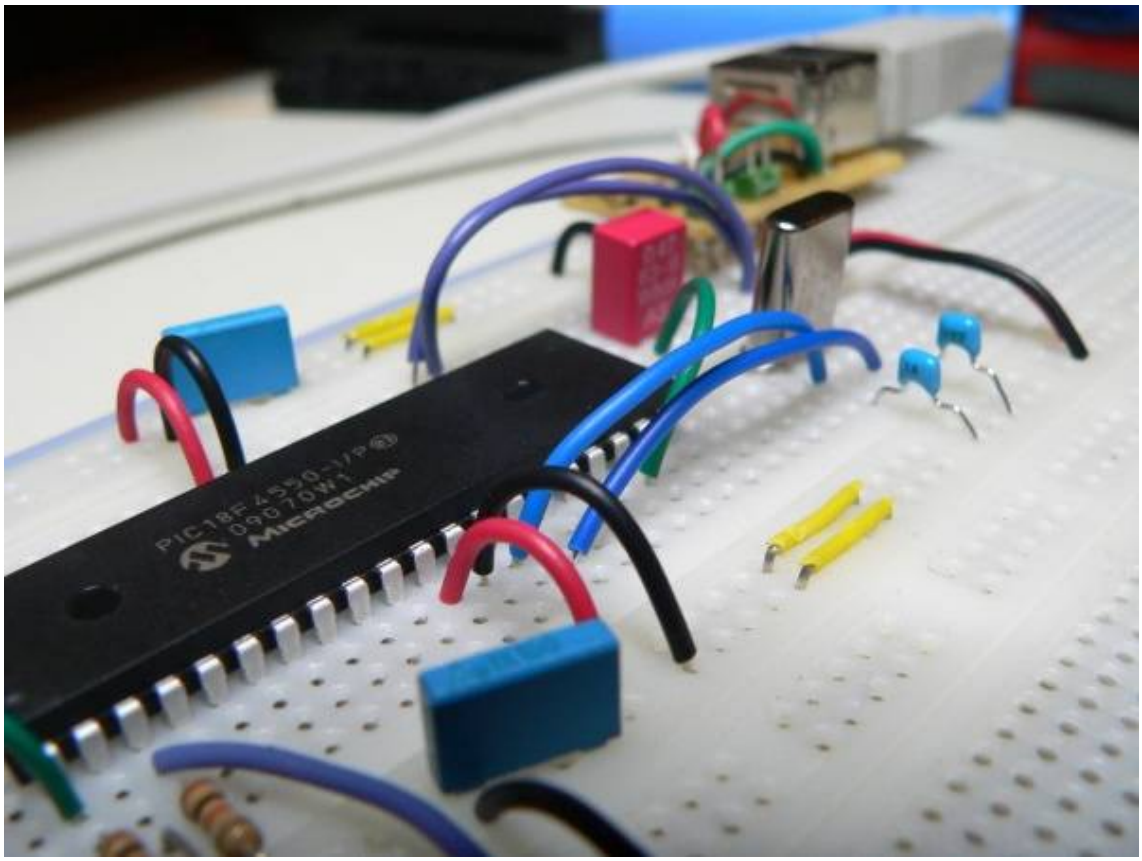
PIC18F USB 장치 만들기

번역 : binoopang (binoopang @ gmail . com)

□ 소개

이 글에서 나는 어떻게 단순한 USB HID 장치를 빵판에 올리고, PIC18F 펌웨어를 만들고 최후에 윈도우에서 장치의 푸시 버튼의 상태를 확인하고 LED를 컨트롤 할 수 있는 인터페이스를 만들 수 있는지 소개할 것이다. 윈도우7 덕분에 만약 커스텀 USB 드라이버를 만들기 위해서는 값비싼 마이크로소프트의 유효한 인증서가 필요하다(이거 없이는 심지어 소프트웨어를 설치 할 수도 없다). 일반적인 HID 장치를 위한 빌트인 드라이버는 윈도우와 리눅스 호환 가능하고 펌웨어와 소프트웨어를 더욱 단순하게 구현 가능하도록 해 준다. HID 표준이 커스텀 드라이버를 필요로 하지 않기 때문에 당신의 드라이버를 위한 인증서를 구매하지 않아도 되고 윈도우와 리눅스는 통신을 도와주기 위한 빌트인 라이브러리를 가지고 있다.



