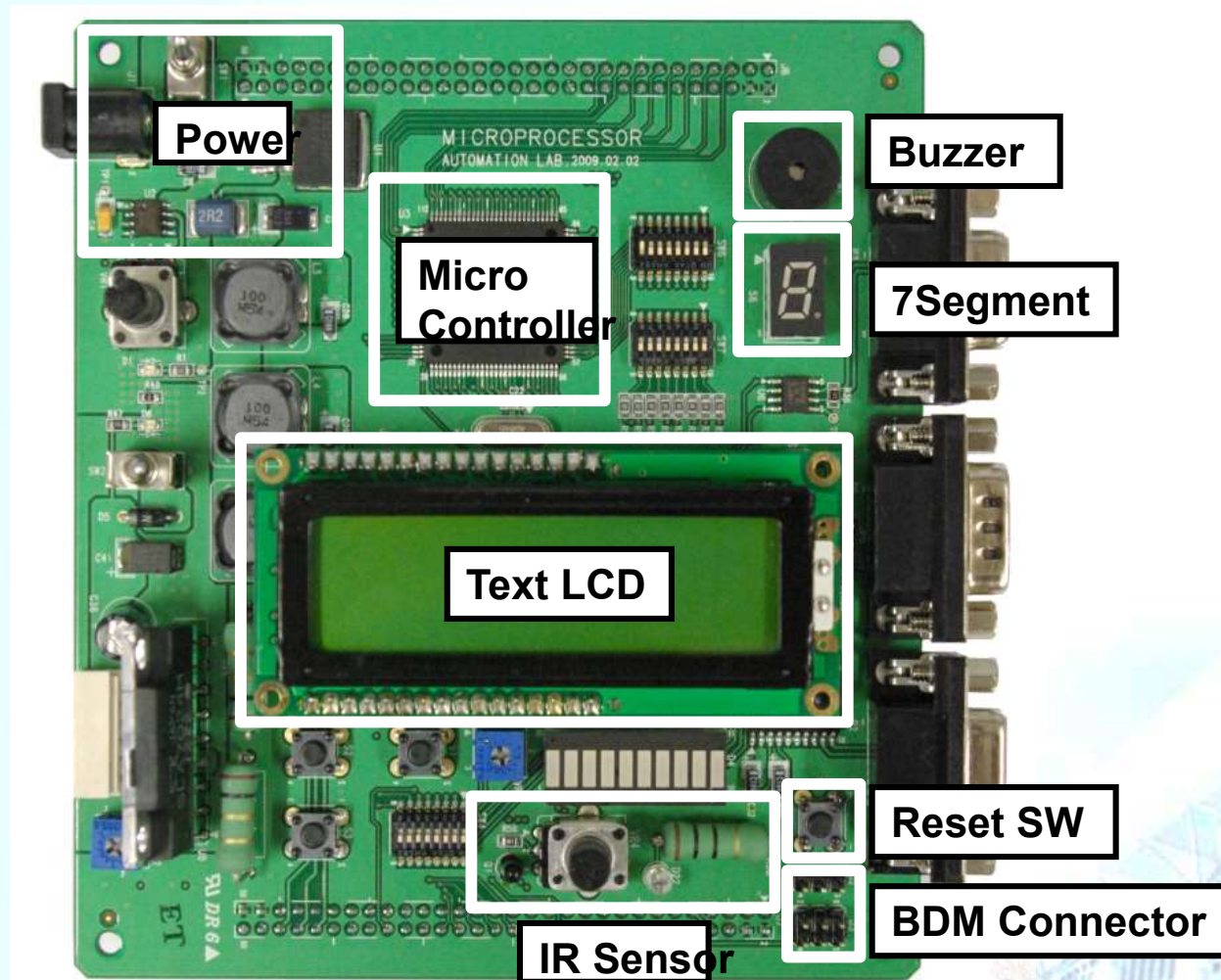


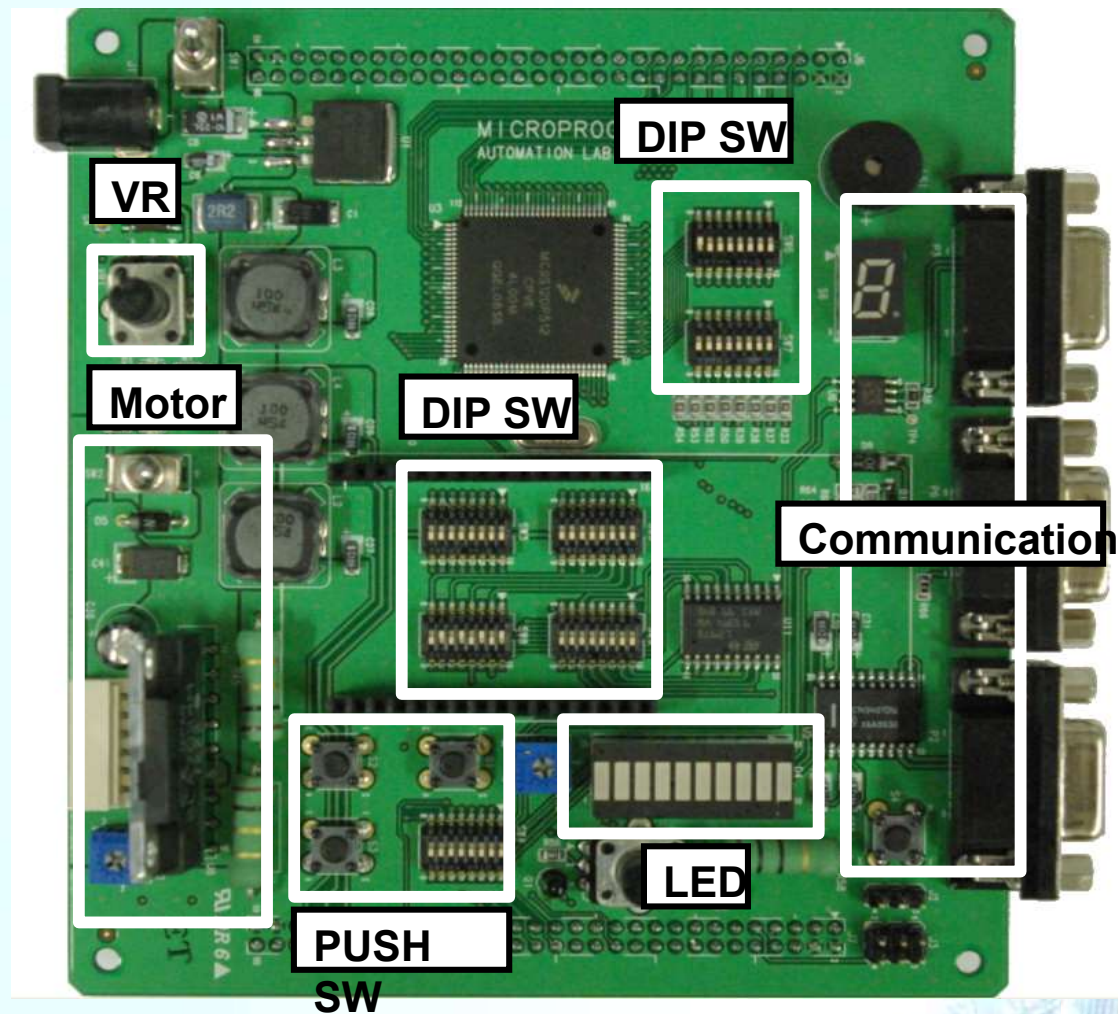
실습 보드

■ 보드 구성



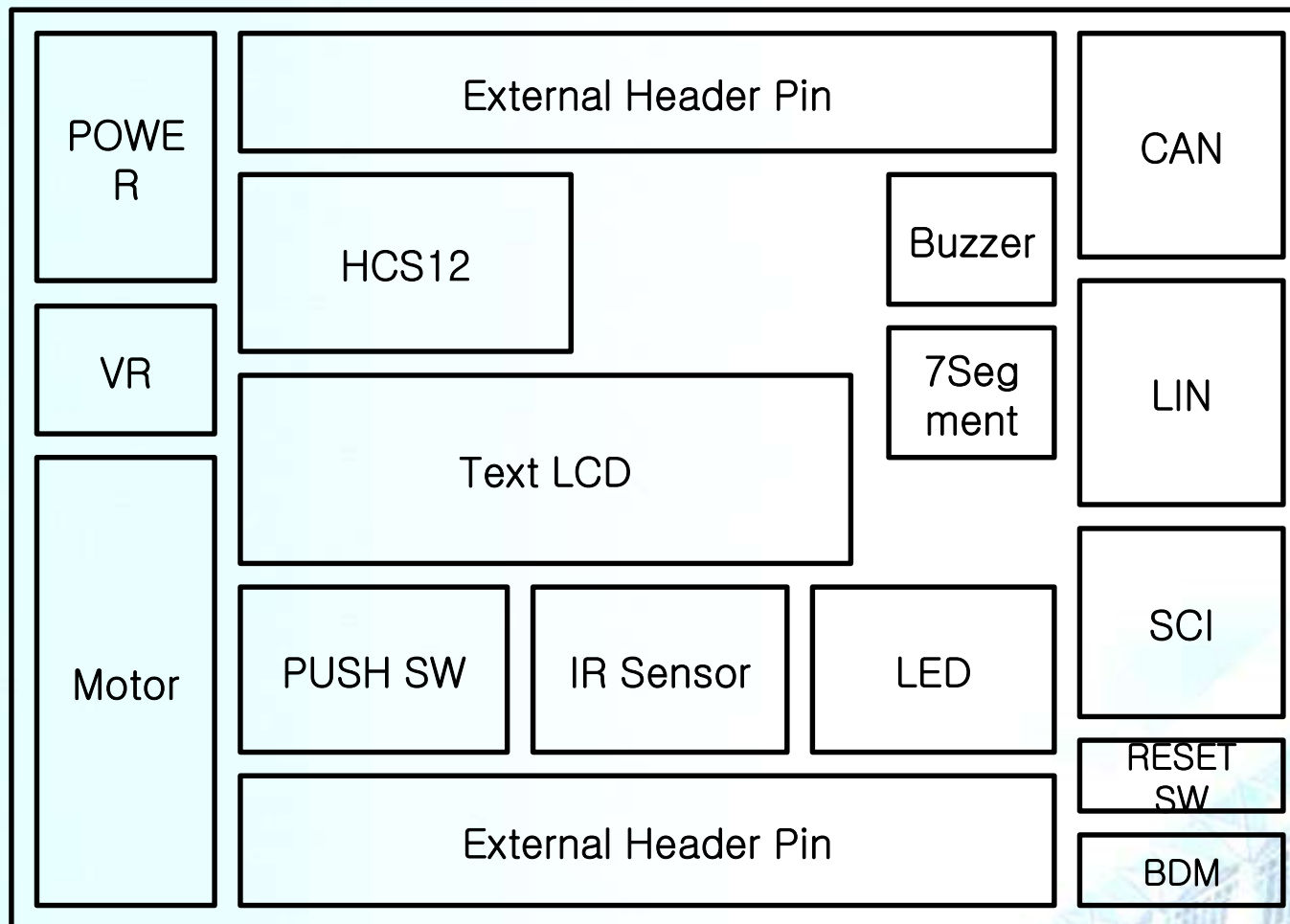
실습 보드

■ 보드 구성



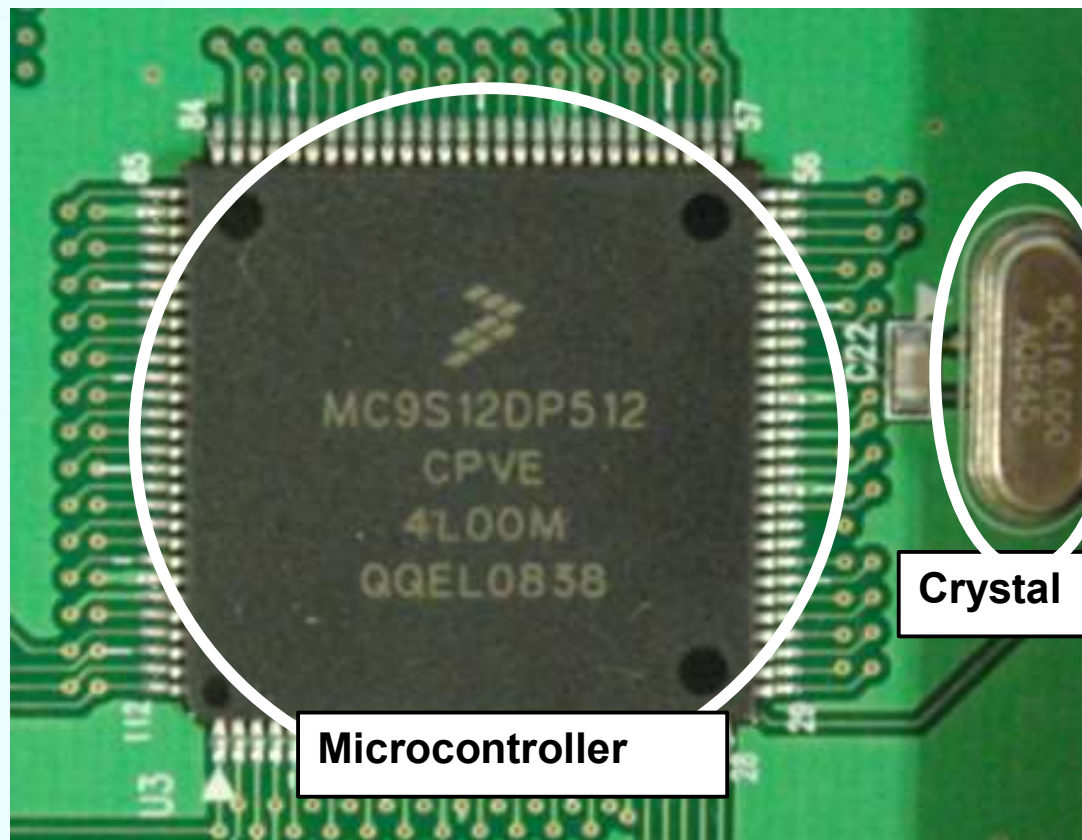
실습 보드

■ 보드 구성



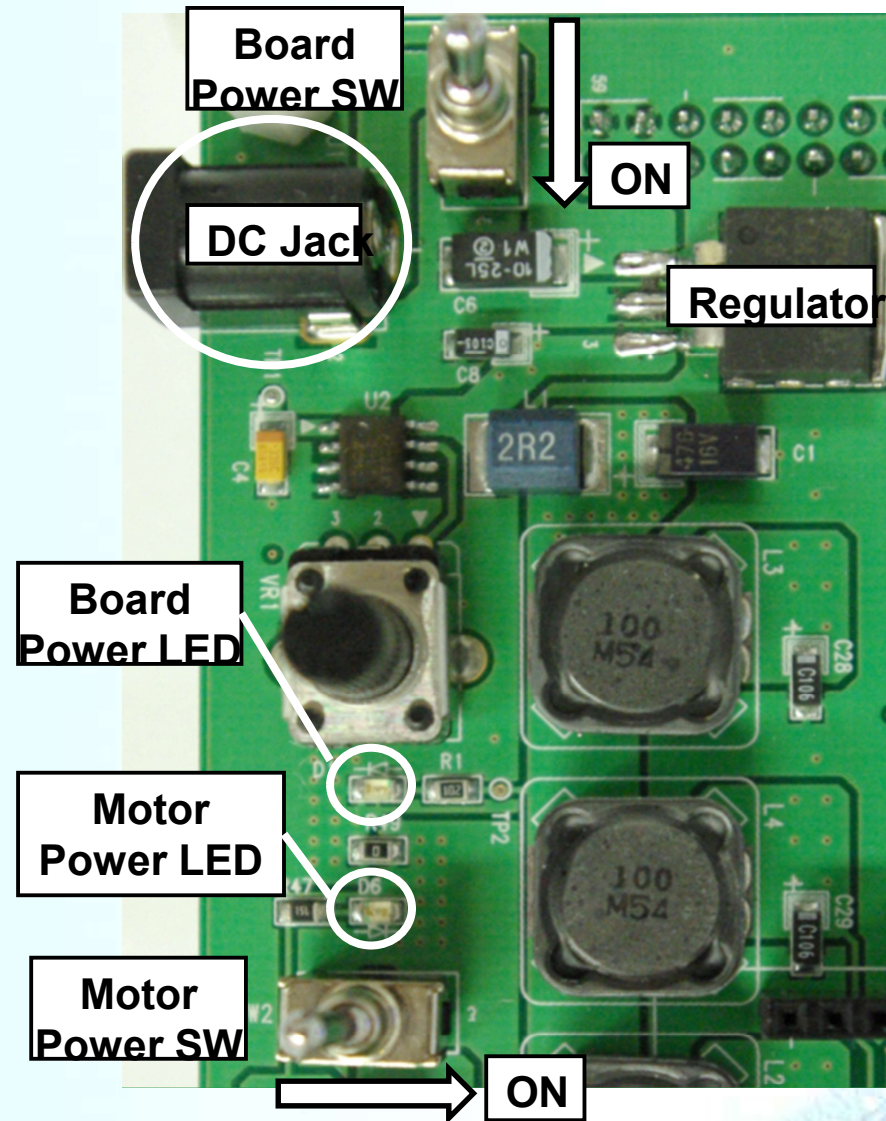
실습 보드

■ MCU



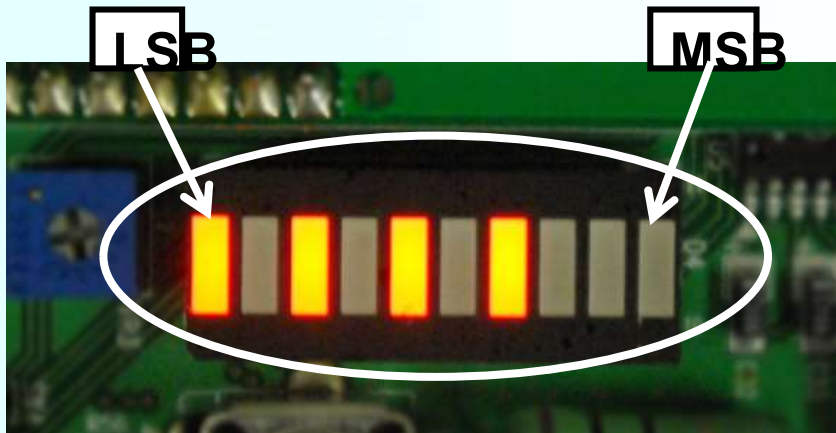
실습 보드

■ 전원 부



실습 보드

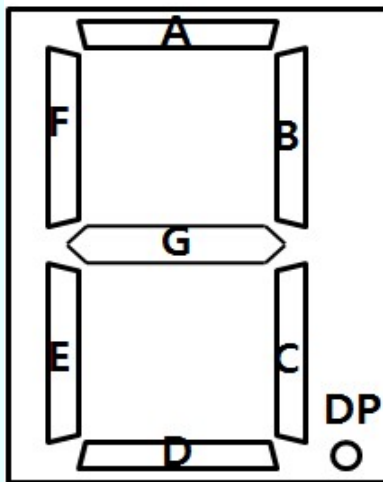
