

Embedded System Software

(Programming on an embedded systems 1)

http://jcnet.co.kr/ 의 자료실의 강의노트를 참조함

Fall, 2017

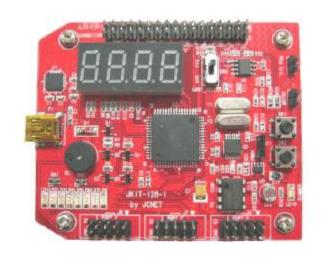


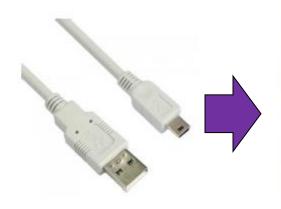
- 개관
- 실습 환경 설정
- LED
- FND



개관 (임베디드보드)

교차개발 환경





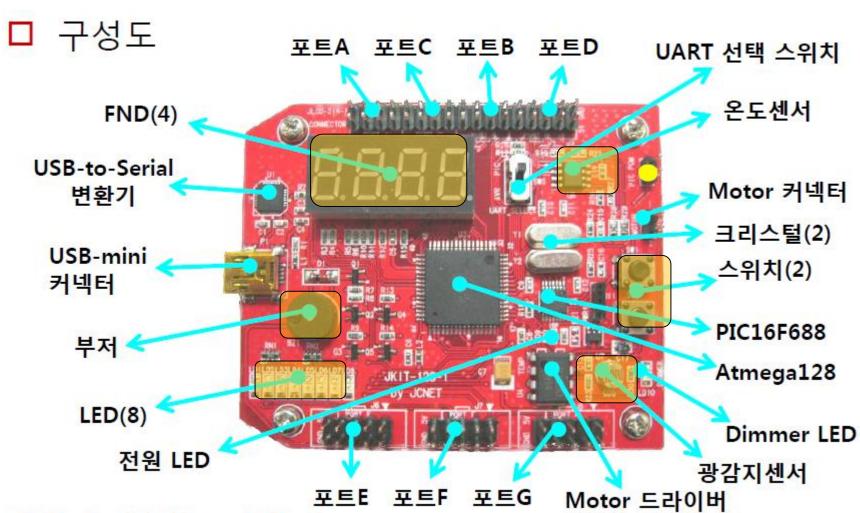


보드

케이블

연결







ATmega128 GPIO 관련 레지스터

- 1> DDRx (Data Direction Register)
 - 각 포트에 대한 데이터 입출력 방향 지정용 레지스터
 - DDRA~DDRG 레지스터의 해당 비트에 '1을 쓰면 출력 (default), '0'을 쓰면 입력으로 설정
- 2> PORTx (Port Output Register)
 - 데이터 출력 레지스터
 - 출력을 원하는 데이터값을 PORTx 레지스터에 쓰면 됨
- 3> PINx (Port Input Register)
 - 데이터 입력 레지스터
 - PINx 레지스터의 값을 읽으면 그것이 입력된 값임
- 4> SFIOR(Special Function IO Register) 레지스터
 - SFIOR의 비트2(PUD: Pull-Up Disable)를 '1'로 세트하면 풀 업 저항이 비활성화되고(기본 상태) '0'으로 하면 활성화 됨



개발 환경 설정

- 개발 환경 구축
 - 1> AVR Studio 4 설치
 - 2> WinAVR 설치
 - "WinAVR-20100110-install.exe "를 찾아 다운로드
 - "WinAVR-20100110-install.exe "를 실행하여 WinAVR-20100110 설치
 - 3> CP2102 드라이버 준비
 - SILAB사의 홈페이지 (http://www.silabs.com)에 접속
 - <CP210x_VCP_Win_XP_S2K3_Vista_7.exe > 프로그램
 을 찾아 다운로드 후 실행하여 드라이버 설치



개발 환경 설정

- 윈도우 10 컴파일 에러시,
- If there's an compile error like below,

0 [main] sh 7096 sync_with_child: child 6880(0x1A8) died before initialization with status code 0xC0000142
60240 [main] sh 7096 sync_with_child: *** child state waiting for longjmp
/usr/bin/sh: fork: Resource temporarily unavailable

avr-gcc -mmcu=atmegal28 -Wall -gdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -MD -MP -MT test.o -MF dep/test.o.d -c ../test.c

../test.c:17: fatal error: opening dependency file dep/test.o.d: No such file or directory

compilation terminated.
make: *** [test.o] Error 1

Build failed with 1 errors and 0 warnings...



