마이크로프로세서 (강의자료 #3)



교과목명 : 마이크로프로세서 (01) 담당교수 : 이 수 형 E-mail : <u>soohyong@uu.ac.kr</u> 교재명 : 스마트기기 개발을 위한 아두이노, 이수형. (LINC+ 사업단 배포)

수강생 안내

- 본 교과목은 기본적으로 "대면"수업입니다.
- '코로나바이러스감염증-19' 등의 이유로 대면수업을 받지 못하는 학생들을 위해 서 '비대면 실시간 수업'을 병행합니다. 기본적으로 대면 수업이므로 대면 수업에 참석한 학생들은 교실에서 출석을 체크하면 되며, 참석하지 못하고 '비대면 수업' 을 듣는 학생들은 '실시간 온라인 강의'에 참석하면 출석이 반영됩니다.
- 수강생 여러분들은 강의 자료를 온라인 배포 및 게재 등의 행위를 할 시 저작권 문 제가 발생할 수 있사오니 이에 유념하여 강의 자료를 공유하는 행위는 삼가 바랍니 다.
- 교재는 "스마트기기 개발을 위한 아두이노"이며, 나누어준 키트는 각자 잘 관리하 면서 실습실(대면) 및 집(비대면)에서 실험/실습을 수행할 수 있도록 진행합니다.

지난시간에는?

아두이노란

- 아두이노(arduino)는 이탈리아의 IDII(Interaction Design Institutelvera)에서 2005년에 만든 초보자를 위한 하드웨어 (hardware) 제작용 마이크로컨트롤러(microcontroller) 보드
- 비전공자 및 초보자도 쉽게 하드웨어에 접근
- 하드웨어 및 소프트웨어를 오픈소스로 공개 → 확산
- 간단하게 센서, 스위치 등으로 제어 가능
- C/C++ 기반 개발환경 제공

- 4 -





• 다양한 센서 모듈의 발전

- 스위치, 포텐쇼미터

- 의학용 : 근전도 센서, 심박동 센서

- 모터, LED, LCD Display 등

- 환경 : 온도 센서, 압력 센서, 습도 센서, 기압 센서, 광센서

- 직렬통신(RS232), 무선인터넷(WIFI), 블루투스, …

- 움직임 : 모션 센서, 움직임 센서, 가속도 센서, 근접 센서, 거리 센서

Div. of Energy and Electrical Engineering, Uiduk University

• 제어장치

• 통신

- 5 -

아두이노의 종류

	Uno	Leonardo	Due	Mega2560	
용도	대표적인 보드	키보드/마우스 장치로 구동	32비트 MPU	Uno의 확장 및 대용량	
MPU ATmega328		ATmega32u4	AT91SAM3X8E	ATMega2560	
동작전압	5V	5V	3.3V	5V	
디지털 I/O핀	14	20	54	54	
PWM핀	6	12	12	16	
아날로그 입력	6	12	12	16	
아날로그 출력	-	-	2	-	
SRAM	2KB	2.5KB	96KB	8KB	
Flash Memory	32KB	32KB	512KB	256KB	
EEPROM	1KB	1KB	-	4KB	
공급전원	USB/외부전원	USB/외부전원	USB/외부전원	USB/외부전원	

Div. of Energy and Electrical Engineering, Uiduk University

3. 아두이노 IDE의 이해

◎ ex3-1 아두이노 1.8.7	_	×
파일 편집 스케치 툴 도움말		
		ø
ex3-1		
int led = 13;		^
// 처음 시작시 한변만 수행하는 함수, 설정을 담당한다 void setup() { pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다 }		
// 반복해서 호출되는 함수 void loop() { digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다 delay(1000); // 1000 ms = 1초를 기다린다. digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다 delay(1000); // 1초를 기다린다. }		

아두이노 우노 보드





아두이노 IDE 실행 결과

◎ ex3-1 아두이노 1.8.7 -		◎ ex3-1 아두이노 1.8.7	- 0	×
파일 편집 스케치 툴 도움말		파일 편집 스케치 툴 도움말		
	₽ ₽			ø
ex3-1 §		ex3-1 §		
	^	int led = 13;		^
		// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다 void setup() { pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다 }		
	~	// 반복해서 호출되는 함수 void loop() { digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다 delay(1000); // 1000 ms = 1초를 기다린다. digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다 delay(1000); // 1초를 기다린다. }		•
저장 완료.		저장 완료.		
1 Arduino/Genuino	Uno on COM4	13 Arduin	/Genuino Uno on I	сом4