LED



번호	연결된 핀
①	포트 B 0번 핀
②	포트 B 1번 핀
③	포트 B 2번 핀
④	포트 B 3번 핀
⑤	포트 B 4번 핀
⑥	포트 B 5번 핀
⑦	포트 B 6번 핀
⑧	포트 B 7번 핀
⑨	포트 M 6번 핀
⑩	포트 M 7번 핀

실습 보드

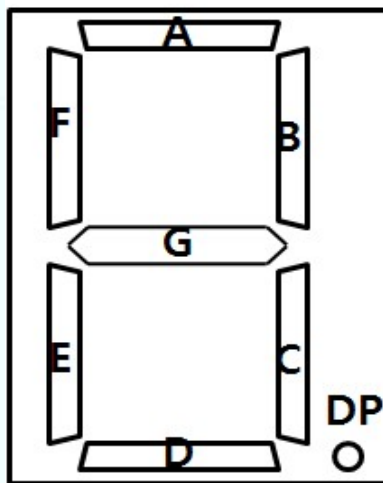
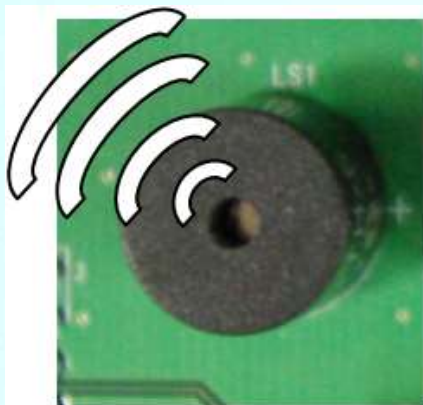
7SEGMENT



번호	연결된 핀
A	포트 H 0번 핀
B	포트 H 1번 핀
C	포트 H 2번 핀
D	포트 H 3번 핀
E	포트 H 4번 핀
F	포트 H 5번 핀
G	포트 H 6번 핀
DP	포트 H 7번 핀