- msys-1.0.dll 을 WinAVR설치경로\utils\bin 이하에 복사
- Copy msys-1.0.dll into <Path_WinAVR>\utils\bin
- (C:\WinAVR-20100110\utils\bin)



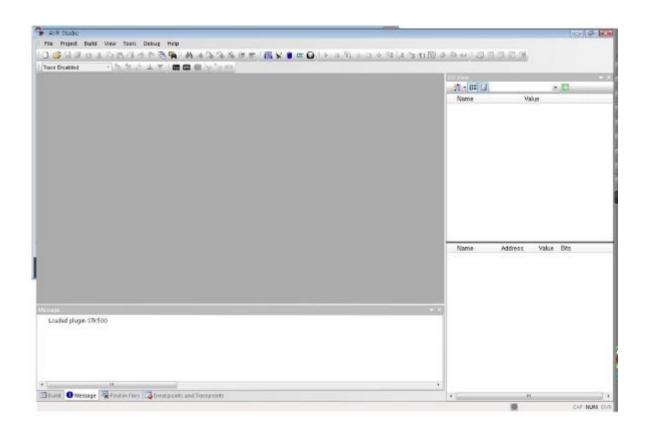
How to run a program on the board

- 1> Wring a program
- 2> Compile
 - [Build]
- 3> Download
 - [Tools] -> [Program AVR] -> [Connect]



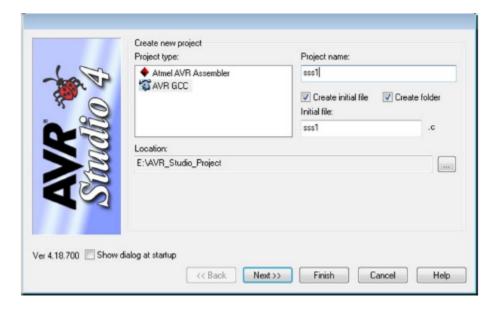
Make your first project

■ 처음 실행 화면 : [Project] [New]

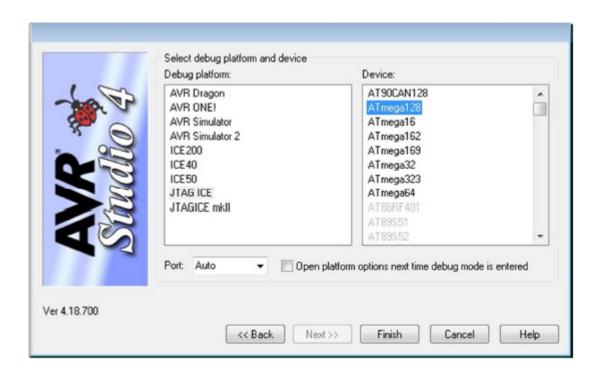




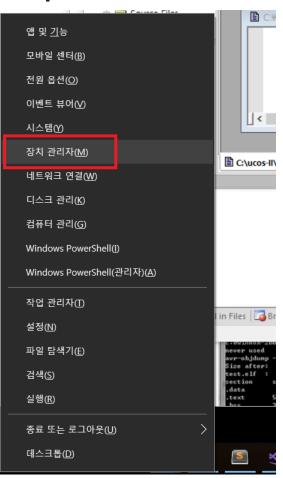
- Project Type : AVR GCC
 - Project Name :
 - Initial File :
 - Location : 기본값 (필요시 조정)
 - [Next>>]