프로그램의 구조

Review

- C/C++을 이용하지만, 일반적인 프로그램과 다른 구조
 - setup() 함수와 loop()함수로 이루어짐
 - setup() : 처음 한번만 수행, loop() : 무한히 반복해서 수행

ex1.m
<pre>void setup() { // put your setup code here, to run once:</pre>
}
<pre>void loop() { // put your main code here, to run repeatedly:</pre>
}

첫번째 스케치

Review

• 아두이노 연결 : USB를 이용하여 컴퓨터에 연결

```
Example 3-1
int led = 13;
// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
   pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다
}
// 반복해서 호출되는 함수
void loop() {
   digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다
              // 1000 ms = 1초를 기다린다.
   delay(1000);
   digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다
              // 1초를 기다린다.
   delay(1000);
```

Review

- pinMode() 함수
 - -디지털입/출력 핀의 사용 방식 설정 (입력/출력) pinMode(13, OUTPUT); pinMode(12, INPUT);
 - 보통 초기에 한번만 수행하면 됨 ⇒ setup() 함수
- digitalWrite() 함수
 - 디지털 출력핀으로 출력값을 지정 (HIGH, LOW 또는 1, 0)
 - -HIGH:5V,LOW:OV digitalWrite(13, HIGH);
- delay() 함수
 - 시간지연 함수, 단위 mili-second. delay(1000);

4. 디지털 입출력의 기초

◎ ex3-1 아두이노 1.8.7	_	×
파일 편집 스케치 둘 도움말		
		Ø
ex3-1		
int led = 13;		^
// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다 void setup() { pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다 }		
// 반복해서 호출되는 함수 void loop() { digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다 delay(1000); // 1000 ms = 1초를 기다린다. digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다 delay(1000); // 13번 핀으로 0V 디지털 신호를 출력한다 } }		

4.1 LED를 이용한 디지털 신호 출력

- LED (Light Emitting Diode)
 - 1962 닉 홀로니악이 발명
 - 전원이 방향에 따라 빛을 방출
 - Anode (양극), Cathode (음극)
 - 초기 : 빨간색, 초록색
 - 최근 : 노란색, 파란색, 흰색, 3색
 - 내부저항이 적으며 20~30mA의 전류 사용
 - → 5V연결시 200옴 정도의 저항을 직렬로 연결





• 브레드보드

- 전원선으로 사용되는 가로선 (빨간색 : +, 파란색 : -)

- 구멍 사이의 간격은 0.1인치 (2.54mm) : DIP타입의 IC 칩의 간격과 동일



- 저항
 - 전류의 흐름을 방해하는 소자
 - 저항값: 색 띠로 계산



구분		첫번째 띠	두번째 띠	세번째 띠	내번째 띠	다섯번째 띠
색상		숫자 1	숫자 2	숫자 3	승수	오차
	검은색	0	0	0	100 = 1	
	갈색	1	1	1	10 ¹ = 10	±1%
	빨간색	2	2	2	$10^2 = 100$	±2%
	주황색	3	3	3	$10^3 = 1000$	
	노란색	4	4	4	$10^4 = 10000$	
	초록색	5	5	5	$10^5 = 100000$	±0.5%
	파란색	6	6	6	$10^6 = 1000000$	±0.25%
	보라색	7	7	7	107 = 10000000	±0.1%
	회색	8	8	8	$10^8 = 100000000$	±0.05%
	흰색	9	9	9		
	황금색					±5%
	은색					±10%

• 회로 구성



• 스케치

```
예제 4-1. 첫번째 스케치
```

```
int led = 13;
// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
   pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다
}
// 반복해서 호출되는 함수
void loop() {
   digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다
   delay(1000);
                    // 1000 ms = 1초를 기다린다.
   digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다
              // 1초를 기다린다.
   delay(1000);
}
```



• 13번 핀이 아닌 다른 곳에 연결하여 동일한 동작을 하도록 회로 및 스 케치를 수정하라.

4.2 두개의 LED를 제어하기

• 회로 구성



예제 4-2. 두 개의 LED 순차 점등

```
int ledA = 13;
int ledB = 12;
// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다.
void setup() {
   pinMode(ledA, OUTPUT); // 13번 핀을 출력으로 설정한다
   pinMode(ledB, OUTPUT); // 12번 핀을 출력으로 설정한다
}
// 반복해서 호출되는 함수
void loop() {
  // LED 1만 켜기
   digitalWrite(ledA, HIGH); // LED 1 켜기
   digitalWrite(ledB, LOW); // LED 2 끄기
               // 1초를 기다린다.
   delay(1000);
  // LED 2만 켜기
   digitalWrite(ledA, LOW); // LED 1 끄기
   digitalWrite(ledB, HIGH); // LED 2 켜기
              // 1초를 기다린다.
   delay(1000);
}
```

• 추가 실습

- 3개의 LED를 사용하여 신호등처럼 동작하는 회로를 구성하고 스케치를 완성하 라.

