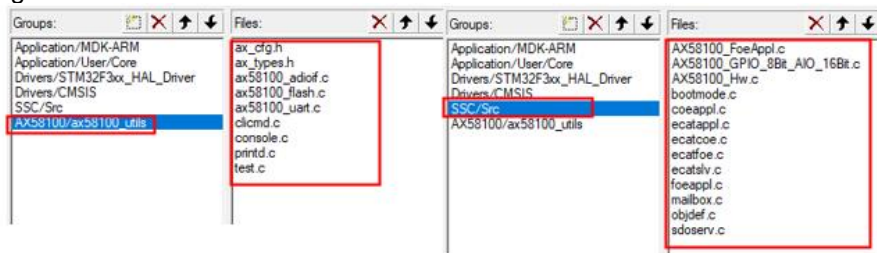


Figure 5-26

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.

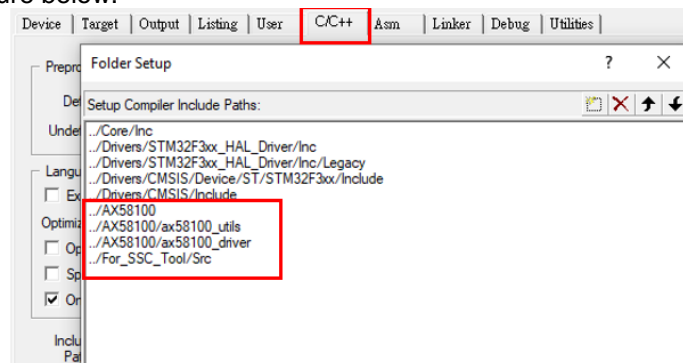


Figure 5-27

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.

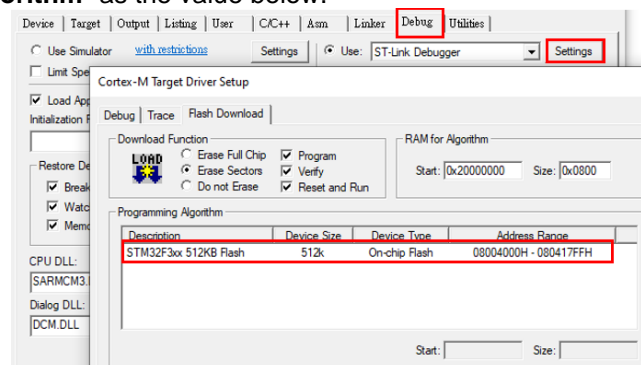


Figure 5-28

Start = 0x08004000, Size = 0x0003D800

Step 4: Change “Linker” settings.

Find the setting of “Project > Options for Target ‘xxx’... > Linker”, then checked the “Use Memory Layout from Target Dialog”.

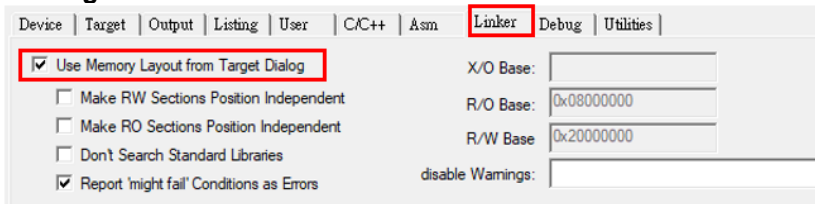


Figure 5-29

Step 5: Add post build actions.

Find the setting of “Project > Options for Target ‘xxx’... > User > After Build/Rebuild”, then add actions as the value below.

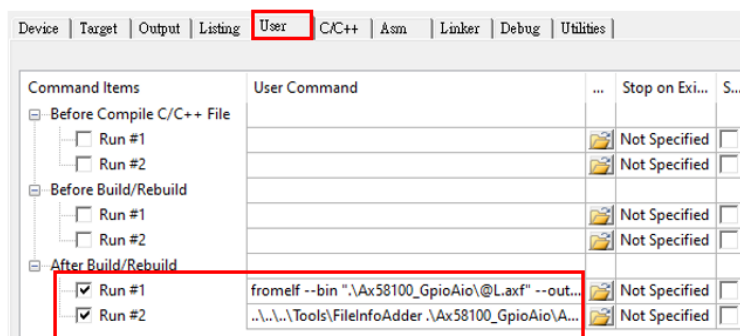


Figure 5-30

Run #1: `fromelf --bin ".\Ax58100_GpioAio\@L.axf" --output ".\Ax58100_GpioAio\@L.bin"`

Run #2:

`..\..\Tools\FileInfoAdder .\Ax58100_GpioAio\Ax58100_GpioAio.bin ..\For_TwinCAT\FoE\AX58100_GpioAio.efw`

Step 6: Modify “Target Options”.

Find the setting of “Project > Options for Target ‘xxx’... > Target > Read/Write Memory Areas”, then modify it as the value below.

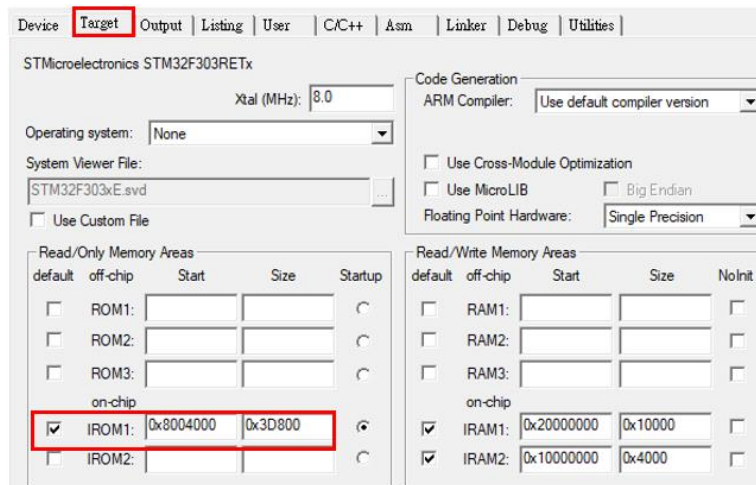


Figure 5-31

IROM1 Start = 0x8004000, Size = 0x3D800

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

- **For STM32CubeIDE:**

Double clicks the file as below to open project.

BSP_ROOT\SampleCode\Ax58100_GpioAio\STM32CubeIDE\project

Step 1: Add source code.

On “**Project Explorer**”, add virtual folders and linking the source files.