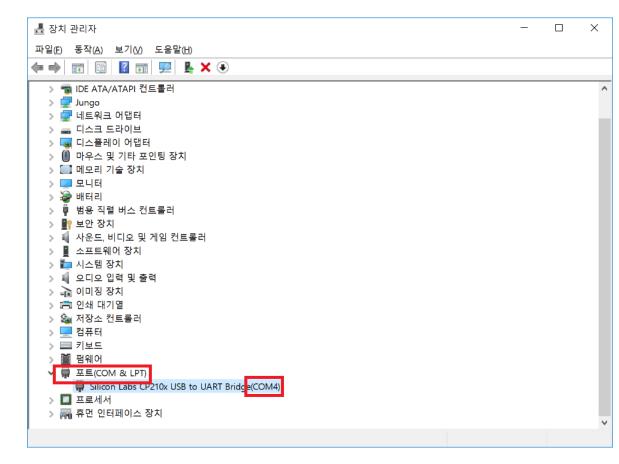


Debug platform & device

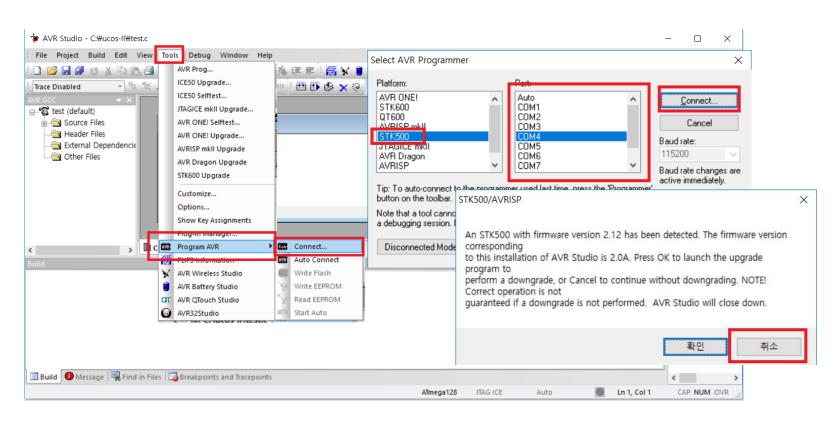






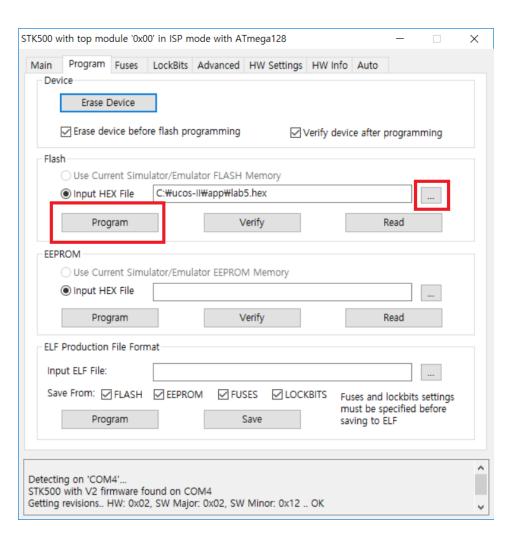


window + x 를 눌러 장치 관리자를 열고 포트 번호를 확인한다.



Program AVR -> Connect를 실행하고 STK500. 확인한 포트 넘버를 선택하고 Connect한다. 그리고 팝업되는 화면에서 취소를 누른다.

■ [Flash]의 [Input HEX File]을 선택하고 [Program] 버튼 선택





- LED 작동 원리
 - LED를 켜기 위해서는 LED 신호에 '1'을 인가하여 야 함 (끄기 위해서는 '0'을 인가하여야 함)
 - 즉, ATmega128 GPIO PA포트의 해당 비트에 '1'을 출력 하여야 함
 - ATmega128 GPIO PA 포트의 특정 비트에 '1'을 출력하려면
 - 1. DDRA 레지스터의 해당 비트에 '1'을 write하여 방향을 ' 출력' 상태로 만들고,
 - 2. PORTA 레지스터의 해당 비트에 '1'을 write하면 됨



