

REV01
User Manual 2020/01/02

Firmware Development Guide for SES_PUMP DSP C6711

Table of Contents

CHAPTER 1. CCS8.1 에서 개발 시작하기	오류! 책갈피가 정의되어 있지 않습니다.
1.1 개발 환경 구축하기.....	오류! 책갈피가 정의되어 있지 않습니다.
1.2 새 프로젝트 만들기.....	오류! 책갈피가 정의되어 있지 않습니다.
1.3 제이텍 연결하기.....	오류! 책갈피가 정의되어 있지 않습니다.
1.4 DSP/BIOS 설정.....	오류! 책갈피가 정의되어 있지 않습니다.
CHAPTER 2. TMS320C6657 기능.....	오류! 책갈피가 정의되어 있지 않습니다.
2-1. GPIO IN/OUT 사용 방법	오류! 책갈피가 정의되어 있지 않습니다.
2-2. UART 사용 방법.....	오류! 책갈피가 정의되어 있지 않습니다.
2-3. Ethernet 사용 방법.....	오류! 책갈피가 정의되어 있지 않습니다.
2-4. I2C 온도 센서 사용 방법.....	오류! 책갈피가 정의되어 있지 않습니다.
2-5. Memory 사용 방법	오류! 책갈피가 정의되어 있지 않습니다.

CHAPTER 1. 펌웨어 포팅

1.1 라이브러리 및 헤더 파일 변경

SES_PUMP DSP C6711 개발환경을 구축하기 위해서는 우선 개발 툴인 CCS(Code Composer Studio)를 설치 하여야 합니다. 현재 개발 환경 구축에 관한 내용은 CCS 8.1 버전 기준입니다.

1. ComiLib.lib 파일을 프로젝트 내 복사 합니다.

.launches	2020-01-02 오전...	파일 폴더	
.settings	2020-01-02 오전...	파일 폴더	
Debug	2020-01-02 오전...	파일 폴더	
ESCSStack	2020-01-02 오전...	파일 폴더	
mk_lib	2020-01-02 오전...	파일 폴더	
targetConfigs	2020-01-02 오전...	파일 폴더	
.ccsproject	2019-12-02 오후...	CCSPROJECT 파일	1KB
.cproject	2019-12-02 오후...	CPROJECT 파일	24KB
.project	2019-05-02 오후...	PROJECT 파일	1KB
.xdchelp	2018-03-12 오후...	XDCHHELP 파일	0KB
boot_c671x.asm...	2019-06-11 오후...	ASM_ 파일	7KB
C6711.cmd	2019-05-02 오후...	Windows 명령어 ...	3KB
c6711_hrrm.h	2019-05-31 오후...	C/C++ Header	10KB
ceNM_PEGel	2008-08-12 오후...	GEL 파일	17KB
ceNM_PETcf	2019-07-16 오후...	TextualConfig	14KB
ComiLib.lib	2019-12-20 오후...	Object File Library	1,402KB
Defines.h	2013-08-26 오후...	C/C++ Header	7KB
dvC671xBootLoader.asm	2019-05-24 오후...	Assembler Source	8KB
Functions.h	2008-10-27 오후...	C/C++ Header	1KB
GlobVar.h	2008-11-04 오후...	C/C++ Header	1KB
main.c	2020-01-02 오전...	C Source	5KB
main.c.bak	2019-05-31 오후...	BAK 파일	8KB
project.log	2019-12-02 오후...	텍스트 문서	1KB

