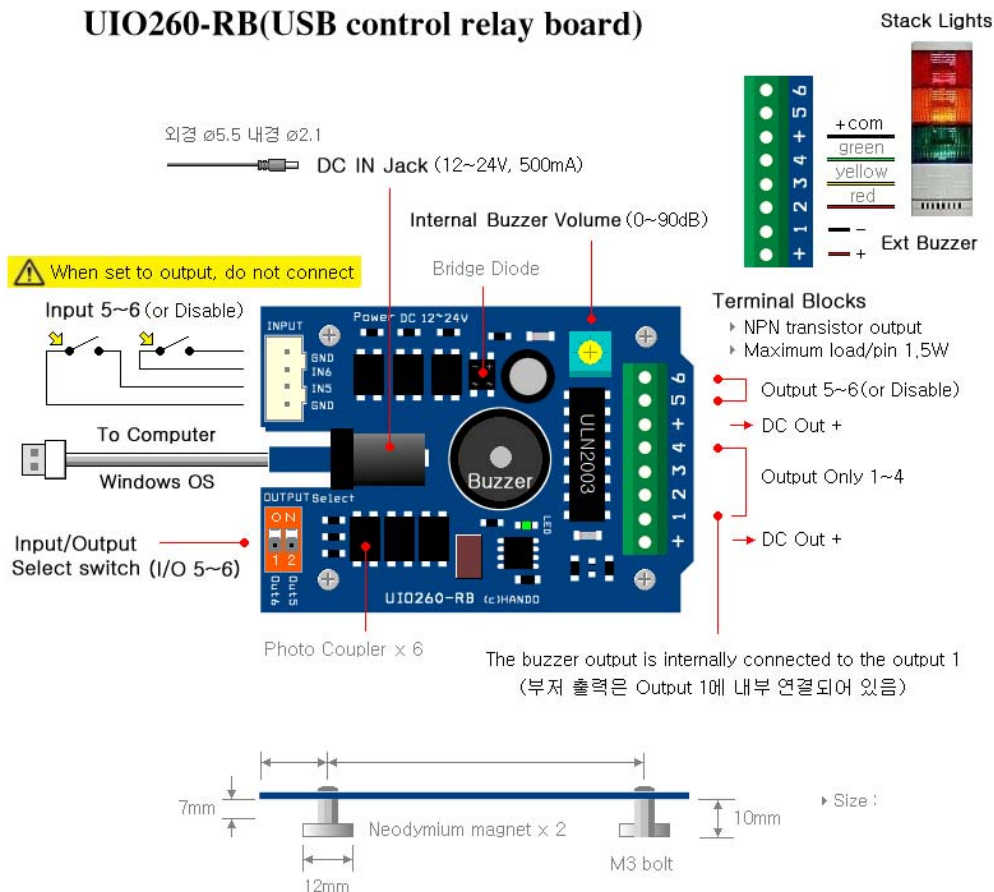


> 제품 상세설명

- 입출력 포트: 출력 전용 포트 4개, 입력 또는 출력 포트 2개
- 출력 포트는 포토커플러에 의해 회로 차단되어 있으며, 24V 외부어댑터 전압으로 구동되는 1W 정도의 LED를 제어할 수 있습니다.
- 입력 포트 사용시 7K옴 풀업되어 있으며, 입력 변화 후 0.005초 이상 신호가 안정되면 입력 인터럽트가 발생(일반 on/off 스위치 직접연결 가능)
- USB Product ID 변경하면 한 대의 컴퓨터에 9개까지 모듈 사용가능
- 운영 체제 : 출력 사용시 Win98SE ~ Window10 x32, x64 지원(입력 사용시 WinXP 이상의 운영체제 필요)
- 코드길이: 1.5m (전선규격 28AWG 6C)

UIO260-RB(USB control relay board)



주의 : 출력포트(push/pull)로 설정된 상태에서는 데이터 선이 DC Out+선에 직접 연결되지 않도록 주의하시기 바랍니다. (고장시 A/S불가)

> USB 입출력장치의 사용 전에 필요한 개념

1. UIO260은 TTL레벨의 기본 입출력 모듈로서 여기에 포토커플러와 모터 드라이버 IC(ULN2003)로 구성된 보드와 결합해서 판매되는 제품입니다.
2. USB 컨넥터에는 조임 나사가 없어 연결선이 빠질 가능성이 있고 USB 통신은 PC가 일대다수의 통신을 하기 때문에 오동작하는 다른 USB 장치가 동시에 연결되면 USB 통신 전체가 장애를 받을 수 있습니다. 따라서 USB 모듈을 통해 위험한 모터 기기를 작동하는 것은 삼가해 주기 바랍니다.
3. USB Product ID란? USB입출력 모듈은 ID를 달리하면 9개를 동시에 연결이 가능하며, 각각 제어가 가능합니다. ID는 사용자가 직접 변경할 수 있습니다.
UIO260은 16진수 261~269 중의 장치 설정 가능 합니다.(출고시 261로 설정)

> 입출력 설정

공장 출고시 기본 입출력 설정은 다음과 같으며, 예제 응용프로그램으로 제어하는 테스트를 완료하여 개념 파악이 된 후에 1번~7번의 작업을 수행하십시오.

