



Arduino

조진성

경희대학교 컴퓨터공학과

Mobile & Embedded System Lab.

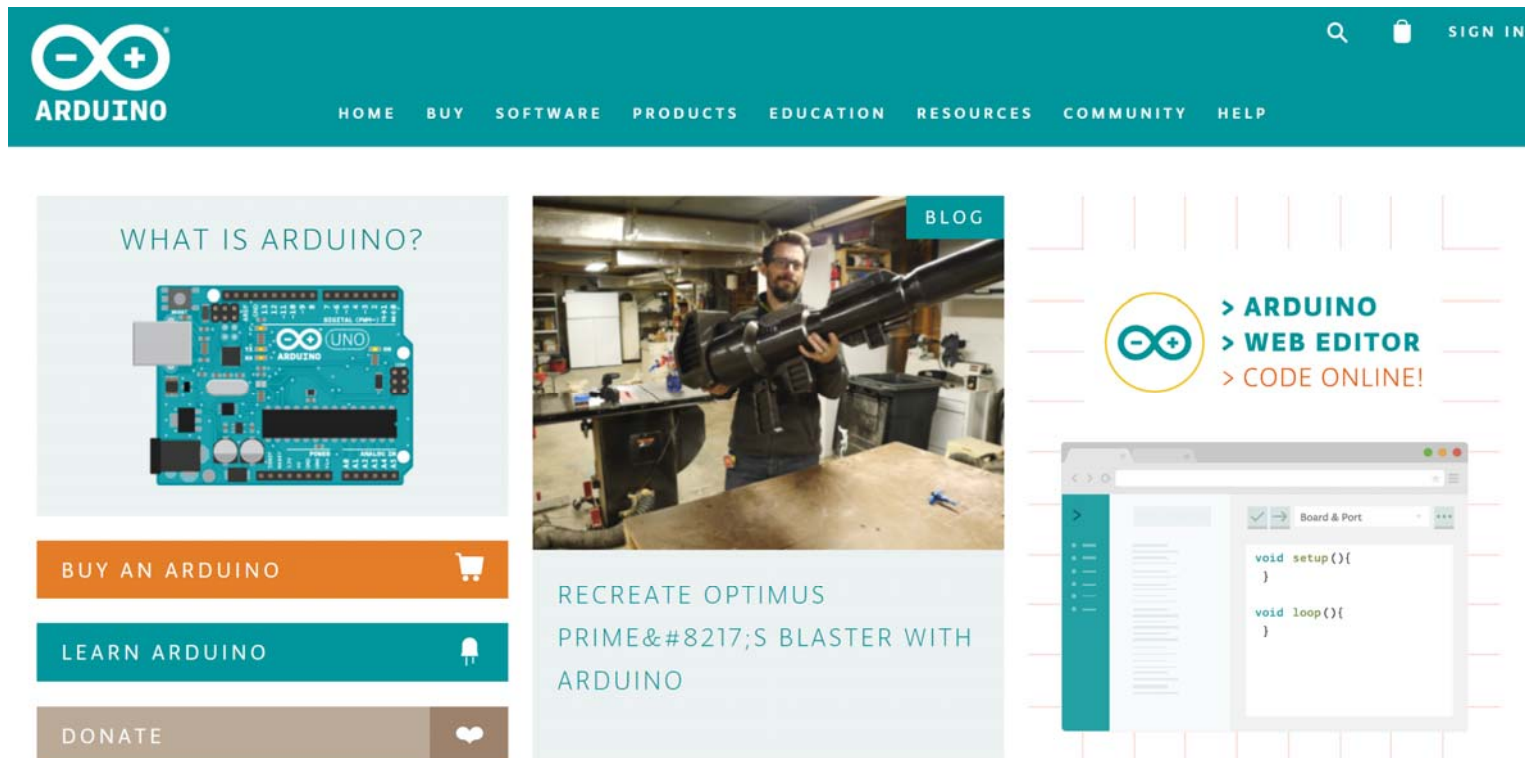


Computer Engineering in KyungHee University

Mobile & Embedded System Lab.