□ 구동 프로그램 코딩 : LED ON (led_1_1.c)

```
/* GPIO로 LED 켜고 끄기 1번 예
 LED 8개가 연결되어 있는 포트: Port A(PA)
      - 비트7 : LED7(LD1), 비트6 : LED6(LD2)
      - 비트1: LED1(LD7), 비트0: LED0(LD8) */
                            // ATmega128 register 정의
#include <avr/io.h>
int main()
                            // 포트 A를 출령 포트로 사용
       DDRA = 0xff;
                            ৴/_0xff = 0b11111111, LED 모두 ON
       PORTA = 0xff;
```

4

_delay_ms(500);

```
/* GPIO로 LED 움직임 표현하기 : 링 카운터, 오른쪽 이동 */
#include <avr/io.h> // ATmega128 register 정의
void main()
     unsigned char value = 128; // 왼쪽 LED 1개만 켜기
     DDRA = 0xff; // 포트 A를 출력 포트로 사용
     for (;;)
           PORTA = value; // LED에 원하는 값 출력
            my_delay(2); // 0.2초 지연
           value >≯= 1; // 값을 오른쪽을 1칸 shift
            if (value == 0) // 오른쪽 끝인지 검사
                  value = 128; // 끝이면 초기화
```

my_delay 함수 작성이 필요함



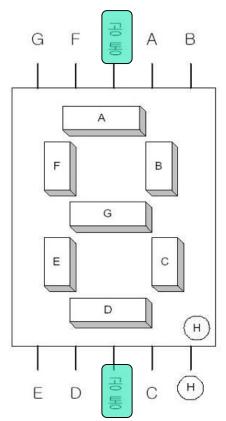
FND

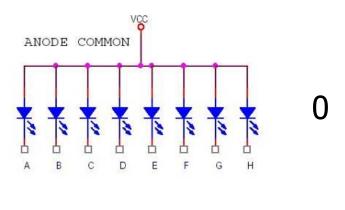
- Flexible Numeric Device의 약자
- 보통 7-Segment LED라고 칭함
- LED 7개(점 포함 8개)를 숫자를 표시하기 쉽도록 배열한 제품

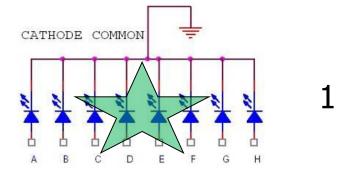


How to control FND ?

■ 7-세그먼트 '1'을 표시하기 위해서는 B, C 의 LED를 On 시키면 됨

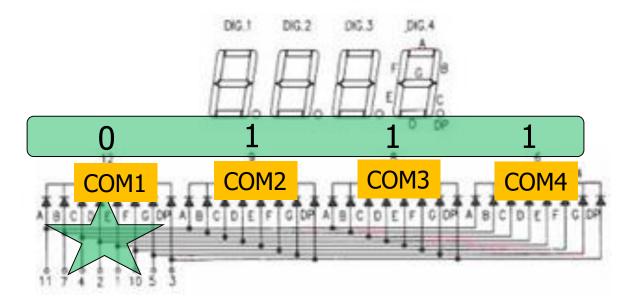








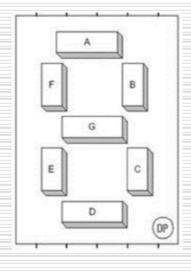
- FND (WCN4-0036SR4-C11)
 - 4개의 7-Segment를 내장한 CC 타입
 - 4개의 7-Segment의 숫자를 나타내는 부분의 신호를 모두 공유하고, COM 신호는 따로 나와 있는데, 이를 돌아가면서 오직 1개만 enable(LOW) 상태로 하면 4개의 숫자를 나타낼 수 있음



4

□ FND 숫자 표현 (Common-Cathode 경우)