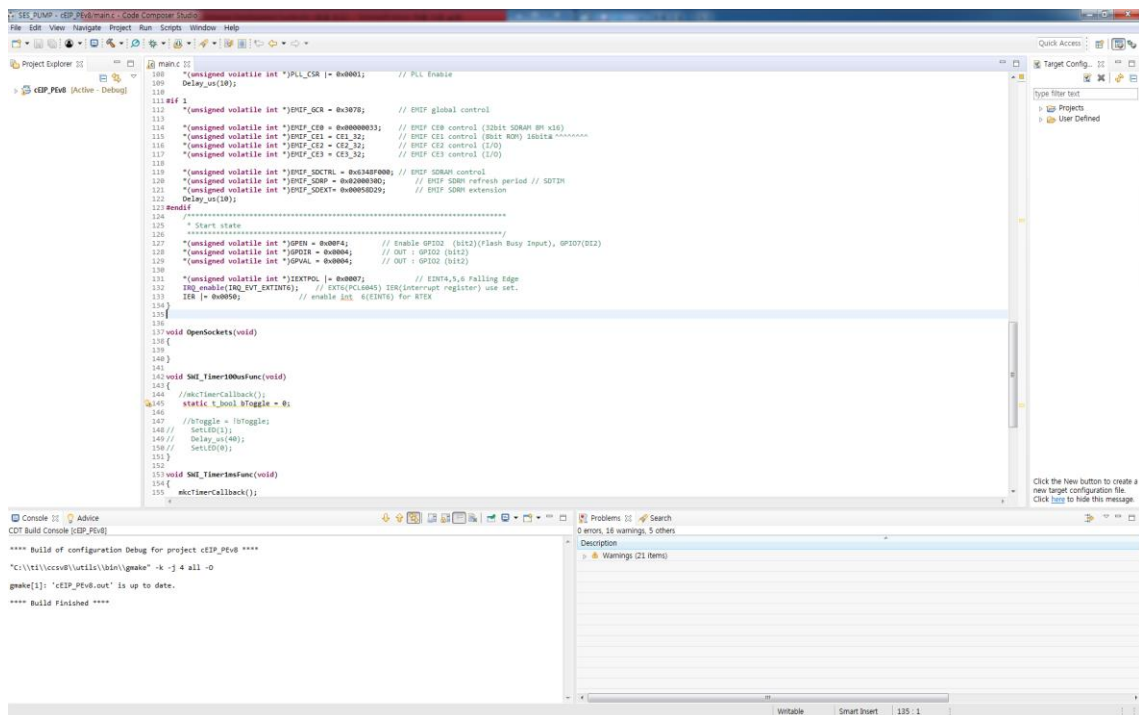
2. 예제 프로젝트 폴더 내에 mk_lib 폴더를 사용 프로젝트에 복사 합니다.

.launches	2020-01-02 오전...	파일 폴더	
.settings	2020-01-02 오전...	파일 폴더	
Debug	2020-01-02 오전...	파일 폴더	
ESCSStack	2020-01-02 오전...	파일 폴더	
mk_lib	2020-01-02 오전...	파일 폴더	
targetConfigs	2020-01-02 오전...	파일 폴더	
.ccsproject	2019-12-02 오후...	CCSPROJECT 파일	1KB
.cproject	2019-12-02 오후...	CPROJECT 파일	24KB
.project	2019-05-02 오후...	PROJECT 파일	1KB
.xdchelp	2018-03-12 오후...	XDCHHELP 파일	0KB
boot_c671x.asm...	2019-06-11 오후...	ASM_ 파일	7KB
C6711.cmd	2019-05-02 오후...	Windows 명령어 ...	3KB
c6711_hrrm.h	2019-05-31 오후...	C/C++ Header	10KB
ceNM_PEGel	2008-08-12 오후...	GEL 파일	17KB
ceNM_PETcf	2019-07-16 오후...	TextualConfig	14KB
ComiLib.lib	2019-12-20 오후...	Object File Library	1,402KB
Defines.h	2013-08-26 오후...	C/C++ Header	7KB
dvC671xBootLoader.asm	2019-05-24 오후...	Assembler Source	8KB
Functions.h	2008-10-27 오후...	C/C++ Header	1KB
GlobVar.h	2008-11-04 오후...	C/C++ Header	1KB
main.c	2020-01-02 오전...	C Source	5KB
main.c.bak	2019-05-31 오후...	BAK 파일	8KB
project.log	2019-12-02 오후...	텍스트 문서	1KB

3. 예제 프로젝트 폴더 내에 ESCStack 폴더를 사용 프로젝트에 복사 합니다.