이 글을 위해서 우리는 기본적은 USB 장치에 대해 알아야 한다. 장치는 윈도우에서 LED를 컨트롤 할 수 있고 장치의 푸시 스위치의 상태를 확인할 수 있다. 이 두 가지 기능의 USB 통신 이론을 이용하면 이를 이용해서 더욱 복잡한 프로젝트를 할 수 있게 될 것이다.

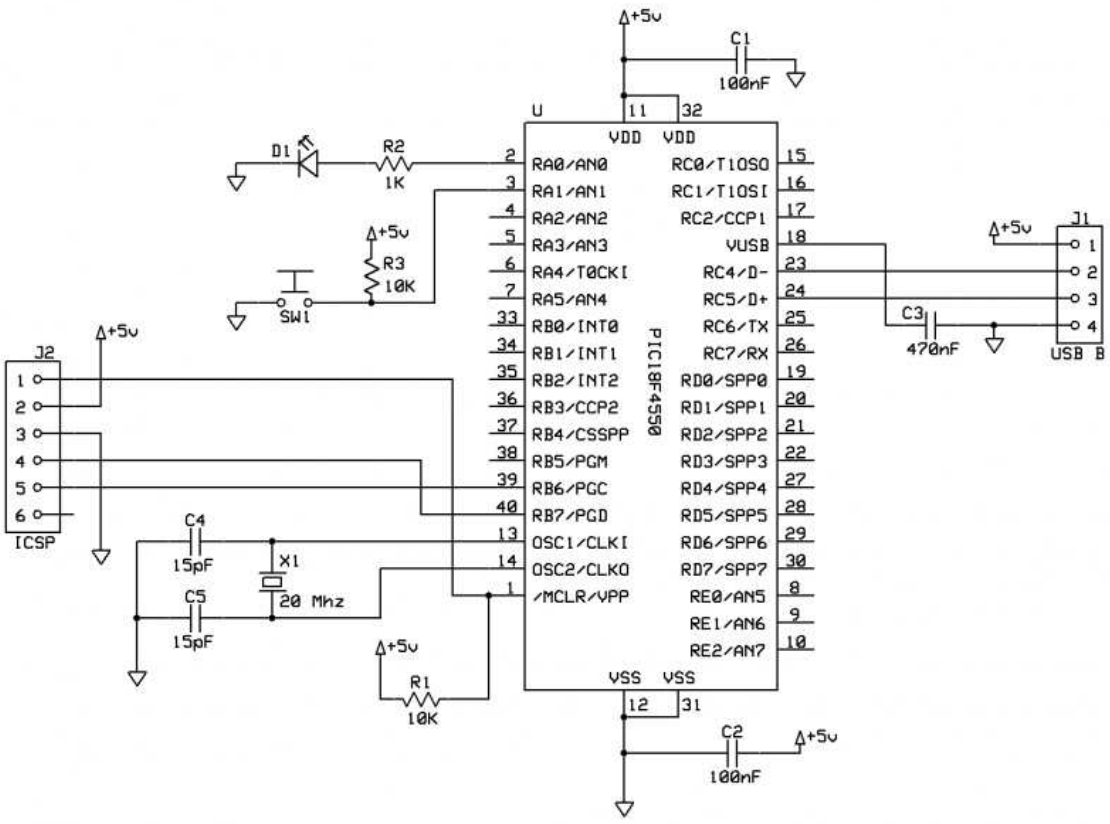
가능한 비용과 난이도를 낮추기 위하여 나는 빵판에 소수의 콤포넌트를 이용하여 하드웨어를 구성하고 PIC18F 펌웨어는 MPLAB과 Hitech C 컴파일러를 이용할 것이며, 윈도우 소프트웨어는 마이크로소프트 Visual C++ 2008 Express를 이용 할 것이다.