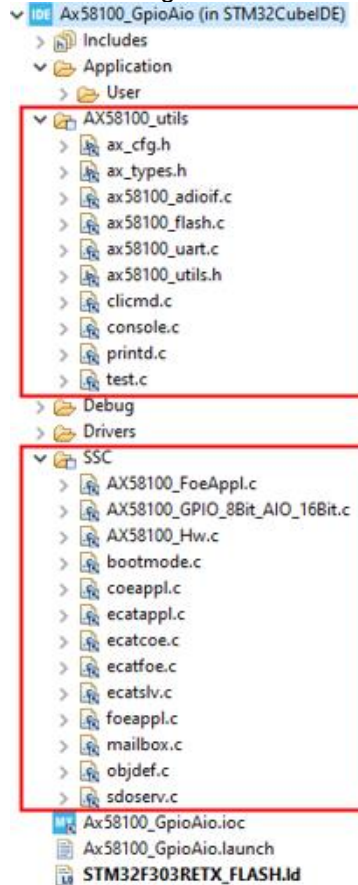


Figure 5-32

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

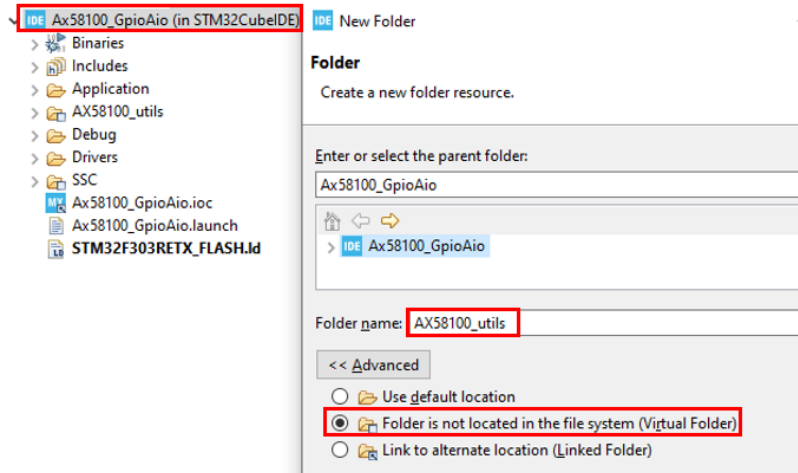


Figure 5-33

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

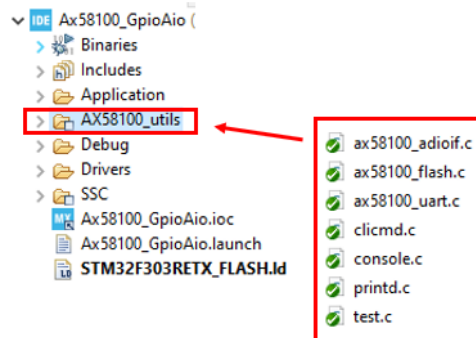


Figure 5-34

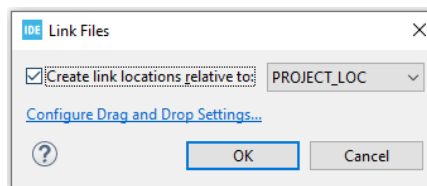


Figure 5-35

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.

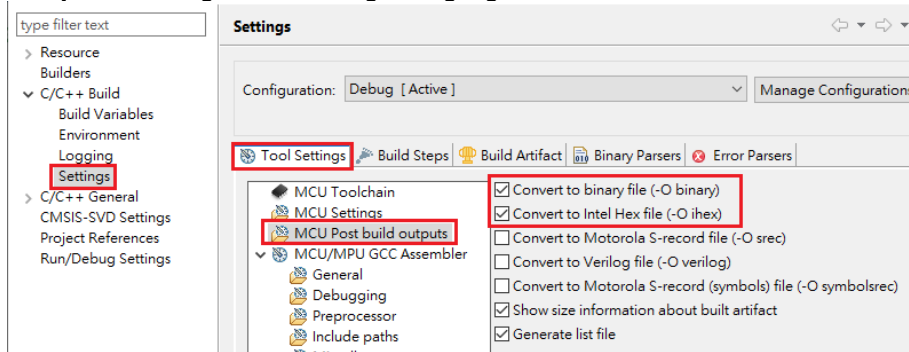


Figure 5-36

Step 3: Add include path.

Select target project, then find setting on “**Project > Properties > C/C++ Build > Settings > Tool Settings > MCU/MCP GCC Compiler > Include paths**”, then add paths as the figure below.

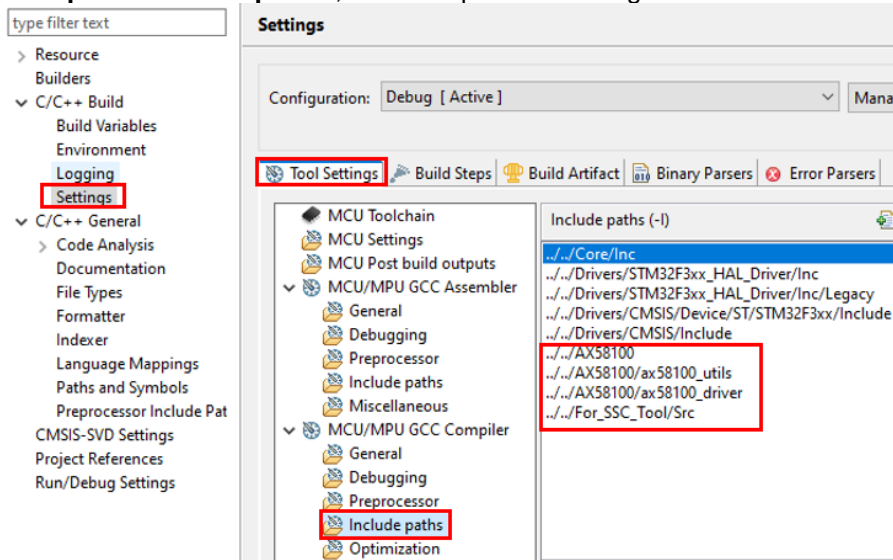


Figure 5-37

Step 4: Add post build actions.

Find the setting of “**Project > Properties > C/C++ Build > Settings > Build Steps > Post-build steps**”, then add actions as the value below.

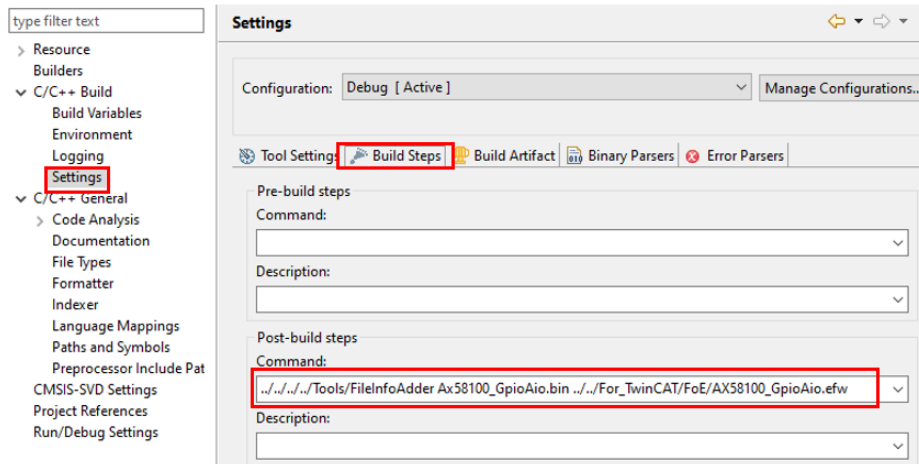


Figure 5-38

Post-build steps Command:

../../../../Tools/FileInfoAdder Ax58100_GpioAio.bin ../../../../../../For_TwinCAT/FoE/AX58100_GpioAio.efw

Now, entire building procedure has done for STM32CubeIDE.

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

6. Build Up and Download the Firmware

Here will introduce how to build up full firmware source in your local IDE.

6-1 Build up Firmware Binary

- **For Keil MDK IDE:**

Step 1: Double clicks on “Ax58100_GpioAio.uvprojx” file in the folder below to open project.
BSP_ROOT\SampleCode\Ax58100_GpioAioMDK-ARM.

Step 2: Execute “Rebuild all target files” under “Menu Bar > Project”.

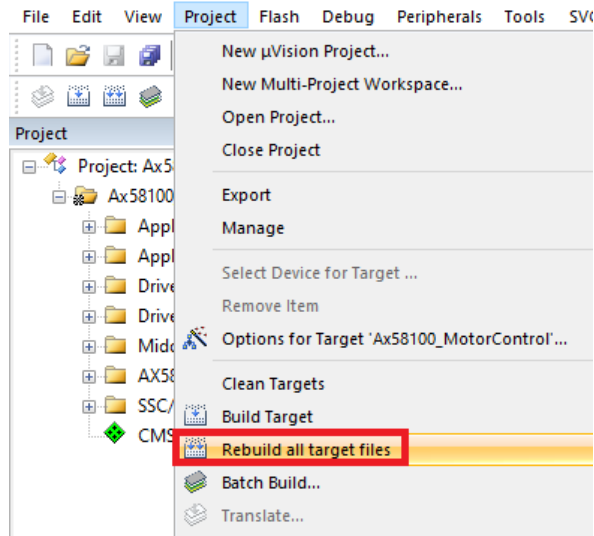


Figure 6-1

- **For STM32CubeIDE:**

Step 1: Double clicks the file as below to open project.
BSP_ROOT\SampleCode\Ax58100_GpioAio\STM32CubeIDE\project

Step 2: Select the project node and execute “Build Project” under “Menu Bar > Project”.

