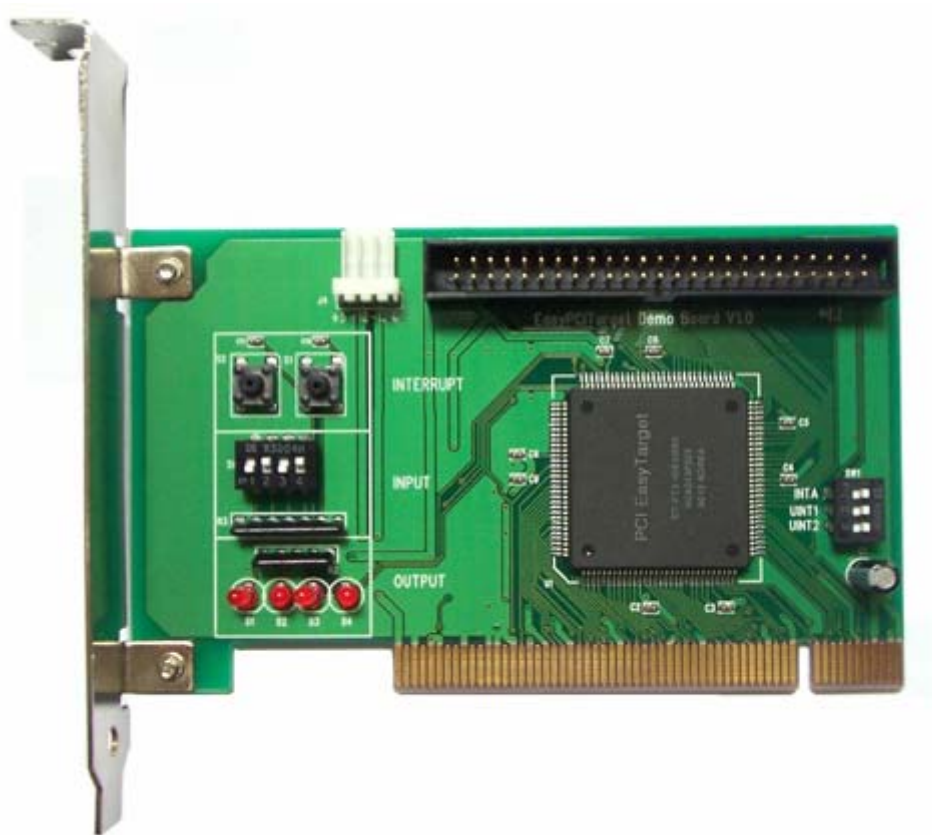


## Easy Target PCI KIT V3.0



(C) 2006 에듀키트 [www.edukit.co.kr](http://www.edukit.co.kr)

TEL : 0505-586-8086

## 목 차

1. Easy Target PCI KIT .....	3
2. EasyTarget Chip 사양 .....	4
3. 콘넥터, DIP 스위치 설명 .....	9
4. 내부 테스트 회로 .....	12
5. WDM 드라이버의 설치 .....	13
6. 테스트프로그램 설명 .....	15
7. WDM 디바이스 드라이버 설명 .....	16

## 1. EasyTarget PCI KIT

### 아주 쉬운 PCI 인터페이스

PCI Target 인터페이스 칩인 EasyTarget 소자를 사용한 PCI 실험 키트이다. EasyTarget 칩은 51개의 다목적 제어핀과 2개의 외부 인터럽트 입력핀이 있는 소자로, PCI 규격이나 PCI Interface에 대한 아무런 학습이 없어도 원하는 제어를 할 수 있다. 디바이스 드라이버는 전혀 손댈 필요없이 응용프로그램의 조작만으로 제어를 할 수 있다.

### 제어 예제 및 확장 콘넥터 보드내에 내장

보드내에 4비트 DIP 스위치 입력 테스트, 4비트의 LED 출력 테스트, 인터럽트 단자 테스트를 할 수 있고

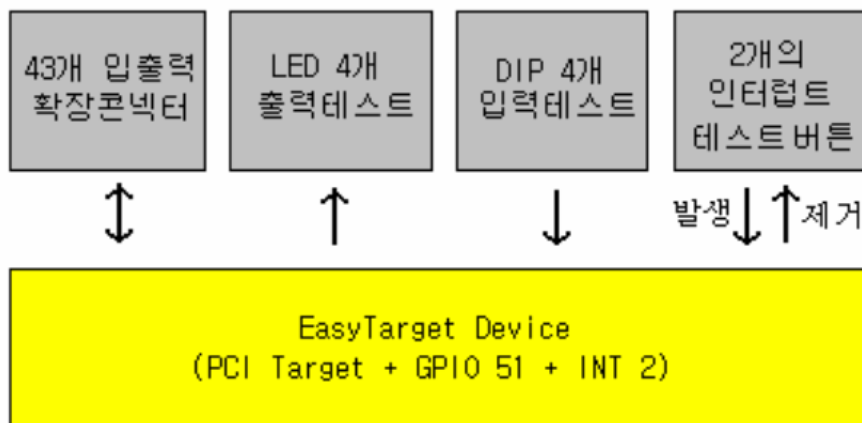
확장 콘넥터를 통하여 50핀 케이블 또는 확장 보드를 통하여 사용자의 제어 회로를 추가할 수 있다.

