뻔뻔한 아두이노 – 예제

- 1. [시작하기] 아두이노 소프트웨어 설치 및 브레드보드 이용하기
- 2. [출력1] LED 깜빡이기
- 3. [출력2] 시리얼 모니터에 텍스트 출력하기
- 4. [출력3] PWM을 이용한 LED 밝기 조절
- 5. [출력4] 여러 개 LED 동시에 제어하기
- 6. [출력5] 쉬프트 레지스터 (74HC595)
- 7. [출력6] 서보 모터 구동하기
- 8. [출력7] H-Bridge를 이용한 DC 모터 방향 제어하기
- 9. [입력1] 키보드로 문자 입력 받아 아두이노에 전송하기
- 10.[입력2] 스위치 사용하여 LED 컨트롤
- 11.[입력3] 근접 물체 감지하기 (QRD1114)
- 12.[입력4] 온도 센서로 온도 감지하기 (TMP36)
- 13.[입력5] MPU-6050 이용 각속도/가속도/온도 측정하기
- 14.[입력6] Flex 센서를 이용한 구부림 측정하기
- 15.[입력7] FSR 압력 센서를 이용한 압력 측정하기
- 16.[응용1] 밝기에 따른 DC 모터 속도 제어
- 17.[응용2] 장애물 감지 후 모터의 방향 제어
- 18.[응용3] 손가락으로 모터 제어하기
- 19.[응용4] 압력에 따른 LED의 밝기 제어

20.[응용5] 키보드로 모터 방향 제어하기

21.[응용6] Visual Studio Express C#와 아두이노 인터페이스

22.[응용7] 매틀랩을 이용한 센서값 디스플레이

1. [시작하기] 아두이노 소프트웨어 설치 및 브레드보드 이용하기

1) 아두이노 소프트웨어 다운로드

아두이노 소프트웨어는 일반적으로 스케치라고 부르며 이 스케치는 arduino.cc/en/Main/Software 에서 다운로드 받으실 수 있다. 주소창에 위의 주소를 입력하면 다음과 같이 다운로드 페이지가 열린다. Windows, Mac OS X, Linux 등의 플랫폼에 맞는 스케치를 다운로드 받을 수 있으며, 이번 매뉴얼은 많은 분들이 사용하시는 윈도우를 기반으로 작성되었기에 윈도우용 스케치를 다운받겠다.

← → C 🗋 arduino.cc/en/Main/Software			술 🛃 :
Main Site Blog Playground Forum Labs Store	and the second se		Help Sign in or Register
	Download the Arduino Software		
	The open-source Arduino environment makes it easy to write or Mac OS X, and Linux. The environment is written in Java and ba software.	de and upload it to the i/o board. It runs on Windows, sed on Processing, avr-gcc, and other open source	
	THE Arduino SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MA IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCT OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLEO WARRAN MERCHARTABLITY, PRIVES FOR A PARTUCULAR PURPOSE, OR IND DORESSY DISCLAIM ANY LIABLITY WHATSOEVER FOR ANY OBEE CONSEQUENTIAL, INCIDENTIAL OR SPECIAL DAMAGES, INCLUDING LOST REVENUES, LOST PROFITS, LOSSE RESULTING FROM BUSING LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LIGGAL THE LUBALITY MAY BE ASSIRTED, EVEN IF ADVISED OF THE POSSI OF SUCH DAMAGES	KE NO EXPRESS OR IONALITY, OPERABILITY, ITTES OF T, WORKCT, WE THOUT LIMITATION, S WITSRUPTION OR THEORY UNDER WIECH RUTY OR LIKELIHOOD	
	By downloading the software from this page, you agree to the sp	cified terms.	
	Download	Next steps	
	Arduino 1.0.3 (release notes), hosted by Google Code: + <u>Windows</u> + Mac OS X + Linux: 32 bit, 64 bit + source	Getting Started Reference Environment Examples Foundations FAQ	
	Download Arduino DUE Board Software		
	If you have the new Due Board click here		
arduino.googlecode.com/files/arduino-1.0.3-windows.zip			

아두이노 스케치 다운로드 페이지

사용하는 OS가 윈도우가 아니면, 위에서 사용할 플랫폼에 맞게 다운로드 받을 수 있다. 그리고 다운로드 받은 플랫폼은 저장하고 싶은 위치에 저장하면 된다. 아두이노 스케치는 별도의 설치가 필요 없으며, 다음의 아두이노 아이콘을 이용하여 소프트웨어를 실행할 수 있다. Mac을 사용한다면, 아두이노 다운로드는 디스크 이미지(.dmg)로 제공되며, 다운로드가 완료된 후 파일을 두 번 클릭하면 이미지가 마운트된다. 리눅스는 리눅스 버전에 따라 달라지기 때문에 자세한 정보는 아두이노 위키 (<u>http://www.arduino.cc/playground/Learning/Linux</u>)를 참조하거나 메카솔루션의 기술상담 게시판을 이용할 수 있다. 다운로드 및 설치가 끝나고 폴더를 열어서 아두이노 아이콘을 더블 클릭하면 다음과 같이 스케치가 열리며 처음에는 시간이 약간 걸릴 수 있기 때문에 기다리도록 하자.

and a local							X
Comput	er ▶ 로컬 디스크 (C:) ▶ Arduino	▶ arduino-1.0.3 ▶			arduino-1.0.3 검색		P
구성 ▼ 🖬 열기	새 폴더					•	0
☆ 즐겨찾기	이름	수정한 날짜	유형	크기			
Stopbox	퉲 drivers	2013-01-23 오전	파일 폴더				
🚺 다운로드	examples	2013-01-24 오후	파일 폴더				
🌉 바탕 화면	l hardware	2013-01-23 오전	파일 폴더				
💹 최근 위치	퉬 java	2013-01-23 오전	파일 폴더				
	퉬 lib	2013-01-23 오전	파일 폴더				
🍃 라이브러리	퉬 libraries	2013-01-24 오후	파일 폴더				
📑 문서	퉬 reference	2013-01-23 오전	파일 폴더				
😸 비디오	퉬 tools	2013-01-23 오전	파일 폴더				
🔜 사진	💿 arduino	2013-01-23 오전	응용 프로그램	840KB			
🚽 음악	cygiconv-2.dll	2013-01-23 오전	응용 프로그램 확장	947KB			
	🚳 cygwin1.dll	2013-01-23 오전	응용 프로그램 확장	1,829KB			
🔣 홈 그룹	🚳 libusb0.dll	2013-01-23 오전	응용 프로그램 확장	43KB			
	revisions	2013-01-23 오전	텍스트 문서	36KB			
Normal Computer	🚳 rxtxSerial.dll	2013-01-23 오전	응용 프로그램 확장	76KB			
🏭 로컬 디스크 (C:)							
📬 네트워크							
arduino 응용 프로그램	수정한 날짜: 2013-01-23 오전 7:5 크기: 840KB	56 만든 날짜: 2012-12-10 오후 12:5	0				

아두이노 실행 폴더: 위의 아이콘을 더블 클릭하면 아두이노 스케치가 열린다.



아두이노 스케치 초기화면

아두이노 소프트웨어를 오픈하였을 때 보이는 창이다. 현재, 아두이노는 Arduino Pro Mini로 설정되어 있고, 시리얼포트는 COM5 설정되어 있음이 오른쪽 하단에 표기되어 있다. 우리가 사용할 아두이노 우노와 적합한 시리얼포트를 설정하기 위해 아두이노 우노 드라이버가 잘 설치 되었는지 확인한다.

2) USB로 아두이노와 컴퓨터 연결

만약, USB를 이용하여 아두이노에 연결하기 전과 연결한 후의 시리얼포트 인식 변화가 없다면, 아두이노의 드라이버가 설치되어 있지 않은 것이다. 아두이노 소프트웨어에서 **도구 > 시리얼 포트**를 클릭한다. 만약, 회색톤으로 시리얼포트가 활성화되어 있지 않다면, 아두이노 드라이버가 연결되어 있지 않은 것이다. 또한, USB를 꼽았다가 빼었다를 하면서 시리얼포트에 변화가 없다면 이 또한, 아두이노 드라이버를 인식하지 못한 것이기 때문에, 아두이노 드라이버를 매뉴얼로 설정해준다.

💿 sketch_feb03a 아프	루이노 1.0.3		
파일 편집 스케치 도	구 도움말		
sketch_feb03a	자동 포맷 스케치 보관하기 인코딩 수정 & 새로 고침	Ctrl+T	ي ج ا
	시리얼 모니터	Ctrl+Shift+M	
	보드	۱.	
	시리얼 포트	Þ	
	프로그래머 부트로더 굽기	•	
			-
			,
1	Arduino Pro or Pro Mini (3	3.3V, 8 MHz) w/ ATi	mega328 on COM5

시리얼 포트가 설정되지 않은 상황: 드라이버를 설치해야 한다.

3) 드라이버 설치

장치관리자에 들어가기 위해 왼쪽 아래에 있는 실행버튼을 누르고 프로그램 및 파일 검색이라는 조그만 검색창에 devmgmt.msc 를 입력하고 검색한다. 그럼, 다음과 같은 장치관리자가 나타나게 된다. 혹은 시작버튼 - 제어판 - 시스템 및 보안 - 시스템 - 장치관리자를 이용해서 장치관리자에 접근할 수 있다. 장치관리자가 열리면 다음과 같이 느낌표로 드라이버 설치를 요구하는 아두이노를 발견할 수 있고, 오른쪽 마우스버튼을 클릭하여 드라이버 소프트웨어 업데이트를 클릭한다.

2 E IS	드라이버 소프트웨어 업데이트(P)
5	사용 안 함(D)
플러	제거(U)
-	하드웨어 변경 사항 검색(A)
ii ii	속성(R)

장치관리자 화면과 아두이노 소프트웨어 업데이트

🕞 🗕 드라이버 소프트웨어 업데이트 - Arduino UNO(COM3)	×
드라이버 소프트웨어는 어떻게 검색합니까?	
→ 업데이트된 드라이버 소프트웨어 자동으로 검색(5) 컴퓨터와 인터넷에서 장치에 대한 최신 드라이버 소프트웨어를 검색합니다. 이렇 게 하지 않으려면 장치 설치 설정에서 이 기능을 사용하지 않도록 설정해야 합니 다.	
→ 컴퓨터에서 드라이버 소프트웨어 찾아보기(R) 수동으로 드라이버 소프트웨어를 찾아 설치하십시오.	
	취소

드라이버 소프트웨어 업데이트: 수동으로 업데이트 하기

그 후, 컴퓨터에서 드라이버 소프트웨어 찾아보기를 클릭하여 수동으로 검색을 한다.