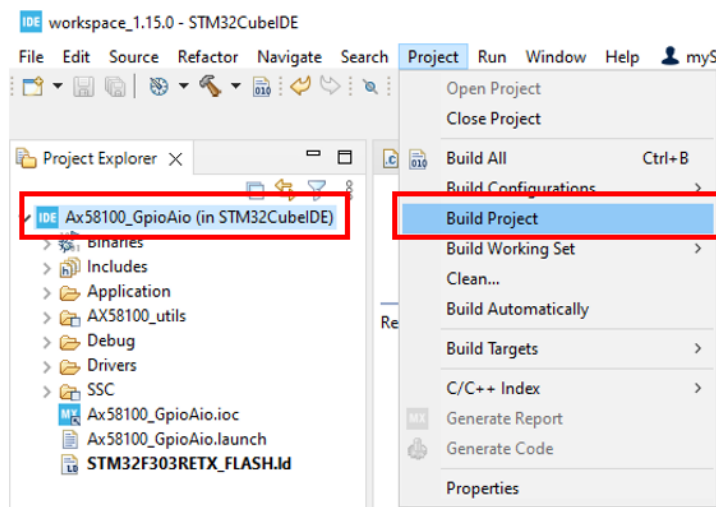


Figure 6-2

6-2 Download Bootloader Firmware

Step 1: Open STM32CubeProgrammer utility and connect to the MCU board.

Step 2: Select the “Erasing & Programming” page, then click “Browse” button to select bootloader firmware binary file in the path below.

BSP_ROOT\SampleCode\Ax58100_Bldr\Binary\Ax58100_Bldr_vX.X.X

Please ensure the “Start address” is 0x08000000.

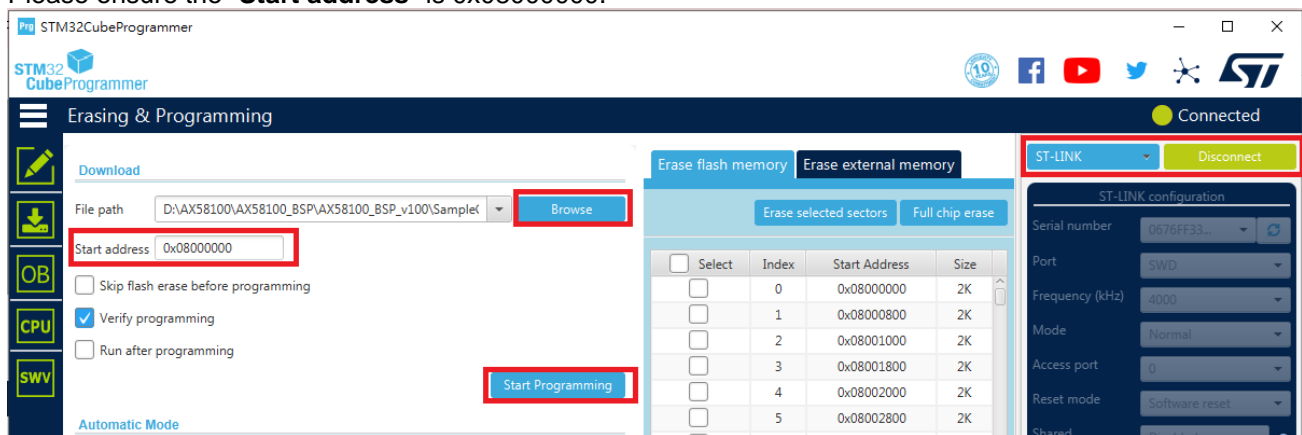


Figure 6-3

Step 3: Press “Start Programming” button to download bootloader firmware.

6-3 Download GPIO / AIO Firmware

- For Keil MDK IDE:

Step 1: Click “Menu bar > Flash > Download” to download firmware into MCU.

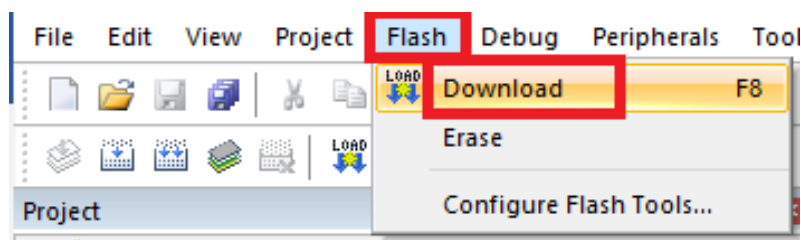


Figure 6-4

- For STM32CubeIDE:

Step 1: Click “Menu bar > Run > Run As > STM32 C/C++ Application” to download firmware into MCU.

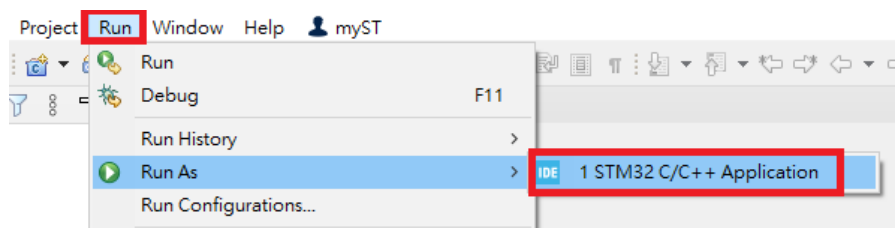


Figure 6-5

- **Using STM32CubeProgrammer Utility:**

Step 1: Similarly, please ensure the utility has connected to target MCU board.

Step 2: Select the “**Erasing & Programming**” page, then click “**Browse**” button to select GPIO / AIO firmware binary file in the path below.

- **For Keil MDK IDE:**
BSP_ROOT\SampleCode\Ax58100_GpioAio\MDK-ARM\Ax58100_GpioAio\Ax58100_GpioAio.bin
- **For STM32CubeIDE:**
BSP_ROOT\SampleCode\Ax58100_GpioAio\STM32CubeIDE\Debug\Ax58100_GpioAio.bin

Please ensure the “**Start address**” is 0x08004000.

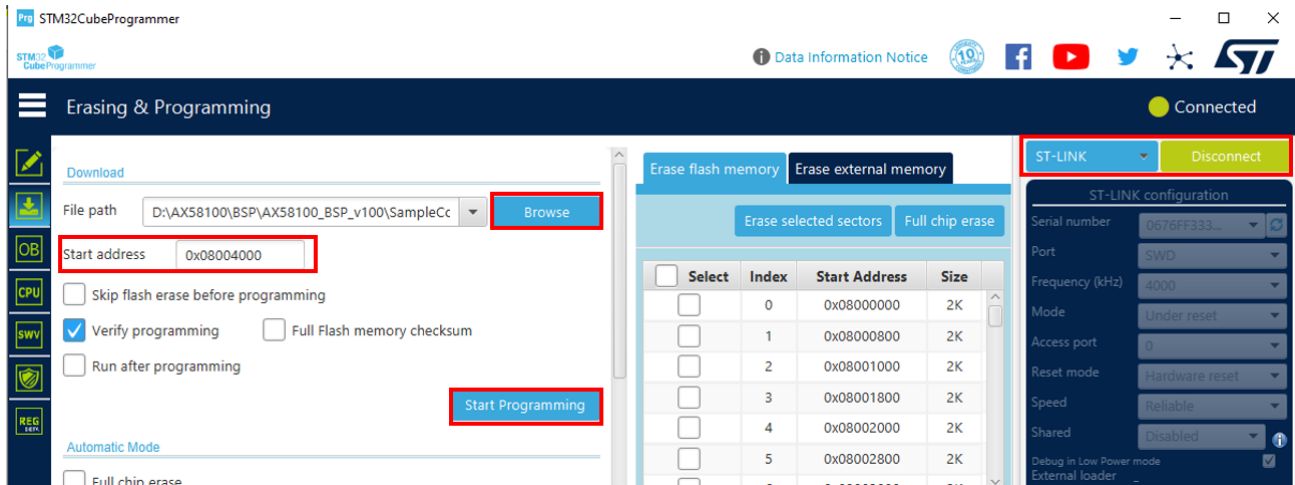


Figure 6-6

Step 3: Press “**Start Programming**” button to download GPIO / AIO firmware.