### 각종 OS 드라이버, 제어 함수 제공

Windows 98, Me, 2000, XP 등을 지원하는 통합 디바이스 드라이버, Visual C++, Visual Basic 데모 프로그램의 소스가 제공되며, 디바이스 드라이버를 전혀 수정하지 않고도 응용프로그램에 제공되는 제어 함수 10개 정도의 사용법만 익히면 원하는 제어를 할 수 있다.

EasyTarget PCI KIT는 [EasyTarget PCI 소자](#)를 사용한 PCI 실험 키트이다. EasyTarget PCI KIT는 PCI Board와 데모 프로그램과 각종 윈도우

디바이스 드라이버 소스가 들어있는 CD, 설명서로 구성되어 있다. PCI Board는 43개의 입출력 및 인터럽트 제어핀 전원핀 등이 나와있는 확장 콘넥터가 있으며, LED 4개 출력, 4개의 DIP 스위치 입력, 인터럽트 버튼 2개가 있다.



## 2. EasyTarget Chip 사양

### 2-1. 사양

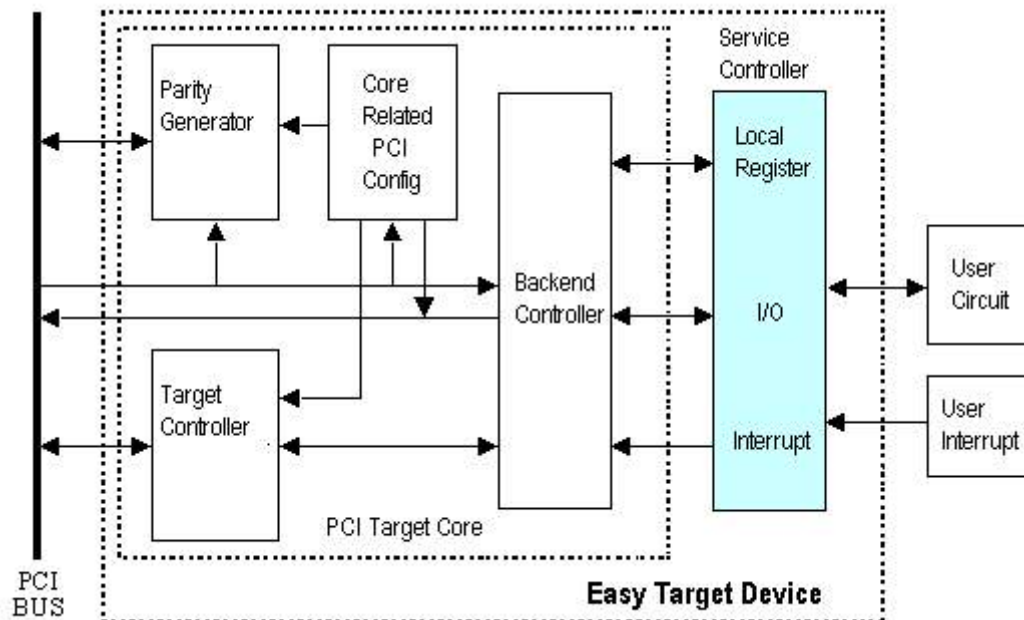
- 게이트 공정기술 : 0.65um - 동작전압 : 5V  $\pm$ 5%
- 패키지 : 120핀 QFP
- 폭, 두께, 높이 : 28mm x 28mm x 4mm
- 리드 폭 : 0.4mm , 리드 피치 : 0.4mm



- 최대 동작 주파수 :  
입력 50MHz 출력 50MHz, 정상 33MHz
- PCI BUS 모드 :  
PCI 버전 2.1 만족 , Transparent 타겟 모드 지원  
PCI 타겟 fast back-to-back 처리 지원

## 2-2. 내부 구조

5개의 모듈로 구성되어 있으며, Block diagram 으로 표시하면 다음과 같다.



### - Target Controller

주어진 상황에 적절한 TRDY#, STOP#, DEVSEL# 등의 제어신호를 만들어 낸다. 또한 현재 동작하고 있는 PCI BUS의 상태를 알려주는 기능을 하여서 다른 모듈들이 적절한 동작을 할 수 있도록 한다.

### - Parity Controller

parity generator 기능을 수행한다. 버스의 상태에 따라서 PAR, PERR#, SERR#의 신호를 발생한다. 또한 PCI config 모듈에 에러 report 신호를 보낸다. 이를 통해서 에러의 상황을 호스트가 파악할 수 있도록 한다.

### - Core related PCI config

target 기능을 수행하는 데에 밀접히 관련된 PCI config 관련 정보를 처리할 수 있도록 한 것이다. 이것이 가지는 정보는 CONFIG 영역에서 04h번지에 있는 Config Status, Config Command 레지스터를 제어한다.

### - Backend Controller

사용자가 PCI 신호에 관련한 신호를 직접 발생시키지 않고 편리하게 사용하기 위한 로직을 구현해 놓은 모듈이다.

### - Service Controller

51개의 범용 제어핀, 인터럽트 로직, Local 레지스터 등을 구현하였다.