찾아보기를 클릭하여 압축을 푼 아두이노 폴더에 들어가서 drivers라는 폴더를 지정한다. 그리고 다음버튼을 누르면 드라이버가 설치됩다. 가령 C:\#Arduino에 다운받은 아두이노를 압축 풀었다면 C:\#Arduino\#arduino-1.0.3\#drivers까지 설정하고 설치를 하면 된다. 설치가 되지 않으면 메카솔루션의 기술상담으로 문의할 수 있다.

4) 아두이노 보드와 시리얼포트 설정

설치된 아두이노에 몇 번의 시리얼(COM포트)이 입력되어 있는지 확인하기 위해서 USB를 끼웠다 빼었다하면 도구 - 시리얼포트에 변화가 생기는 것을 볼 수 있다. 변화가 있는 시리얼포트가 현재 연결되어 있는 아두이노의 시리얼포트입니다. 시리얼포트를 클릭하여 설정한다.

5) 브레드보드와 익숙해지기.

일명 빵판이라 불리우는 브레드보드는 납땜없이 테스팅을 하기 위해 만들어져 있다.

일반적인 브레드보드는 다음과 같이 연결되어 있다.





브레드보드 정면과 내부 케리커처 (출처: SIK GUIDE, Sparkfun)

양쪽 끝에 있는 파란색, 그리고 빨간색의 세로열은 하나로 연결되어 있어서, VCC 및 GND로 사용되고, 가로열은 일반적으로 5-6개의 핀을 하나로 연결시킴으로서 5-6개의 핀에 연결된 케이블 및 핀들이 동일한 신호를 공유할 수 있도록 해 준다.

6) 프로그램을 아두이노에 업로드

앞으로 설명될 예제 및 여러분들이 작성한 프로그램들은 다음과 같은 절차를 거쳐 아두이노 보드에 업로드 된다. 먼저, 1) 프로그램을 작성 혹은 다운 받고, 2) 프로그램을 확인(Verify)를 한 후, 3) 프로그램을 업로드(Upload)한다. 이 세 단계가 아두이노에서 가장 중요한 스텝이다.

2. [출력1] LED 깜빡이기

가장 기본적인 출력이면서 동시에 아두이노가 제대로 동작하는지를 확인할 수 있다. 아두이노 우노는 13번핀이 LED에 연결되어 있는 채 출고되기 때문에 별다른 연결 없이 USB 케이블과 아두이노만으로 LED 깜빡이기 예제를 수행할 수 있다.

준비물: 아두이노 우노, USB 케이블

따라하기:

- USB를 이용하여 아두이노와 컴퓨터의 USB포트를 연결하기.
- 아두이노 소프트웨어의 파일 예제 Basics Blink를 클릭.
- LED 깜빡이는 예제가 다음과 같이 열린다.

፩ Blink 아두이노 1.0.3		x
파일 편집 스케치 도구 도울	(말	
	2	D -
Blink	- I	
// Pin 13 has an LED connec	ted on most Arduino boards.	~
int led = 13;		
// the setup routine run; O	nce when you press reset:	
Vold Setup() { // ipitialize the digital	nin se an outout	
pinMode(led, NUTPUT);	pin as an output.	
}		
// the loop routine runs ov	er and over again forever:	=
<pre>void loop() {</pre>		-
digitalWrite(led, HIGH);	// turn the LED on (HIGH is the voltage level)	
delay(1000);	// wait for a second	
digitalWrite(led, LUW);	// turn the LEU off by making the voltage LUW	
deray(1000); l	// Walt for a second	
ſ		-
•		•
1	Arduino Uno on COM	19

첫번째 부분은 변수 선언 파트로 13번 핀을 led로 사용하겠다는 의미이며, 두번째 void setup()는 아두이노에 전원이 공급되면 한번 실행되는 메소드이며 핀에 대한 입출력 설정 및 시리얼 함수

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

초기화 등을 한다. pinMode(led, OUTPUT)이라는 커맨드는 led를 출력으로 사용하겠다는 의미이다. 여기서 led는 핀번호를 의미하기 때문에 13번 핀을 출력으로 설정이라고 생각하면 된다.

그리고 세번째 void loop()는 아두이노에 전원이 들어온 후 꺼질 때까지 계속 반복해서 실행되는 메소드이며 위의 소스를 보면 digitalWrite(led, HIGH)라는 커맨드에서 led(13번 핀)에 출력을 HIGH(5V)로 출력하여 LED를 켜고 delay(1000)이라는 함수로 1000ms(1초)동안 켠 상태를 유지한 다. 그리고 다시 digitalWrite(led, LOW)라는 커맨드로 led에 출력을 LOW(0V)로 하여 LED를 끄고 delay(1000)이라는 함수로 1000ms(1초)동안 끈 상태를 유지한다. 결과적으로 void loop()는 LED가 1초에 한번씩 꺼졌다 켜졌다를 반복하는 깜빡이는 동작을 하게 한다.

- 왼쪽 상단의 확인(Verify) 버튼을 누른 후, 완료가 되면 업로드 버튼을 눌러서 LED 깜빡이기 프로그램을 아두이노에 올려보자.
- 확인이 되었으면, delay(1000)을 delay(100)으로 바꿔서 LED가 더 빨리 깜빡이는지도 확인해보자.

3. [출력2] 시리얼 모니터에 텍스트 출력하기

아두이노에는 USB 케이블을 이용하여 간단히 컴퓨터와 데이터를 주고 받을 수 있다. 특히, 시리얼 모니터라는 기능을 유용하게 이용할 수 있는데, 간단한 예제를 통해서 시리얼 데이터를 아두이노에서 컴퓨터로 보내본다.

준비물: 아두이노 우노, USB 케이블

따라하기:

- 앞서 LED 깜빡이기 예제를 수행하였다면 별다른 하드웨어 변경은 필요없다. USB 케이블을 이용하여 아두이노와 컴퓨터를 연결한다.
- 다음의 간단한 프로그램을 아두이노의 스케치에 복사한 후, 확인(컴파일) 그리고 업로드를 한다.

void setup()
{
Serial.begin(9600); // 시리얼 포트를 사용하겠다는 선언
}
void loop()
{
Serial.println("Mechasolution"); // Mechasolution이라는 텍스트를 줄바꾸면서 출력한다.
}

 업로드 후, 스케치의 오른쪽 상단에 있는 시리얼 모니터 아이콘을 클릭하거나, 도구-시리얼 모니터, 혹은 Ctrl+Shift+M을 눌러서 시리얼 모니터 창을 띄워본다.



4. [출력3] PWM을 이용한 LED 밝기 조절

LED의 밝기를 조절하려면 어떻게 해야 할까. PWM(Pulse Width Modulation)이라는 것을 이용하여 LED의 밝기를 조절할 수 있다.



위의 그림에서와 PWM은 신호의 On/Off의 비율을 변경시키는 방법을 통해 높고 낮은 레벨의 출력을 할 수 있도록 한다. 즉, Off의 비율이 많으면 점점 흐릿해지는 LED를 출력하게 되고, On의 비율이 많으면 점점 밝아지는 LED 출력을 하게 된다. 이 PWM은 아날로그 출력이라는 함수를 사용하게 되는데 이 출력에 사용할 수 있는 핀의 개수가 제한되어 있다. 아두이노 우노에서는 3,5,6,9,10,11번의 6개 핀을 사용할 수 있고, 디지털 핀 쪽의 물결표시(~)로 확인할 수 있다.

준비물: 아두이노 우노, USB 케이블, 브레드보드, LED, 330옴 저항, 점퍼 케이블

따라하기:

LED는 애노드와 캐소드라는 두 개의 리드가 있는 반도체 장치이다. 다리가 긴 쪽이 애노드(+)이고, 다리가 짧은 쪽이 캐소드(-)이다. 통상적으로 LED에 필요한 전압은 1.8~2.3V이므로 5V에서 3V가량을 전압강하 해야 한다. 그리고 LED가 끌어쓰는 전류를 감안해서 약 330옴의 저항을 연결한다.

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)



• 다음의 소스를 스케치에 복사 혹은 타이핑을 한 후 확인-업로드를 해 보자.

```
int led = 3;
int brightness = 0;
int increment = 1;
void setup()
{
   // analogWrite을 통해 구동되는 핀은 출력으로 선언하지 않아도 된다.
}
void loop()
{
   if(brightness > 255)
   {
      increment = -1; // 255에 도달하면 값은 낮춘다.
   }
   else if(brightness < 1)
   {
      increment = 1; // 0으로 떨어지면 값을 높인다.
   }
   brightness = brightness + increment; // 밝기(brightness)를 증가 혹은 감소한다.
   analogWrite(led, brightness); // 밝기를 출력한다
   delay(10);
```

매번 LED의 밝기가 밝아졌다가 어두워졌다가 하는 것을 확인할 수 있다. 아날로그 신호를 디지털 신호로 읽기 위해서는 analogRead()를 사용하며 0~1023(10bit)까지의 값으로 변환해서 읽을 수 있고, 이번 예제에서 사용한 analogWrite()는 0~255(8bit)까지변환하여 출력할 수 있다.

5. [출력4] 여러 개 LED 동시에 제어하기

LED를 깜빡이고, PWM을 이용한 밝기 조절을 하였다면 여러 개의 LED를 제어하는 것은 어려운 일이 아니다. 이번 예제를 통해 8개의 LED를 순차적으로 깜빡이는 것을 for 루프와 배열에 대하여 공부도 해보도록 하자.

- for 루프 간략하게 설명을 하자면 여러분이 어떤 코드를 여러 번 실행해야 할 때,
 사용하는 매우 유용한 반복문의 일종이다.
- 배열 배열(array)는 여러 개의 변수를 지정할 때 사용하며 배열의 인덱스를 이용하여
 그 값을 지정할 수 있고 배열에 들어있는 값을 요소(element)라 한다.