Arduino

- Physical Computing을 위한 오픈소스 HW & SW
 - 단일 보드 마이크로컨트롤러
 - 통합 개발환경을 제공
 - 2005년 이탈리아에서 시작
 - 방대한 참고자료, 라이브러리 (<http://www.arduino.cc>)



Arduino Open Source Hardware

■ Arduino Boards



Arduino Uno



Arduino Leonardo



Arduino Robot



Arduino Esplora



Arduino Due



Arduino Yún



Arduino Mega ADK



Arduino Ethernet



Arduino Tre



Arduino Micro



Arduino Mega 2560



Arduino Mini



LilyPad Arduino USB



LilyPad Arduino Simple



LilyPad Arduino SimpleSnap



LilyPad Arduino Simple



Arduino Nano



Arduino Pro Mini



Arduino Pro



Arduino Fio

Arduino Open Source Hardware

■ Arduino Shields



Arduino GSM Shield



Arduino Ethernet
Shield



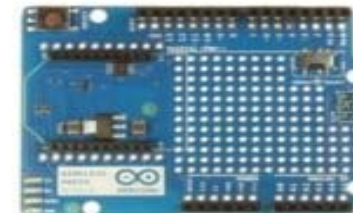
Arduino WiFi Shield



Arduino Wireless SD
Shield



Arduino Motor Shield



Arduino Wireless
Proto Shield

Arduino Open Source Hardware

■ Arduino Starter Kit



The Arduino Starter
Kit

■ Arduino Accessories



TFT LCD screen



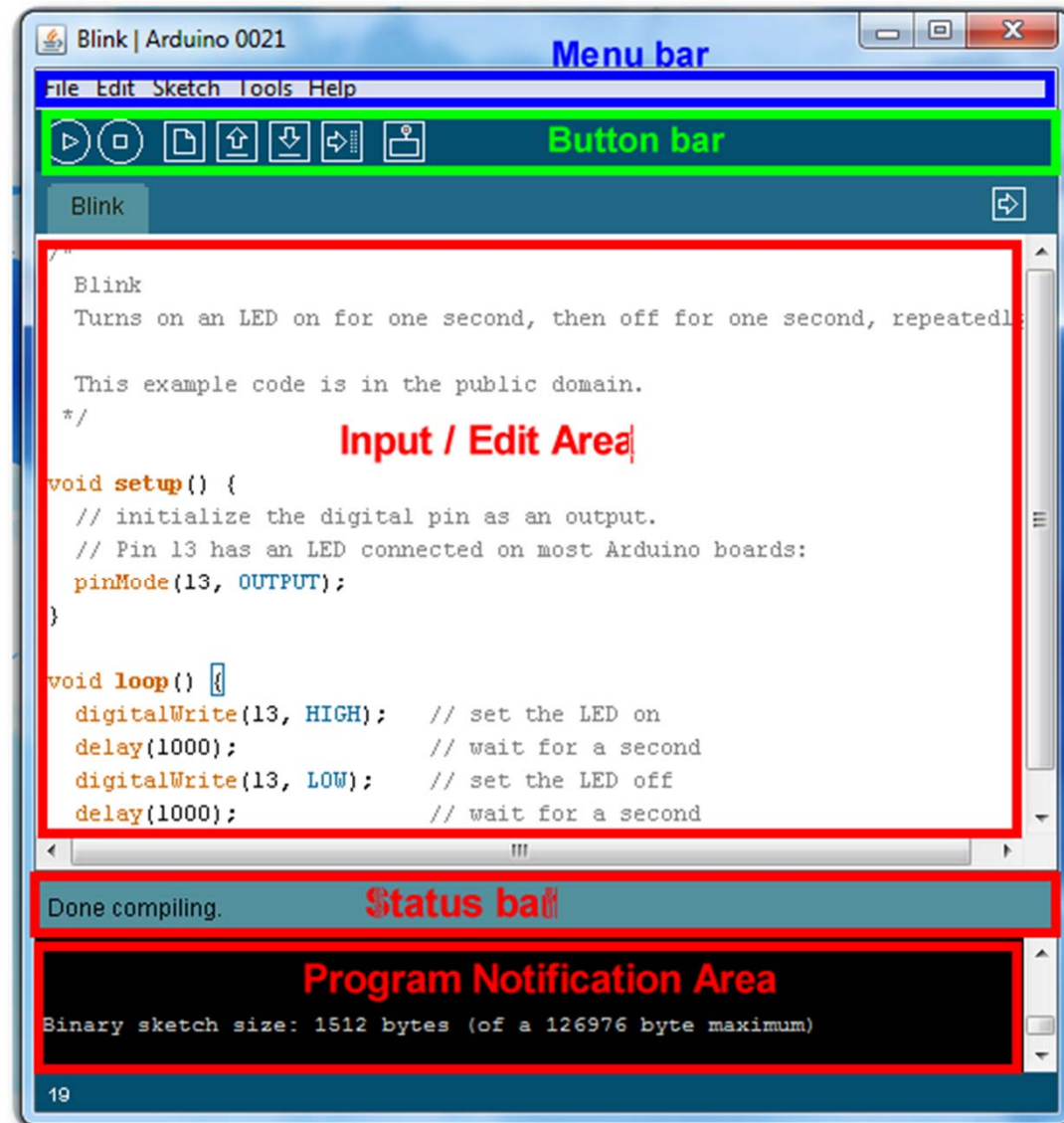
USB/Serial Light
Adapter



Mini USB/Serial
Adapter

Arduino Open Source Software

■ Arduino IDE



Arduino Open Source Software

■ Arduino Library

- 다양한 센서, 모터, 모듈을 쉽게 사용할 수 있도록 지원
- 일반적으로 ZIP 압축 파일로 배포
- 헤더 파일 include 한 뒤, 초기화 해서 사용

Ex)