실습 보드

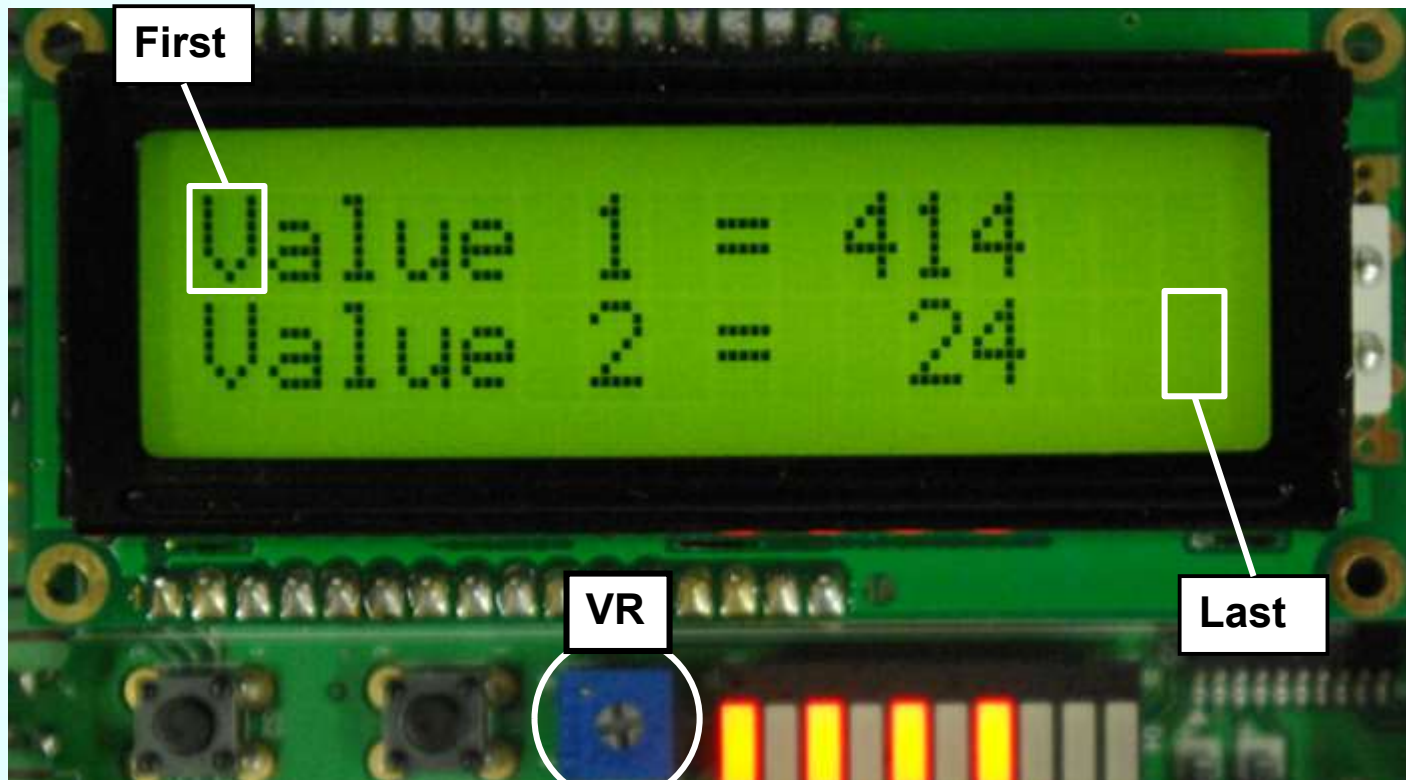
7SEGMENT & buzzer



번호	연결된 핀
A	포트 H 0번 핀
B	포트 H 1번 핀
C	포트 H 2번 핀
D	포트 H 3번 핀
E	포트 H 4번 핀
F	포트 H 5번 핀
G	포트 H 6번 핀
DP	포트 H 7번 핀
buzzer	포트 A 7번 핀