이 글이 비록 PIC18F4550 마이크로컨트롤러를 기반으로 하고 있지만 당신은 4550과 코드 호환 가능한 PIC18F2550으로 포팅할 수 있을 것이다.

만약 당신이 이 글을 따라 하길 바란다면 나는 이 글의 가장 하단으로 이동하여 첨부한 소프트웨어를 다운받기를 권하고 싶다. 또한 당신은 MPLAB, PIC18F용 HiTech C 컴파일러 그리고 마이크로소프트 Visual Studio 2008 Express가 설치되어 있어야 한다.

□ 하드웨어

시작하기 위해서 우리는 통신 할 USB 장치를 제작해야 한다. 이어지는 회로 다이어그램을 통해서 최소한의 사용가능한 USB 장치의 사양을 확인할 수 있다. 장치는 ICSP(In Circuit Serial Programming) 헤더와 USB Type B 커넥션을 포함한다. 추가적으로 한 개의 LED와 장치의 입출력을 나타내기 위한 한 개의 푸시 스위치를 포함한다.



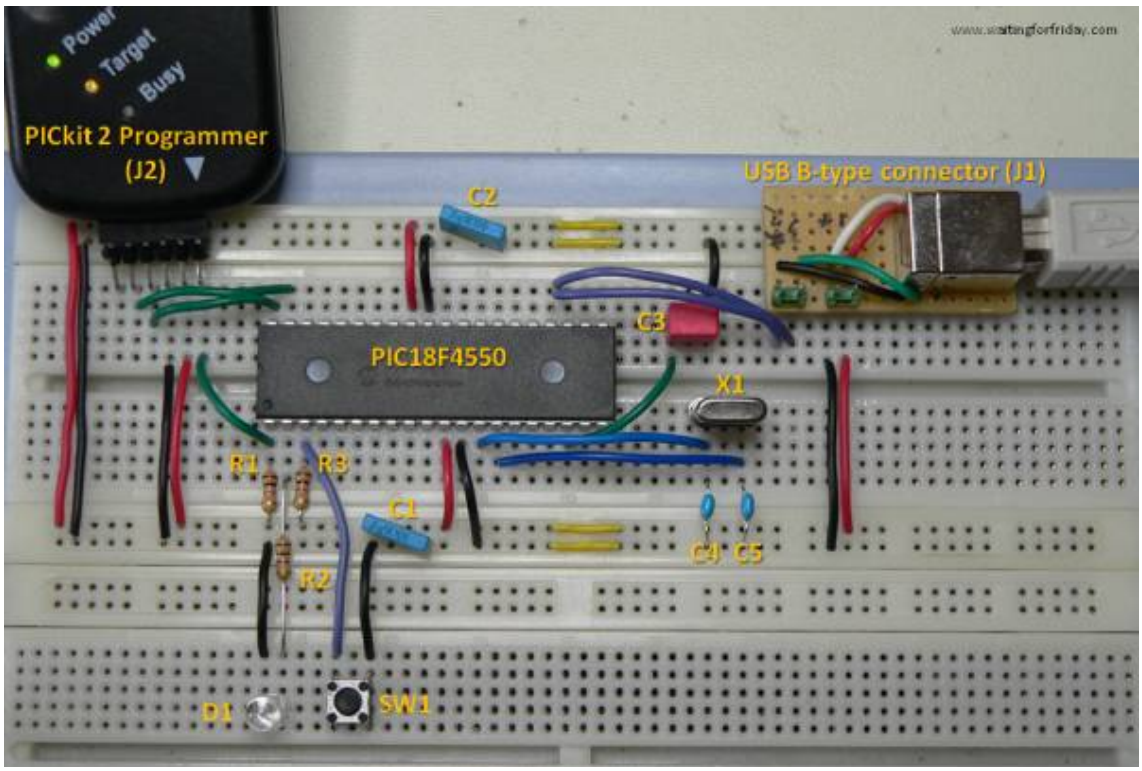
회로는 매우 직관적이다(만약 이 정도 수준의 마이크로컨트롤러 회로가 잘 이해 안된다면 몇 가지 LED 예제와 푸시 버튼 공부를 하길 권한다.) PIC18F4550은 "버스파워"를 이용한다;