.launches	2020-01-02 오전...	파일 폴더	
.settings	2020-01-02 오전...	파일 폴더	
Debug	2020-01-02 오전...	파일 폴더	
ESCStack	2020-01-02 오전...	파일 폴더	
mk_lib	2020-01-02 오전...	파일 폴더	
targetConfigs	2020-01-02 오전...	파일 폴더	
.ccsproject	2019-12-02 오후...	CCSPROJECT 파일	1KB
.cproject	2019-12-02 오후...	CPROJECT 파일	24KB
.project	2019-05-02 오후...	PROJECT 파일	1KB
.xdchelp	2018-03-12 오후...	XDCHelp 파일	0KB
boot_c671x.asm	2019-06-11 오후...	ASM_ 파일	7KB
C6711.cmd	2019-05-02 오후...	Windows 명령어 ...	3KB
c6711_hrrm.h	2019-05-31 오후...	C/C++ Header	10KB
ceNM_PE.gel	2008-08-12 오후...	GEL 파일	17KB
ceNM_PETcf	2019-07-16 오후...	TextualConfig	14KB
ComLib.lib	2019-12-20 오후...	Object File Library	1,402KB
Defines.h	2013-08-26 오후...	C/C++ Header	7KB
dvC671xBootLoader.asm	2019-05-24 오후...	Assembler Source	8KB
Functions.h	2008-10-27 오후...	C/C++ Header	1KB
GlobVar.h	2008-11-04 오후...	C/C++ Header	1KB
main.c	2020-01-02 오전...	C Source	5KB
main.c.bak	2019-05-31 오후...	BAK 파일	8KB
project.log	2019-12-02 오후...	텍스트 문서	1KB

4. 프로젝트를 실행하여 빌드합니다.



```

100 * (unsigned volatile int *)JPLI_CSR |= 0x0001; // PLL Enable
101 Delay_us(10);
102
103 #if 1
104 * (unsigned volatile int *)JSHF_GCR = 0x3078; // SHF global control
105
106 * (unsigned volatile int *)JSHF_CER = 0x00000033; // SHF CER control (Sbit SORAH BR x16)
107 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
108 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
109 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
110 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
111 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
112 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
113 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
114 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
115 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
116 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
117 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
118 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
119 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
120 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
121 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
122 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
123 #endif
124
125 // Start state
126
127 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
128 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
129 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
130 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
131 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
132 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
133 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
134 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
135 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
136 * (unsigned volatile int *)JSHF_CEL = C_E1_32; // SHF CEL control (Sbit NDI) Sbita *****
137 void OpenSockets(void)
138 {
139 }
140
141
142 void SHF_Timer100usFunc(void)
143 {
144 // TimerCallback();
145 static unsigned long RToggle = 0;
146 // RToggle = !RToggle;
147 // SetLED(1);
148 // Delay_us(100);
149 // SetLED(0);
150 }
151
152 void SHF_Timer1msFunc(void)
153 {
154 // TimerCallback();
155 }

```

Console Output:

```

C:\ti\ccsv8\utils\bin\gmake -k -j 4 all -O
gmake[1]: 'cESP_PV08.out' is up to date.
**** Build Finished ****

```

1.2 DIO, AIO 채널 확인

알파 버전과 베타 버전 보드 내의 Digital I/O, Analog I/O의 접점수에 변화가 있기에 구현 코드의 Digital, Analog의 채널번호의 변화를 확인 하여야 합니다.

	알파 버전	베타 버전
제어 PUMP 모듈	8	4
제어 VALVE 모듈	8	4
Digital Input	총 제어 : 24 Pump / Valve 1Set : 3	총 제어 : 12 Pump / Valve 1Set : 3
Digital Output	총 제어 : 32 Pump / Valve 1Set : 4	총 제어 : 32 Pump / Valve 1Set : 4
Analog Input	총 제어 : 88 Pump / Valve 1Set : 11	총 제어 : 36 Pump / Valve 1Set : 9
Analog Output	총 제어 : 16 Pump / Valve 1Set : 2	총 제어 : 16 Pump / Valve 1Set : 4
Motion	총 제어 : 8 Pump / Valve 1Set : 1	총 제어 : 4 Pump / Valve 1Set : 1

1. Digital Input 채널번호

- A. 1번 Pump/Valve 0~2
- B. 2번 Pump/Valve 3~5
- C. 3번 Pump/Valve 6~8
- D. 4번 Pump/Valve 9~11

2. Digital Output 채널번호

- A. 예비 0~15
- B. 1번 Pump/Valve 16~19
- C. 2번 Pump/Valve 20~23
- D. 3번 Pump/Valve 24~27
- E. 4번 Pump/Valve 28~31

3. Analog Input 채널번호

Pump/Valve 별 마지막 채널(8,17,26,35)은 RTD 채널 입니다.