7-Segment 에서 16진수 표시방법



16 진수	7-세그먼트의 비트값								데이터 값
10 位十	DP	G	F	Е	D	С	В	Α	(HEX)
0	0	0	1	1	1	1	1	1	0X3F
1	0	0	0	0	0	1	1	0	0X06
2	0	1	0	1	1	0	1	1	0X5B
3	0	1	0	0	1	1	1	1	0X4F
4	0	1	1	0	0	1	1	0	0X66
5	0	1	1	0	1	1	0	1	0X6D
6	0	1	1	1	1	1	0	1	0X7D
7	0	0	1	0	0	1	1	1	0X27
8	0	1	1	1	1	1	1	1	0X7F
9	0	1	1	0	1	1	1	1	0X6F
Α	0	1	1	1	0	1	1	1	0X77
В	0	1	1	1	1	1	0	0	0X7C
С	0	0	1	1	1	0	0	1	0X39
D	0	1	0	1	1	1	1	0	0X5E
E	0	1	1	1	1	0	0	1	0X79
F	0	1	1	1	0	0	0	1	0X71



How ?

- Jkit-128-1 키트의 C 포트를 FND의 불을 켜기 위한 출력 포트로 설정하고, G 포트는 어떤 FND를 선택 할 것인지를 결정하는 선택신호 출력 포트로 설정
 - [FND 숫자 표현] 표를 참조하여 PORTC, PORTG 레지스 터에 알맞은 값을 출력
 - 4개의 숫자를 동시에 선택하여 표현할 수 없으므로 시간을 1/4로 쪼개서 지속적으로 1개의 FND 디스플레이 시간은 전체시간의 1/4만을 사용함



예제 1

```
#include <avr/io.h>
                                    // ATmega128 register 정의
void main()
      unsigned char FND_DATA[19]=\{0x3f, 0x06, 0x5b, 0x4f,
      0x66, 0x6d, 0x7d, 0x27, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e,
      0x79, 0x71, 0x80, 0x40, 0x08);
      DDRC = 0xff;
                                    // C 포트는 모두 출력
      DDRG = 0x0f;
                                // G 포트도 4개는 출력
      PORTC = FND_DATA[7]; // C 포트는 FND 데이터 신호
      PORTG = 0x01;
                                    // G 포트는 FND 선택 신호
```



예제 2

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>
unsigned char digit[10]=\{0x3f, 0x06,
0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27,
0x7f, 0x6f};
unsigned char fnd_sel[4] = \{0x01,
0x02, 0x04, 0x08;
int main()
   int i=0;
   DDRC = 0xff;
   DDRG = 0x0f;
```

```
while (1)
     for (i=0; i<4; i++)
        PORTC = digit[4-i];
        // 숫자 '1'-'4'를 출력
        PORTG = fnd_sel[i];
        _delay_ms(2);
```



Assignments

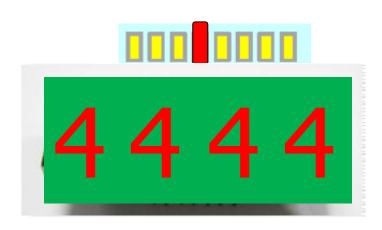
- I-Class에 다음 슬라이드의 과제 제출
- 제출 파일
 - 3 * 소스 코드(2 * mandatory + 1 optional)
 - 보고서(.pdf)
 - 실행 결과 사진 첨부
 - You have to take a picture of device and paste the picture on your report
- 파일명
 - 10주차_학번_이름(압축하여 하나의 파일로 제출)
- 제출 기한
 - 다음주 화요일 자정까지



- LED 실습 과제 1
 - 좌우로 LED 를 번갈아 켜는 프로그램을 작성하시 오.

FND (실습)

- <FND 실습 과제 2>
 - 1/100 초 단위 초시계 만들기
- <LED + FND 실습 과제 3> (optional)
 - 0~7의 랜덤 값을 생성하여 해당 LED를 킨다.
 - 켜진 LED 위치에 매핑되는 숫자를 출력하기





- 실습 후 Erase device 해주세요.
- Before you go out, please click "Erase device" button.