## 2-3. 핀 구성

PCI 어드레스/데이터 : 36 개

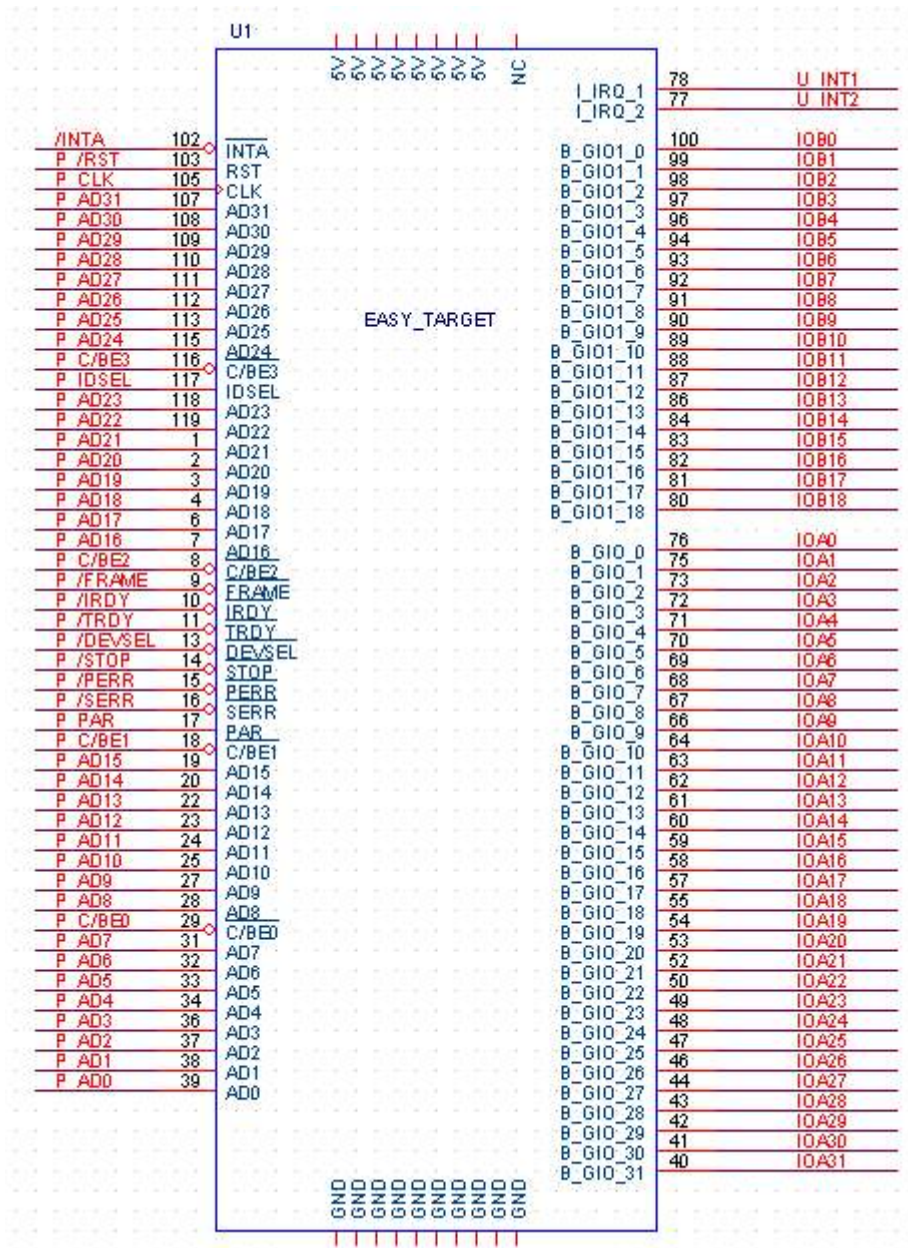
PCI 제어 : 12개

외부 IRQ : 2개 ( Rising/Falling Edge 선택가능 )

범용 I/O : 51 개 (입/출력 프로그램 가능)

전원단자 : GND 10개, 5V 8개

안쓰는 핀 : 1개 ( test핀 , 핀번호 128 )



## 2-4. 내부 레지스터

주소	비트	신호	Def	r/w	설명
0x00					
	[31:0]	B_GIO_o[31:0]	0	w	general output to pin
	[31:0]	B_GIO_i[31:0]		r	general input from pin
0x04					
	[31:0]	B_GIO_oe[31:0]	0	r/w	B_GIO_o output enable 0: input, 1 : output
0x08					
	[18:0]	B_GIO1_o[18:0]	0	w	general output to pin B_GIO1
	[18:0]	B_GIO1_i[18:0]		r	general input from pin B_GIO1
0x0c					
	[31]	irq_2_ack	0	w	writing with '1' generate the ack strobe
	[30]	irq_1_ack	0	w	writing with '1' generate the ack strobe
	[29]	irq_2_en	0	r/w	1 : enable, 0 : disable
	[28]	irq_1_en	0	r/w	1 : enable, 0 : disable
	[27]	irq_2_trg	1	r/w	1 : pos edge irq 2 trigger 0 : neg edge
	[26]	irq_1_trg	1	r/w	1 : pos edge irq 1 trigger 0 : neg edge trigger
	[25]	irq_2_o	0	r	irq 2 flag , 1 : occur
	[24]	irq_1_o	0	r	irq 1 flag , 1 : occur
	[18:0]	B_GIO1_oe[18:0]		r/w	B_GIO1 output enable 0: input, 1 : output

def : 전원인가시 default 값을 말함

참고 : 모든 레지스터는 byte, word, dword 액세스 가능

C언어에서 다음과 같이 정의하여 쓴다.

```
// Easy PCI Target Register Address Map

#define AT_IOA_D0      0x 0
#define AT_IOA_D8      0x 1
#define AT_IOA_D16     0x 2
#define AT_IOA_D24     0x 3
#define AT_IOA_OE0     0x 4
#define AT_IOA_OE8     0x 5
#define AT_IOA_OE16    0x 6
#define AT_IOA_OE24    0x 7
#define AT_IOB_D0      0x 8
#define AT_IOB_D8      0x 9
#define AT_IOB_D16     0xa
#define AT_IOB_OE0     0xc
#define AT_IOB_OE8     0xd
#define AT_IOB_OE16    0xe
#define AT_IRQ_CTL     0xf
```

AT\_IRQ\_CTL 레지스터 설명

D0 (R) : irq 1 interrupt flag

D1 (R) : irq 2 interrupt flag

D2 (W,R) : irq 1의 edge를 정하는 것으로

1로 설정하면, 외부 irq 1단자가 0->1로 바뀔때

인터럽트 flag를 설정하고 인터럽트를 발생시킨다.