- A. 1번 Pump/Valve 0~8
- B. 2번 Pump/Valve 9~17
- C. 3번 Pump/Valve 18~26
- D. 4번 Pump/Valve 27~35

4. Analog Output 채널번호

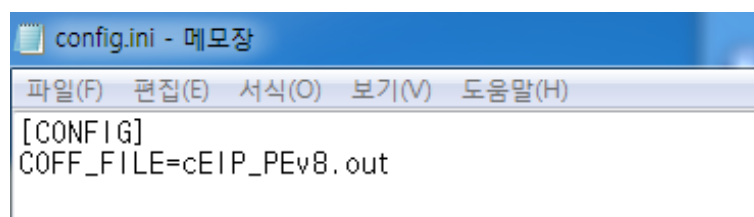
- A. 1번 Pump/Valve 0~3
- B. 2번 Pump/Valve 4~7
- C. 3번 Pump/Valve 8~11
- D. 4번 Pump/Valve 12~15

CHAPTER 2. 펌웨어 업데이트

2-1. .out 파일 .efw 만들기

TwinCAT 프로그램을 통해 펌웨어 업데이트 하기 위해서는 프로젝트 출력파일 .out 파일은 .efw 파일로 변경하여야 합니다.

1. Config.ini 파일 설정



Config.ini 파일을 오픈 하여 프로젝트 출력파일의 이름을 입력

2. 프로그램 실행

cEIP_PEv8.out	2020-01-02 오전...	Wireshark capture...	1,151KB
config.ini	2020-01-02 오전...	구성 설정	1KB
SES_Flasher.exe	2020-01-02 오후...	응용 프로그램	16KB

SES_Flasher.exe 프로그램 실행

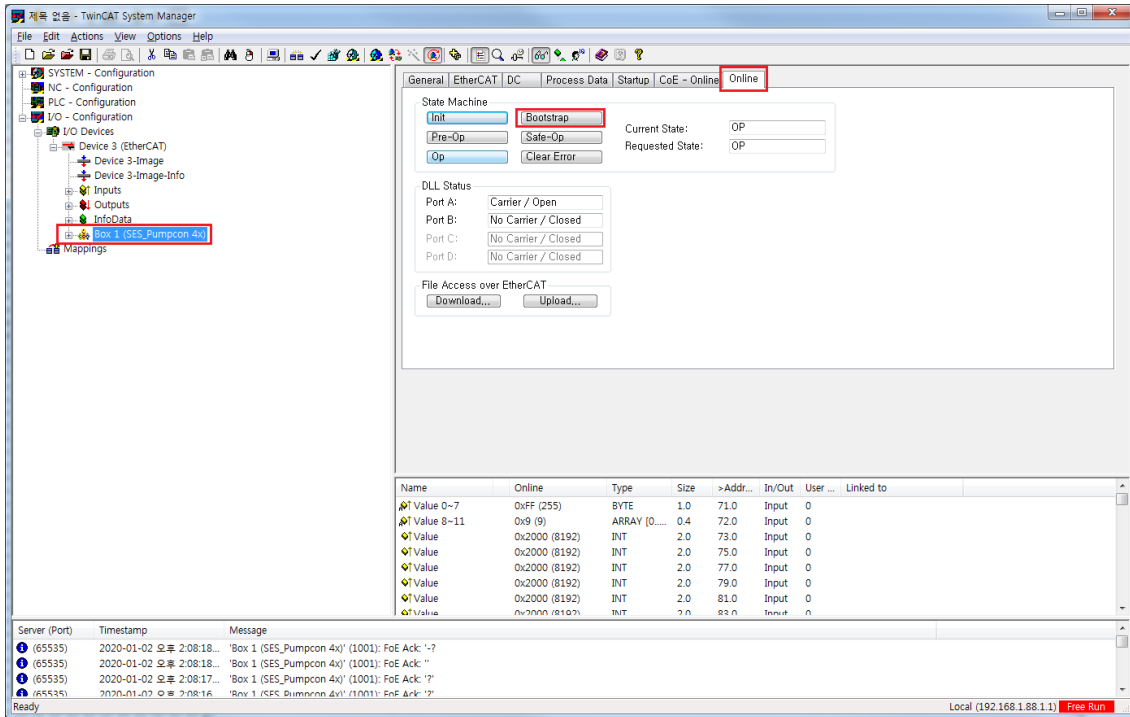
cEIP_PEv8.out	2020-01-02 오전...	Wireshark capture...	1,151KB
COMIZOA-cEIP_PEv8.efw	2020-01-02 오후...	EFW 파일	335KB
config.ini	2020-01-02 오전...	구성 설정	1KB
SES_Flasher.exe	2020-01-02 오후...	응용 프로그램	16KB