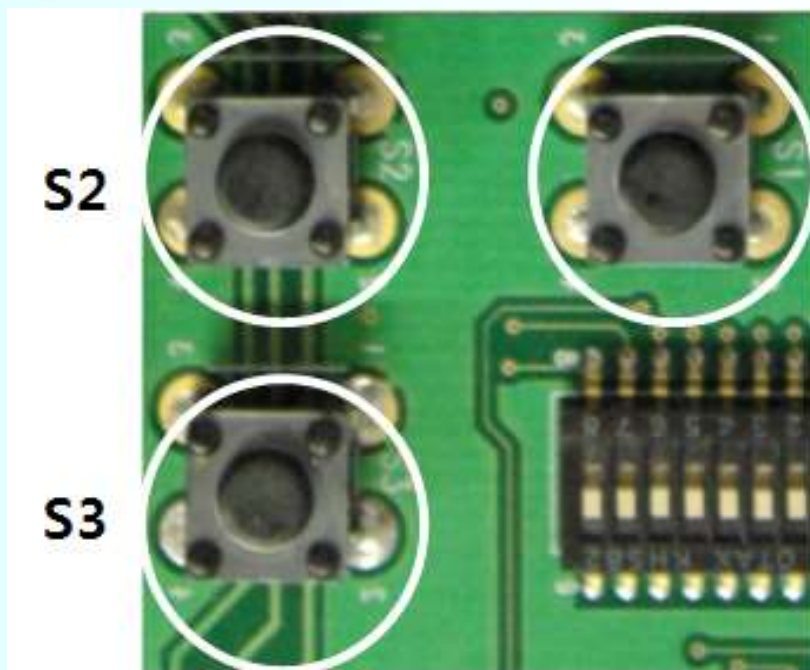
실습 보드

■ LCD



실습 보드

푸시 스위치



S1

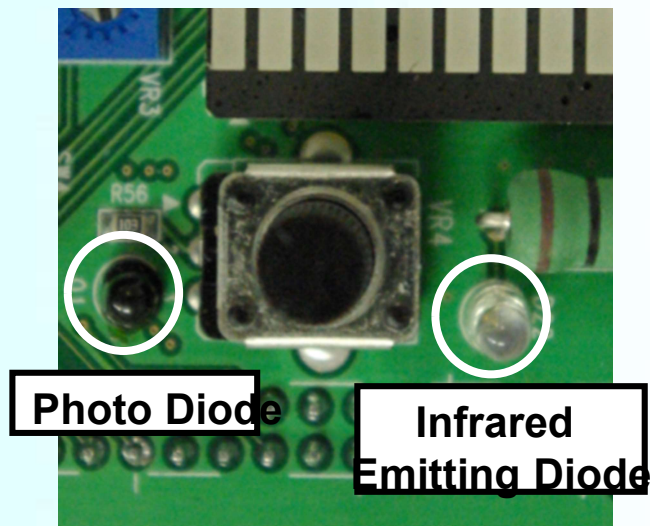
S2

S3

번호	연결된 핀
S1	XIRQ 핀
S2	포트 J 0번 핀
S3	포트 J 1번 핀

실습 보드

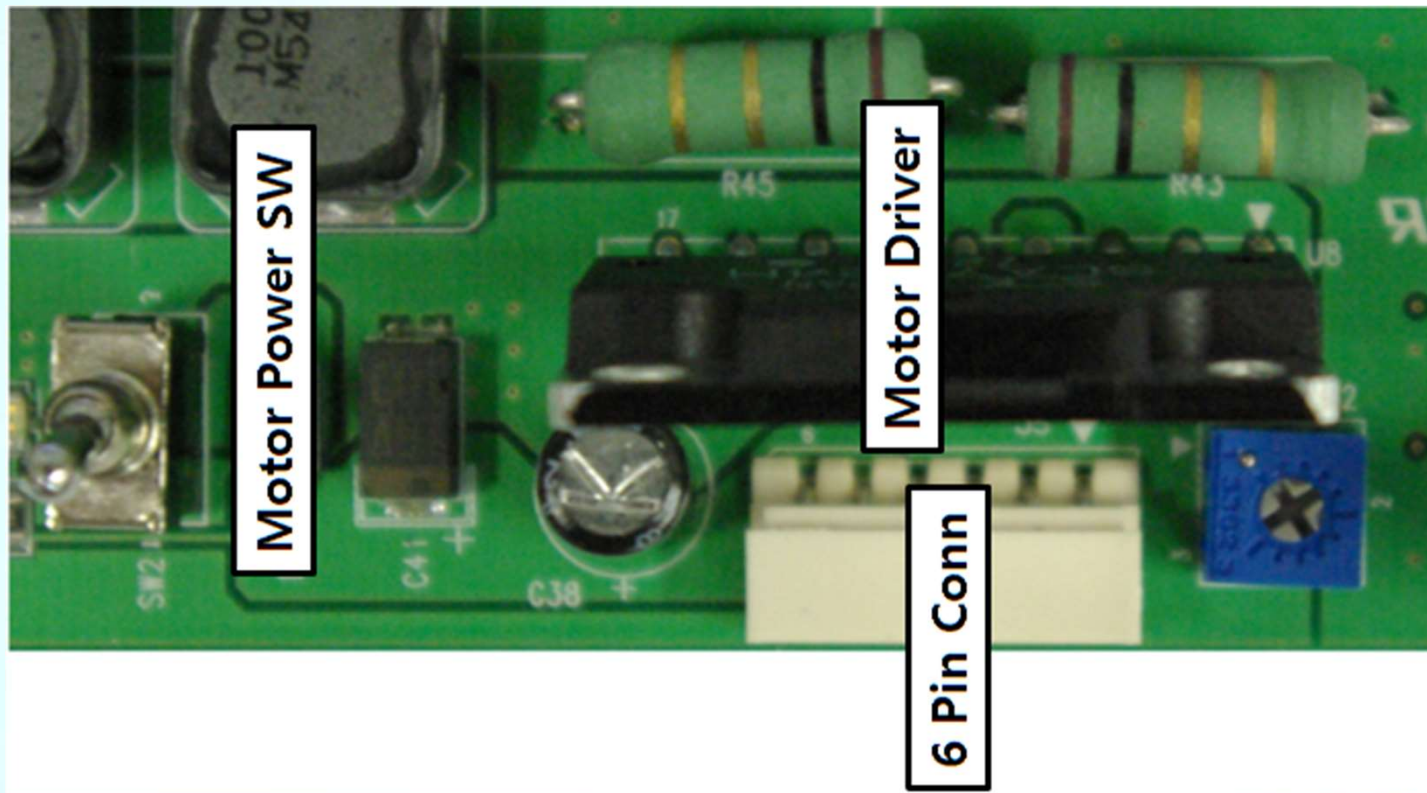
■ INFRARED SENSOR & VR



번호	연결된 핀
발광부	포트 K 0번 핀
수광부	포트 AD 1번 핀
전원브측 가변저항	포트 AD 0번 핀

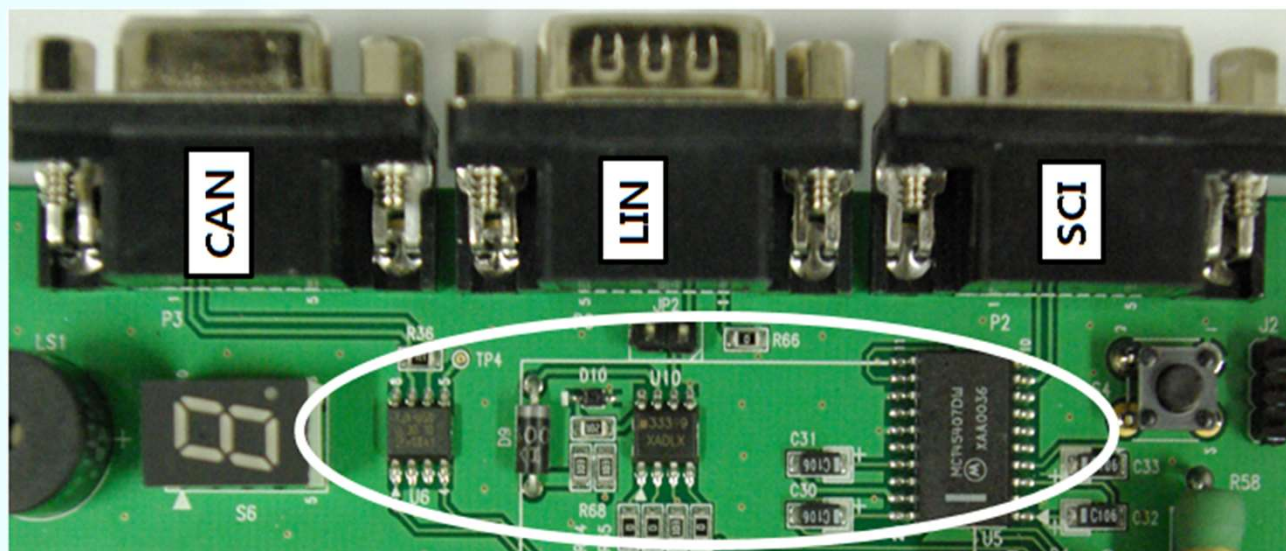
실습 보드

■ MOTOR MODULE



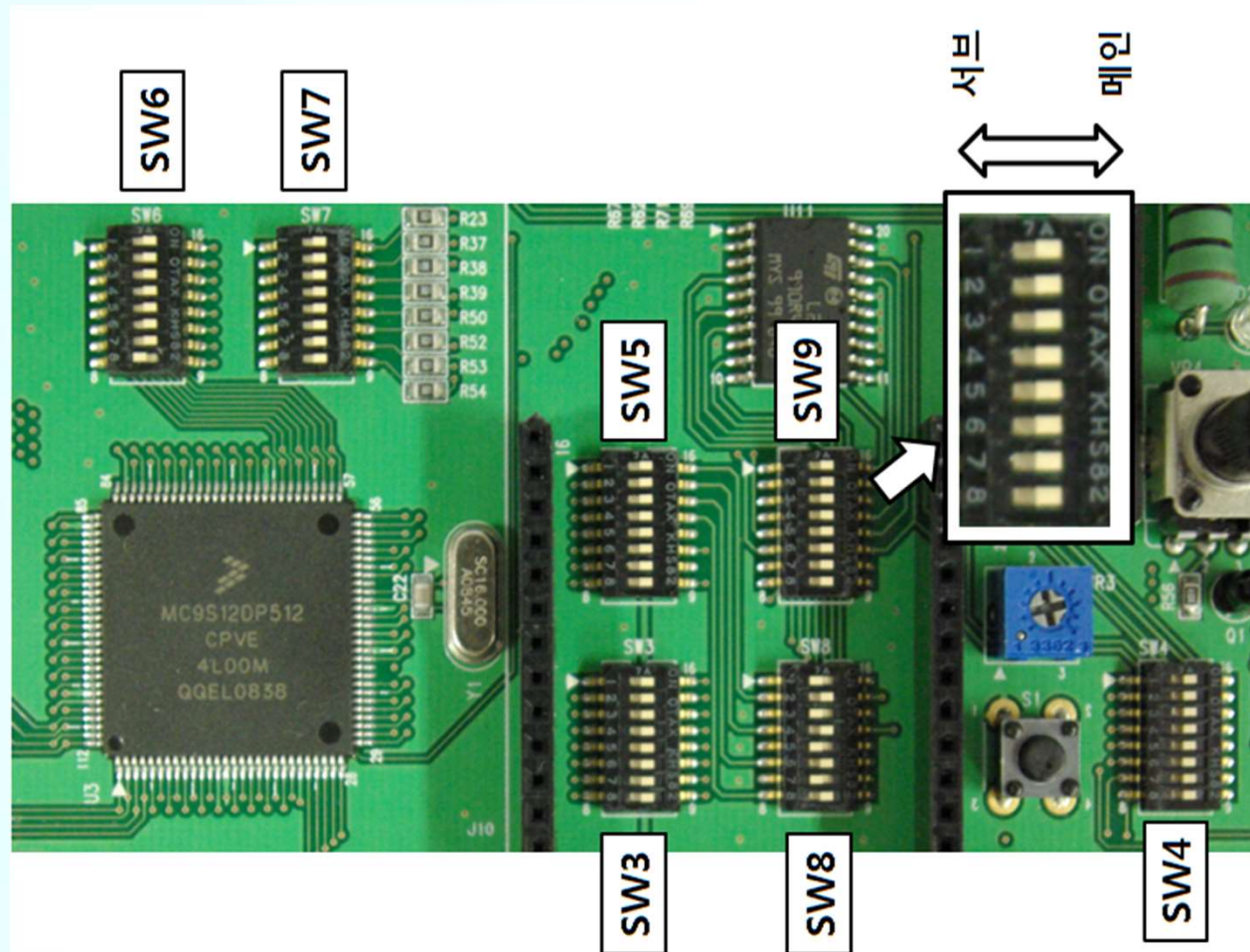
실습 보드

■ COMMUNICATION



실습 보드

■ DIP 스위치



실습 보드

■ DIP 스위치

번호	제어 부분
SW3	LED ①~⑧ 번 출력 신호
SW4	SCI, CAN, LIN 통신 송수신 신호, Push 스위치 2번, 3번 입력 신호
SW5	PWM 신호, 아날로그-디지털 변환 신호
SW6	텍스트 LCD 출력 및 제어 신호, 부저 출력 신호
SW7	7세그먼트 출력 신호
SW8	DC 모터/스텝 모터 선택 신호
SW9	LED ⑨,⑩번 출력 신호

SW3과 연결된 핀

번호	연결된 핀
1	포트 B 0번 핀
2	포트 B 1번 핀
3	포트 B 2번 핀
4	포트 B 3번 핀
5	포트 B 4번 핀
6	포트 B 5번 핀
7	포트 B 6번 핀
8	포트 B 7번 핀

실습 보드

■ DIP 스위치

SW4와 연결된 핀

번호	연결된 핀
1	RXD0 핀
2	TXD0 핀
3	RXD1 핀
4	TXD1 핀
5	RXCAN0 핀
6	TXCAN0 핀
7	포트 J 0번 핀
8	포트 J 1번 핀

SW5와 연결된 핀

번호	연결된 핀
1	PWM 0번 핀
2	PWM 1번 핀
3	PWM 2번 핀
4	PWM 3번 핀
5	포트 AD 0번 핀
6	포트 AD 1번 핀
7	포트 AD 2번 핀
8	포트 E 0번 핀

SW6과 연결된 핀

번호	연결된 핀
1	포트 A 0번 핀
2	포트 A 1번 핀
3	포트 A 2번 핀
4	포트 A 3번 핀
5	포트 A 4번 핀
6	포트 A 5번 핀
7	포트 A 6번 핀
8	포트 A 7번 핀

SW7과 연결된 핀

번호	연결된 핀
1	포트 H 0번 핀
2	포트 H 1번 핀
3	포트 H 2번 핀
4	포트 H 3번 핀
5	포트 H 4번 핀
6	포트 H 5번 핀
7	포트 H 6번 핀
8	포트 H 7번 핀

SW8과 연결된 핀

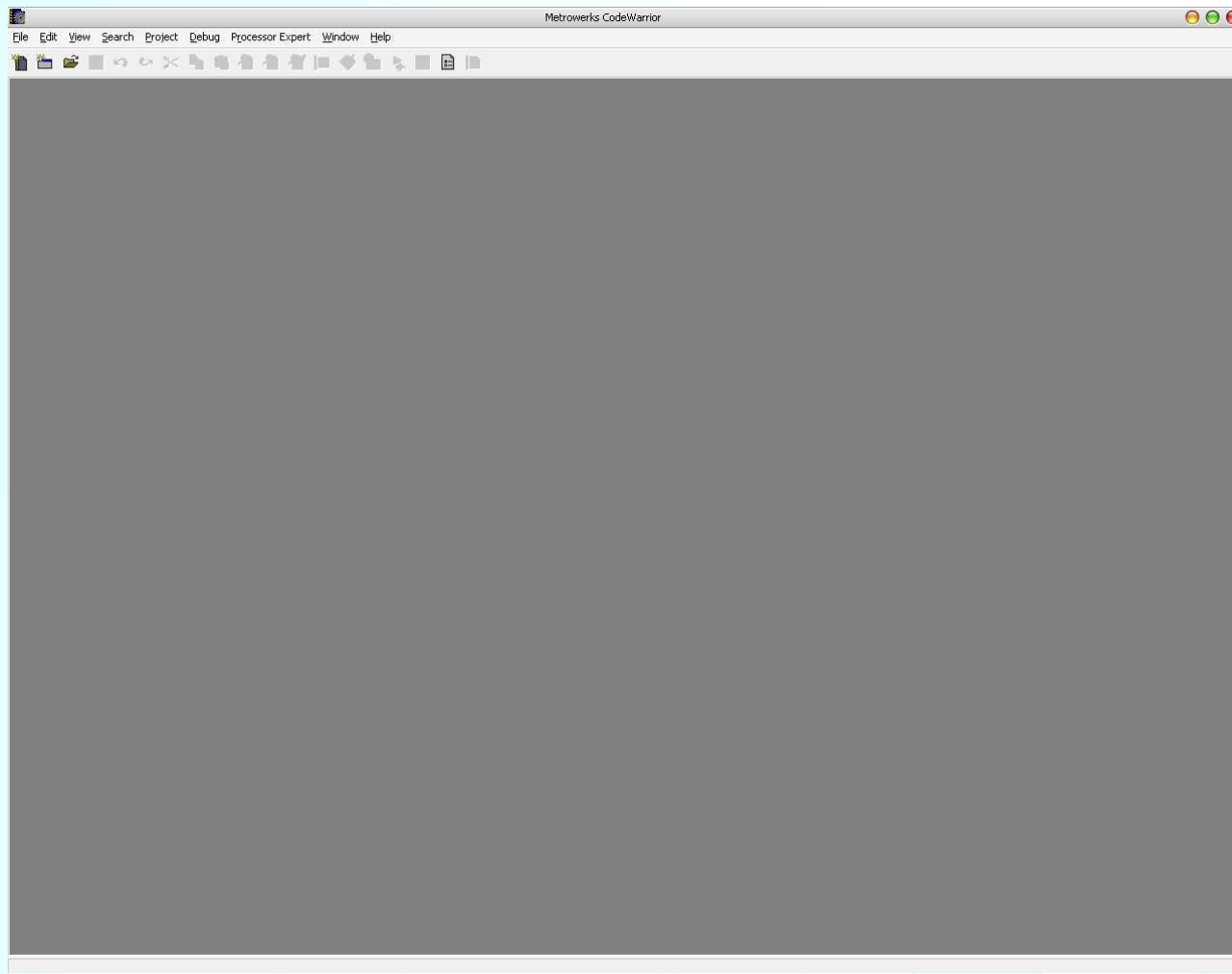
번호	연결된 핀
1	DC 모터 0번 핀
2	스텝 모터 0번 핀
3	DC 모터 1번 핀
4	스텝 모터 1번 핀
5	DC 모터 2번 핀
6	스텝 모터 2번 핀
7	DC 모터 3번 핀
8	스텝 모터 3번 핀

SW9와 연결된 핀

번호	연결된 핀
1	포트 M 2번 핀
2	포트 M 3번 핀
3	포트 M 4번 핀
4	포트 M 5번 핀
5	포트 M 6번 핀
6	포트 M 7번 핀
7	연결 없음
8	연결 없음