이 말은 장치가 PC로부터 전원을 공급받으며 전원 레귤레이션이 필요 없다는 것을 의미한다. PIC가 내부 회로를 동작 시킬 수 있도록 470nF 캐퍼시터(C3)가 필요하다(이것은 PIC 내부의 온보드 USB 인터페이스가 필요로 하는 USB 전압을 레귤레이팅 하는데 도움을 준다).

ICSP 헤더는 PIC 프로그래머를 연결하는데 사용되는데 개인적으로 비싸지 않은 PICkit2 프로그래머를 추천하지만 다른 ICSP 호환 프로그래머들도 잘 작동한다. USB 어플리케이션을 위해서 20MHz 클럭이 필요하다. 이것은 USB 통신에 요구되는 48MHz의 클럭 스피드로 높여주는 PIC의 PLL을 사용할 수 있게 해 준다.

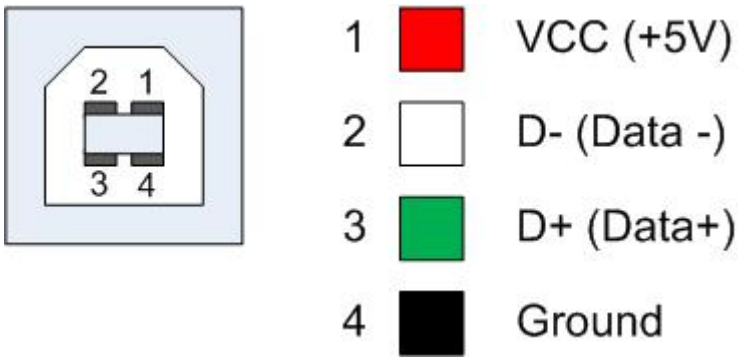
(독자 [Jason]은 PIC18F의 오보드 USB를 이용하기 위해서 20MHz 크리스탈만 사용할 수 있는 것이 아니라고 지적하였다. 당신은 PIC 퓨즈 설정을 변경하여 다양한 종류의 크리스탈을 이용할 수 있다. PIC18F4550 데이터시트의 29~30 페이지를 통해서 더 많은 정보를 확인하기 바란다. - 감사 제이슨!!)

이어지는 사진은 취미 빵판에 구성된 회로를 보여준다. 나는 콤포넌트가 어디에 위치해 있는지 식별하기 쉽게 하기 위하여 몇 개의 라벨을 사진에 붙여 놓았다. 프로그래밍을 위해서 프로그래머에게 5V 전압을 공급해야 한 다는 사실을 알고 있기 바란다. 버스 파워 USB 장치이기 때문에 5V 라인은 USB 커넥터에도 연결이 된다. 이것은 만약 프로그래머와 USB 케이블에 동시에 연결이 되면 잠재적으로 프로그래머에서 USB 호스트로 5V 전압이 전달될 수 있다; 이것은 USB 표준에서 권장되지 않은 사양이다. 나는 지금까지 이런 경우를 확인하지 못하였다. 원한다면 USB 커넥터에 다이오드를 이용한 장벽을 세워 이 같은 일을 방지할 수 있다. 나의 프로젝트에서는 1N5817 다이오드를 이용한 장벽을 세워놓았다.



만약 당신이 빵판을 위한 USB 플러그를 가지고 있지 않다면 작은 크기의 커넥터를 만들거나 전선을 직접 빵판에 연결하여 USB 케이블을 사용할 수 있다. 나는 시간을 가지고 아답타를 만들길 권장하며 이 것은 USB 케이블을 PC에 연결하거나 해제 할 때 빵판의 전선이 루즈해 지는 위험을 줄여준다.