6-4 Check Firmware Status

After upgrading the firmware, you can verify that the firmware upgrade has been successfully programmed. Please refer to the following steps.

Users need to use “**Tera Term**” console tool or another console connection tool. The default setting of UART port as following:

Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

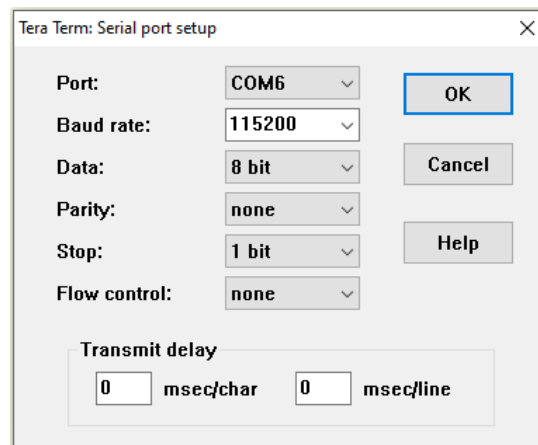


Figure 6-7

Open the COM port. If no characters appear in COM port, you can press “**Reset**” button to confirm again. Below screen shows the firmware of the EtherCAT Slave Stack working successfully.

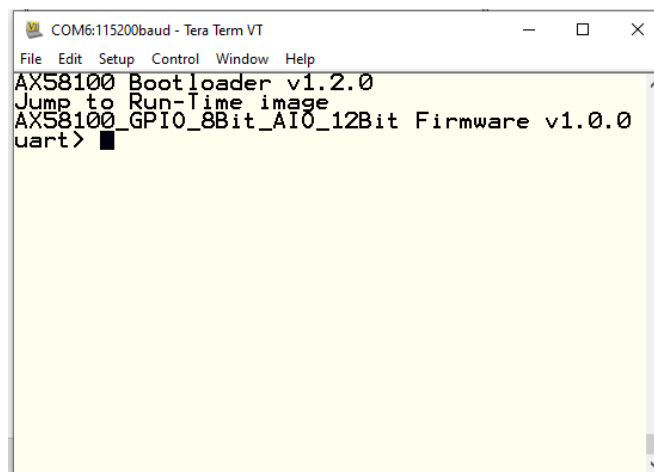


Figure 6-8

7. Basic Operation of TwinCAT

Please refer to the file “AX58x00_TwinCAT_UserGuide” Chapter 3.

8. PLC Application

Please refer to the file “AX58x00_TwinCAT_UserGuide” Chapter 5.

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 STM32F303RE as below.

Flash area	Flash memory addresses	Size (bytes)	Name
Bootloader area (Page 0~7)	0x0800 0000 - 0x0800 07FF	2 K	Page 0
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 1800 - 0x0800 1FFF	2 K	Page 3
	0x0800 3800 - 0x0800 3FFF		Page 7
Main memory	0x0800 4000 - 0x0800 47FF	.	Page 8
	0x0804 1000 - 0x0804 17FF	.	Page 130
	0x0804 1800 - 0x0804 1FFF	.	Page 131
	0x0807 F000 - 0x0807 F7FF	.	Page 254
Information block	0x0807 F800 - 0x0807 FFFF	2 K	Page 255
	0x1FFF D800 - 0x1FFF F7FF	8 K	System memory
	0x1FFF F800 - 0x1FFF F80F	16	Option bytes

Figure 9-1

- **Bootloader area (Address=0x08000000, Size=16Kbytes):**
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.
- **Runtime image area (Address=0x08004000, Size=246Kbytes):**
Used to run application firmware (as the GPIO / AIO firmware in here), it is also responsible for store the received .efw file into “Flash buffer area” via FoE data transfer.
- **Flash buffer area (Address=0x08041800, Size=247Kbytes):**
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\Ax58100_GpioAioFor_TwinCAT\FoE\AX58100_GpioAio.efw

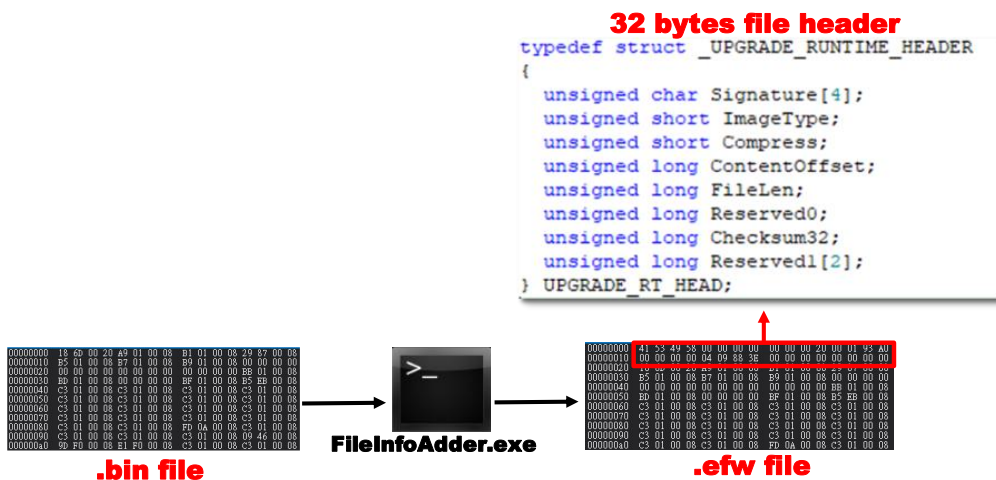


Figure 9-2

9-1 Firmware Upgrade

Here assume the EtherCAT network has established and the device has been under OP mode, then please follow the steps below to do firmware upgrade via FoE.

Step 1: Right click on the online device that you want to upgrade firmware, then select “**Firmware Update**”.

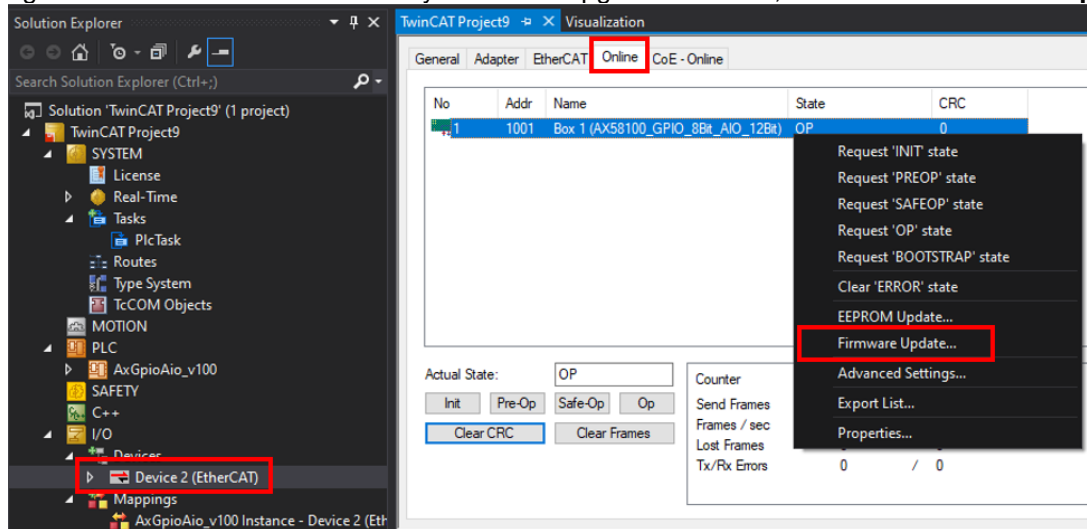


Figure 9-3

Step 2: Select firmware binary file for FoE in the path below.

“**BSP_ROOT\SampleCode\Ax58100_GpioAio\For_TwinCAT\FoE\AX58100_GpioAio.efw**”

Step 3: Click OK to start firmware upgrading.