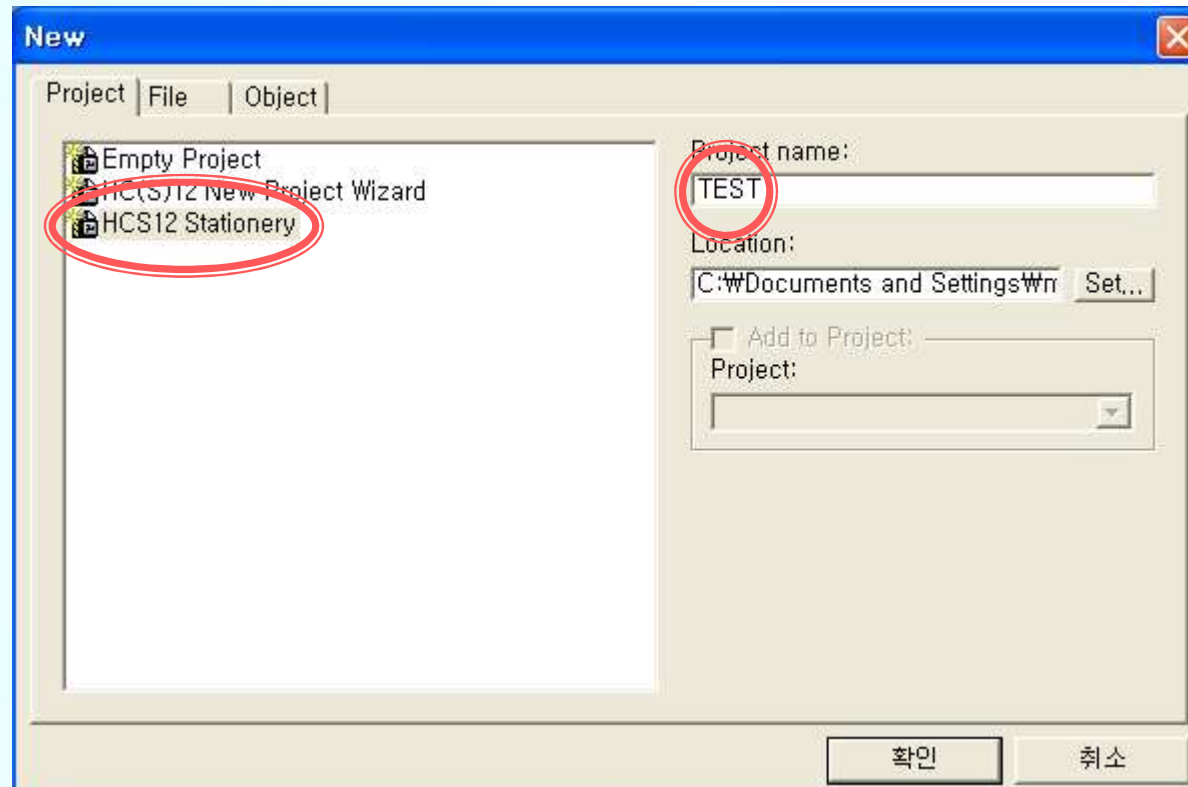
Development Tool

■ Codewarrior IDE



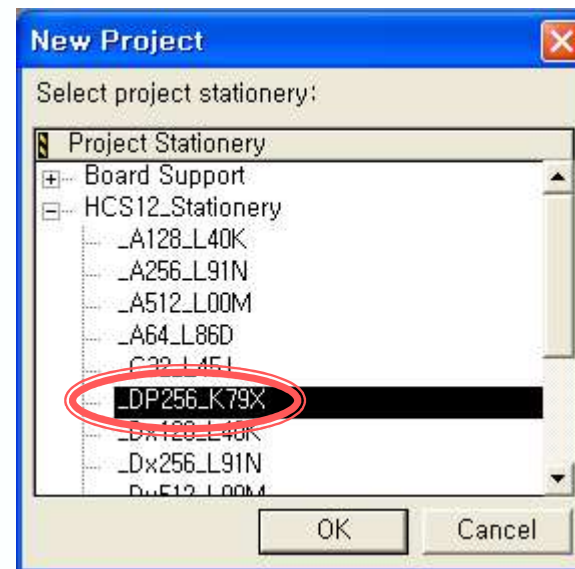
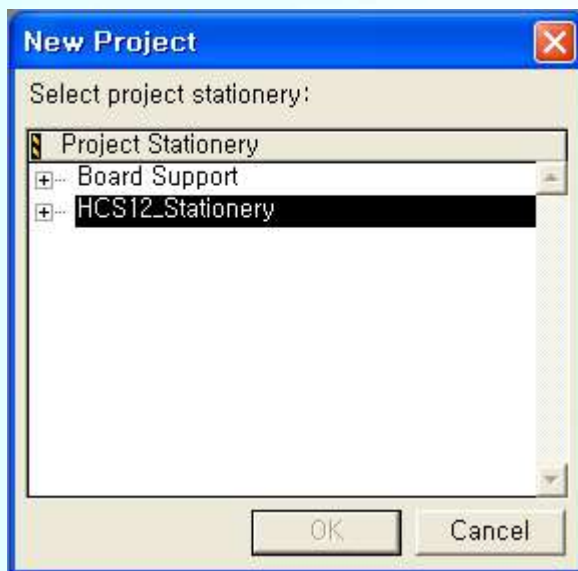
Development Tool

- Create New Project
 - File -> New..
 - 'HCS12 Stationery' option
 - Fill project name



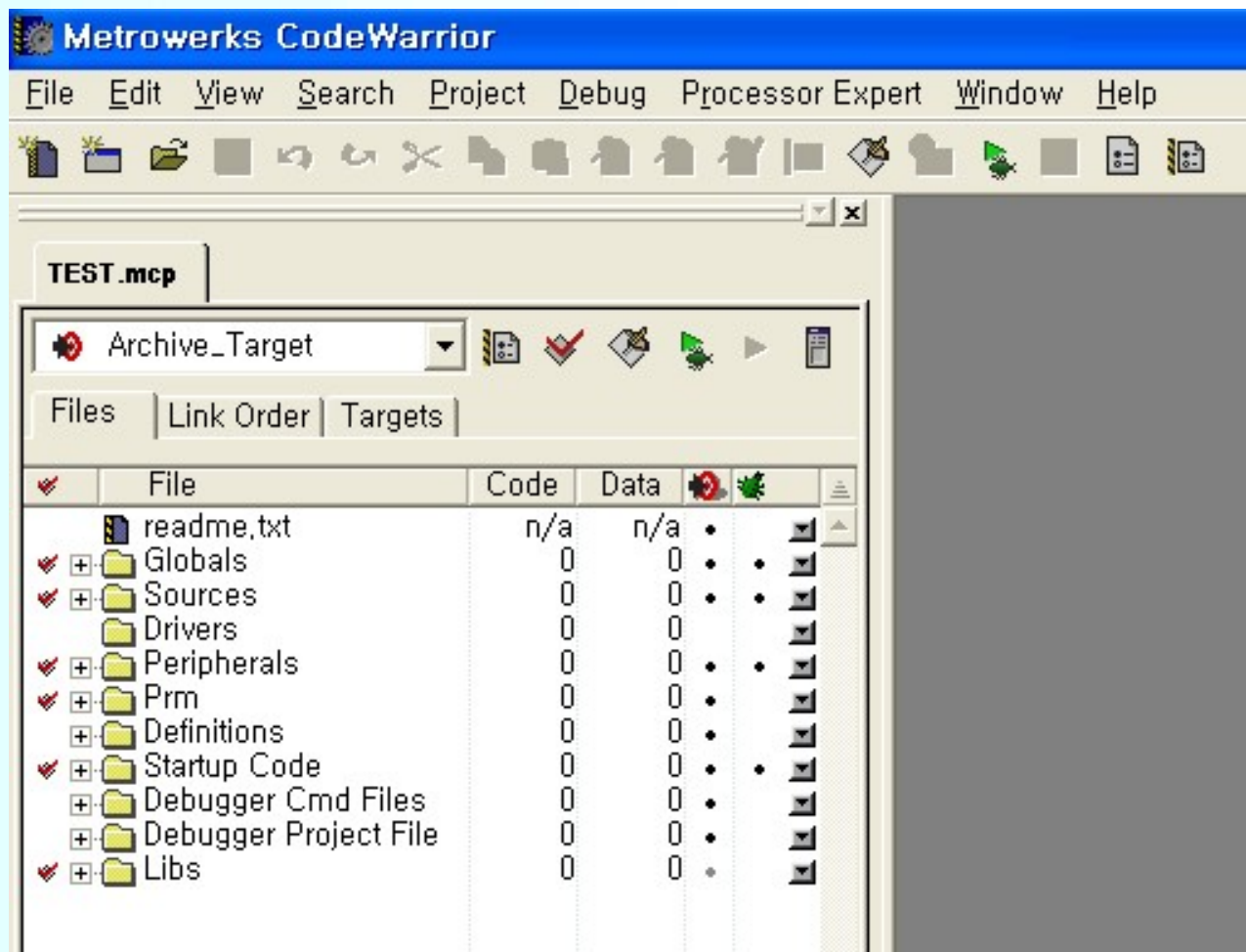
Development Tool

- Select project stationery
 - MCU : MC9S12DP256B
 - Select_DP256_K79X



Development Tool

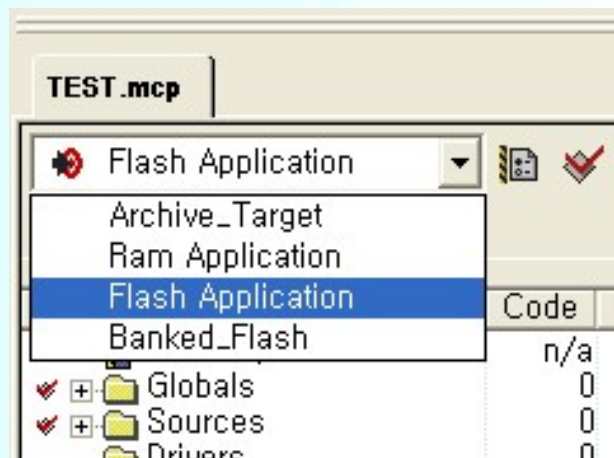
- After creating the new project



Development Tool

■ Select target

- Ram Application, Flash Application, Banked_Flash
- Banked_Flash
 - Large size program
 - Paging technique
- Flash Application : small program
- Select Flash Application



Development Tool

■ Files

■ Globals

- projectglobals.h : global variables and definitions for all files
- projectvector.c , projectvector.h : interrupt vectors

■ Sources : source files ex)main.c

■ Drivers : device driver

■ Peripherals : declaration of peripheral variables

■ Prm : linker parameter file

■ Definitions : defines each module as a data structure

- ex) S12ATD10B8CV2.h : ATD module register data structure

■ Startup Code : entry point

- START12.c

■ Debugger Cmd Files : settings for the debugger

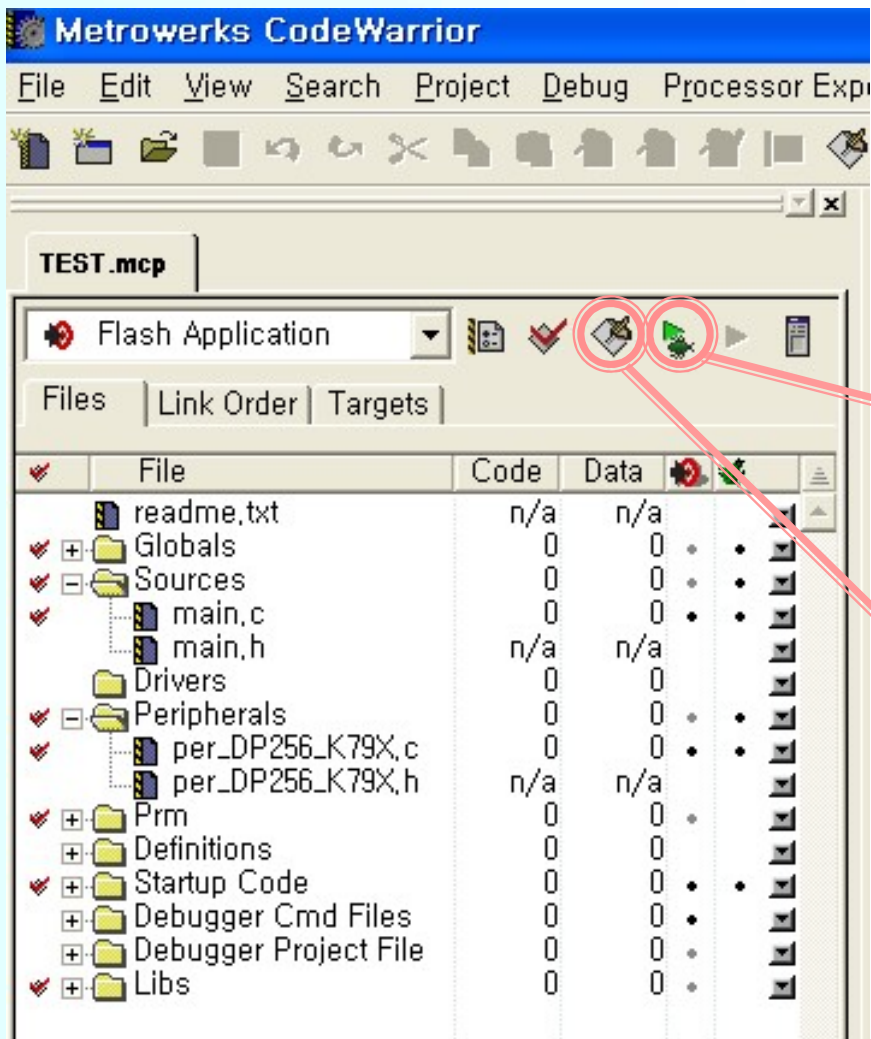
■ Debugger Project File : settings for the debugger