만약 USB 케이블 와이어링에 확신을 가지지 못한다면 이어지는 사진을 이용하기 바란다.



위의 회로가 완성되고 나면 당신의 PC에 연결하기 전에 +극과 -극 연결이 잘 되고 있는지 확인 한다; 당신은 당신의 컴퓨터가 망가지는 것을 원치 않을 것이다.

□ 펌웨어

USB 장치를 컴퓨터에 연결하기 위해서 가장 첫 번째로 필요한 것은 PIC18F4550을 위한 펌웨어를 작성하고 컴파일 하는 것이다. 마이크로칩(PIC 마이크로컨트롤러 제작회사)는 이러한 목적으로 사용할 수 있는 USB 스택을 무료로 제공하고 있다. 쉽게 하기 위해서 나는 장치를 동작시킬 수 있는 단순한 펌웨어를 작성하였고 당신은 이 펌웨어를 사용하고 이해함으로써 펌웨어가 어떻게 동작하는지 알 수 있을 것이다. 만약 첫 동작하는 장치를 완성하고 나면 당신은 좀 더 복잡한 어플리케이션에 어떻게 응용할 수 있을지 쉽게 이해할 수 있을 것이다.

펌웨어는 아래에서 설명 할 중요한 두 가지를 수행해야 한다.

장치 열거(Enumeration)

첫 번째는 USB 장치 열거이다 - 이 복잡한 기술은 사실 장치가 호스트에게 무엇이고, 어떻게 통신하길 희망하는지 말해주는 일종의 USB 호스트와의 초기 통신이다. USB 통신은 호스트나 장치로 정보를 전송하는 엔드 포인트를 이용한다. 통신 채널이 설정되고 나면 장치는 장치 이름과 두 가지 중요한 값을 전달해야 한다; VID와 PID이다.

VID는 Vendor ID 이며 장치의 제조사를 식별한다. 당신만의 VID를 얻기 위해서는 수천의 비용을 지불하거나 USB 표준 바디(body)를 이용할 수 있다. 이 예제에서는 마이크로칩의 VID를 이용하여 비용을 아낄 것이다. 만약 당신이 상품을 판매할 것을 심각하게 고려하고 있다면 당신의 것을 하나 등록해야 할 것이다.

PID는 Product ID이다. VID와 함께 당치를 유일하게 식별한다. 당신의 장치가 윈도우에서 처음 열거 되었을 때 윈도우는 장치의 VID와 PID 조합을 저장한다. 이는 만약 장치의 열거 정보를 변경하길 원할 때 중요하다. 최소한 다시 연결하기 전에 PID를 변경해야 하며 그렇지 않을 경우 코드가 흐르고 있다하더라도 'Device not started'에러를 발견하게 될 것이다.

호스트와의 통신

두 번째 중요한 작업은 펌웨어가 호스트와 장치간에 실질적인 통신을 수행하는 것이다. 각각의 통신은 'command'에 의해 구분된다. 일반 HID 표준을 이용할 때 'command'는 명령과 함께 전달되는 정보가 어떻게 해석되어야 하는지 알려준다. 이 정보는 무엇이든 될 수 있으며 이는 당신의 장치가 매우 훌륭한 유연성을 가질 수 있다는 것을 의미한다.

당신의 장치가 열거된 후 호스튼 주기적으로 장치를 폴링하지만 장치는 그렇지 않을 것이다 (좀 더 깊숙이 통신 프로토콜을 살펴보면 예외적인 상황도 존재한다). 각각의 폴링에서 호스트는 명령과 데이터 둘 다 장치로 전송하고 장치로부터 명령과 데이터를 받을 수 있다.

당신이 눈여겨 보어야 할 펌웨어의 주요한 부분은 호스트로부터 폴링 리퀘스트를 처리하는 부분과 장치가 작동할 수 있도록 필요한 액션 부분이다.