다음의 예를 통하여 for 루프와 배열에 대하여 알아보자. 먼저, LED 8개를 Digital핀 2번에서 9번까지 8개를 연결할 것을 설정하였다. 각각의 LED에 대하여 ledPin1에서 ledPin8이라고 설정을 했다. 그리고 setup() 메소드에서 이 LED들을 입력이 아닌 출력으로 사용할 것을 pinMode함수를 통해 지정하였다. 왼쪽의 코드가 잘못된 것은 전혀 아니지만, for 루프와 배열을 이용하여 좀 더 컴팩트한 프로그램을 구현해 보도록 하자. 오른쪽의 코드는 intledPins[] 라는 배열을 선언했다. 8개의 숫자가 ledPins라는 배열에 할당이 된다. 아두이노에서 배열은 0번이 첫번째 요소를 지칭한다. 따라서, int ledPins[0] = 2; int ledPins[1] = 3, 그리고, int ledPins[7]은 9가 된다. 이 8개의 LED들을 OUTPUT으로 설정하기 위하여 for(int i=0; i<=7; i++) 라는 for 루프문을 사용한다. 이는 i라는 인덱스를 이용하여 0에서 7까지 i를 1씩 증가시키겠다는 의미이다. 따라서 for 루프가 시작되는 처음에는 i=0, pinMode(ledpins[0], OUPUT); i를 1씩 증가시켜 다음에는 i=1이 되고 pinMode(ledPins[1], OUTPUT); 그리고 마지막으로 i=7; pinMode(ledPins[7], OUTPUT)이 되고 for 루프문은 종료된다.

int ledPin1 = 2;	int ledPins[] = {2,3,4,5,6,7,8,9};
int ledPin2 = 3;	
int ledPin3 = 4;	void setup()
int ledPin4 = 5;	{
int ledPin5 = 6;	for(int i = 0; i <= 7; i++)
int ledPin6 = 7;	{
int ledPin7 = 8;	pinMode(ledPins[i],OUTPUT);
int ledPin8 = 9;	}
	}
void setup()	
{	

pinMode(ledPin1, OUTPUT);	
pinMode(ledPin2, OUTPUT);	
pinMode(ledPin3, OUTPUT);	
pinMode(ledPin4, OUTPUT);	
pinMode(ledPin5, OUTPUT);	
pinMode(ledPin6, OUTPUT);	
pinMode(ledPin7, OUTPUT);	
pinMode(ledPin8, OUTPUT);	
}	

기본적으로 왼쪽의 코드와 오른쪽의 코드는 같다. 하지만, 왼쪽의 코드는 오른쪽의 코드보다 50바이트 정도 사이즈가 더 크다. 한정된 메모리를 사용하는 아두이노인만큼 최적화된 프로그램을 위해 배열과 for 루프를 적절히 사용하도록 하자.

본격적으로 8개의 LED를 제어해보자.

준비물: 아두이노 우노, USB 케이블, 브레드보드, LED 8개, 330옴 저항 8개, 점퍼 케이블

따라하기:

- LED는 애노드와 캐소드라는 두 개의 리드가 있는 반도체 장치이다. 다리가 긴 쪽이 애노드(+)이고, 다리가 짧은 쪽이 캐소드(-)이다. 통상적으로 LED에 필요한 전압은 1.8~2.3V이므로 5V에서 3V가량을 전압강하 해야 한다. 그리고 LED가 끌어쓰는 전류를 감안해서 약 330옴의 저항을 연결한다.
- 330옴을 LED의 짧은 다리와 GND에 연결한다. 그리고, LED의 다리가 긴 쪽을 아두이노의 디지털핀2번에서 9번까지 연결한다. (다음 그림에서 사용된 저항 이미지는 임의로 사용되었기 때문에 330옴(주황-주황-갈색)을 사용한다. 주황은 3을 갈색은 0을 의미함)



• 다음의 소스를 스케치에 복사 혹은 타이핑을 한 후 확인-업로드를 해 보자.

```
int ledPins[] = {2,3,4,5,6,7,8,9}; // 8개의 LED를 Digital2번부터 Digital9번까지 사용함.
void setup()
{
  for(int i = 0; i <= 7; i++)
  {
    pinMode(ledPins[i],OUTPUT); // 8개의 LED를 출력으로 설정.
  }
}
void loop()
{
  for(int i = 0; i <= 7; i++)
```

}

```
{
digitalWrite(ledPins[i], HIGH); // D2부터 D9에 연결된 LED를 순차적으로 켬
delay(100); // 100ms(0.1초)동안 기다림.
}
for(int i = 7; i >= 0; i--)
{
digitalWrite(ledPins[i], LOW); // D9부터 D2에 연결된 LED를 순차적으로 끔
delay(100); // 100ms(0.1초)동안 기다림.
}
```

6. [출력5] 쉬프트 레지스터 (74HC595)

만약, 여러 개의 LED를 제어해야하지만 그만큼의 사용가능한 핀들이 없다면(많은 핀들을 모터 및 센서에 할당한 경우), 쉬프트 레지스터를 이용하여 다수의 LED를 제어할 수 있다. 쉬프트 레지스터는 IC(integrated circuit)의 일종으로 3개의 핀을 이용하여 8개의 각기 다른 출력을 할 수 있다. 쉬프트 레지스터는 많은 종류가 있지만, 74HC595라는 칩을 사용하도록 하자. 다음은 74HC595칩의 핀 배열 및 논리값이다.

Q _B [Q _C [Q _D [Q _E [Q _F [Q _G [1 2 3 4 5 6 7	υ	16 15 14 13 12 11 10] V _{CC}] Q _A] SER] OE] RCLK] SRCLK] SRCLR
Q _H [GND [7		10 9	
	ľ		Ĭ	n œH,
	-			

SER, SRCLK, RCLK는 74HC595의 제어 인풋이며 아두이노에서 값을 받는다. 그리고 QA~QH까지는 74HC595의 출력이며 최대 8개의 LED에 연결이 가능하다. SER은 data핀, SRCLK는 Clock, RCLK는 latch 핀으로 각각 아두이노의 디지털핀 2,3,4번에 연결할 수 있다. 우리가 사용할 아두이노의 핀은 D2, D3, D4이며 이들은 각각 data, clock, latch 핀이 된다. 이번에는 bitWrite()와 shiftOut()이라는 함수를 이용하여 시리얼 시프트 함수를 작성할 것이다.

- bitWrite(): bitWrite는 숫자를 bit로 바꾸서 기록하는 함수이며 bitWrite(x,n,b)와 같이 사용된다. 여기서 x는 bit로 바꿀 숫자이며 n은 기록될 디지털 핀, 그리고 b는 bit에 쓸 불리언 수(0 혹은 1)을 나타낸다. (참고: http://arduino.cc/en/Reference/BitWrite)
- shiftOut(): shiftOut은 1byte를 한번에 1bit씩 밀어내는 시프트 출력 명령입니다. shiftOut(dataPin, clockPin, bitOrder, value)와 같이 사용되며 dataPin은 74HC595의 SER과 연결된 핀, clockPin은 SRCLK에 연결된 핀을 지정하면 된다. bitOrder는 MSBFIRST (Most Significant Bit First) 혹은 LSBFIRST (Least Significant Bit First) 중 선택이 가능한데 이는 쉬프트하는 방향을 의미한다. 그리고 value는 쉬프트출력을 하려는 data이다.

준비물: 아두이노 우노, USB 케이블, 74HC595, 브레드보드, LED 8개, 330옴 저항 8개, 점퍼 케이블

따라하기:

• 먼저 74HC595와 아두이노, 그리고 8개의 LED와 저항들을 다음에 보이는 회로와 같이 연결한다.



• 다음의 소스를 스케치에 복사 혹은 타이핑을 한 후 확인-업로드를 해 보자.

int datapin = 2;	
int clockpin = 3;	
int latchpin = 4;	
byte data = 0;	
void setup()	
{	

```
pinMode(datapin, OUTPUT);
  pinMode(clockpin, OUTPUT);
  pinMode(latchpin, OUTPUT);
}
void loop()
{
  for(int i = 0; i <= 7; i++)
  {
    shiftWrite(i, HIGH);
    delay(100);
  }
  for(int i = 7; i >= 0; i--)
  {
    shiftWrite(i, LOW);
    delay(100);
  }
}
void shiftWrite(int desiredPin, boolean desiredState)
{
  bitWrite(data,desiredPin,desiredState);
  shiftOut(datapin, clockpin, MSBFIRST, data);
  digitalWrite(latchpin, HIGH);
  digitalWrite(latchpin, LOW);
}
```

7. [출력6] 서보 모터 구동하기

우리가 사용할 서보 모터는 내부에 포텐쇼미터 혹은 볼륨이 있어서 모터가 어느 정도 회전했는지 센싱하고, 모터 내부에 있는 컨트롤러를 이용하여 입력 받은 값만큼 돌려주는 기능을 가지고 있다. 대부분의 서보 모터는 무한 회전을 하지 않고 180도 정도 회전하는데, 아두이노의 PWM을 이용하여 각도를 컨트롤 할 수 있다. 또한, 아두이노의 delay함수를 이용하여 모터의 회전 속도도 조절할 수 있다. 앞에서 보았던 PWM을 이용하여 모터를 회전할 수 있지만, 아두이노의 강점인 오픈 라이브러리를 이용하여 서보 모터를 쉽게 컨트롤해보자. 아두이노 소프트웨어를 다운 받으면 Servo.h라는 라이브러리가 함께 설치된다. 따라서 별도의 라이브러리 추가를 할 필요 없이 그대로 사용하면 된다. Servo.h 라이브러리는 다음의 함수를 포함한다.

- servo.attach(pin): 서보 모터가 연결될 핀을 지정한다. 서보 모터는 PWM으로 제어되기 때문에 디지털 핀 중 물결표시가 있는 핀에 연결하면 된다. 우노의 경우 3, 5, 6, 9, 10, 11번을 사용할 수 있다.
- servo.write(angle): 회전시킬 각도를 입력한다. 이 함수가 실행되면 모터는 angle만큼 회전한다.

준비물: 아두이노 우노, USB 케이블, 서보 모터, 브레드보드, 점퍼 케이블

따라하기:

 서보 모터는 일반적으로 3개의 선을 가지고 있다. VCC(빨강), GND(검정 혹은 갈색), 그리고 신호선(노랑 혹은 주황). VCC와 GND를 아두이노에 연결하고, 신호선을 디지털9번 핀에 연결한다.



• 다음의 소스를 스케치에 복사 혹은 타이핑을 한 후 확인-업로드를 해 보자.

#include <Servo.h> // 서보 라이브러리 Servo motor1; // 서보 모터 오브젝트 void setup() { motor1.attach(9); // 서보 모터를 9번에 연결 } void loop() { for(int position = 0; position < 180; position += 2) // 0도에서 180까지 2도씩 증가

```
{
	motor1.write(position); // 모터를 position만큼 회전
	delay(20); // 모터가 돌 때까지 기다림.
}
for(int position = 180; position >= 0; position -= 2)//180도에서 0도까지 2도씩 감소
{
	motor1.write(position); // 모터를 position만큼 회전
	delay(20); // 모터가 돌 때까지 기다림.
}
```

8. [출력7] H-Bridge를 이용한 DC 모터 방향 제어하기

SN754410은 쉽고 편리한 IC로 DC 모터의 속도와 방향을 컨트롤할 수 있게 한다. 이번 예제를 통해 쉽고 간단하게 DC모터의 방향을 제어해보자. 또한, 본 예제와 가속도 센서를 이용해 세그웨이와 같은 밸런스 로봇도 제작할 수 있고, 모터를 이용한 여러 가지 애플리케이션을 제작할 수 있다.