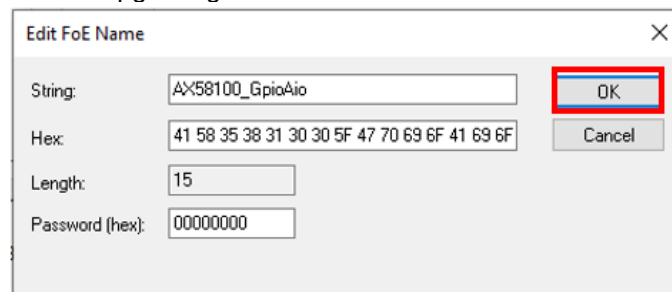


Figure 9-4

Step 4: Wait the process complete, the dialog will pop up as below if upgrading is successfully done.



Figure 9-5

10. Virtual ADIO Mode

This example not only provides physical pins for connecting external devices but also offers virtual commands, making GPIO / AIO evaluation and testing more convenient. Virtual ADIO commands must be used in conjunction with the TwinCAT PLC AxGpioAio example. This section can be referenced in Chapter 8.

If users need to connect external devices, the pinout can be referenced in Section 3.2.

10-1 Console Command Usage

Usage		
adio [-dsb] [-ve DecValue] [-vsw HexValue] [-vdi HexValue] [-vai HexValue]		
Parameters	Available value	Description
-dsb	N/A	Enables the dashboard display to show real-time test status or results. Ctrl+C is used to exit.
-ve	0: Disable (Default) 1: Enable	This command is used to enable virtual ADIO Mode. Once enabled, the -vsw, -vdi, and -vai commands can be used to provide virtual values to their respective parameters. These values will be reflected in the corresponding fields on the TwinCAT PLC AxGpioAio operation interface. If disabled, use the physical peripheral pins instead.
-vsw	0~FF	Sets the virtual switch ID state. The parameter must be in hexadecimal format. virtual ADIO Mode must be enabled.
-vdi	0~FF	Sets the virtual digital input state. The parameter must be in hexadecimal format. virtual ADIO Mode must be enabled.
-vai	0~FFF	Sets the virtual analog input value. The parameter must be in hexadecimal format and limited to 12-bit ADC data (0x000 to 0xFFF). virtual ADIO Mode must be enabled.
Examples:		
In virtual ADIO Mode, the digital I/O value is set to 0xff and enable the dashboard: adio -ve 1 -vdi ff -dsb		
In virtual ADIO Mode, the analog I/O value is set to 0x123 and enable the dashboard: adio -ve 1 -vai 123 -dsb		
In virtual ADIO Mode, analog I/O value is set to 0x123, the digital IO value is set to 0x45 and enable the dashboard: adio -ve 1 -vai 123 -di 45 -dsb		
In physical ADIO Mode, enable the dashboard: adio -ve 0 -dsb		

Table 10-1

11. Switch ID Description

The Switch ID in this application is used for identification function of TwinCAT master. User can specify an ID at EtherCAT slave side and setup identification at TwinCAT master side. TwinCAT master will check if the ID of the slave matched with the expected identification value during startup of the slave. If you want to know how to setup identification, please follow below example.

You can find out the “IdentificationReg134” setting in ESI file after SSC Tool generated source files and ESI file through import ASIX’s configuration file.

```

<Info>
  <StateMachine>
    <Timeout>
      <PreopTimeout>2000</PreopTimeout>
      <SafeopOpTimeout>9000</SafeopOpTimeout>
      <BackToInitTimeout>5000</BackToInitTimeout>
      <BackToSafeopTimeout>200</BackToSafeopTimeout>
    </Timeout>
  </StateMachine>
  <Mailbox>
    <Timeout>
      <RequestTimeout>100</RequestTimeout>
      <ResponseTimeout>2000</ResponseTimeout>
    </Timeout>
  </Mailbox>
  <IdentificationReg134>true</IdentificationReg134>
</Info>

```

Figure 11-1

This example provides both physical pins and the ability to set virtual values to simulate the Rotary Dip Switch through Console commands. However, the physical pins require an actual rotary switch to be connected externally.

The diagram below provides a pinout reference for SW0-7.

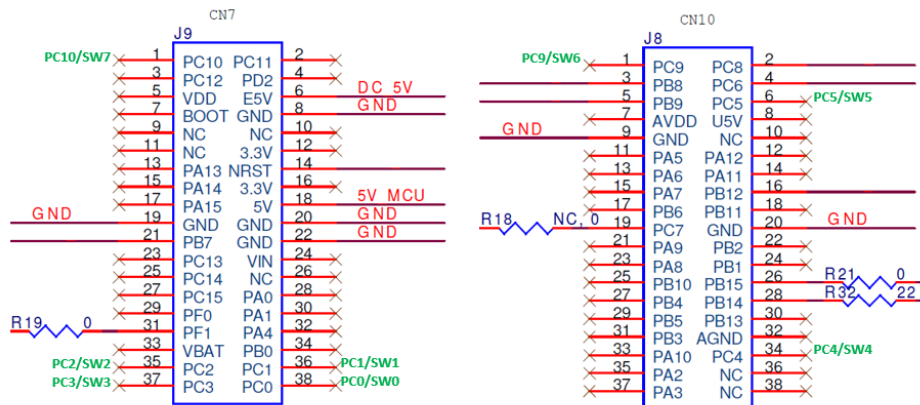


Figure 11-2

This setting will make TwinCAT master enable the identification function for the slave.
Select the slave, right-click and select “**Advanced Settings...**”.

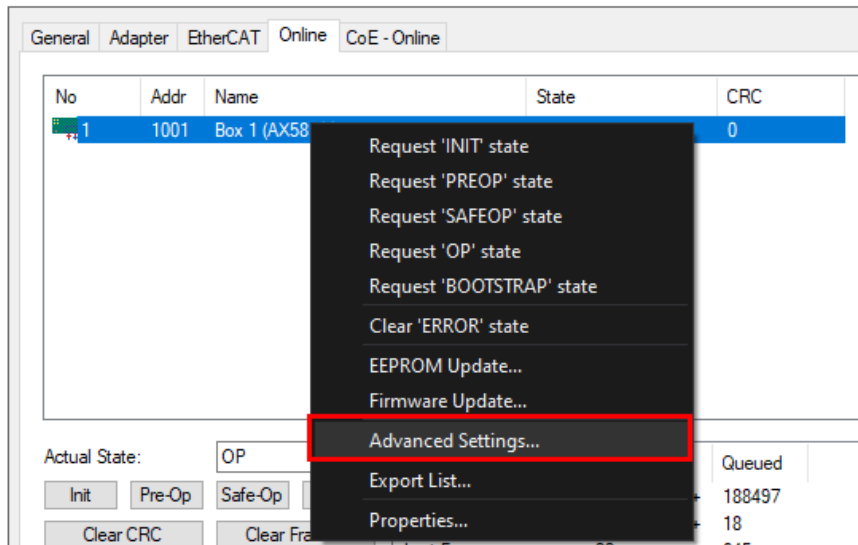


Figure 11-3

Go to “**General -> Behavior**” and turn on the “**Check Identification**”.