펌웨어 소스코드 이해

이 글에 첨부된 'PIC18F Generic HID Device' Zip파일은 PIC18F4550에서 사용가능한 펌웨어를 포함하고 있다. 단순히 파일 압축을 해제하고 MPLAB을 이용하여 프로젝트를 연다. 나는 소스파일과 헤더 파일을 분리해 놓았으며 마이크로칩 스택의 generic 파트를 확인해야 한다 (이것은 'USB stack' 서브 디렉토리에 저장되어 있다). 'USB stack' 디렉터리 아래에 존재하는 파일들은 흥미롭지만 완전 세밀하게 알고 있을 필요는 없다.

VID/PID와 열거 정보들은 이미 준비가 되어 있기 때문에 당신은 프로젝트를 빌드하고 그 결과를 PIC18F로 로딩하면 된다. 물론 이 작업을 위해서 온전한 빌드 환경이 구축되어 있어야 하며 환경에 뭔가 문제가 발생했을 경우 구글을 통해서 해결하기 바란다. 먼저 단순한 프로젝트를 빌드해본 뒤에 우리 프로젝트를 진행하기 바란다.

펌웨어는 다음과 같은 세 가지 명령을 제공한다.

- 0x80 - LED를 토글한다
- 0x81 - 푸시 스위치 상태를 읽는다
- 0x82 - LED 상태를 읽는다

이 명령을 수행하는 부분은 main.c 소스파일에 존재하는 ProcessIO() 함수에 존재한다. 이 함수는 요구되는 명령을 결정한 뒤 데이터를 전송하고 수신하는 책임을 맡는다. 이 부분은 USB 스택이 유연성을 우선하기 때문에 매우 직관적이다. 소스코드를 보면 얼마나 간단한지

알 수 있다. 함수에 의해 유일하게 수행되는 추가 확인작업은 장치가 설정 상태에 들어가 있는지 확인하는 것이다; 이것은 장치가 호스트에 연결되었고 열거가 정상적으로 수행되었다는 것을 의미한다.

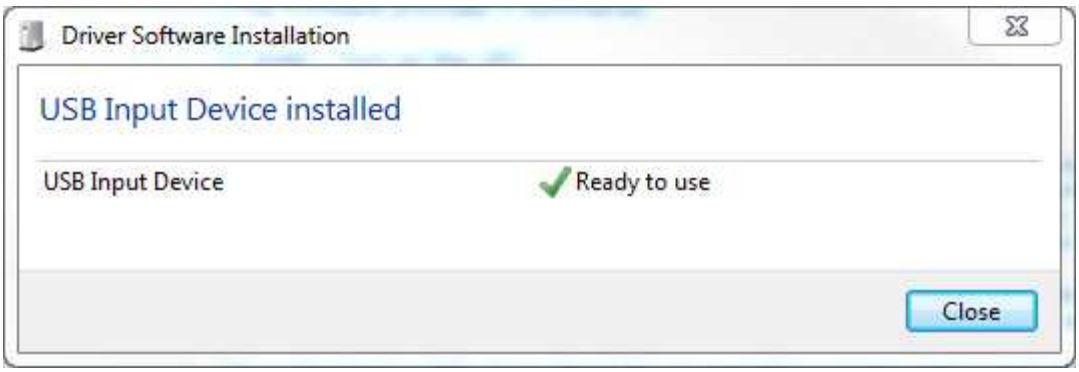
main 함수는 단순히 로우 레벨의 장치 작업을 수행하고 ProcessIO()를 반복적으로 수행하기 위해 USB stack을 호출한다. 이것은 루프 대신 인터럽트를 이용할 수 있다. 하지만 나는 펌웨어를 최대한 단순화 하였다.

열거과정을 좀 더 이해하기 위해서 장치가 처음 연결되었을 때 호스트로 전달되는 정보가 담긴 sub_descriptors.c를 확인하여라. 소스에서 장치가 호스트에게 어떤 타입의 인터페이스이고 인터페이스의 수용가능 정도를 설명하는 설정 디스크립터의 정보를 확인할 수 있다. 엔드포인트는 앞에서 설명한 파이프를 위한 연결자이다. 또한 제조회사와 상품을 문자적으로 표현하는 문자열도 존재한다. 윈도우는 일반적으로 USB 이름을 지어줄 때 이 정보를 사용한다.