.out에서 COMIZOA-파일명.efw 파일 생성

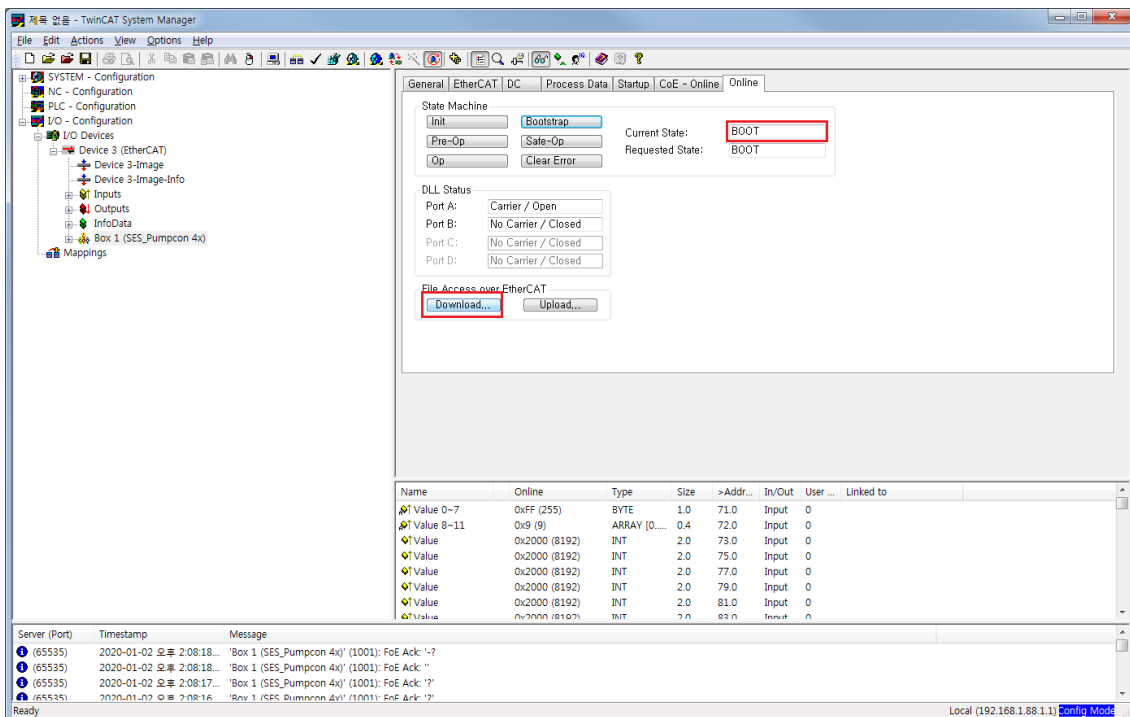
2-2. 펌웨어 다운로드

TwinCAT을 이용하여 .efw 파일 다운로드 방법

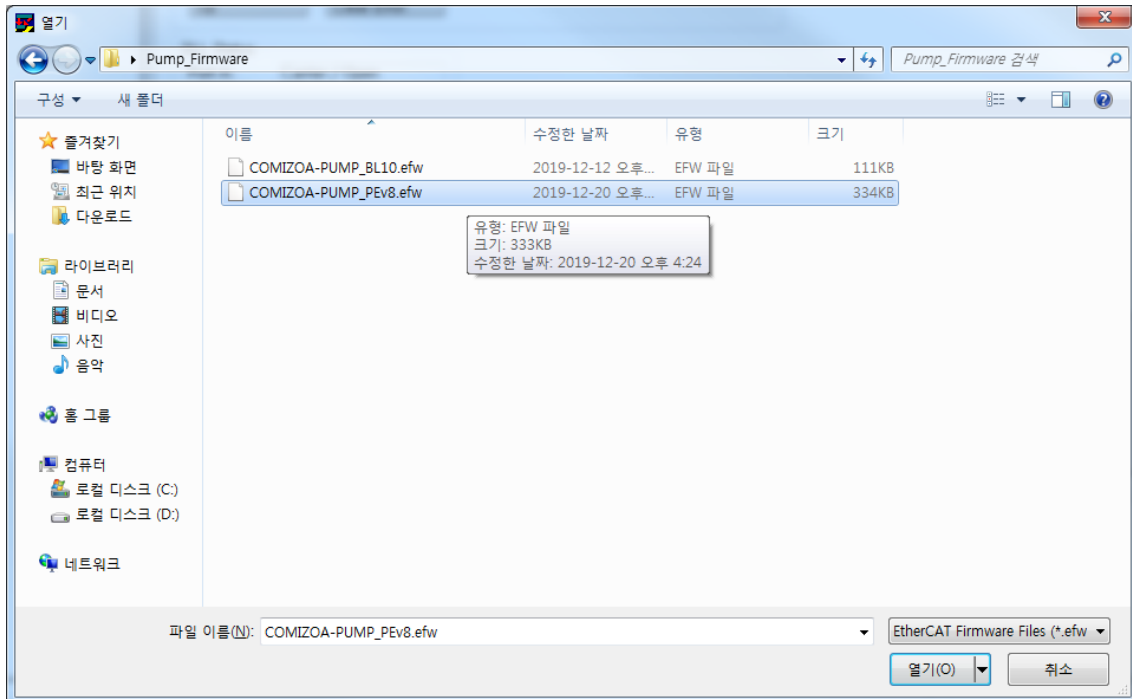
1. TwinCAT에 SES_PUMP 선택하여 Online을 선택하여 Bootstrap으로 설정