준비물: 아두이노 우노, USB 케이블, 브레드보드, SN754410, DC 모터, 점퍼 케이블

배경지식:

SN754410을 이용하여 한 개의 DC모터를 컨트롤 하기 위해서는 1개의 PWM, 2개의 디지털 아웃풋이 요구된다. 1개의 PWM은 스피드를 컨트롤하기 위함이며, 2개의 디지털 아웃풋은 모터의 방향을 결정하기 위함이다. 다음의 SN754410 칩의 함수 테이블을 보면 A가 H(high)일 때, 아웃풋 Y는 HIGH이고, A가 L(low)일 때, 아웃풋은 L이다. 모터 컨트롤러를 왜 사용하는가 생각할 수 있다. 아두이노에서 나오는 아날로그 아웃풋을 이용하여 컨트롤할 수도 있고, 혹은 트랜지스터와 다이오드를 이용하여 컨트롤할 수도 있다. 하지만, 모터에서 사용하는 전류가 보통 100mA이상이므로 40mA를 출력하는 아두이노에 직접적으로 연결하는 것은 옳지 않으며 간단히 모터 컨트롤러 칩을 사용함으로써 두 개의 모터를 제어하는 편리함 때문에 이 번 예제에서는 트랜지스터와 다이오드를 이용한 모터컨트롤 대신 SN754410을 이용한 모터 컨트롤을 하기로 했다. SN754410의 핀 정보를 보면 총 4개의 인풋이 있다 - 1A, 2A, 3A, 4A. 이 4개의 인풋을 이용하여 두 개의 모터 방향을 결정할 수 있다. 즉, 1A와 2A를 이용하여 1Y, 2Y의 값을 컨트롤하고, 3A와 4A를 이용하여 3Y, 4Y를 컨트롤한다. 1Y, 2Y는 첫번째 모터에, 그리고 3Y, 4Y는 두번째 모터에 각각 연결된다. 그리고, 1,2EN핀과 3,4EN핀에 PWM 값을 입력해주면서 전압에 따른 모터의 속도 컨트롤이 가능하다.





따라하기:

 1개의 DC모터를 컨트롤하기 위해서 다음의 회로를 구성해보자. SN754410을 브레드보드에 꼽는다. 아래 회로에서 왼쪽에 있는 작은 홈이 있는 방향이 1번핀이 시작되는 방향이다. 모터에서 나오는 두 개의 신호선은 + 혹은 -에 대한 정의가 없으니, 임의로 결정을 한 후에 브레드보드에 끼워도 무방하며 방향은 컨트롤러에서 결정할 수 있다. 혹은 회전하는 방향에 따라 모터에서 나오는 갈색과 보라색의 선을 바꾸어 끼워도 된다.



• 회로를 구성하였으면 다음의 프로그램을 컴파일 후 업로드 해보자.

```
int speedPin = 3; // SN754410의 1,2EN 핀과 PWM핀에 연결하였다. 속도제어용 핀
int motor1APin = 6; // 1A
int motor2APin = 7; // 2A
int speed_value_motor1; // 모터 스피드를 위한 변수
void setup()
{
  pinMode(speedPin, OUTPUT);
  pinMode(motor1APin, OUTPUT);
  pinMode(motor2APin, OUTPUT);
}
```

{

}

digitalWrite(motor1APin, LOW); // 1A에 LOW digitalWrite(motor2APin, HIGH); // 2A에 HIGH speed_value_motor1 = 127; // 0~255의 수 중 선택가능함. 0: 속도 제로, 255: 최고 속도. analogWrite(speedPin, speed_value_motor1); // PWM 출력을 하여 정한 속도만큼 모터 회전

- 위의 코드에서 loop() 내에 있는 명령문 중, digitalWrite(motor1APin, LOW)와 digitalWrite(motor2APin, HIGH)를 다음과 바꾸어보자. ditialWrite(motor1APin, HIGH), digitalWrite(motor2APin, LOW). 모터가 반대 방향으로 도는 것을 확인할 수 있다.
- 또한, speed_value_motor1 = 127의 값을 255로도 바꾸어보자. 모터가 더 빨리 회전하는 것을 확인할 수 있을 것이다.

9. [입력1] 키보드로 문자 입력 받아 아두이노에 전송하기

키보드를 이용하여 로봇을 제어하거나, MIDI와 같은 장비와의 인터페이스를 구현하기 위해, 아두이노의 Serial.read()함수를 이용하여 컴퓨터와 아두이노의 통신을 구현할 수 있다. 키보드로부터 키를 입력받아, 아두이노의 시리얼모니터링 창에 입력하면, 그 값을 아두이노에 전송되고, 아두이노는 다시 그 값을 컴퓨터에 보냄으로 확인이 가능하다.

준비물: 아두이노 우노, USB 케이블

따라하기:

 USB 케이블을 이용하여 아두이노와 컴퓨터를 연결한다. 보드와 시리얼 포트 설정을 한 후 다음의 프로그램을 업로드해 본다.

```
byte KeyInput;
void setup()
{
   Serial.begin(9600);
}
void loop()
{
   if(Serial.available() > 0)
   {
     KeyInput = Serial.read();
     Serial.write(KeyInput);
    }
}
```

• 업로드 후, 시리얼 모니터를 열고, 키를 입력한 후 엔터를 쳐서 아두이노에 전송해보자.

뻔뻔한 아두이노 매뉴얼 Ver.1 - 메카솔루션 (www.mechasolution.com)

© COM10	
hello	Send
V Autoscroll	lo line ending 🖌 [9600 baud 🔶

엔터를 치면, 아두이노로부터 값을 출력받는다.

© COM10	
1	Send
hello	
Z Autoscroll	No line ending 👻 9600 baud 👻

10.[입력2] 스위치 사용하여 LED 컨트롤



알고리즘으로 LED를 컨트롤할 수도 있지만, 다양한 스위치를 이용하여 LED를 컨트롤할 수도 있다. 스위치는 버튼 스위치, 토글 스위치 등 여러 가지가 있지만, 기본 원리는 회로를 연결 혹은 끊는 것이다.

준비물: 아두이노 우노, USB 케이블, 마이크로 스위치, 10k 저항, 브레드보드, 점퍼 케이블 배경지식:



위의 회로를 보면, 스위치가 열려 있을 때, 아두이노의 인풋(PINx)는 HIGH 인풋을 받게 된다. 하지만, 스위치가 닫히게 되면 PINx는 GND에 연결이 되어 LOW 인풋을 받게 된다.

따라하기:

- 스위치를 브레드보드에 연결 혹은 필요에 따라 납땜을 한다. 그리고 스위치의 한 쪽에 풀업 저항을 이용하여 5V에 연결한다. 반대쪽의 스위치 핀은 GND에 연결한다. 아두이노의 디지털핀을 스위치와 저항이 만나는 점(노드)에 연결한다. 이 디지털핀은 digitalRead()라는 함수를 이용하여 값을 읽을 것이다.
- 아두이노 디지털 핀 13번에 연결되어 있는 내장형 LED를 사용할 것이나 필요에 따라서 앞서 설명된 LED 예제들을 참고할 수 있다.



```
int button1Pin = 2; // 버튼 스위치의 2번 핀
int ledPin = 13; // 아두이노에 내장된 LED
void setup()
{
    pinMode(button1Pin, INPUT); // 버튼을 인풋으로 사용할 것을 설정
    pinMode(ledPin, OUTPUT); // LED는 아웃풋으로 사용.
}
void loop()
{
    int button1State; // 버튼의 입력값을 저장할 변수
    button1State = digitalRead(button1Pin); // 버튼으로부터 받은 0 혹은 1의 값을 저장
    if (button1State == LOW) // 버튼이 LOW(0)이면
    {
```

digitalWrite(ledPin, HIGH); // LED를 켜고 } Else // 그렇지 않으면 { digitalWrite(ledPin, LOW); // LED를 끈다. }

<mark>11. [입력3] 근접 물체 감지하기</mark>

근접 물체를 감지하기 위해서는 여러 가지 센서가 사용될 수 있지만, 저렴하면서 작은 QRD1114를 이용하여 근접 센싱을 할 수 있다. 이 QRD1114는 적외선 LED와 포토 트렌지스터가 작은 모듈에 구현이 되어 있는 것으로 적외선 LED로부터 발광된 빛이 사물에 반사되어 돌아오는 빛의 세기를 포토 트렌지스터로 측정함으로 센싱을 할 수 있도록 설계되어 있다.

준비물: 아두이노 우노, USB 케이블, QRD1114, 10k옴 저항, 220옴 저항, 브레드보드, 점퍼 케이블 배경지식:



1번과 2번으로 구성되어 있는 것은 포토 트렌지스터이고, 3번과 4번으로 구성되어 있는 것은 적외선 LED이다. 3번에서 입력 받은 전원으로 적외선 LED는 발광하게 되고, 사물에 반사된 적외선의 세기를 포토 트렌지스터로 센싱하게 된다. 이 때, 아두이노의 디지털인풋은 풀업 저항에 연결되어 있기 때문에 사물이 없을 경우는 센싱값이 HIGH(1)을 갖게 되고, 물체가 감지되었을 경우에는 값이 LOW(0)을 갖게 된다. 시리얼 모니터를 이용하여 확인할 수 있다.

따라하기:

 QRD1114의 4개의 다리를 브레드보드에 꼽고 저항을 연결하여 다음과 같이 회로를 구성한다.



• 다음의 프로그램을 업로드한다.

int signal = 4; // 디지털 핀4번으로 센서값(0 혹은 1)을 읽을 변수. int onoff; // 4번 핀에서 읽은 값이 High(1)인지 Low(0)인지 저장할 변수 void setup() { Serial.begin(9600); // 시리얼통신을 사용하여 모니터링 할 수 있다. } void loop() { onoff = digitalRead(signal); // 불리언 신호(0 혹은 1)을 읽어서 onoff에 저장

```
Serial.println(onoff); // onoff변수값을 Serial.println을 통해 출력.
```

업로드 후에 시리얼 모니터를 이용하여 센싱되는 값을 시리얼 모니터를 통해 확인해 볼
 수 있다. 시리얼 모니터는 스케치의 오른쪽 상단에 있는 아이콘을 클릭하여 볼 수 있다.



 시리얼 모니터링 창이 열리면 센서를 손으로 막았다 떼었다하면서 값이 변하는 것을 체크해 볼 수 있다.

12. <mark>[입력4] 온도 센서로 온도 감지하기</mark>

TMP36온도 센서는 -40도에서 125도까지 온도를 측정할 수 있는 센서로 약±2도 정도의 정확도를 가지고 있다. 세 개의 다리가 있는데 평평한 곳을 정면으로 가정할 때, 왼쪽 다리는 입력 전원, 오른쪽은 GND, 그리고 가운데 다리를 이용하여 온도를 측정할 수 있다.