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(2, 3) // Rx, Tx pin
```

- <http://www.arduino.cc/reference/en>

■ 라이브러리 경로

- 외부 라이브러리

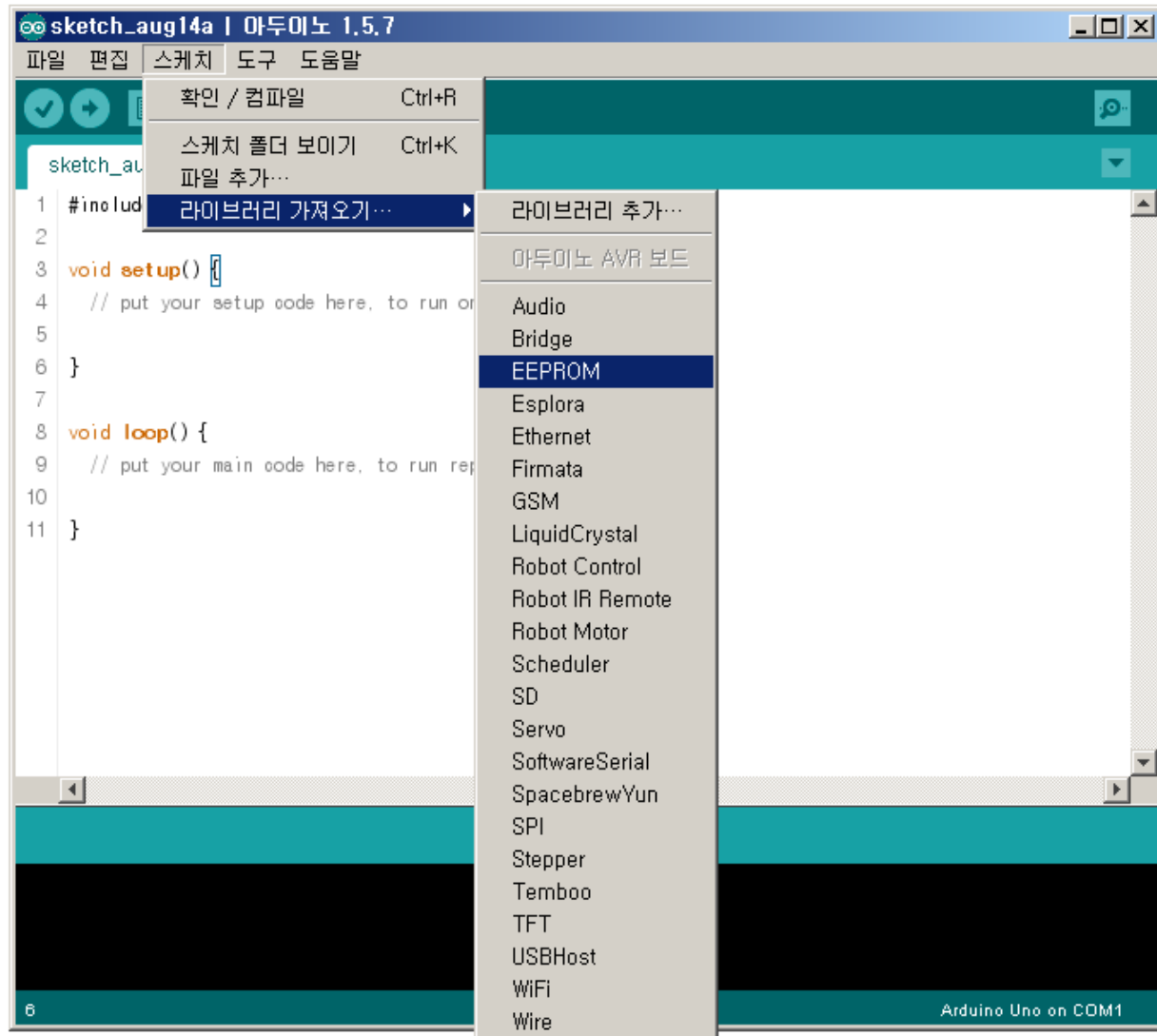
[C:\Program Files\Arduino\libraries]

- 내부 라이브러리 (기본 라이브러리)

[C:\Program Files\Arduino\hardware\arduino\avr\libraries]

Arduino Open Source Software

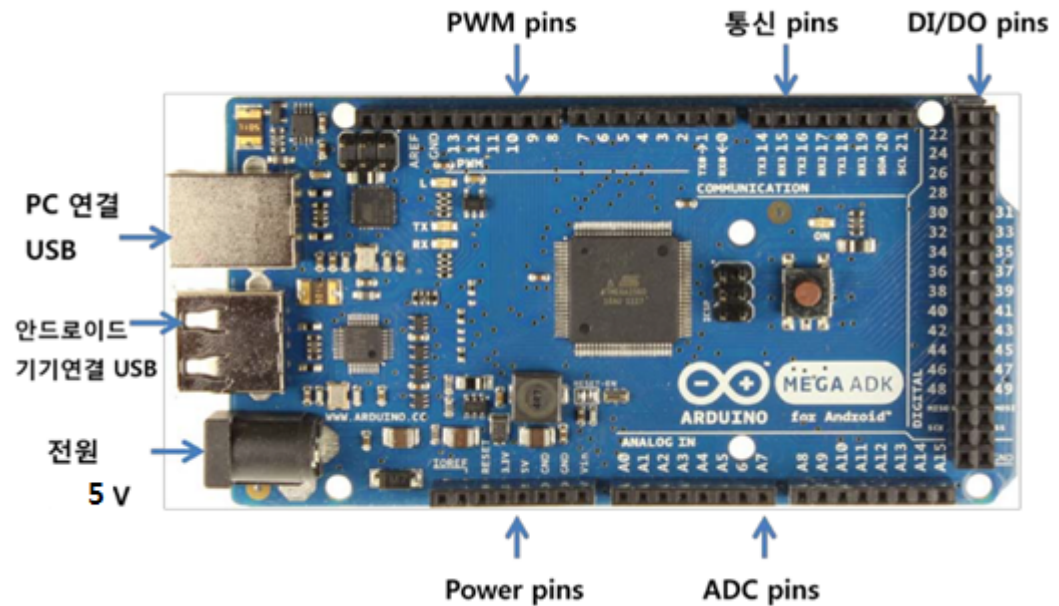
■ Arduino Library



Arduino Mega ADK

■ Arduino Mega ADK 개요

- Atmega 2560 MCU를 기반으로 한 마이크로 컨트롤러 보드
- MAX2421E 라는 IC를 이용하여 휴대폰과 연결 가능
- 54개의 디지털 입/출력핀
- 16개의 아날로그 입력핀
- 4개의 UART
- 16MHz 크리스탈 발진기
- USB 호스트
- 5V 전원 잭
- ISP 헤더 및 리셋 버튼



Arduino Mega ADK



■ Arduino Mega ADK Spec.

마이크로 컨트롤러	ATMEGA2560
작동 전압	5V
입력 전압(권장)	7-12V
입력 전압(한계)	6-20V
디지털 I/O PIN	54개(15 PWM출력 제공)
아날로그 입력 PIN	16개
I/O핀 당 DC 전류	40mA
3.3V핀의 DC 전류	50mA
플래시 메모리	256KB (부트로더에서 8KB사용)
SRAM	8KB
EEPROM	4KB
클럭 속도	16MHz

Arduino Mega ADK



■ Arduino Mega ADK 전원

- Mega ADK는 USB 및 외부 전원으로 구동
- 외부 전원에는 AC-DC 어댑터 또는 배터리 중 하나를 선택