0이면 1->0으로 바뀔 때

D3 (W,R) : irq 2의 edge를 정한다.

D4 (W,R) : 외부 irq 1를 사용할 것인가를 정의한다.

1로 설정하면 사용한다.

D5 (W,R) : 외부 irq 2를 사용할 것인가를 정의한다.

D6 (W) : irq 1의 인터럽트를 flag를 clear시킨다.

인터럽트 Ack에 해당하는 것으로 1을 쓰면 된다.

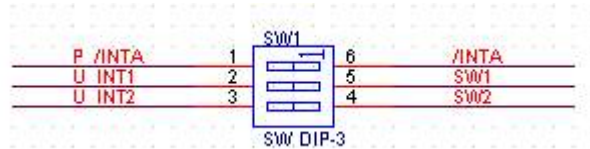
D7 (W) : irq 2의 인터럽트를 flag를 clear시킨다.



### 3. 콘넥터, DIP 스위치 설명

#### 3-1. SW1 설정

J3의 U\_INT1,2, P\_/INTA의 쓰임에 따라 다음의 스위치 설정이 필요하다.



1-6 사이가 ON이 되면, EasyTarget Chip의 인터럽트 출력 /INTA가 PCI Bus의 인터럽트 단자 P\_/INTA가 연결되게 되며, 이것은 EasyTarget Chip의 내부 인터럽트 회로를 사용함을 뜻한다. 이 때는, 외부 회로가 J3의 48번 P\_/INTA를 사용하면 안된다.

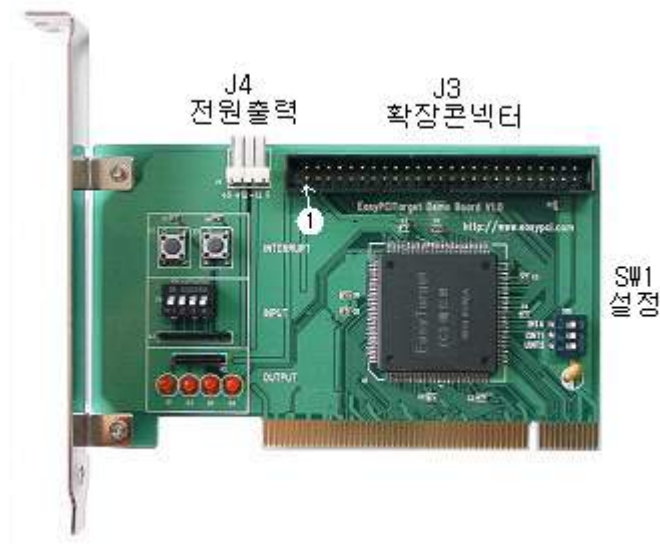
46,48번 U\_INT1,2를 통하여 인터럽트를 인가하여야 한다.

보드내에 있는 인터럽트 스위치가 동작하려면 ON이 되어야 한다.

1-6 사이가 OFF가 되면, EasyTarget Chip의 인터럽트 회로를 사용하지 않고 외부 회로가 J3의 48번(P\_/INTA)을 사용하여 직접 PCI Bus에 인터럽트를 인가하겠다는

뜻이 되며, 이 때는, J3의 46,48번 U\_INT1,2는 사용하지 않는다.

그리고, SW1의 2-5, 3-4의 설정도 의미가 없다.



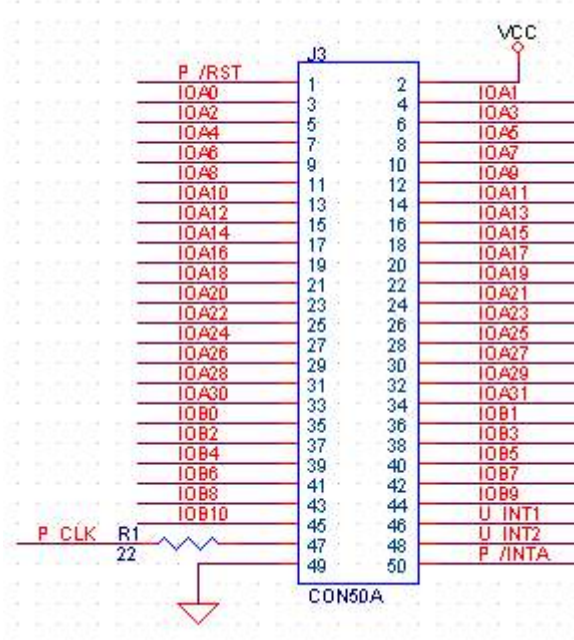
2-5,3-4사이가 ON이 되면, 내부 보드의 SW1,SW2가 U\_INT1,2를 점유하게 되므로, J3의 46,48번 U\_INT1,2를 통한 인터럽트가 부정확하게 된다. J3을 통해 연결되어진 외부 회로가 U\_INT1,2를 사용한다면 2-5,3-4사이는 OFF되어야 한다.