비교적 간단한 회로이지만, 핀 방향을 잘 숙지하고 회로를 구성하도록 하자.

준비물: 아두이노 우노, USB 케이블, TMP36 온도 센서, 브레드보드, 점퍼 케이블

배경지식:

일반적으로 센서를 사용하여 실제적인 물리량(ex. 온도)을 측정하기 위해서는 변환공식이 필요하다. TMP36에 인가해주는 전원은 5V이다. 그리고 아두이노가 센서로부터 읽는 방법은 analogRead()라는 함수를 이용한다. 이 anlogRead()함수는 센서로부터 나오는 아날로그 전압값(0~5V)를 디지털값으로 변환하여 읽게 되는데 그 범위는 10bit, 즉 0에서 1023 사이의 값이다. 따라서 5V는 1023에 매칭이 되고, 5V/1024*analogRead를 이용하여 전원 값을 읽을 수 있다. 그리고, 데이터시트에서 기재된 센서의 특징에서 전압과 온도의 관계를 생각할 수 있다.
Sensor	Offset Voltage (V)	Output Voltage Scaling (mV/°C)	Output Voltage @ 25°C (mV)
TMP35	0	10	250
TMP36	0.5	10	750
TMP37	0	20	500

Table 4. TMP3x Output Characteristics

위의 표에서 사용된 TMP36을 보면 Offset Voltage(V)는 0.5이고, Output Voltage Scaling은 0.01V/C이다. 즉, 1도에 0.01V씩 증가한다는 의미이다.

V(전압)과 C(온도)와의 관계를 생각해보면, V = 0.01*C + 0.5가 되고, 이를 온도에 대하여 다시 풀면, C = 100*(V - 0.5)가 된다.

따라하기:

 TMP36의 가운데 신호 다리를 아날로그0번에 연결하고 전원과 GND를 아두이노에 다음과 같이 연결한다.



• 다음의 프로그램을 업로드하고 시리얼모니터를 통해 온도의 변화를 측정해보자.

```
int temperaturePin = 0;
float V;
float C;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  V = analogRead(temperaturePin) * 5/1024;
  C = 100*(V - 0.5);
  Serial.print("voltage: ");
  Serial.print(V);
  Serial.print(" deg: ");
  Serial.print(C);
  delay(1000);
}
```

13.[입력5] MPU-6050을 이용하여 각속도/가속도/온도 측정하기

MPU-6050은 9축 모션퓨전 알고리즘을 지원하는 가속도, 자이로 센서보드로서 디지털로 출력되는 온도센서 기능도 가지고 있다. I2C라는 통신 프로토콜을 이용하여 다수의 센서 데이터들을 소수의 핀을 이용하여 수집할 수 있도록 한다.



MPU-6050

지금까지 다루었던 센서들은 아날로그핀을 이용하여 값을 읽었다. 이번에 다룰 MPU-6050은 아날로그핀이 아닌 두 개의 디지털핀을 이용하여 3축 각속도, 3축 가속도, 그리고 온도를 측정할 수 있다.

준비물: 아두이노 우노, USB 케이블, MPU-6050, 브레드보드, 점퍼 케이블

배경지식:

I2C(Inter-Integrated Circuit)는 저속의 주변 기기간 SCL(serial clock)과 SDA(serial data) 2개의 커넥션을 가지고 있다. 표준 아두이노 보드는 클럭 시그널을 제공하는 SCL을 위해 아날로그 핀 5을 사용하고 데이터 전송을 위한 SDA을 위해서는 아날로그 핀 4을 사용한다. (Mega는 SDA는 디지털 핀 20 그리고 SCL은 핀 21을 사용한다.) 따라서, MPU-6050의 SCL은 아두이노의 A5, SDA는 아두이노의 A4를 사용한다.

따라하기:

• 아두이노의 3.3V, GND, A4, A5를 MPU-6050의 VCC, GND, SDA, SCL에 각각 연결한다.



• 다음의 프로그램을 업로드하고 시리얼모니터링을 통해 값을 확인한다.

#include <wire.h></wire.h>		
#define MPU6050_ACCEL_XOUT_H	0x3B	// R
#define MPU6050_PWR_MGMT_1	0x6B	// R/W
#define MPU6050_PWR_MGMT_2	0x6C	// R/W
#define MPU6050_WHO_AM_I	0x75	// R
#define MPU6050_I2C_ADDRESS 0x68		
typedef union accel_t_gyro_union		
{		
struct		
{		
uint8_t x_accel_h;		
uint8_t x_accel_l;		
uint8_t y_accel_h;		
uint8_t y_accel_l;		
uint8_t z_accel_h;		
uint8_t z_accel_l;		
uint8_t t_h;		
uint8_t t_l;		
uint8_t x_gyro_h;		

```
uint8_t x_gyro_l;
    uint8_t y_gyro_h;
    uint8_t y_gyro_l;
    uint8_t z_gyro_h;
    uint8_t z_gyro_l;
  } reg;
  struct
  {
    int x_accel;
    int y_accel;
    int z_accel;
    int temperature;
    int x_gyro;
    int y_gyro;
    int z_gyro;
  } value;
};
void setup()
{
  int error;
  uint8_t c;
  Serial.begin(9600);
  Wire.begin();
  // default at power-up:
        Gyro at 250 degrees second
  //
  //
        Acceleration at 2g
        Clock source at internal 8MHz
  //
  //
        The device is in sleep mode.
  //
  error = MPU6050_read (MPU6050_WHO_AM_I, &c, 1);
  Serial.print(F("WHO_AM_I : "));
  Serial.print(c,HEX);
  Serial.print(F(", error = "));
```

```
Serial.println(error,DEC);
  // According to the datasheet, the 'sleep' bit
  // should read a '1'. But I read a '0'.
  // That bit has to be cleared, since the sensor
  // is in sleep mode at power-up. Even if the
  // bit reads '0'.
  error = MPU6050_read (MPU6050_PWR_MGMT_2, &c, 1);
  Serial.print(F("PWR_MGMT_2 : "));
  Serial.print(c,HEX);
  Serial.print(F(", error = "));
  Serial.println(error,DEC);
  // Clear the 'sleep' bit to start the sensor.
  MPU6050_write_reg (MPU6050_PWR_MGMT_1, 0);
}
void loop()
{
  int error;
  double dT;
  accel_t_gyro_union accel_t_gyro;
  Serial.println(F(""));
  Serial.println(F("MPU-6050"));
 // Read the raw values.
 // Read 14 bytes at once,
 // containing acceleration, temperature and gyro.
 // With the default settings of the MPU-6050,
 // there is no filter enabled, and the values
 // are not very stable.
  error
           =
                MPU6050_read
                                    (MPU6050_ACCEL_XOUT_H,
                                                                   (uint8_t
                                                                             *)
                                                                                     &accel_t_gyro,
sizeof(accel_t_gyro));
  Serial.print(F("Read accel, temp and gyro, error = "));
```

Serial.println(error,DEC);

// Swap all high and low bytes. // After this, the registers values are swapped, // so the structure name like x_accel_l does no // longer contain the lower byte. uint8_t swap; #define SWAP(x,y) swap = x; x = y; y = swap

SWAP (accel_t_gyro.reg.x_accel_h, accel_t_gyro.reg.x_accel_l); SWAP (accel_t_gyro.reg.y_accel_h, accel_t_gyro.reg.y_accel_l); SWAP (accel_t_gyro.reg.z_accel_h, accel_t_gyro.reg.z_accel_l); SWAP (accel_t_gyro.reg.t_h, accel_t_gyro.reg.t_l); SWAP (accel_t_gyro.reg.x_gyro_h, accel_t_gyro.reg.x_gyro_l); SWAP (accel_t_gyro.reg.y_gyro_h, accel_t_gyro.reg.y_gyro_l); SWAP (accel_t_gyro.reg.z_gyro_h, accel_t_gyro.reg.z_gyro_l);

// Print the raw acceleration values

Serial.print(F("accel x,y,z: ")); Serial.print(accel_t_gyro.value.x_accel, DEC); Serial.print(F(", ")); Serial.print(accel_t_gyro.value.y_accel, DEC); Serial.print(F(", ")); Serial.print(accel_t_gyro.value.z_accel, DEC); Serial.println(F(""));

// The temperature sensor is -40 to +85 degrees Celsius.

// It is a signed integer.

// According to the datasheet:

// 340 per degrees Celsius, -512 at 35 degrees.

// At 0 degrees: -512 - (340 * 35) = -12412

Serial.print(F("temperature: "));

dT = ((double) accel_t_gyro.value.temperature + 12412.0) / 340.0;

Serial.print(dT, 3);

```
Serial.print(F(" degrees Celsius"));
  Serial.println(F(""));
  // Print the raw gyro values.
  Serial.print(F("gyro x,y,z : "));
  Serial.print(accel_t_gyro.value.x_gyro, DEC);
  Serial.print(F(", "));
  Serial.print(accel_t_gyro.value.y_gyro, DEC);
  Serial.print(F(", "));
  Serial.print(accel_t_gyro.value.z_gyro, DEC);
  Serial.print(F(", "));
  Serial.println(F(""));
  delay(500);
}
int MPU6050_read(int start, uint8_t *buffer, int size)
{
  int i, n, error;
  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  n = Wire.write(start);
  if (n != 1)
    return (-10);
                                      // hold the I2C-bus
  n = Wire.endTransmission(false);
  if (n != 0)
    return (n);
  // Third parameter is true: relase I2C-bus after data is read.
  Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);
  i = 0;
  while(Wire.available() && i<size)
  {
    buffer[i++]=Wire.read();
  }
```

```
if ( i != size)
    return (-11);
  return (0); // return : no error
}
int MPU6050_write(int start, const uint8_t *pData, int size)
{
  int n, error;
  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  n = Wire.write(start);
                         // write the start address
  if (n != 1)
    return (-20);
  n = Wire.write(pData, size); // write data bytes
  if (n != size)
    return (-21);
  error = Wire.endTransmission(true); // release the I2C-bus
  if (error != 0)
    return (error);
  return (0);
              // return : no error
}
int MPU6050_write_reg(int reg, uint8_t data)
{
  int error;
  error = MPU6050_write(reg, &data, 1);
  return (error);
}
```

 아래 보이는 시리얼모니터링의 창처럼, error = 0이고, 3축의 가속도, 온도, 그리고 3축의 자이로 값이 출력되는 것을 확인할 수 있다.