■ Libs : ANSI-C library

- General ANSI-C library is available

Development Tool

■ Compile and Download



Compile & Download

Compile

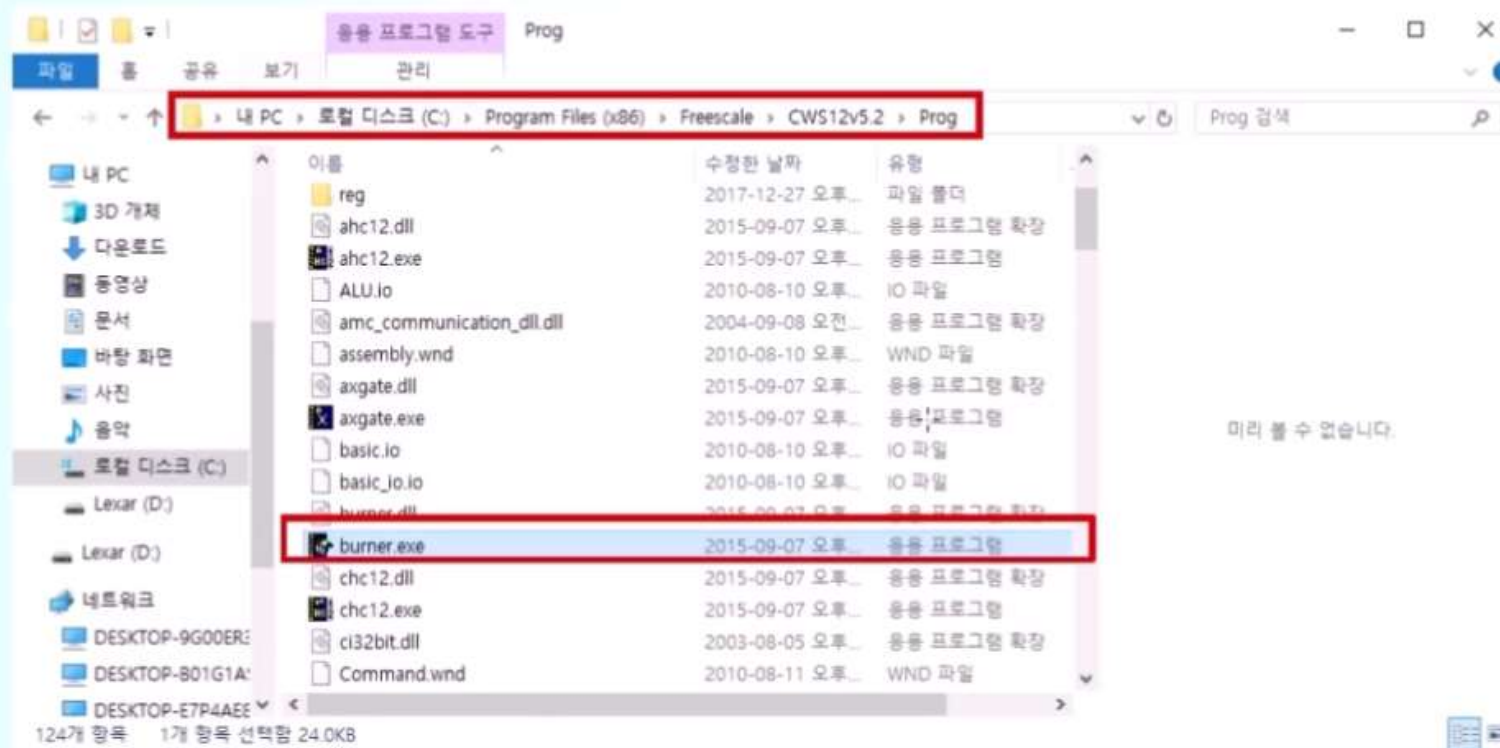
Program download

■ HW setting



Program download

■ Double click Burner.exe



Program download

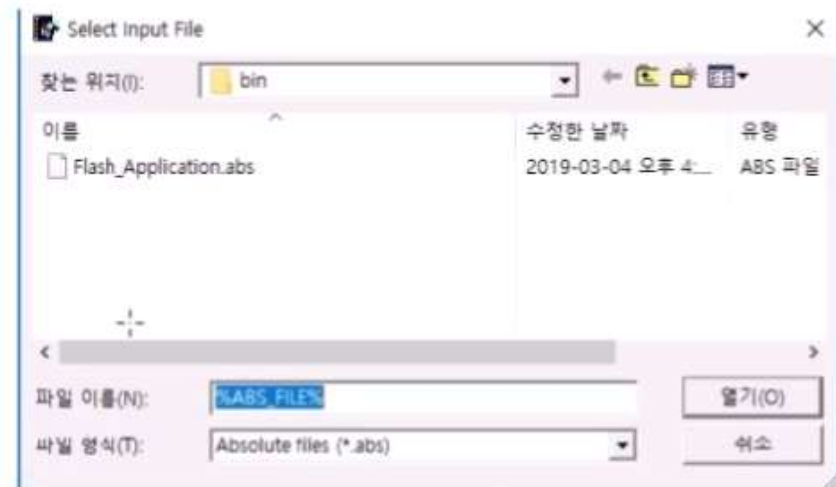
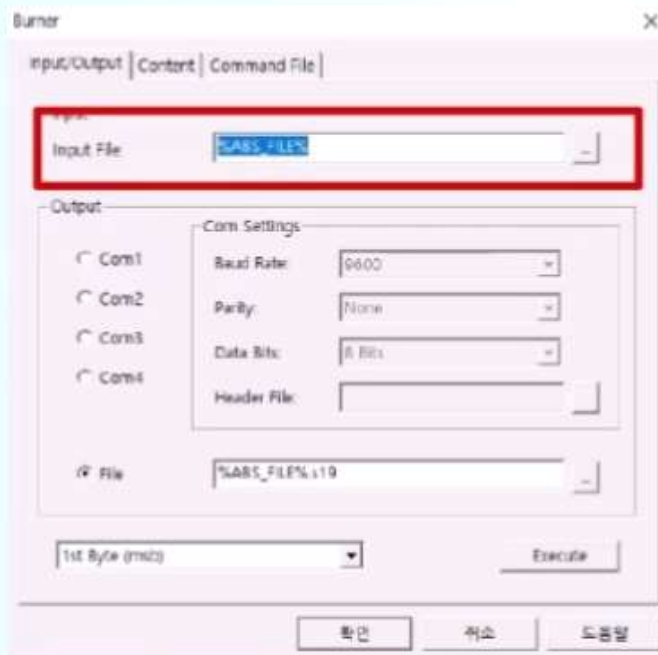
■ Click the icon