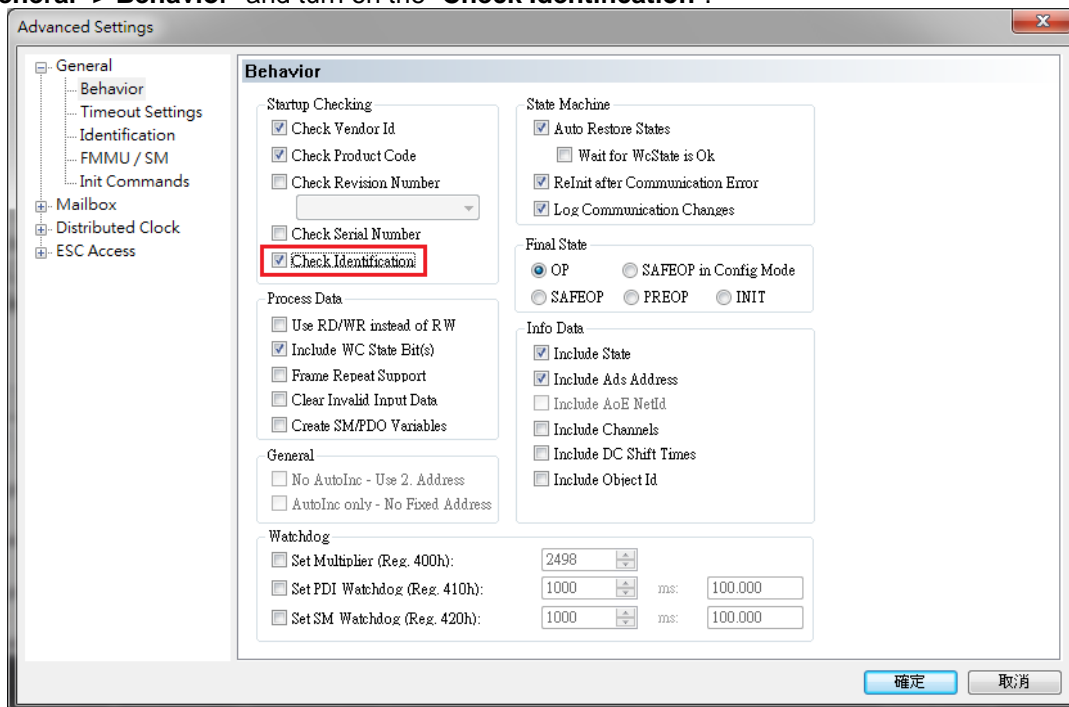


Figure 11-4

Confirm current identification ADO (Address Offset) is “(ADO 0x0134)”.
Don't forget to assign the value in decimal. The value “55” is just an example here.

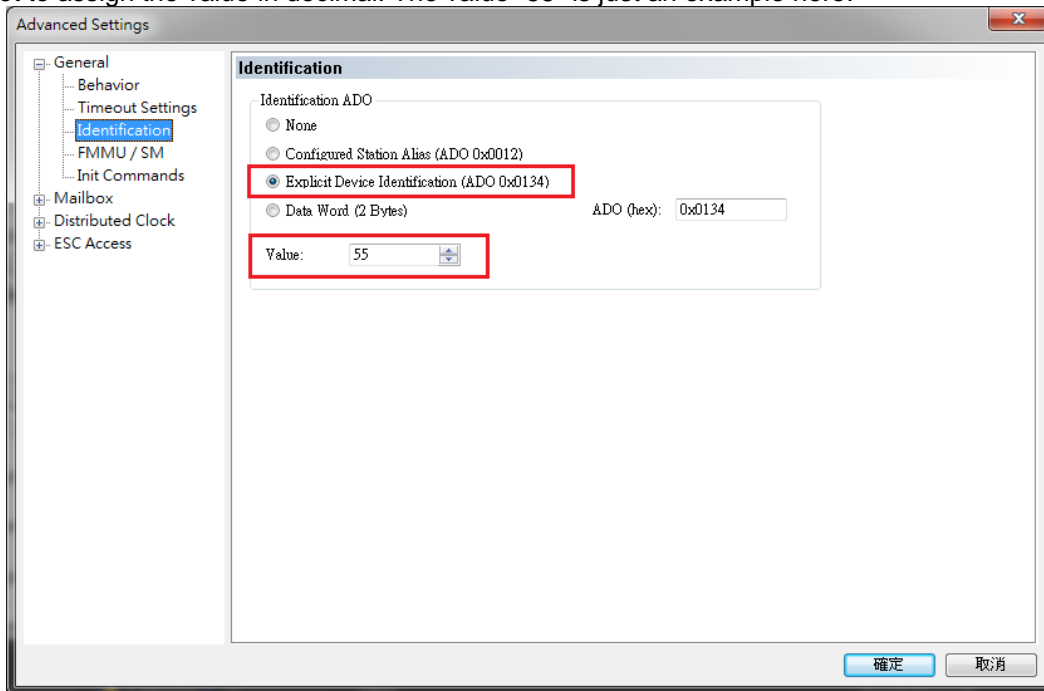


Figure 11-5

Of course, you should also set up the same value by simulating the Rotary Dip Switch through Console commands like following:

Assuming users are already familiar with executing commands via the console, otherwise, users can refer to Section 6.4 and Chapter 10.

Command: `adio -ve 1 -vsw 37 -dsb`

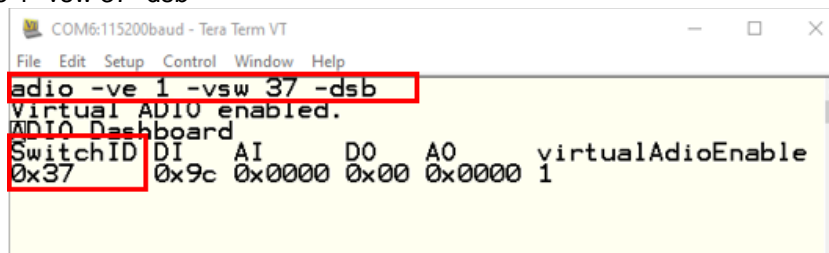


Figure 11-6

Click the "Reload Devices" button, and the state will change to "OP."

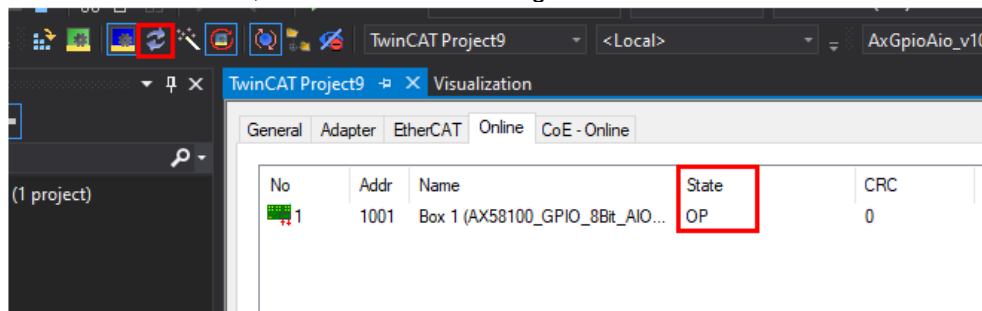


Figure 11-7

Clicked “Activate Configuration”.

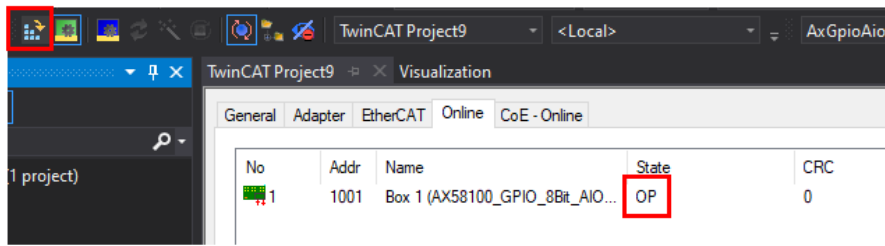


Figure 11-8

Since this example does not have an actual Rotary Dip Switch connected, we can observe that the value of ESC 0x012 will be a random value, differing each time the system is powered on. This behavior is normal. Next, we will arrange the commands to observe this behavior.

Go to “ESC Access -> Memory” in “Advanced Settings” and check out the address 0x0012.

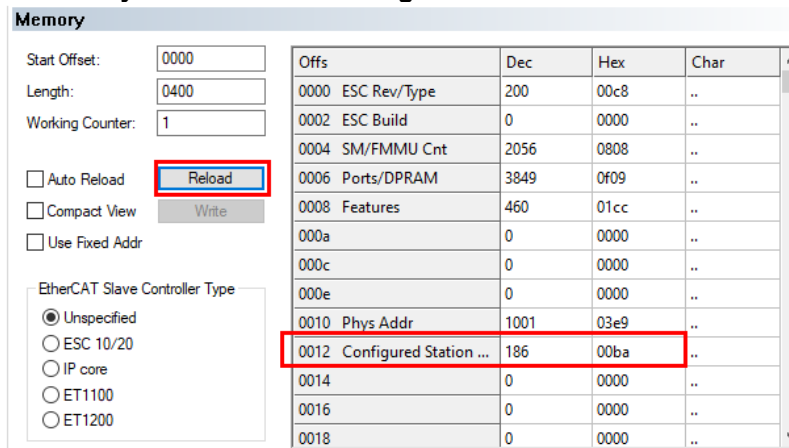


Figure 11-9

You can append an EtherCAT command to TwinCAT master for confirm the second address. Click “Device N (EtherCAT) -> Append EtherCAT Cmd”.