- 어댑터를 보드의 전원 잭에 2.1MM 전원 플러그를 연결하여 사용
- 권장 전압은 5V

■ Arduino Mega ADK 통신

- ATmega2560은 4개의 UART TTL(5V) 시리얼 통신
- ATmega8U2 보드 채널은 USB를 통해 컴퓨터에 가상의 COM포트를 제공
- ATmega2560은 TWI와 SPI통신을 지원

Arduino Mega ADK GPIO



■ GPIO (General Purpose Input Output)

- GPIO is a generic pin on a chip whose behavior can be controlled by the user at run time
- The GPIO connector has a number of different types of connection:
 - True GPIO (General Purpose Input Output) pins that you can use to turn LEDs on and off etc.
 - I2C interface pins that allow you to connect hardware modules with just two control pins
 - SPI interface with SPI devices, a similar concept to I2C but uses a different standard
 - Serial Rx and Tx pins for communication with serial peripherals
- GPIO pins can be used as both digital outputs and digital inputs
 - Output: turn a particular pin HIGH or LOW
 - Input: detect the pin being at HIGH or LOW

Arduino Mega ADK GPIO

ATmega2560 GPIO

- 86개의 GPIO를 지원
- Port A/B/C/D/E/F/G/H/J/K/L (각 포트는 8bit, G 포트만 6bit)
- Atmel AVR MCU는 다음과 같은 register(8bits)로 GPIO 제어
 - DDRx – 입/출력 설정 Register (Data Direction Register)
 - PORTx – DDRx가 출력일 경우 high 또는 low 출력 (Port Output Register)
 - PINx - DDRx가 입력일 경우 현재 핀의 상태를 읽음 (Port Input Register)

13.4.2 PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.3 DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.4 PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Arduino Mega ADK GPIO

ATmega2560 GPIO ↔ Arduino GPIO

Pin Map

Port Pin Name (ATmega2560) ↔ Mapped Pin Name (Arduino)

Pin Map Table(일부)

Pin Name의 P'X'가 Port'X'를 나타냄

예) PB = Port B

전체 Table: <https://www.arduino.cc/en/Hacking/PinMapping2560>

Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)

- Pin Map



Arduino Mega ADK GPIO

GPIO Register

사용 예 – Blink.ino

■ Arduino에서 제공하는 기본 예제

- Arduino에 탑재된 LED ON/OFF

■ 왼쪽: Arduino 제공 예제

■ 오른쪽: Register를 사용하여 동일하게 동작하도록 작성