보드내에 있는 인터럽트 스위치가 동작하려면 ON이 되어야 한다.

### 3-2. J3 확장 콘넥터

외부에 추가의 회로를 연결하여 제어하고자 할 때 사용한다.

50핀 케이블을 꽂아서 PC케이스 바깥으로 빼거나, 확장보드를 꽂아 사용할 수 있다.



1번핀 : 보드사진의 의 좌측하단이 1번핀이며, 1번핀 P\_/RST는 PCI Bus의 /RST( 리셋:출력)로, 외부회로에 리셋신호가 필요할 때 사용한다.

2번핀 : +5V단자가 출력된다.

3~32번핀 : EasyTarget Chip의 Aport인 IOA0 ~ IOA31단자이다.

32비트로서 Byte, Word, Dword단위로 동시에 입출력 할 수 있다.

33~45번핀 : EasyTarget Chip의 Bport인 IOB0 ~ IOB10 단자이다. 11비트로서

IOB11 ~ IOB18은 보드 내부의 DIP 스위치 및 LED 출력용으로 사용하고 있다.

46,48번 : EasyTarget Chip의 U\_INT1,2 단자로서 외부의 인터럽트 입력을 받을 때 사용한다. 만일 외부에서 이 것을 사용한다면, 보드 내부의 인터럽트 스위치의

영향을 받지 않도록 SW1을 이용하여 내부의 인터럽트와 차단하여야 한다.

47번핀 : PCI Bus의 CLK (클락:출력) 신호로서 외부에서 Clk신호가 필요할 때 사용하면 된다.

50번핀 : PCI Bus의 /INTA (인터럽트 : 입력) 단자로서, 외부에서 EasyTarget의 U\_INT1,2 을 거쳐 인터럽트를 걸지않고 직접 PCI Bus에 인터럽트를 요청하고자 할때 사용한다. 이럴 때는 SW1의 설정을 바꾸어 EasyTarget의 인터럽트 출력단자가 PCI Bus의 /INTA에 연결되지 않도록 하여야 한다. 이것의 사용예는, EasyTarget Chip이 인터럽트단자가 2개이므로, 실제로 필요한 것은 8개라면

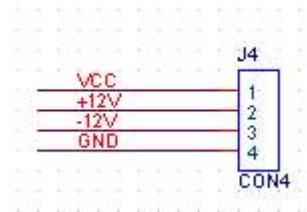
외부의 8개의 인터럽트 신호를 OR하여 **EasyTarget**칩을 거치지 않고 곧 바로 PCI Bus에 인터럽트를 거는 경우이다.

### 3-3 J4 전원 출력 콘넥터

J4는 PC의 전원출력으로 +5v,+12V,-12V, Gnd 4단자로 되어있다.

외부보드에서 +12V,-12V전원이 필요할 때 사용하면 편리하다.

+5V,Gnd만 필요로 할 때는 J3에서도 공급되므로 J4는 필요 없다.

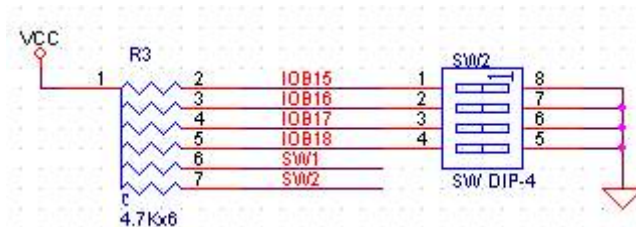


## 4. 내부 테스트 회로

### 4-1. DIP S/W 입력

보드내의 DIP 스위치는 EasyTarget Chip의 Bport 15~18에 연결되어 있다.

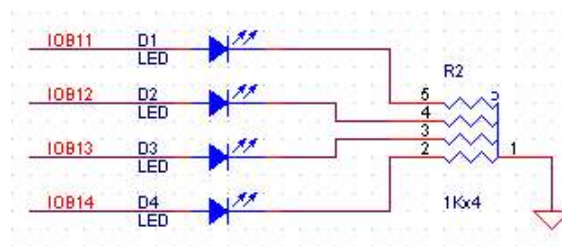
평소에는 Pull Up 되어 있어 1이 되며, 스위치를 On시키면 0 이 된다.



### 4-2. LED 출력

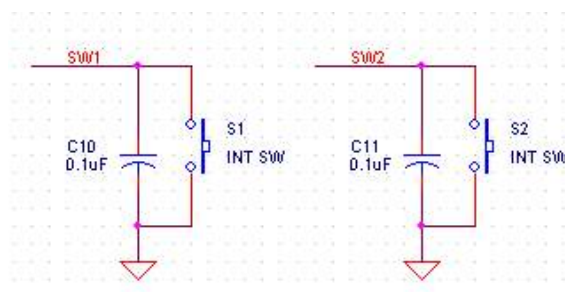
보드내에 있는 LED는 EasyTarget Chip의 Bport 11~14 에 연결되어 있다.

1을 출력하면 LED가 점등된다.



### 4-3. 인터럽트 입력

보드내에 있는 인터럽트 테스트 버튼이 2개 이며, 보드에 있는 W1설정스위치를 거쳐 EasyTarget Chip의 U\_INT1,2 단자에 연결되어 있다. 외부회로에서 J3 확장콘넥터 U\_INT1,2 를 사용한다면, 내부의 인터럽트 회로의 간섭을 없애기 위해 보드의 SW1의 설정스위치를 OFF시켜야 한다.