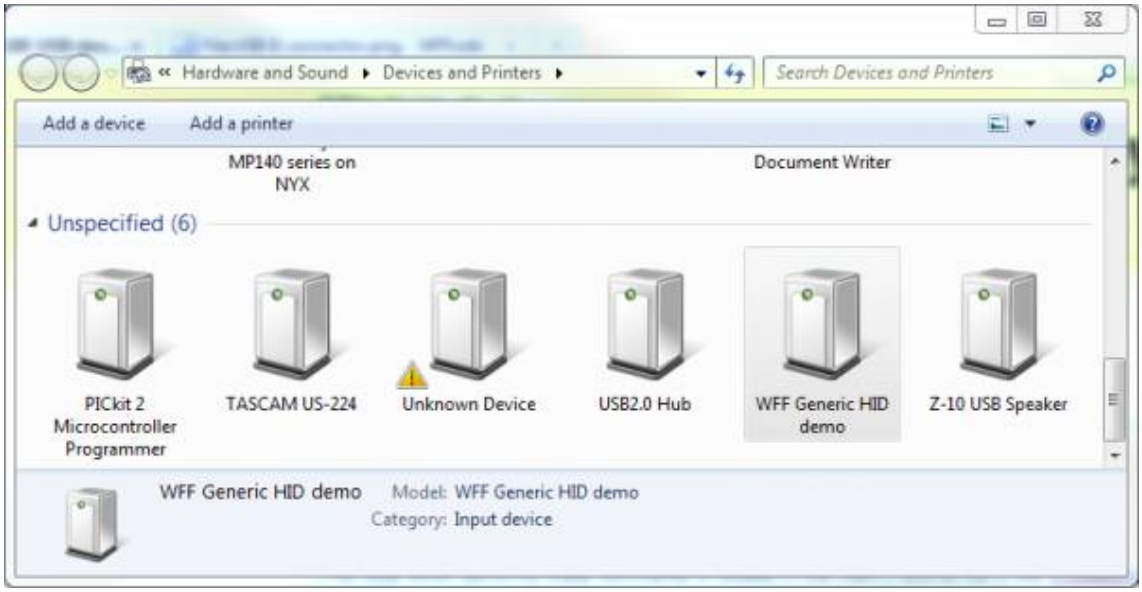
장치 연결

위에서 다운로드 한 펌웨어를 USB 장치에 로드하였다면 PC에 연결할 준비가 끝난 것이다. 우리는 일반 HID USB 장치를 이용하기 때문에 연결하기 전에 PC에 어떤 것도 설치 할 필요가 없다. 단순히 USB 케이블을 장치와 PC에 연결한다.

윈도우7은 새로운 장치를 탐지하고 새로운 하드웨어를 설치중이라는 메시지를 보여주어야 한다. 몇 초 후에 다음과 같은 대화상자를 확인할 수 있다.



만약 시작 메뉴에 존재하는 장치와 프린터를 선택하면 다음과 같은 화면을 볼 수 있다.



당신의 USB 장치는 열거되었고 사용 할 준비가 되었다. 이제 우리는 호스트 쪽 프로그래밍을 하고 마이크로소프트 Visual C++ 2008을 이용하여 어떻게 장치와 통신하는지 확인해야 한다.

□ 호스트 소프트웨어

호스트 소프트웨어는 매우 직관적이고 기본적으로 세 가지 부분으로 나뉘어 진다.

- USB 장치를 모니터링 하여 확실히 연결되었는지 확인
- 사용자가 어플리케이션과 상호작용할 수 있도록 사용자 인터페이스를 표시하고 처리
- USB 장치와 통신하고 상태를 업데이트 함

나는 호스트 소프트웨어를 USB 스택에 존재하는 알반 PnP HID 소프트웨어를 기반으로 하였다. 그러나 펌웨어때와 마찬가지로 나는 최대한 쉽게 만들려고 시도하였다.