```
void setup() {  
  // LED_BUILTIN: Arduino에 탑재된 LED 제어 핀(13번 핀)  
  // LED_BUILTIN을 출력으로 설정  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  // LED_BUILTIN에 5V인가 - LED ON  
  digitalWrite(LED_BUILTIN, HIGH);  
  // 1초 지연  
  delay(1000);  
  // LED_BUILTIN에 0V인가 - LED OFF  
  digitalWrite(LED_BUILTIN, LOW);  
  // 1초 지연  
  delay(1000);  
}
```

```
void setup() {  
  // Digital 13번 핀을 출력으로 설정  
  // 16진수 또는 2진수를 이용하여 표현 가능  
  DDRB = 0x80; // DDRB = 0b10000000;  
}  
  
void loop() {  
  // Digital 13번 핀에 5V인가 - LED ON  
  PORTB = 0x80; // PORTB = 0b10000000;  
  // 1초 지연  
  delay(1000);  
  
  // Digital 13번 핀에 0V인가 - LED OFF  
  PORTB = 0x00; // PORTB = 0b00000000;  
  // 1초 지연  
  delay(1000);  
}
```

Arduino Mega ADK GPIO

GPIO Register

Arduino에서 제공하는 GPIO 제어 함수

pinMode(uint8_t pin, uint8_t mode)

- Arduino Pin을 입력 또는 출력 모드로 설정

```
void pinMode(uint8_t pin, uint8_t mode)
{
    uint8_t bit = digitalPinToBitMask(pin); // 전달된 pin에 대응하는 레지스터를 표시하는 bit를 찾음
    uint8_t port = digitalPinToPort(pin); // 전달된 pin에 해당하는 PORT를 찾음
    volatile uint8_t *reg;

    if (port == NOT_A_PIN) return; // 전달된 pin에 해당하는 port가 존재하지 않으면 함수 종료
    // 사용자가 pin 번호를 잘못 입력했을 경우를 대비

    reg = portModeRegister(port); // PORT에 해당하는 DDR을 찾음

    if (mode == INPUT) {
        ...
        *reg &= ~bit; // INPUT 모드 설정
        ...
    }
    else {
        ...
        *reg |= bit; // OUTPUT 모드 설정
        ...
    }
}
```

Arduino Mega ADK GPIO

GPIO Register

Arduino에서 제공하는 GPIO 제어 함수

digitalWrite(uint8_t pin, uint8_t val)

- Arduino Pin을 통해 값 출력

```
void digitalWrite(uint8_t pin, uint8_t val)
{
    uint8_t bit = digitalPinToBitMask(pin); // 전달된 pin에 대응하는 레지스터를 표시하는 bit를 찾음
    uint8_t port = digitalPinToPort(pin); // 전달된 pin에 해당하는 PORT를 찾음
    volatile uint8_t *out;

    if (port == NOT_A_PIN) return; // 전달된 pin에 해당하는 port가 존재하지 않으면 함수 종료
    // 사용자가 pin 번호를 잘못 입력했을 경우를 대비

    out = portOutputRegister(port); // PORT에 해당하는 Port Output Register를 찾음

    if (val == LOW)
        *out &= ~bit; // 0V 출력
    else {
        *out |= bit; // 5V 출력
    }
}
```


Arduino Mega ADK GPIO

GPIO Register

Arduino에서 제공하는 GPIO 제어 함수

■ digitalWrite(uint8_t pin)

- Arduino Pin을 통해 값 입력

```
int digitalWrite(uint8_t pin)
{
    uint8_t bit = digitalPinToBitMask(pin); // 전달된 pin에 대응하는 레지스터를 표시하는 bit를 찾음
    uint8_t port = digitalPinToPort(pin); // 전달된 pin에 해당하는 PORT를 찾음
    volatile uint8_t *out;

    if (port == NOT_A_PIN) return LOW; // 전달된 pin에 해당하는 port가 존재하지 않으면 함수 종료
    // 사용자가 pin 번호를 잘못 입력했을 경우를 대비

    // PORT에 해당하는 Port Input Register를 찾음
    if (*portInputRegister(port) & bit) return HIGH;
    return LOW;
}
```

Q & A



<http://mesl.khu.ac.kr>