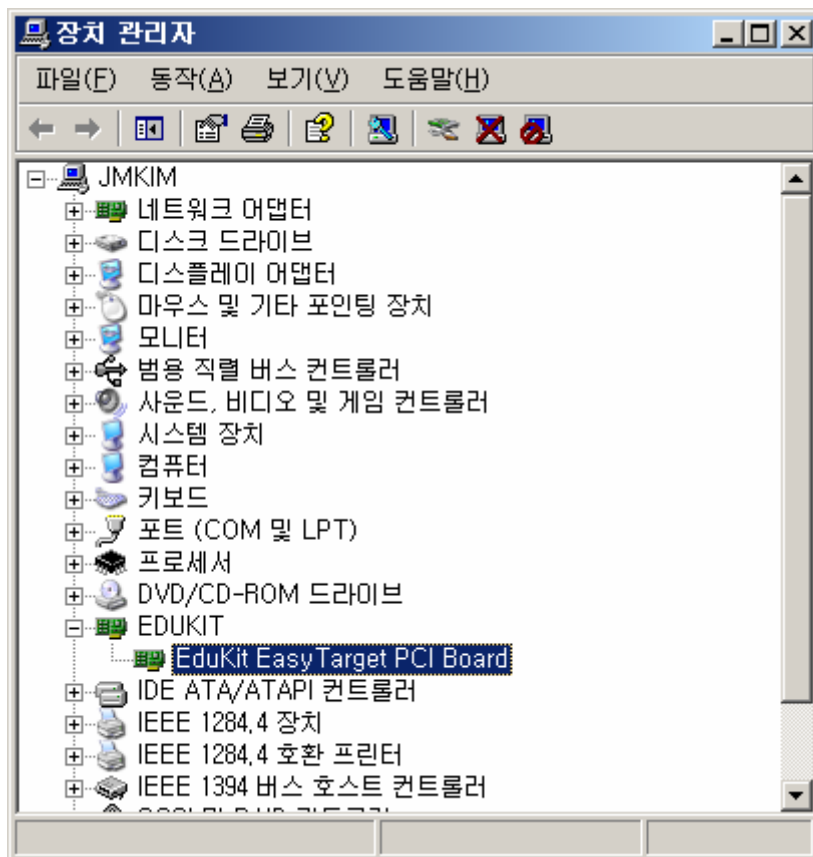


## 5. WDM 드라이버의 설치

윈도우가 시작되면, 새로운 하드웨어 발견이라고 하면서 드라이버를 요구 한다. 제공되는 CD의 Run\Driver를 지정하여 주면 edueztarget.sys가 윈도우의 system32\driversss 디렉토리에 복사된다.

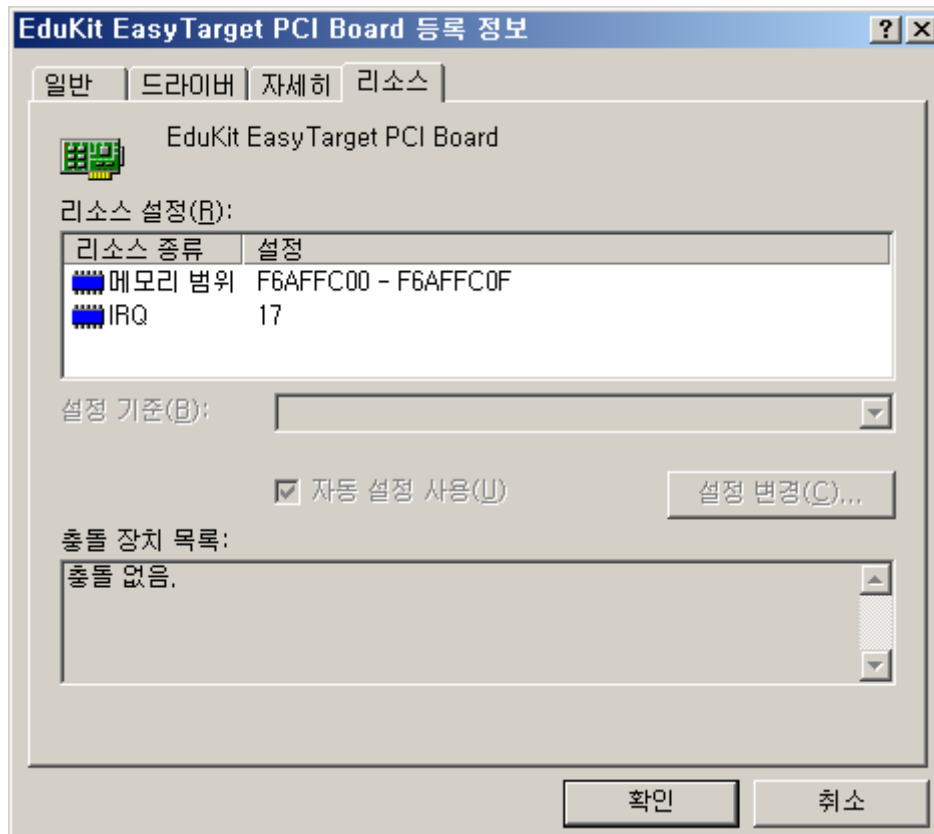
디바이스 드라이버가 올바르게 설치되어 있다면, 제어판 - 시스템 - 하드웨어탭 - 장치관리자를 선택하여 보면

EDUKIT에 Edukit EasyTarget PCI Board 드라이버가 제대로 설치되어 있음을 확인 할 수 있다.