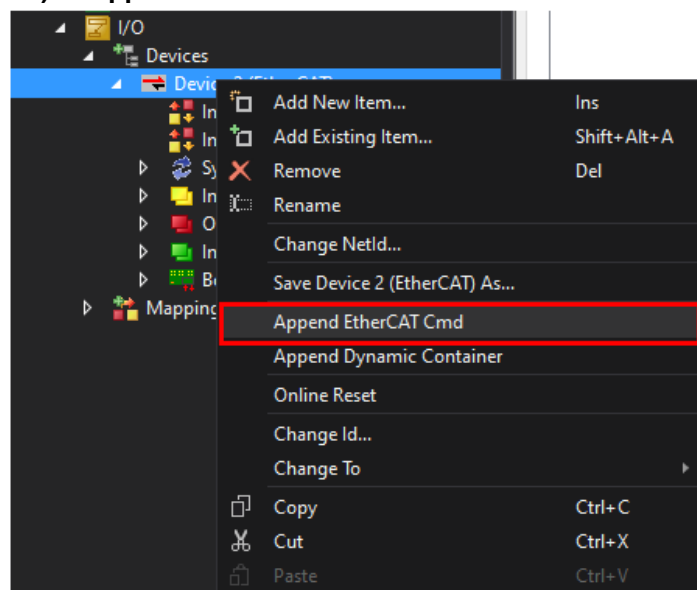


Figure 11-10

Please assign “Command” and “Type”.
I demonstrate “FPRD” and “UINT” in this example.

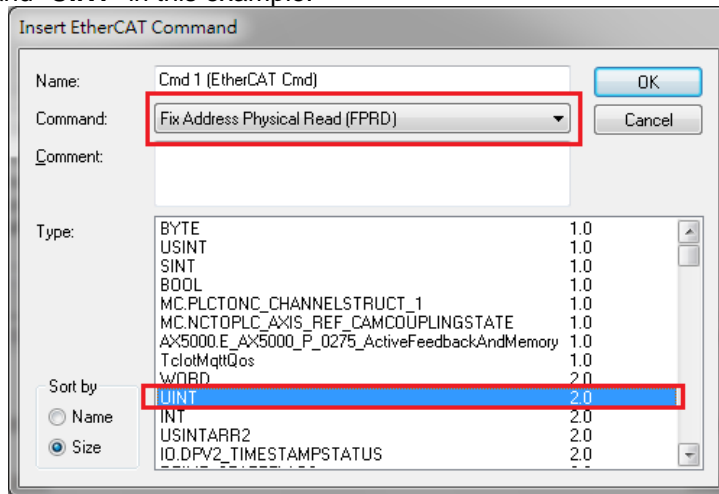


Figure 11-11

Double-click “Cmd 1” and setup “Slave Address” and “Address Offset”.
I setup slave address = 0xba and address offset = 0x12 in this example.

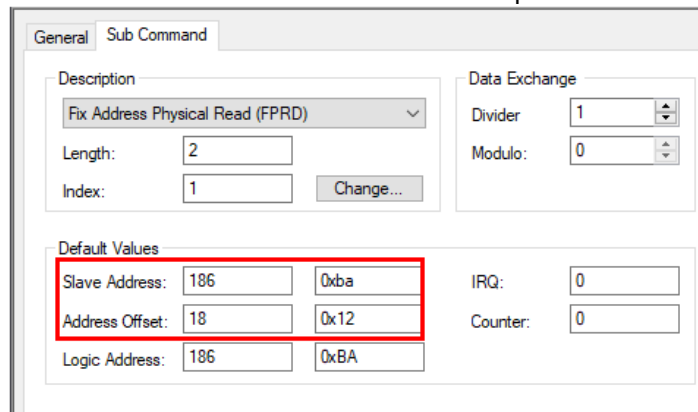


Figure 11-12

Clicked “Activate Configuration”.

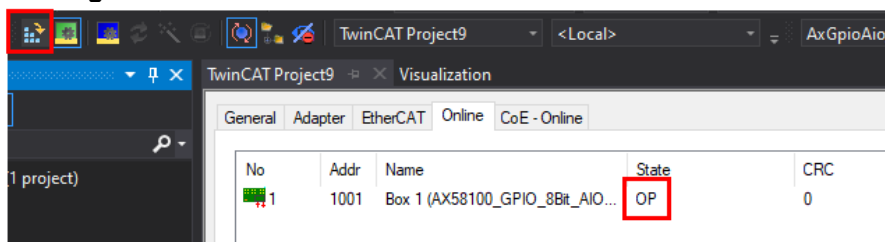
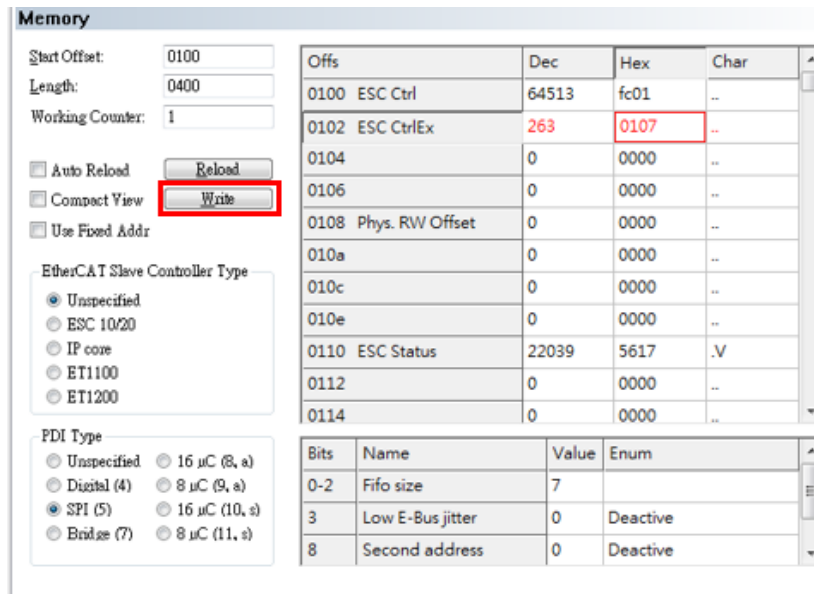


Figure 11-13

If you expect that the slave can also use this Switch ID value as second slave address. You just turn on **“Seconds address”** bit (bit8 = 1) in ESC register 0x0102.



Offs	Dec	Hex	Char
0100	64513	fc01	..
0102	263	0107	..
0104	0	0000	..
0106	0	0000	..
0108	0	0000	..
010a	0	0000	..
010c	0	0000	..
010e	0	0000	..
0110	22039	5617	.V
0112	0	0000	..
0114	0	0000	..

Bits	Name	Value	Enum
0-2	Fifo size	7	
3	Low E-Bus jitter	0	Deactive
8	Second address	0	Deactive

Figure 11-14

Click **“Device N (EtherCAT) -> Cmd 1 -> DATA”**, you will see the command get the value of ESC register 0x0012 back successfully. It is correct, the **“0x00ba”** in UINT data type.