Program download

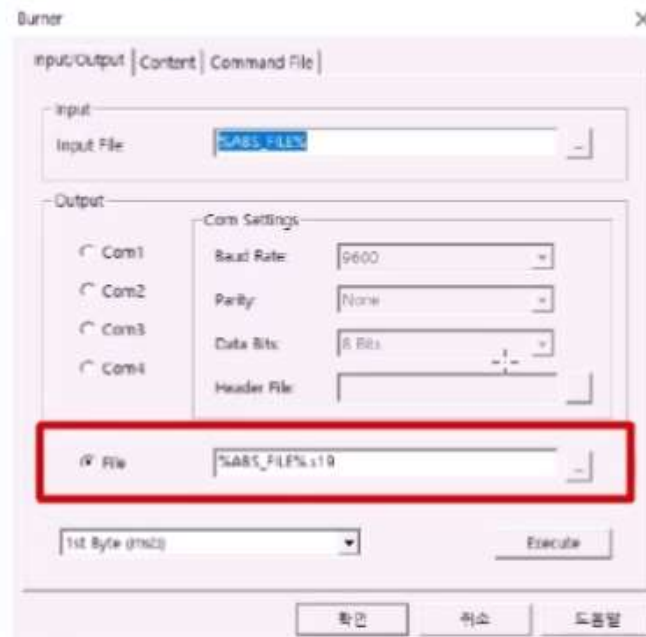
Find input file

- .abs file in the 'bin' folder



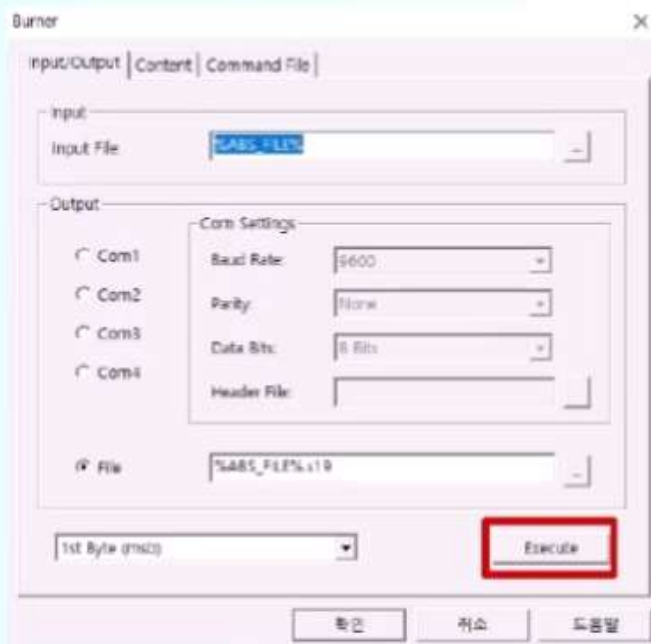
Program download

■ Set output file directory



Program download

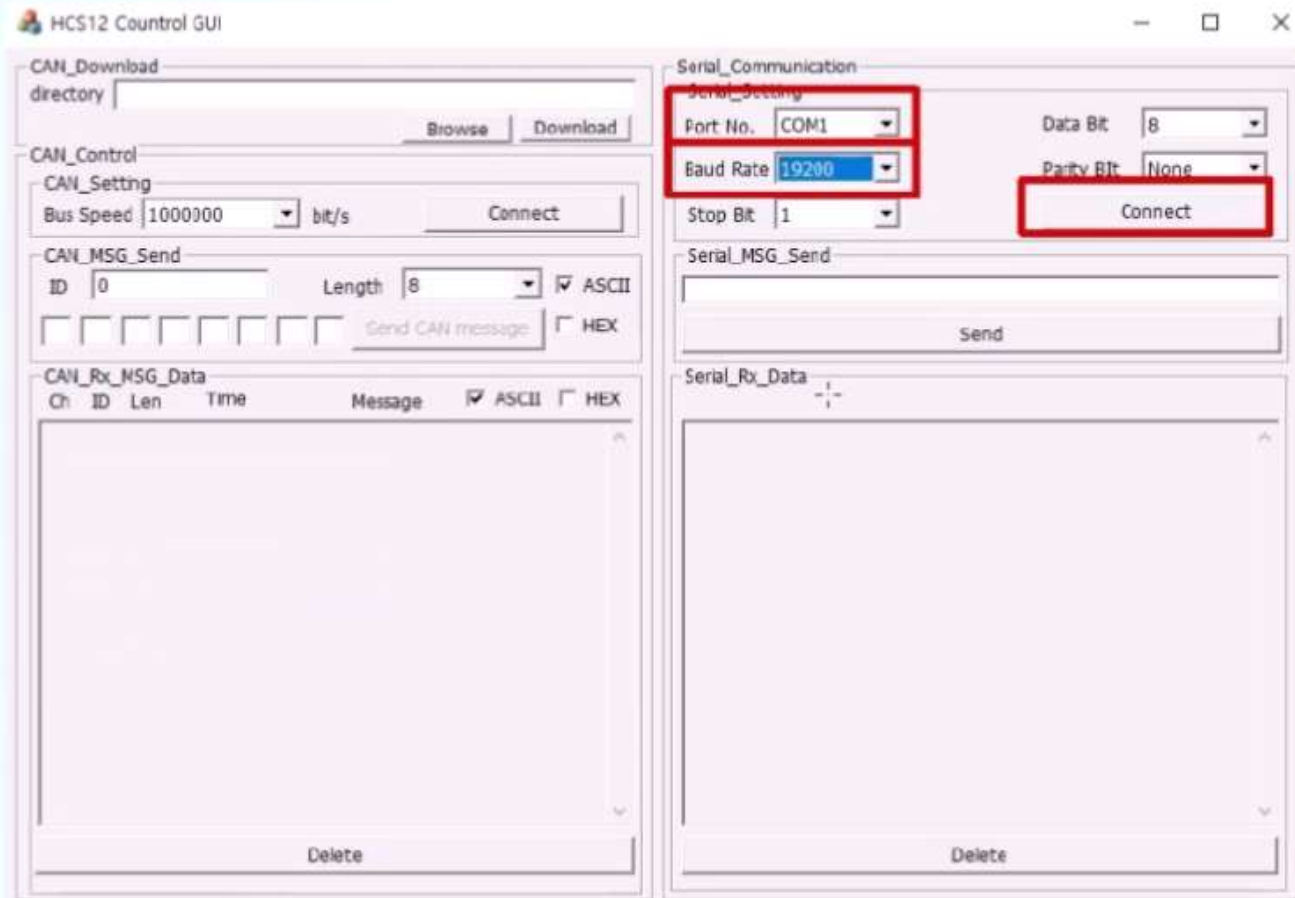
Click execute



Program download

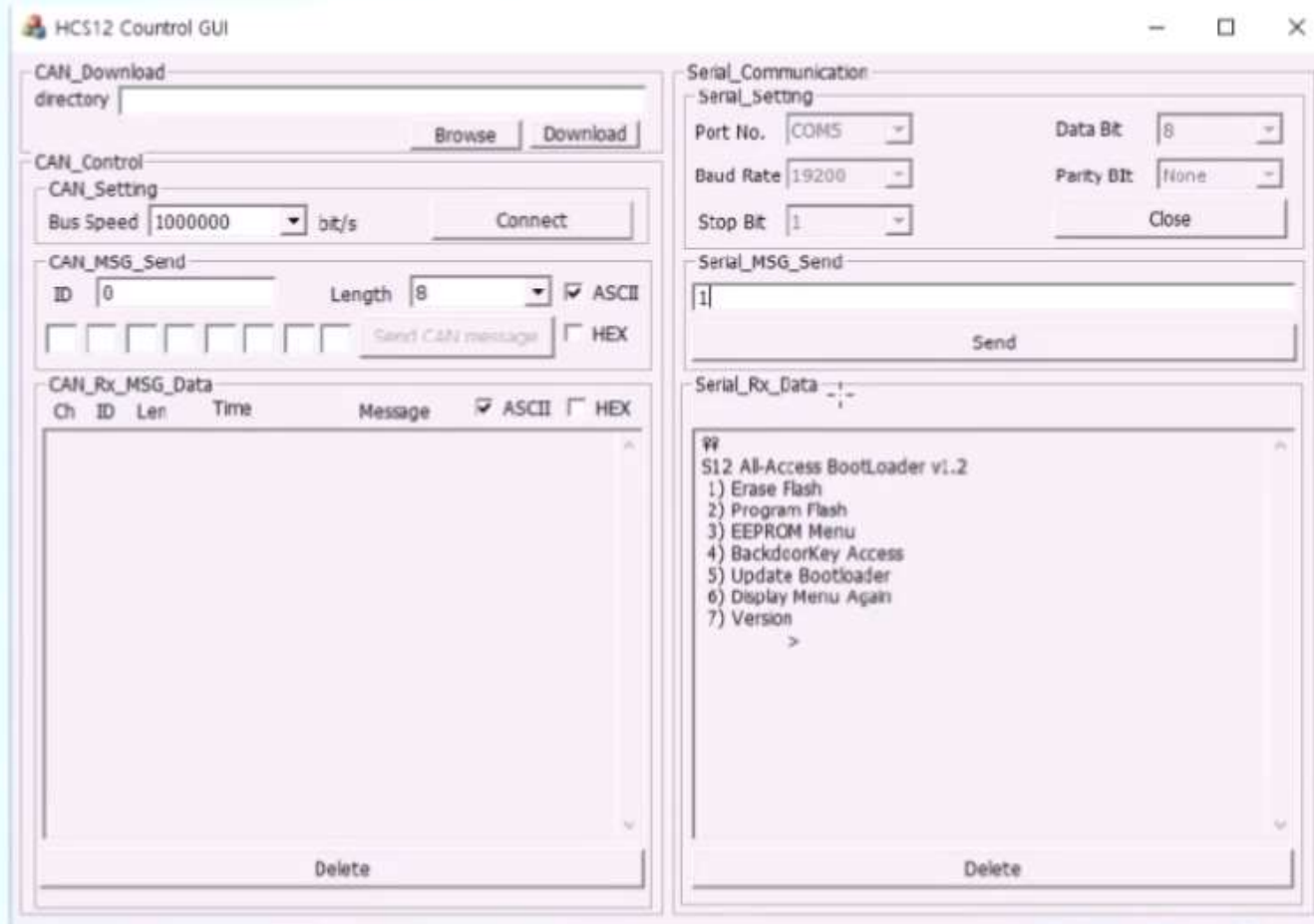
■ GUI

- Set baud rate, port No. and connect serial communication



Program download

- Erase flash
- Send '1'



Program download

■ Program flash flash

■ Send '2'

The screenshot displays the HCS12 Control GUI with two main panels: CAN communication and Serial communication.

CAN Communication Panel:

- CAN_Download:** Includes a 'directory' field, 'Browse', and 'Download' buttons.
- CAN_Control:** Includes a 'CAN_Setting' section with 'Bus Speed' set to 1000000 bit/s and a 'Connect' button.
- CAN_MSG_Send:** Includes 'ID' (0), 'Length' (8), and checkboxes for 'ASCII' (checked) and 'HEX'. A 'Send CAN message' button is present.
- CAN_Rx_MSG_Data:** A table with columns 'Ch', 'ID', 'Len', 'Time', and 'Message'. It includes checkboxes for 'ASCII' (checked) and 'HEX'. A 'Delete' button is at the bottom.

Serial Communication Panel:

- Serial_Setting:** Includes 'Port No.' (COM5), 'Data Bit' (8), 'Baud Rate' (19200), 'Parity Bit' (None), and 'Stop Bit' (1). A 'Close' button is present.
- Serial_MSG_Send:** Includes a text field with '2' and a 'Send' button.
- Serial_Rx_Data:** A text area showing the received data:

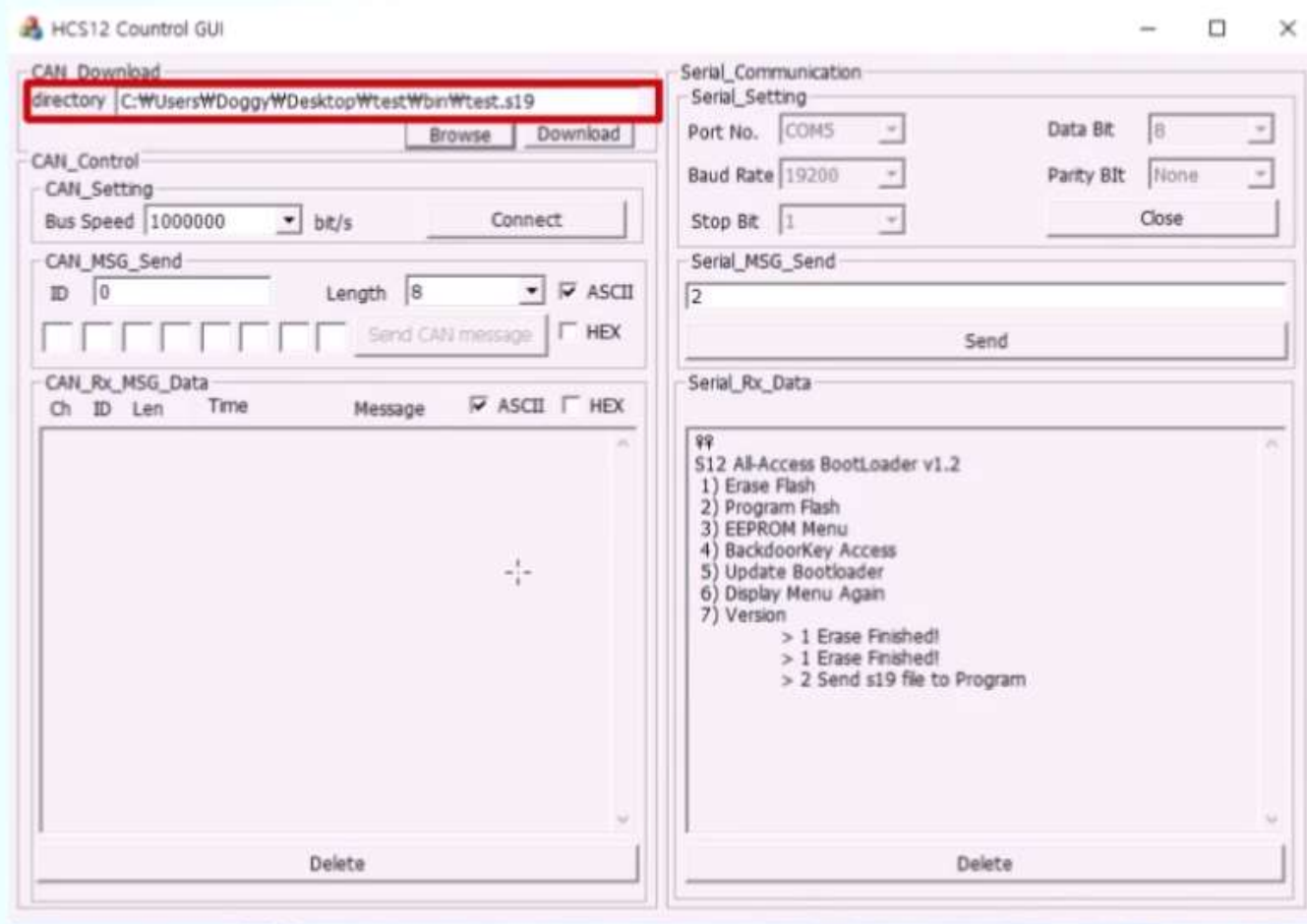

```

S12 All-Access BootLoader v1.2
1) Erase Flash
2) Program Flash
3) EEPROM Menu
4) BackdoorKey Access
5) Update Bootloader
6) Display Menu Again
7) Version
  > 1 Erase Finished!
  > 1 Erase Finished!
  > 2 Send s19 file to Program
      
```

 A 'Delete' button is at the bottom.