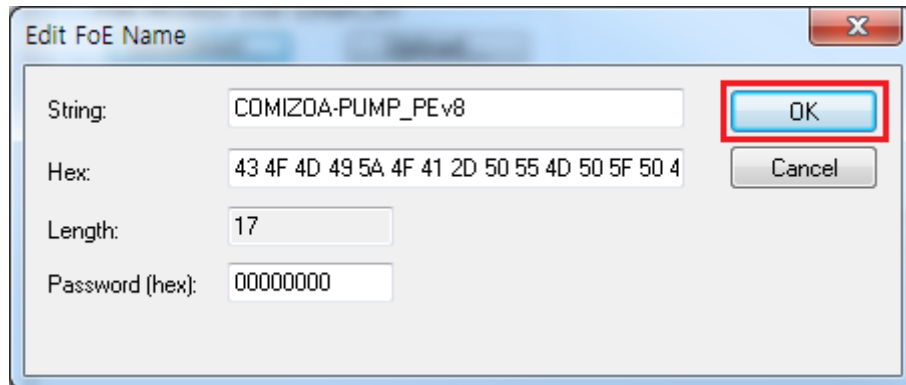
2. 현재 상태가 BOOT인지 확인 후 Download 선택



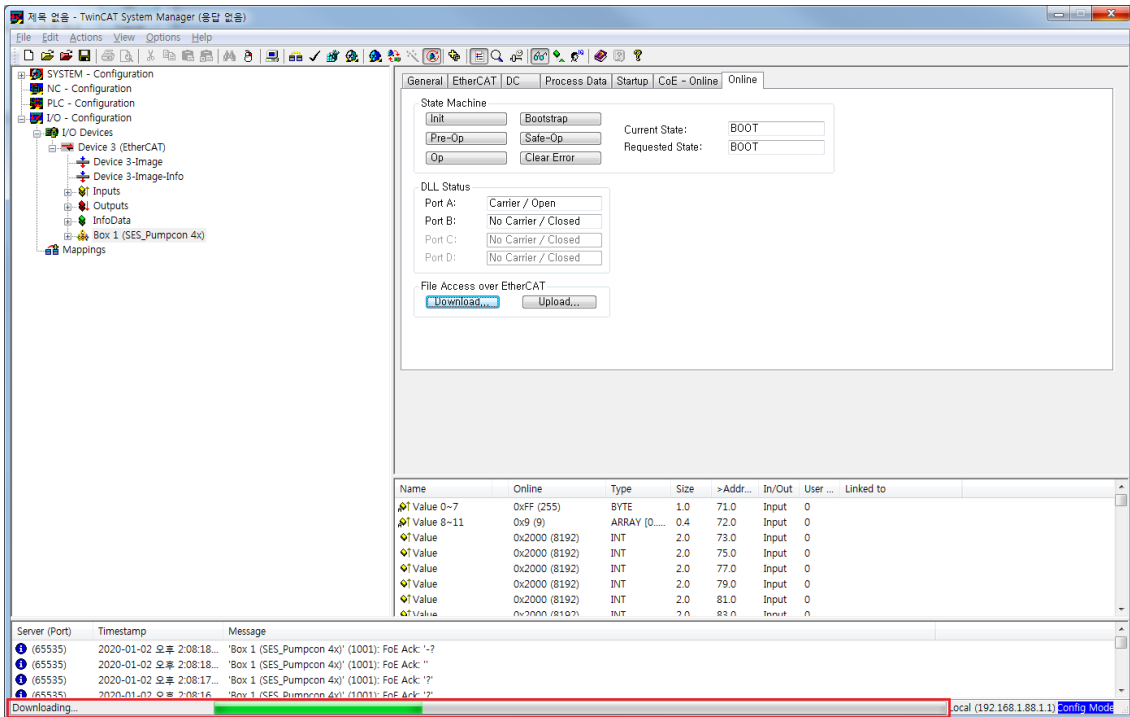
3. 생성된 .efw를 선택



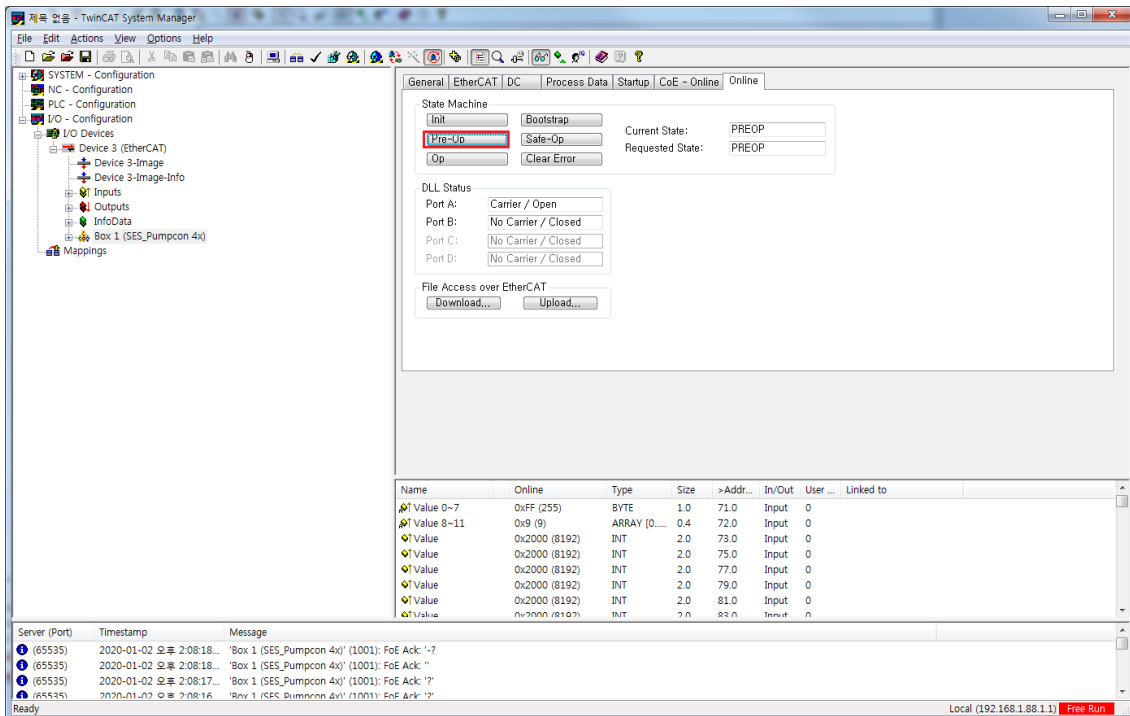
4. 확인버튼 클릭



5. 다운로드 완료 까지 대기



6. 완료 후 Pre-Op을 선택하여 상태 변화 확인



저작권자 : (주)커미조아

Copyright (c) by COMIZOA CO.,LTD. All rights reserved.

2020년 01월 02일 초판 발행.

이 사용자 설명서는 저작권법에 의해 보호되고 있습니다.

(주)커미조아의 사전 서면 동의 없이 사용자설명서의 일부 또는 전체를 어떤 형태로든
복사, 전재할 수 없습니다.

Hardware Support : hwteam@comizoa.com

Software Support : swteam@comizoa.com



(주)커미조아

www.comizoa.com

www.comizoa.co.kr

Tel) 042 - 936 - 6500

Fax) 042 - 936 - 6507