≦ COM10		x
	Sen	d
gyro x,y,z : −126, −27, 85,		^
MPU-6050		
Read accel, temp and gyro, error = 0		
accel x,y,z: -1352, 392, 16072		
temperature: 28,082 degrees Celsius		
gyru x,y,2 · -105, 5, 66,		
MPU-6050		
Read accel, temp and gyro, error = O		
accel x,y,z: -1360, 136, 16024		
temperature: 28.082 degrees Celsius		
gyro x,y,z : -125, -16, 63,		≡
HOLL COTO		
		-
V Autoscroll	No line ending 👻 9600 baud	•

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

14.[입력6] Flex 센서를 이용한 구부림 측정하기

웨어러블 수트가 등장하면서 각광받은 Flex 휨센서는 구부러진 정도를 측정할 수 있는 센서이다. FSR 압력센서와 마찬가지로 저항의 일종인 Flex 휨센서는 구부러진 정도에 따라 저항값이 바뀌는 특징을 이용한다.

준비물: 아두이노 우노, USB 케이블, 10k옴 저항, Flex 센서, 브레드보드, 점퍼 케이블

배경지식:

Flex센서는 다음과 같이 구부러진 정도가 커지면 저항값이 커진다.

45° Bend (increased resistance)

90° Bend (resistance increased further)



따라하기:

• Flex 센서와 저항을 이용하여 다음과 같은 회로를 구성한다.



다음의 프로그램을 업로드한 후 시리얼모니터링을 통하여 값의 변화를 체크해보자.

```
int flexpin = 0; // 센서값을 읽기 위해 아날로그핀 0번을 flexpin에 지정한다.
void setup()
{
  Serial.begin(9600); // 센서값을 읽기 위해 시리얼 모니터를 사용할 것을 설정.
}
void loop()
{
  int flexVal; // 센서값을 저장할 변수
  flexVal = analogRead(flexpin); // 아날로그를 입력 받음 (0~1023)
  Serial.print("sensor: "); // sensor: 라는 텍스트를 프린트한다.
  Serial.println(flexVal); // println은 줄바꿈 명령이다. flexVal의 값을 출력한다.
```

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

• 시리얼 모니터링에서 데이터가 빠르게 진행되서 값을 체크하기 어렵다면 delay()함수를 사용해볼 수 있다. Serial.println(flexVal); 다음 줄에 delay()를 사용해보자. (ex. delay(200);)

15.[입력7] FSR 압력 센서를 이용한 압력 측정하기

FSR(Force Sensing Resistor)는 얇은 필름 방식으로 제작된 압력센서로서 간단한 접촉의 압력 측정이 가능하다. 굽힘 센서와 마찬가지로 저항의 일종이며 10k옴 저항을 사용하여 값을 측정할 수 있다. 정밀한 압력 및 힘의 측정에는 로드셀을 사용할 수 있으나, 비교적 저렴하면서 쉽게 구현이 가능하기 때문에 널리 사용되고 있다.

준비물: 아두이노 우노, USB 케이블, 10k옴 저항, FSR, 브레드보드, 점퍼 케이블

배경지식:

데이터시트에서 제공된 실험 결과를 보면 저항값과 힘의 관계가 그래프에 표현되어 있다.

$$\frac{V_{O}}{V_{I}} = \frac{R_{FSR}}{R_{1} + R_{FSR}}$$

여기서 V_I=5V, 즉 1023이고, R₁=10000옴이라고 하면, V₀와 R_{FSR}의 관계식은 다음과 같이 된다.

$$R_{FSR} = \frac{9.78 * V_0}{1 - (\frac{V_0}{1023})}$$

Vo 는 아두이노의 analogRead()함수를 이용하여 센서값을 얻고, 위 공식을 바탕으로 FSR의 저항값을 측정할 수 있다. 그리고, 측정된 저항값(R_{FSR})을 아래 저항과 힘의 관계 그래프를 통하여 가해진 힘을 확인할 수 있다.



따라하기:

• FSR과 저항을 이용하여 다음과 같은 회로를 구성한다.



• 다음의 프로그램을 업로드한 후 시리얼모니터링을 통하여 값의 변화를 체크해보자.

```
int FSRpin = 0; // FSRpin을 아날로그0(A0)에 연결

int Vo; // 센서값을 저장할 변수

float Rfsr;

void setup() {
		Serial.begin(9600); // 센서값을 읽기 위해 시리얼 모니터를 사용할 것을 설정.
	}
	void loop()
{
		Vo = analogRead(FSRpin); // 아날로그를 입력 받음 (0~1023)
		Rfsr = ((9.78 * Vo)/(1-(Vo/1023.0)));
		Serial.print("Rfsr: "); // sensor: 라는 텍스트를 프린트한다.
		Serial.print(Rfsr); // println은 줄바꿈 명령이다. Rfsr의 값을 출력한다.
	}
```

• 시리얼 모니터링에서 데이터가 빠르게 진행되서 값을 체크하기 어렵다면 delay()함수를 사용해볼 수 있다. Serial.println(Rfsr); 다음 줄에 delay()를 사용해보자. (ex. delay(200);)

16.[응용1] 밝기에 따른 DC 모터 속도 제어

조도센서는 CdS라는 광전도셀을 이용하여 가시광선이 없는 곳에서는 절연체와 같이 전류가 흐르지 않다가 가시광선이 닿으면 도체와 같이 전류가 잘 흐르는 성질을 가지고 있다. 빛을 감지하여 회로를 ON/OFF하거나 센서 및 방범용 모듈 등에 많이 사용된다. 이번 예제에서는 조도센서를 이용하여 빛의 세기를 측정하고, 그 세기에 따라 DC모터의 속도를 제어하는 프로그램을 실행해 보자.

준비물: 아두이노 우노, USB 케이블, 10k옴 저항, 조도센서, SN754410, DC 모터, 브레드보드, 점퍼 케이블

따라하기:

Flex 센서, FSR과 같이 조도센서도 두 개의 다리를 가지고 있는 일종의 저항이다. 여기에 앞서 다루었던 DC 모터 방향제어하기 예제를 응용하여 다음의 회로를 구성한다.



```
• 다음의 프로그램을 업로드한다.
```

```
int sensorPin = 0;
                  // 조도센서가 연결될 A0핀
int speedPin = 3; // SN754410의 1,2EN 핀과 PWM핀에 연결하였다. 속도제어용 핀
int motor1APin = 6; // 1A
int motor2APin = 7;
                  // 2A
int speed_value_motor1; // 모터 스피드를 위한 변수
             // 조도센서로부터 읽은 값을 저장할 변수
int lightLevel;
void setup()
{
 pinMode(speedPin, OUTPUT);
 pinMode(motor1APin, OUTPUT);
 pinMode(motor2APin, OUTPUT);
}
void loop()
{
   lightLevel = analogRead(sensorPin); // 센서로부터 읽은 값을 저장(0~1023)
   digitalWrite(motor1APin, LOW); // 1A에 LOW
   digitalWrite(motor2APin, HIGH); // 2A에 HIGH
   lightLevel = map(lightLevel, 0, 1023, 0, 255); // PWM 출력 레벨(8bit)로 센서값을 맵핑
   analogWrite(speedPin, lightLevel); // 밝기에 따라 모터 회전 속도 출력
```

17.[응용2] 장애물 감지 후 모터의 방향 제어

앞에서 소개된 QRD1114는 1cm 내에 있는 사물을 감지하는데 유용하게 쓰이는 센서이다. 이 센서를 이용하여 소형 로봇의 장애물 감지를 할 수 있으며 모터의 방향도 제어할 수 있다. 보다 먼 거리의 사물을 감지하기 위해서는 적외선 센서 혹은 초음파 센서를 이용할 수 있다.

준비물: 아두이노 우노, USB 케이블, QRD1114, 10k옴 저항, 220옴 저항, SN754410, DC 모터, 브레드보드, 점퍼 케이블

따라하기:

- *** $\bullet \bullet \bullet$ * * * * * * Speed < Direction Ŧ d g (UNO)(+)메카솔루션
- 앞서 다루었던 QRD1114예제와 DC모터 제어 예제를 이용하여 다음의 회로를 구성한다.

 QRD1114가 사물을 감지했을 때, 모터의 방향을 바꾸는 예제를 if 명령어를 이용하여 다음과 같이 프로그램을 작성할 수 있다. 프로그램을 업로드 한 후, QRD1114센서를 손가락으로 막았다 떼었다하면서 모터의 회전 방향을 체크할 수 있다.

```
int speedPin = 3; // SN754410의 1,2EN 핀과 PWM핀에 연결하였다. 속도제어용 핀
int signal = 4; // 디지털 핀4번으로 센서값(0 혹은 1)을 읽을 변수.
           // 4번 핀에서 읽은 값이 High(1)인지 Low(0)인지 저장할 변수
int onoff;
int motor1APin = 6;
                   // 1A
int motor2APin = 7;
                    // 2A
int speed_value_motor1; // 모터 스피드를 위한 변수
void setup()
{
pinMode(speedPin, OUTPUT);
 pinMode(motor1APin, OUTPUT);
 pinMode(motor2APin, OUTPUT);
}
void loop()
{
   onoff = digitalRead(signal); // 불리언 신호(0 혹은 1)을 읽어서 onoff에 저장
   if(onoff == 1)
   {
   digitalWrite(motor1APin, LOW); // 1A에 LOW
   digitalWrite(motor2APin, HIGH); // 2A에 HIGH
   }
   else
   {
   digitalWrite(motor1APin, HIGH); // 1A에 LOW
   digitalWrite(motor2APin, LOW); // 2A에 HIGH
   }
   speed_value_motor1 = 127; // 0~255의 수 중 선택가능함. 0: 속도 제로, 255: 최고 속도.
   analogWrite(speedPin, speed_value_motor1); // PWM 출력을 하여 정한 속도만큼 모터 회전
```

18. [응용3] 손가락으로 서보 모터 제어하기

손가락에 Flex 센서를 연결하여 굽힌 정도에 따라 서보 모터를 제어할 수 있다. 앞에서 다룬 Flex 센서 예제와 서보 모터 예제를 통하여 손가락으로 서보 모터 제어를 구현해보자.



준비물: 아두이노 우노, USB 케이블, Flex 센서, 10k옴 저항, 서보 모터, 브레드보드, 점퍼 케이블 따라하기:

 앞서 다루었던 QRD1114예제와 서보 모터 구동 예제를 이용하여 다음의 회로를 구성한다.



다음의 프로그램을 업로드하여 손가락의 굽힌 정도에 따라 서보 모터가 회전하는지 체크해보자. 매 0.2초마다 값을 읽기 때문에 모터의 속도가 느리다면 모터가 회전하는 중간에 센서값을 읽을 수도 있다. 테스트를 하면서 delay()함수를 변경하여 부드러운 컨트롤을 할 수 있다.