Easy Target 장치 드라이버의 위에서 우측 마우스 버튼을 눌러 등록정보를 보면 그림과 같이 보드가 점유한 메모리 및 인터럽트 번호가 보인다.

인터럽트 번호는 다른 하드웨어 장치와 공유 될수 있다.



# 참고 : 드라이버 설치가 잘 안될 때

1. East target Board의 슬롯에 닿는 부분을 지우개로 깨끗하게 닦는다.
2. VGA card만 제외한 모든 카드를 컴퓨터의 슬롯에서 빼고, Board를 꽂아 본다.
3. Board를 다른 PCI 슬롯에 꽂아 본다.
4. 장치관리자에서 "EduKit EasyTarget PCI Board"에 느낌표나 X가 표시되어 있으면 드라이버를 제거하고, "새로 고침" 버튼을 눌러 다시 설치한 후에, 윈도우를 재 시작한다.

## 6. 테스트프로그램 설명

ezdemo.exe 프로그램을 실행하면 다음과 같은 실행화면이 나타난다.

Card 는 동일한 EasyTarget board 가 있을 때 그 중 하나를 선택하여 제어할 때 사용한다.  
원하는 카드의 버튼을 눌러서 선택한다.



### 6-1. LED 테스트

3,2,1,0 등의 버튼을 누르면 해당 LED 가 점등한다.

### 6-2. DIP SW 입력 테스트

RD 버튼을 누르면, DIP SW 의 상태가 비트별로 체크 된다.

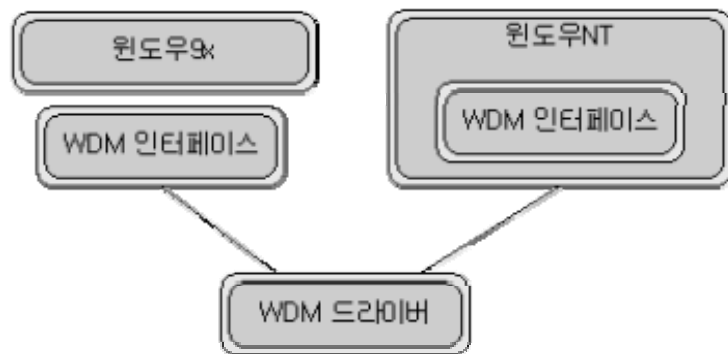
### 6-3. Interrupt Test

데모 보드에 있는 인터럽트 발생 푸시버튼 스위치를 누르면 그때마다, 프로그램이 인터럽트를 인식하여 인터럽트가 걸린 횟수를 표시하는 기능. SET 버튼을 누르면 인터럽트를 작동할 수 있게 되며, 이때부터 데모 보드에 있는 푸시 버튼을 누르면 눌린 회수를 표시하게 된다. 인터럽트 버튼이 2개 있는데 각각 따로 인터럽트 걸린 횟수를 계수한다. CLR 버튼을 누르면 인터럽트가 작동하지 않게 된다.

## 7. WDM 디바이스 드라이버 설명

### 7-1 WDM 드라이버

WDM은 윈도우 OS 버전간의 차이에 고민할 필요 없이 컴파일만 새로 하면 각각의 OS에 적용할 수 있는 윈도우 2000, XP, 윈도우 9x, 미래의 윈도우 OS에 적용되는 편리한 디바이스 드라이버 모델이다. 플러그 앤 플레이를 지원 하기 때문에 PC의 BIOS가 자동으로 할당된 자원을 넘겨받아 사용하므로 개발자가 일일이 하드웨어 자원을 찾아 할당하는 번거로운 과정이 필요없다. WDM 드라이버는 기존 NT 드라이버를 기반으로 하는 실행 파일(sys 형태)로 작성되며 대부분의 소스 코드가 윈도우 NT와 비슷하므로 먼저 NT 드라이버 작성에 필요한 사항을 알고 있는 경우에는 유리하다.



<그림 > WDM 드라이버 모델

### 7-2 계층형 드라이버와 필터 드라이버

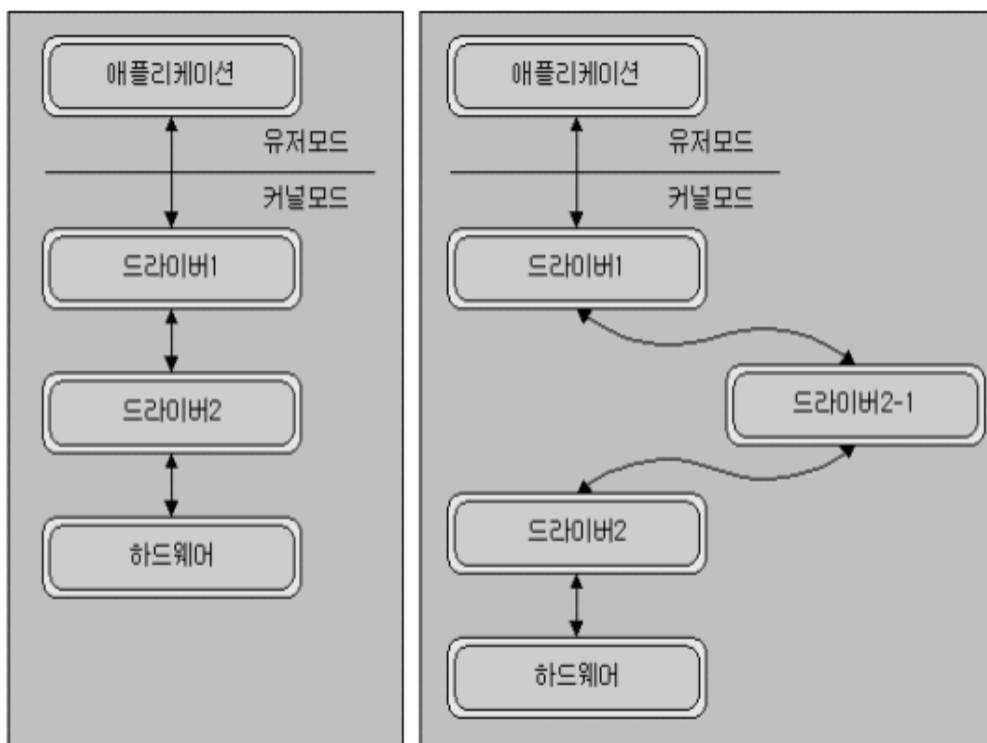
윈도우 NT에서는 Win32 기반의 사용자 모드와 커널 모드로 구분되어 소프트웨어가 실행된다. 커널모드의 드라이버는 HAL이라는 인터페이스를 통해 실제 디바이스와 통신한다.