- 파란색 3초, 노랑색 0.5초씩 2회 깜박임, 빨간색 3초

4.3 스위치를 이용한 디지털 입력



• 회로도







예제 4-3. 디지털 입력 프로그램

```
// 디지털 입/출력 핀 2번을 스위치의 입력으로 설정한다.
int pushButton = 2;
void setup() {
 // 시리얼 통신을 위하여 초기화하는 과정, 9600보레이트(baud-rate)로 설정한다.
 Serial.begin(9600);
 // 버튼 스위치가 연결된 2번 핀을 입력 모드로 설정한다.
 pinMode(pushButton, INPUT);
}
void loop() {
 // 현재 버튼이 연결되어 있는 핀의 값을 읽어들인다.
 int buttonState = digitalRead(pushButton);
 // 입력받은 값을 시리얼 통신을 통해서 PC로 전송한다.
 Serial.println(buttonState);
   // 입력 상태를 안정화하기 위하여 1미리초 동안 대기한다
 delay(1);
```

• 채터링(chattering)

- 스위치의 물리적인 접촉이 일어나는 경우 잡음이 발생





10uF

UART 통신

- 시리얼 통신
 - 아두이노와 PC와의 통신
 - UART (Universal Asynchronous Receiver and Transmittor)
 - RS-232, RS-422, RS-485 등의 표준방식이 존재
 - 아두이노에는 디지털 포트 0번과 1번이 준비되어 있음
 - 추가 시리얼 통신은 소프트웨어로 해결



- Serial 객체
 - PC와의 시리얼통신을 위해 사용함
 - -setup()함수에서 초기화
 - >Serial.begin(9600);
 - ▶ 9600 보레이트(baud-rate)로 설정
 - ✓ Baud-rate : 300, 1200, 2400, 4800, <u>9600</u>, 19200, 38400, …
 - -Serial.println(값);
 - ▶ 주어진 값을 PC로 전송 print + line (줄넘김 포함)
 - ✓ Serial.println("Hello");
 - ✓ Serial.print(3.14);
 - ✓ Serial.print('A');
 - ▶ 읽기 : Serial.available() 함수와 Serial.read() 함수를 사용



 스위치 회로에 LED를 추가하여 스위치를 누르는 경우 LED에 불이 켜 지도록 회로를 구성하고 스케치를 작성하라.

시리얼 입력

- Serial.available()
 - 현재 시리얼 포트에 새로운 값이 전송되었는지 판단하는 함수
 - 이 함수를 이용하여 새로운 값이 있는 경우만 읽어 들임
- Serial.read()
 - 시리얼 포트에 저장되어 있는 값을 읽어 들임



-

✓ 9600 보드레이트 ✓

새 중

×

전송

출력 지우기

예제: 시리얼 입력을 통한 LED 제어

```
void setup() {
  // LED_BUILTIN : 내장된 LED 포트 번호 == 13
 Serial.begin(9600);
 pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  // 시리얼 통신 검사 후 데이터 읽기
  if(Serial.available()) {
   char ch = Serial.read();
   if(ch == '1')
     digitalWrite(LED_BUILTIN, HIGH);
   else if(ch == '2')
     digitalWrite(LED_BUILTIN, LOW);
 delay(10);
```

• 시리얼 모니터를 이용하여 전송

◎ testserial 아두이노 1.8.10 -	- 0	×	💿 сомз — С	x ı
파일 편집 스케치 둘 도움말			1	전송
		ø		
testserial				
<pre>void setup() { // LED_BUILTIN : 내장된 LED 포트 번호 == 13 Serial.begin(9600); pinMode(LED_BUILTIN, OUTPUT); } </pre>		^		
<pre>void loop() { // 시리얼 통신 검사 후 데이터 읽기 if(Serial.available()) { char ch = Serial.read(); if(ch == 'l') digitalWrite(LED_BUILTIN, HIGH); else if(ch == '2') digitalWrite(LED_BUILTIN, LOW); } delen(10); </pre>				
<pre>delay(10); }</pre>		v		
저장 완료. 스케치는 프로그램 저장 공간 1958 바이트(6%)를 사용. 최대 3 전역 변수는 동적 메모리 196바이트(9%)를 사용, 1852바이트의	32256 바이9 지역변수기	Ē. ^ ├ 남음 └		
12 Arduino/Ge	nuino Uno <u>o</u> n	сомз	☑ 자동 스크롤 □ 타임스탬프 표시 새 줄 9600 보드레이트 (출력 지우기