#include <Servo.h> // 서보 라이브러리 Servo motor1; // 서보 모터 오브젝트 int flexpin = 0; // 센서값을 읽기 위해 아날로그핀 0번을 flexpin에 지정한다.

```
int motorangle;
void setup()
{
motor1.attach(9); // 서보 모터를 9번에 연결
}
void loop()
{
int flexVal; // 센서값을 저장할 변수
flexVal = analogRead(flexpin); // 아날로그를 입력 받음 (0~1023)
motorangle = map(flexVal, 0, 1023, 0, 180);
motor1.write(motorangle); // 모터를 position만큼 회전
delay(200); // 모터가 돌 때까지 기다림.
}
```

19.[응용4] 압력에 따른 LED의 밝기 제어

FSR(압력센서: Force Sensing Resistor)을 이용하여 쉽게 LED의 밝기를 조절할 수 있다. 앞서 PWM을 이용한 LED 밝기 조절 예제와 FSR을 이용한 압력 측정 예제를 통하여 이를 구현할 수 있다.

준비물: 아두이노 우노, USB 케이블, FSR 센서, 10k옴 저항, LED, 브레드보드, 점퍼 케이블

따라하기:

 FSR과 저항(10k옴)을 이용하여 압력을 측정하는 예제를 구현하고, LED를 PWM핀에 연결한다. 다음의 회로를 참고한다.



뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

• 압력에 따른 LED 밝기 제어 프로그램을 업로드한다.

```
int led = 3;
int brightness = 0;
int increment = 1;
int FSRpin = 0; // FSRpin을 아날로그0(A0)에 연결
int Vo; // 센서값을 저장할 변수
float Rfsr;
void setup()
{
   // analogWrite을 통해 구동되는 핀은 출력으로 선언하지 않아도 된다.
}
void loop()
{
   Vo = analogRead(FSRpin); // 아날로그를 입력 받음 (0~1023)
   brightness = map(Vo, 0, 1023, 0, 255);
   analogWrite(led, brightness); // 밝기를 출력한다
   delay(10);
}
```

20.[응용5] 키보드로 모터 방향 제어하기

키보드로부터 입력받은 값을 이용하여 DC 모터의 회전 방향을 제어할 수 있다. 이번 예제에서는 A를 입력받으면 정방향 회전, B를 입력받으면 역방향 회전, 그리고, C를 입력받으면 정지하는 시스템을 구현하였다.

준비물: 아두이노 우노, USB 케이블, SN754410, DC 모터, 브레드보드, 점퍼 케이블

따라하기:

- Image: Structure
 Image: Structure

 Image: Structure
 Image: Structure
- 다음과 같이 회로를 구성하고, 아래의 프로그램을 업로드한다.

int speedPin = 3;	// H-bridge enable pin for speed control	
int motor1APin = 6;	// H-bridge leg 1	
int motor2APin = 7;	// H-bridge leg 2	
int ledPin = 13;	// status LED	
int speed_value_motor1; // value for motor speed		

```
byte KeyInput;
void setup() {
  // set digital i/o pins as outputs:
  pinMode(speedPin, OUTPUT);
  pinMode(motor1APin, OUTPUT);
  pinMode(motor2APin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(ledPin, HIGH); // status LED is always on
  KeyInput = Serial.read();
  if(KeyInput == 'A')
  {
     // put motor in forward motion
    digitalWrite(motor1APin, HIGH); // set leg 1 of the H-bridge low
    digitalWrite(motor2APin, LOW); // set leg 2 of the H-bridge high
  }
 else if(KeyInput == 'B')
  {
    digitalWrite(motor1APin, LOW); // set leg 1 of the H-bridge low
    digitalWrite(motor2APin, HIGH); // set leg 2 of the H-bridge high
  }
   else if(KeyInput == 'C')
  {
    digitalWrite(motor1APin, LOW); // set leg 1 of the H-bridge low
    digitalWrite(motor2APin, LOW); // set leg 2 of the H-bridge high
 }
    speed_value_motor1 = 127; // half speed
    analogWrite(speedPin, speed_value_motor1); // output speed as
```

 시리얼모니터 창을 열어서 A, B, 혹은 C를 타이핑한 후 엔터 입력을 하여 모터의 회전방향을 체크할 수 있다.



21.[응용6] Visual Studio Express C#와 아두이노 인터페이스

Microsoft의 Visual Studio Express C#은 무료 개발툴로서 Microsoft홈페이지에서 다운로드가 가능하다. C#이란, C++의 연산력(computing power)과 Visual Basic의 편의성(convenience)을 결합하기 위한 목적으로 마이크로소프트사에서 개발한 객체지향 프로그래밍 언어이다. C++가 기반이며, 자바랑 비슷한 특징을 지니고 있다.

준비물: 아두이노 우노, USB 케이블, 10k옴 저항, FSR센서

따라하기:

• 다음의 링크에서 Visual C# 2010 Express를 다운로드 받는다.

http://www.microsoft.com/visualstudio/kor/downloads#d-express-windows-desktop

● 지금 설치 버튼을 누르면, vcs_web.exe 파일이 다운로드 되고, 이를 더블 클릭하여 실행한다.

제icrosoft Visual C# 2010 Express 설치	
설치 시작	Visual C# 2010 Express
Microsoft Visual C# 2010 Express 설치 마법사를 2010 Express은(는) C# 프로그래머가 Windows F Foundation), 클래스 라이브러리 및 관습기반 응용 된 배우기 쉽고, 흥미로운 간단한 개발 도구입니다 다. 제품에 필요한 구성 요소가 컴퓨터에 설치되어 도 설치할 수 있습니다.	시작합니다, Microsoft Visual C# orms, WPF(Windows Presentation 8. 프로그램을 만들 수 있도록 디자인 이 마법사는 설치 과정을 안내합니 있지 않은 경우 해당 필수 구성 요소
설치 환경 개선 프로그램 Microsoft에 Visual Studio 설치 경험에 대한 정보 아래 상자를 선택하십시오.	를 보내실 수 있습니다. 참여하시려면
☑ 예, Microsoft에 설치 경험에 대한 정보를 보냅니 ① 자세한 내용은 <u>개인 정보 취급 방침</u> 을 참조하	니다(<u>8</u>). [십시오.
< 0	I전(P) 다음(N) > 취소

• 다음을 클릭

뻔뻔한 아두이노 매뉴얼 Ver.1 - 메카솔루션 (www.mechasolution.com)

교 Microsoft Visual C# 2010 Express 설치	
사용 약관 📀	Microsoft* Visual C#* 2010 Express
반드시 사용 약관에 설명된 모든 권리와 제한 사항을 자세히 웨어를 설치하려면 사용 약관에 동의해야 합니다.	읽고 이해하십시오, 소프트
MICROSOFT 소프트웨어 사용권 계약서 MICROSOFT VISUAL C# 2010 EXPRESS	^
본 사용권 계약은 Microsoft Corporation(또는 거주 지역에 귀하 간에 체결되는 계약입니다. 읽어 주시기 바랍니다. 본 된 소프트웨어 및 이 소프트웨어가 포함된 미디어가 있는 경	따라 계열사 중 하나)과 사용권 계약은 위에 명시 응우 해당 미디어에 적용 🖵
자세한 내용을 보려면 <page down=""> 키를 누르십시오.</page>	인쇄(<u>B</u>)
 ● 동의함(A) ○ 동의 안 함(0) 	
<u>< 미젼(P)</u>	다음(<u>N</u>) > 취소

• 다음을 클릭

🐼 Microsoft Visual C# 2010 Express 설치		
설치 옵션	Visual C# 2010 Express	
설치할 선택적 제품 선택:		
✓ Microsoft Silverlight(다운로드 크기: 4.7 Microsoft Silverlight는 웹 환경을 개선할 수 있 Silverlight를 설치하는 것은 <u>Silverlight 사용권</u>	MB) 는 작은 브라우저 플러그 인입니다. <u>셰약</u> 에 동의함을 의미합니다.	
■ Microsoft SQL Server 2008 Express 서비스 팩 1(x64)(다운로드 크기: 113 MB) SQL Server Express는 Visual Studio와 통합하여 기본 클라이언트 데이터베이스 및 서버 데이터베이스 기능을 제공합니다.		
 자세한 내용은 <u>추가 정보</u> 파일을 참조하십시오. 		
	전(P) [다음(N) > 취소	

• 다음을 클릭

뻔뻔한 아두이노 매뉴얼 Ver.1 - 메카솔루션 (www.mechasolution.com)

교 Microsoft Visual C# 2010 Express 설치	
대상 폴더	Visual C#* 2010 Express
Microsoft Visual C# 2010 Express을(를) 설치할 위치를 선택하	H십시오,
⊇⊼ໄ⊇ ອີດ(ປາ: C:₩Program Files (x86)₩Microsoft Visual Studio 10,0₩	황아보기(<u>B</u>)
다운로드 및 설치 항목:	
• Microsoft 응용 프로그램 오류 보고 • VC 10,0 Runtime (x64) • VC 9,0 Runtime (x86)	▲ E
• Microsoft, NET Framework 4 • Microsoft, NET Framework 4 한국어 언어 팩	
• Microsoft Visual Studio 2010 Express 필수 구경 묘소 xt 필요한 디스크 공간: C: 2,5 GB 전체 다운로드 크기: 168 MB	
<u> くの酒(P)</u> [4	치(<u>N)</u> > 취소

• 저장할 폴더를 설정한 후 설치 클릭

🗔 Microsoft Visual C# 2010 Express 설치	
다운로드 및 설치 진행률	isual C ^{#*} 2010 press
이 컴퓨터에 다운로드하여 설치 중인 항목:	
 ♥ ♥ ♥ Microsoft 응용 프로그램 오류 보고 ♥ VC 10.0 Runtime (x64) ♥ VC 9.0 Runtime (x86) ♥ VC 9.0 Runtime (x86) ♥ Microsoft .NET Framework 4 ♥ Microsoft .NET Framework 4 한국어 언어 팩 	
	취소

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)



● 설치 완료

클 시작 팩이지 - Microsoft Visual C# 2010 Express (관리자)	AND DESCRIPTION OF A DESCRIPTION		= 0 X
파일타 편집(6) 보기(V) 디버그(D) 도구(T) 창(W) 도움달	· · · · · · · · · · · · · · · · · · ·		
↗ 시작 페이지 ×		 술루선 탑색기 	- 4 ×
1999 (전 Visual C * 2010 Express) 4 프로젝트) 프로젝트 알기 최근에 사용한 프로젝트	A 전 A A A A A A A A A A A A A A A A A A		
) 프로젝트 로드 후 페이지 당기) 시작할 때 페이지 표시 			

● 설치된 Microsoft Visual C# Express 2010을 실행하면 위의 창이 열린다.