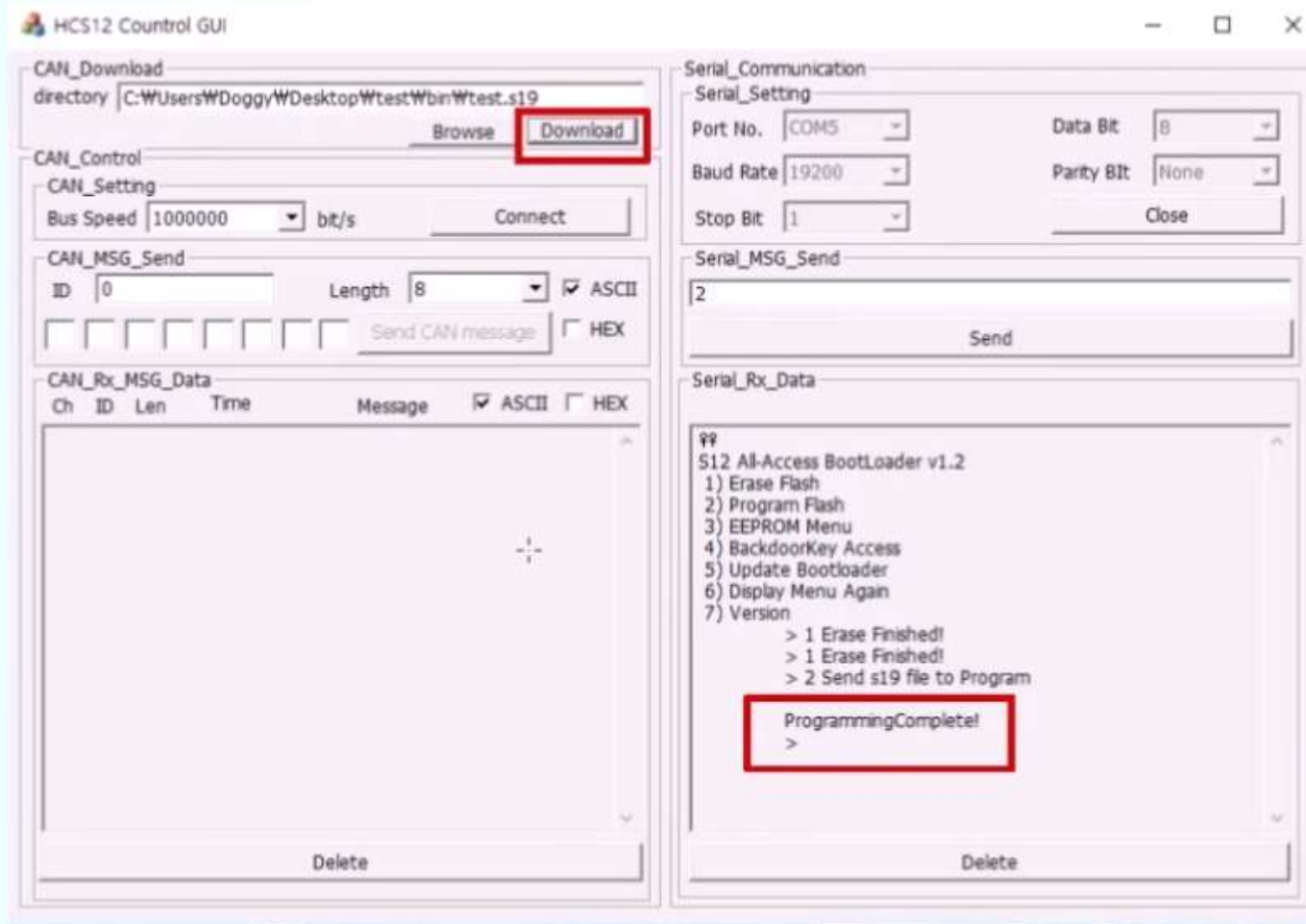
Program download

Find .s19 file directory



Program download

■ File download



Register Definition

- Declaration of peripheral variables (per_DP256_K79X.c)
 - Global variables : Regs, Atd0, Pwm, etc..

```
extern tREGISTER Regs @ (0x0000 + REG_BASE);
extern tPAGE Page @ (0x0030 + REG_BASE);
extern tCRG Crg @ (0x0034 + REG_BASE);
extern tTIMER Tim0 @ (0x0040 + REG_BASE);
extern tATD Atd0 @ (0x0080 + REG_BASE);
extern tPWM Pwm @ (0x00A0 + REG_BASE);
extern tSCI Sci0 @ (0x00C8 + REG_BASE);
extern tSCI Sci1 @ (0x00D0 + REG_BASE);
extern tSPI Spi0 @ (0x00D8 + REG_BASE);
extern tIIC Iic @ (0x00E0 + REG_BASE);
extern tBDLC Bdlc @ (0x00E8 + REG_BASE);
extern tSPI Spi1 @ (0x00F0 + REG_BASE);
extern tSPI Spi2 @ (0x00F8 + REG_BASE);
extern tFLASH Flash @ (0x0100 + REG_BASE);
extern tEEPROM Eeprom @ (0x0110 + REG_BASE);
extern tATD Atd1 @ (0x0120 + REG_BASE);
extern tMSCAN Can0 @ (0x0140 + REG_BASE);
extern tMSCAN Can1 @ (0x0180 + REG_BASE);
extern tMSCAN Can2 @ (0x01C0 + REG_BASE);
extern tMSCAN Can3 @ (0x0200 + REG_BASE);
extern tPIM Pim @ (0x0240 + REG_BASE);
extern tMSCAN Can4 @ (0x0280 + REG_BASE);
```


Register Definition

- Header file for HCS12 register block (S12CPU15V1_2.h)
 - tREGISTER : register data structure
 - porta : Port A data register

```
typedef struct /*port and internal resource control */
{
    volatile tPORTA porta; /*port A data register */
    volatile tPORTB portb; /*port B data register */
    volatile tDDRA ddra; /*port A data direction register */
    volatile tDDRB ddrb; /*port B data direction register */
    tU08 rsv1[4]; /*reserved (maintaining memory map) */
    volatile tPORTE porte; /*port E data register */
    volatile tDDRE ddre; /*port E data direction register */
    volatile tPEAR pear; /*port E assignment register */
    volatile tMODE mode; /*mode register */
    volatile tPUCR pucr; /*pull-up control register */
    volatile tRDRIV rdriv; /*reduced drive register */
    volatile tEBICTL ebictl; /*external bus control */
    tU08 rsv2; /*reserved (maintaining memory map) */
    volatile tINITRM initrm; /*initialise internal ram position */
    volatile tINITRG initrg; /*initialise internal register position */
    volatile tINITEE initee; /*initialise internal eeprom position */
    volatile tMISC misc; /*miscellaneous system control register */
    volatile tMTST mtst0; /*memory test */
    volatile tITCR itcr; /*interrupt test control */
    volatile tITEST itest; /*interrupt test status */
    volatile tMTST mtst1; /*memory test */
    tU08 rsv3[2]; /*reserved (maintaining memory map) */
    volatile tPARTID partid; /*word containing part id information */
    volatile tMEMSIZ0 memsiz0; /* */
    volatile tMEMSIZ1 memsiz1; /* */
    volatile tIRQCR irqcr; /*interrupt control register */
    volatile tHPRIO hprio; /*highest priority I interrupt */
    tU08 rsv4[8]; /*reserved (maintaining memory map) */
    volatile tBKPCT0 bkpct0; /*breakpoint control register 0 */
    volatile tBKPCT1 bkpct1; /*breakpoint control register 1 */
    volatile tBKPOX bkp0x; /* */
    volatile tBKPO bkp0; /*breakpoint address registers */
    volatile tBKPIX bkp1x; /* */
    volatile tBKPI bkp1; /*breakpoint data registers */
}tREGISTER;
```


Register Definition

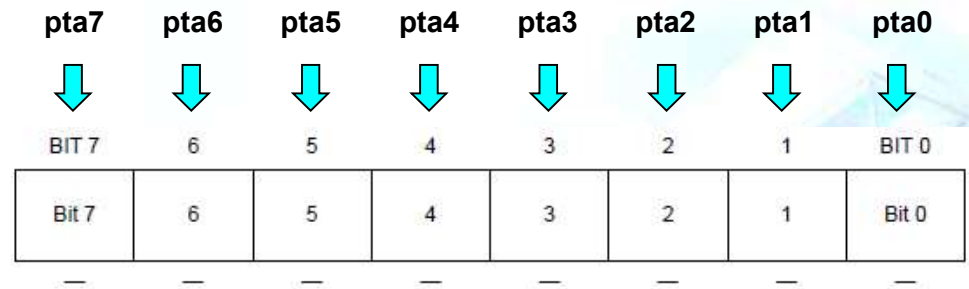
■ Port A data register (S12CPU15V1_2.h)

■ pta0 ~ pta7 : 8 pins, 1 bit

```
typedef union uPORTA
{
    tU08 byte;
    struct
    {
        tU08 pta0 :1;    /*i/o port pins */
        tU08 pta1 :1;
        tU08 pta2 :1;
        tU08 pta3 :1;
        tU08 pta4 :1;
        tU08 pta5 :1;
        tU08 pta6 :1;
        tU08 pta7 :1;
    }bit;
}tPORTA;

#define PTA0    0x01    /*bit masks */
#define PTA1    0x02
#define PTA2    0x04
#define PTA3    0x08
#define PTA4    0x10
#define PTA5    0x20
#define PTA6    0x40
#define PTA7    0x80
```

Read:
Write:
Reset:



Port Control

- Port Control
 - using memory reference instruction
 - Registers control each port
 - Input port, output port or both
 - Controlled by program code

Port Control

■ Example) Port B control

```
#include "main.h"

void main ()
{
    Regs.ddrb.byte = 0b11111111;
    Regs.portb.byte = 0b11111110;
    for (;;) {
    }
}
```

■ Port B

- 8 pins
- Input or output

■ Data representation in C language

- Decimal : 10 (no prefix)
- Hexadecimal : 0x0A (prefix 0x)
- Binary: 0b00001010 (prefix 0b)

Port Control

- DDR(Data Direction Register) : data direction setting

- Input pin : 0

- Output pin : 1

- DDRB setting

- Data Direction Register for Port B

- All pins of Port B are output pin

```
Regs.ddrb.byte = 0b11111111;  
Regs.portb.byte = 0b11111110;  
for (;;) {  
}
```

- Port B value

- Pin 1 : 0, Pin 2~8 : 1

```
Regs.ddrb.byte = 0b11111111;  
Regs.portb.byte = 0b11111110;  
for (;;) {  
}
```