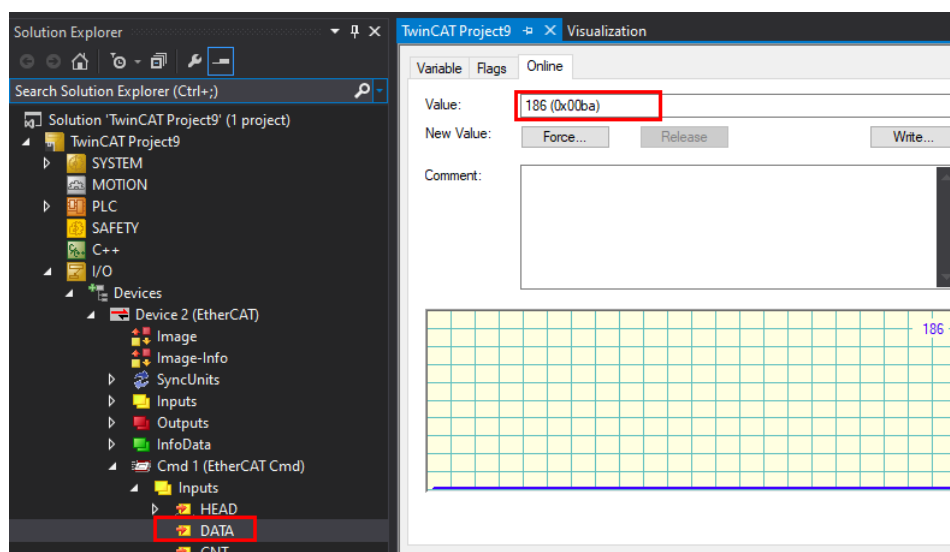
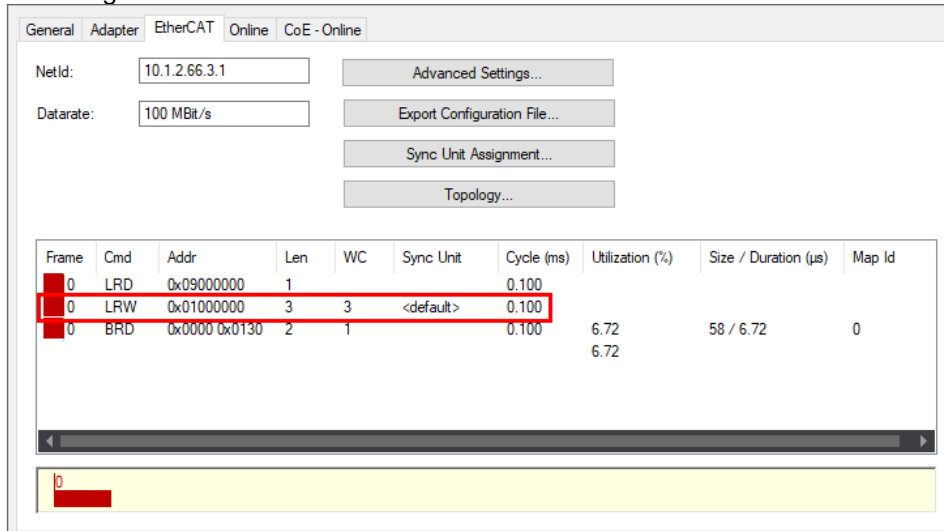


Figure 11-15

12. Shortest EtherCAT Cycle Evaluation

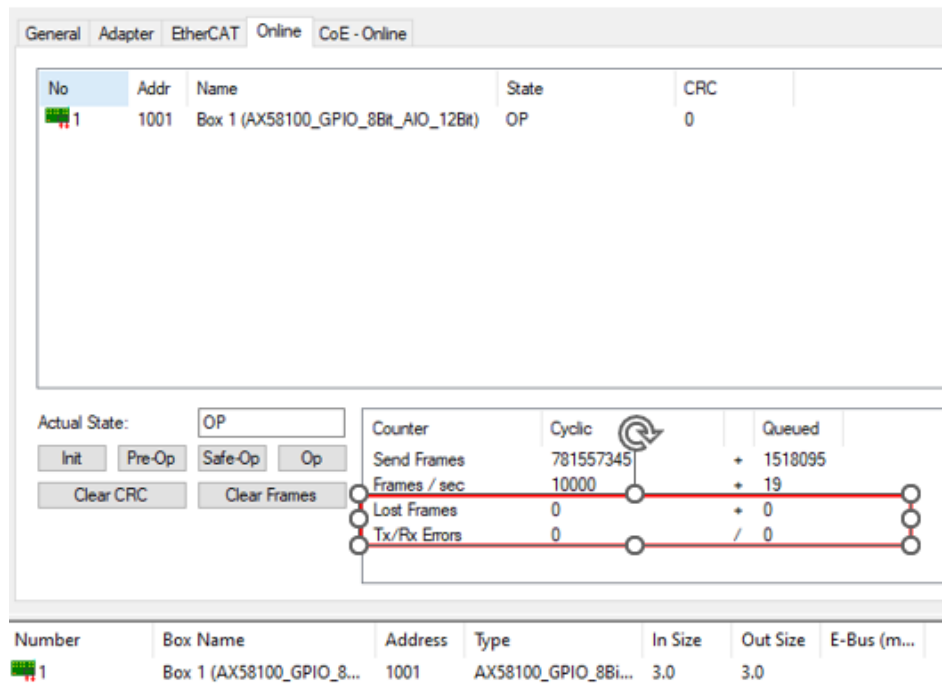
We tried to decrease the TwinCAT cycle to evaluate shortest cycle of AX58100 GPIO / AIO. Shortest workable cycle of AX58100 GPIO / AIO is “**100us**” now. This reference design allocates TX-PDO (3 bytes) and RX-PDO (3 bytes) totally. Please refer to below figures.



Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (us)	Map Id
0	LRD	0x09000000	1			0.100			
0	LRW	0x01000000	3	3	<default>	0.100			
0	BRD	0x0000 0x0130	2	1		0.100	6.72 6.72	58 / 6.72	0

Figure 12-1

Under this condition, the “**Lost Frames**” and “**Tx/Rx Errors**” does not count.



No	Addr	Name	State	CRC
1	1001	Box 1 (AX58100_GPIO_8Bit_AIO_12Bit)	OP	0

Counter	Cyclic	Queued
Send Frames	781557345	+ 1518095
Frames / sec	10000	+ 19
Lost Frames	0	+ 0
Tx/Rx Errors	0	/ 0

Number	Box Name	Address	Type	In Size	Out Size	E-Bus (m...)
1	Box 1 (AX58100_GPIO_8...	1001	AX58100_GPIO_8Bi...	3.0	3.0	

Figure 12-2

The slave stack also does not indicate any errors in SM output parameter / SM input parameter objects of CoE-Online table.

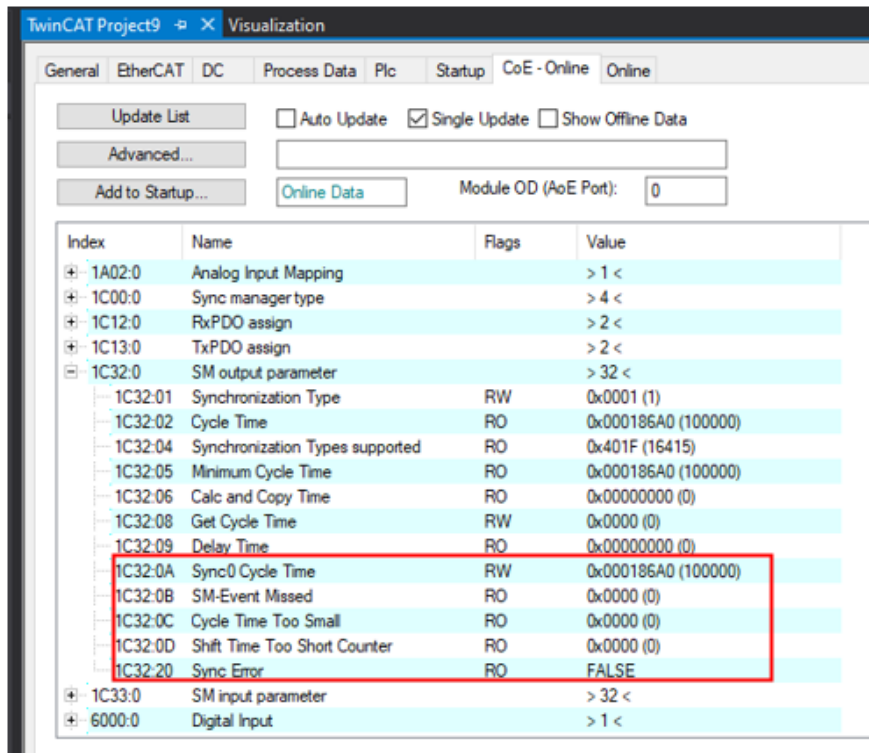


Figure 12-3

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 © 2025 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.



**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>