새 프로젝트			? <mark>×</mark>
최신 템플릿	정렬 기준: 기본값 ▼ 📖 🛄		설치된 템플릿 검색 🔎
<mark>설치된 템플릿</mark> Visual C#	Windows Forms 응용 프로그램	Visual C#	유형: Visual C# Windows Forms 사용자 인터페이스를 사
온라인 템플릿	WPF 응용 프로그램	Visual C#	용하여 응용 프로그램을 만드는 프로젝트 입니다.
	· 콘솔 응용 프로그램	Visual C#	
		Visual C#	
	[∞] C [#] WPF 브라우저 응용 프로그램	Visual C#	
	C♯ 빈 프로젝트	Visual C#	
이름(N): ArduinoInterface	3		
			확인 취소

 새 프로젝트를 클릭하여 Window Forms 응용프로그램을 클릭 후 이름을 ArduinoInterface라 하고 확인을 클릭한다.

에 ArduinoInterface - Microsoft Visual C# 2010 Express (관리구남)	
파울() 관립() 보기() 프로젝트() 디버그() 데이터(A) 도구() 왕() 도운왕(4)	
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	
·바이나 (100 ··································	
2 Form1cs(E(XPP)) ×	 술루선 탐색기 ▼ 쿠 ×
	(3. ST)
Formi State	중 문부선 'Arduinotnterface' (1 프로젝트)
	 ArduinoInterface
	Properties Qa #3 #
	Form1.cs
	Program.cs
Au	

 위와 같은 창이 열리며 왼쪽에 숨겨있는 도구 상자를 클릭한 후, 자동숨기기 버튼(압정모양의 버튼)을 클릭하여 도구 상자를 아래 그림과 같이 펼친다.

ArduinoInterface - Microsoft Visual C#	2010 Express (관리자)	
파일() 편집() 보기() 프로젝트()	디버그(D) 데이터(A) 도구(T) 참(W) 도응앏(H)	
1 1 · · · · · · · · · · · · · · · · · ·	- (C -) (G) - (Q =) (B =)	
10 IL & ALT & ALT & ALT &	网白小沙沙沙 21 4 4 4 4 1月1月1日 4 4 4 -	
日本十 3内 ManuStrip	・ + × Formit's (日本包)・ ×	 · · · · · · · · · · · · · · · · · · ·
StatusStrip		
E ToolStrip	🚀 Form1 👝 🖻 🖾	· · · · · · · · · · · · · · · · · · ·
ToolStripContainer		s Re Properties
- CEOLEI		> Ba 82
▶ 포인터		 Form1.cs
🚵 Chart		Soft Form1.Designer.cs
Fill EindingNavigator		FormLresx
BindingSource		Sa Programas
DataGridView		
DataSet		
▲ 구성 요소		
▶ 포인터		
BackgroundWorker		
DirectoryEntry		
E DirectorySearcher		
ErrorProvider		
EventLog		
FileSystemWatcher		
FI HelpProvider		
ImageList		
MessageQueue	E .	
PerformanceCounter		
Process		
SerialPort		
ServiceController		
G Timer		
- 인쇄 표이터		
표 전 대 Page Cath in Dialog		
DrintDialog		
DrintDocument	9	
PrintPreviewControl	# senaiPort1	
PrintPreviewDialog		
and the strend and		
준비		

• 왼쪽의 도구 상자에서 SerialPort를 드래그하여 오른쪽의 Form1에 놓는다.



 도구 상자에서 버튼을 클릭하여 Form1에 두 개의 버튼을 넣어준다. 그리고 마우스 오른쪽 버튼을 클릭한 후 속성에 들어가서 button1과 button2의 이름(Text)를 LED ON, LED OFF로 바꾸어 준다.

EPOTI DUTON2	Get 4 ArduinoInterface (1 프로 C ArduinoInterface ► @ ArduinoInterface ► @ Properties ► @ Promiles ♥ Formilesx ♥ Program.cs	젝트)
	속성	- ų ×
	button1 System.Windows.Forms.Butt	on -
	21 21 🔟 🖋 🖻	
	Locked False ▷ Margin 3, 3, 3 MaximumSize 0, 0 ▷ MinimumSize 0, 0 Modifiers Private ▷ Padding 0, 0, 0, 0 RightToLeft No ▷ Size 99, 50 Tabindex 0 Tabindex 0	
₽ serialPort1	Text LED ON Text 컨트롤에 연결된 텍스트입니다.	-



- 라벨을 이용하여 Sensor Value라는 텍스트를 넣어주고, RichTextBox를 이용하여 센서값이 출력될 빈 텍스트 상자를 넣어준다.
- 이 때, Form1을 더블클릭하면 다음과 같이 소스코드 창으로 넘어가게 된다. Form1을 더블클릭하였기 때문에 Form1_load라는 함수가 생겼다. 마찬가지로 두 개의 LED ON, LED OFF 버튼을 더블 클릭하여 소스코드에 버튼 클릭 함수를 넣어주자.



 이번 예제에서 구현할 1) C#에서 버튼으로 LED컨트롤하기, 2) FSR 센서값 C#에 디스플레이를 하기 위해, 다음의 회로를 구성한다.


• 아두이노에 다음의 프로그램을 업로드한다.

```
int FSRpin = 0; // FSRpin을 아날로그0(A0)에 연결
int Vo; // 센서값을 저장할 변수
float Rfsr;
void setup()
{
    pinMode(13, OUTPUT);
    Serial.begin(9600); // 센서값을 읽기 위해 시리얼 모니터를 사용할 것을 설정.
}
void loop()
{
    if(Serial.available())
```

```
{
    int c = Serial.read();
    if (c == '1')
    {
        digitalWrite(13,HIGH);
    }
    else if (c == '0')
    {
        digitalWrite(13,LOW);
    }
}
Vo = analogRead(FSRpin); // 아날로그를 입력 받음 (0~1023)
Rfsr = ((9.78 * Vo)/(1-(Vo/1023.0)));
Serial.println(Rfsr); // println은 줄바꿈 명령이다. Rfsr의 값을 출력한다.
```

- C#과 아두이노 인터페이스를 구성하기 위해 시리얼 포트를 설정해야 한다. 시리얼 포트를 설정하기 위해서는 Baudrate과 포트 번호를 아두이노와 동일하게 설정하여야 한다. Serial.begin(9600)에서 9600bps를 사용하였기 때문에 9600으로 입력하고, COM10을 사용하였기 때문에, 다음과 같이 입력한다. 여러분이 사용하고 있는 포트 번호를 입력한다.
- 다음의 C#코드를 앞서 Visual Studio C#에 치환한다.

C# 코드

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
namespace ArduinoInterface
{
    public partial class Form1 : Form
    {
```

```
string sensorval;
   public Form1()
    {
        InitializeComponent();
        serialPort1.PortName = "COM10";
        serialPort1.BaudRate = 9600;
        serialPort1.0pen();
   }
   private void Form1_Load(object sender, EventArgs e)
    }
   private void button1_Click(object sender, EventArgs e)
    {
        serialPort1.Write("1");
       button1.Enabled = false;
       button2.Enabled = true;
   }
   private void button2_Click(object sender, EventArgs e)
   {
        serialPort1.Write("0");
       button1.Enabled = true;
       button2.Enabled = false;
    }
   private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
    {
        sensorval = serialPort1.ReadLine();
        this.Invoke(new EventHandler(DisplayText));
    }
   private void DisplayText(object sender, EventArgs e)
    {
        richTextBox1.AppendText(sensorval);
        richTextBox1.ScrollToCaret();
   }
}
```



● 상단의 디버깅 시작 버튼 혹은 F5를 누르면 프로그램이 실행되고 다음의 창이 열린다.

- Form1	
LED ON	LED OFF
Sensor Value 5386, 12 5363, 01 5317, 00	•
5271,26 5248,49	-

 LED ON 버튼을 누르면 아두이노의 LED가 켜지고, LED OFF 버튼을 누르면 LED가 꺼지는 것을 확인할 수 있다. 센서 값도 실시간으로 업데이트가 되는 것을 확인할 수 있다.

22.[응용7] 매틀랩을 이용한 센서값 디스플레이

매틀랩(MATLAB)은 Mathworks사에서 개발된 수치해석 및 프로그래밍 환경을 제공하는 공학용 프로그램이다. 많은 공학도들이 매틀랩을 이용한 과제 및 연구를 하고 있기에 이번 예제에서는 실시간으로 변하는 센서값을 매틀랩에서 보여주는 기본적인 프로그램을 작성하겠다. 앞서 소개된 FSR을 이용한 회로도를 다시 구성해보도록 하겠다.

준비물: 아두이노 우노, USB 케이블, 10k옴 저항, FSR센서

따라하기:

• FSR과 10k옴 저항을 이용하여 다음의 회로를 구성한다.



아두이노에 다음의 프로그램을 업로드한다.

int FSRpin = 0; // FSRpin을 아날로그0(A0)에 연결 int Vo; // 센서값을 저장할 변수 float Rfsr; 뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

```
void setup()
{
    Serial.begin(9600); // 센서값을 읽기 위해 시리얼 모니터를 사용할 것을 설정.
}
void loop()
{
    Vo = analogRead(FSRpin); // 아날로그를 입력 받음 (0~1023)
    Rfsr = ((9.78 * Vo)/(1-(Vo/1023.0)));
    Serial.println(Rfsr); // println은 줄바꿈 명령이다. Rfsr의 값을 출력한다.
}
```

• 매틀랩을 실행한다.



매틀랩 초기 시작 화면

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)

📣 MATLAB R201	2b					_				Medel and			
HOME	PLOTS	APPS							2		🕒 🕐 Search	n Documentat	ion 🔎 🖬
New New O Script	Find Files	Import Data W	Save Jorkspace	New Variable Open Variable Clear Workspace	Analyze Code	Simulink Library	Layout	 (i) Preferences iii) Set Path iiii) Parallel ▼ 	? Help	û Community ঔ Request Support			
F A A CR CR	FILE		1	VARIABLE	CODE	SIMULINK	El	VIRONMENT		RESOURCES			
Image: Second secon										Workspace		• •	
Name A		U		Inanu Window	s Video, see Everaples, er r	and Cotting	Started			e e	vvorkspace	141	
Details		~	f, >	> 							Command Hi %	"" story -02-07 오전 -02-11 오전	• 9:07× 12:24×
Ready													OVR .::

• 매틀랩 스크립트를 열어서 다음의 코드를 작성한다.

```
instrreset
serialObj = serial('COM10','BaudRate',9600);
fopen(serialObj);
i=1;
while(1)
    a = str2num(fscanf(serialObj));
    plot(i,a,'*');
    hold on
    drawnow
    i=i+1;
end
```

• 그리고, 실행을 하면, 다음과 같이 센서값이 실시간으로 출력되는 것을 확인할 수 있다.

뻔뻔한 아두이노 매뉴얼 Ver. 1 - 메카솔루션 (www.mechasolution.com)



• 이 때, while(1)이라는 무한루프에서 빠져나오기 위해서 Ctrl + C 이용할 수 있다.