<그림 > NT 드라이버 계층도

계층형 구조를 가진 윈도우는 특정 디바이스에 대한 드라이버를 하드웨어에 근접한 순으로 여러 층에 걸쳐 구성하고 상위 드라이버가 애플리케이션에서 입출력 요청을 받아 차례대로 하위 드라이버에 처리 요청을 전달한다. 필터 드라이버는 계층형 드라이버 중간에 드라이버를 삽입시켜 이 드라이버가 입출력 요구를 처리하도록 만든 구조이다. WDM 드라이버가 필터 드라이버에 해당한다.



## 7-2. WDM 드라이버의 컴파일

각 윈도우 버전에 따라 WDM Device Driver 소스는 동일하나 각 OS환경에서 컴파일 하여야 한다. 방법은 다음과 같다.

### 7-3-1 Windows 98

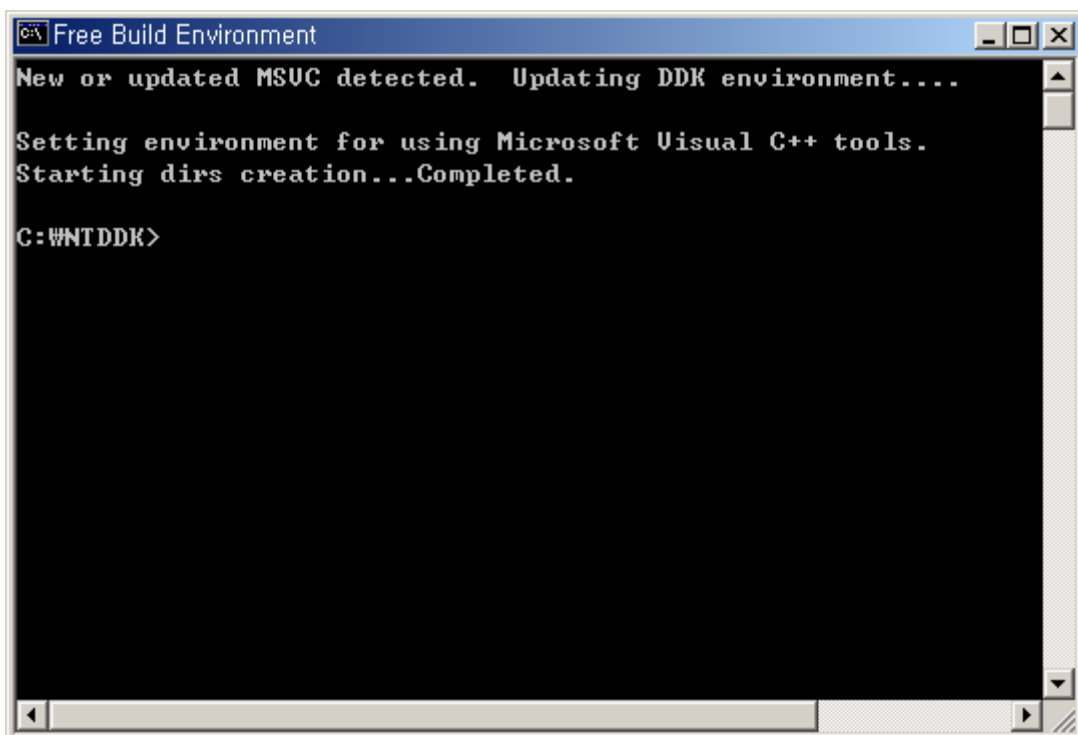
VC++ 6.0, 98DDK를 설치한후에 98ddk의 Free Build 도스창으로 간후에 wdm의 경로에서 run.bat를 실행한다.

### 7-3-2 Windows ME

VC++ 6.0, Win 2000 DDK를 설치한후에 2000 ddk의 Free Build 도스창으로 간후에 wdm\_nt의 경로에서 run.bat를 실행한다.

### 7-3-3 Windows 2000

VC++ 6.0, Win 2000 DDK를 설치한후에 2000 ddk의 Free Build 도스창으로 간후에 wdm 경로에서 run.bat를 실행한다.



### 7-3-4 Windows XP

VC++ 6.0, Win XP DDK를 설치한후에 XP ddk의 Free Build 도스창으로 간후에 wdm 경로에서 run.bat를 실행한다.