<UIO260-RB 모델의 기본설정 상태>

USB Product ID = 0x261

출력포트 (1번, 검정색) - push/pull 출력 1번

출력포트 (2번, 빨강색) - push/pull 출력 2번

출력포트 (3번, 노란색) - push/pull 출력 3번

출력포트 (4번, 녹색) - push/pull 출력 4번

입력포트 (5번, 파란색) - input 입력(bit3)

입력포트 (6번, 흰색) - input 입력(bit2)

1. USB 입출력 모듈을 PC의 USB 포트에 연결하고 녹색 LED가 점등되는 것을 확인한다.
2. 설정 변경 프로그램 `usbmodule.exe`을 실행한다.
<프로그램 다운로드: 홈페이지->한도개발품->USB 입출력모듈>
3. 프로그램 메뉴 'USB>ID Change(261->260)'을 선택하면 설정 모드로 전환됩니다.
4. 프로그램 메뉴 'Data>Read(USB->PC)'를 선택하여 현재의 설정 상태를 읽어 봅니다.
5. 1~4번 포트는 출력 전용이고, 5~6번은 입력 또는 출력(Input, push/full) 설정합니다.
입출력 설정 변경 후 보드에 있는 DIP 스위치도 입출력에 맞게 설정해야 됩니다.
6. 프로그램 메뉴 'Data>Write(PC->USB)'를 선택하여 설정 값을 모듈로 전송합니다.
7. 프로그램 메뉴 'USB>Reset'을 선택하거나 장치를 다시 연결하면 설정된 Product ID의 입출력 모듈로 동작합니다.

> 배선연결

- 출력포트 사용시: 어댑터 공급 전압(1W 이하)의 제어기기를 구동할 수 있습니다.
출력에 해당하는 터미널에는 구동기기의 -를 연결하고,
DC Out 공통 단자에는 구동기기의 +를 연결합니다.
- 입력포트 사용시: 데이터 선은 5V 7.5K옴으로 풀업되어 있으므로 스위치를 연결할 경우 데이터선과 GND 선을 직접 연결시키면 됩니다.
- 입력포트 사용시: 다른 기기의 센서 출력을 입력으로 받으려면 포토커플러를 거쳐서 입력 받도록 설계해야 합니다.

> 응용 프로그램 사용법

- 제공되는 MFC 또는 C#으로 된 예제를 이용하여 입출력 테스트는 프로그램은 다소 복잡하지만 호출 함수는 4가지 정도이므로 귀사의 응용 프로그램에 쉽게 접목할 수 있습니다. 예제의 설명을 참조하기 바랍니다.

USB 모듈 연결확인 함수: usb_io_init(ProductID);

결과값: 0=장치없음, n=연결된 장치번호

모듈 출력 초기화 함수: usb_io_reset(ProductID);

출력 포트 상태를 초기화시킨다.

결과값: False=장치없음, True=명령어 전송 완료

입력상태 확인 함수: usb_in_requet(ProductID);

결과값: False=장치없음, True=명령어 전송 완료

입력 상태 값은 리턴 값으로 받지 못하고 WM_INPUT 인터럽트를 통해서 받는다.

장치 출력제어 함수: usb_io_output(ProductID, Cmd, PortNo, PortNo, PortNo, PortNo);

출력포트에 원하는 on/off/blink 상태로 만드는 함수로 1~4개의 포트를 동시에 제어하는 함수입니다.

- on/off 명령

출력 포트를 on 또는 off시킨다.

cmd=0

Ex)usb_io_output(0x261 0, 1, 0, 0 , 0);

0x261장치의 출력포트 1번은 High로 만드는 예

Ex)usb_io_output(0x261, 0, -2, 0, 0, 0);

0x261 장치의 출력포트 2번은 Low로 만드는 예

- blink 명령(on/off 50%)

출력 포트를 cmd 숫자값*0.2초의 주기로 점멸시킨다.

cmd=1~15

Ex)usb_io_output(0x261, 1, 1, 0, 0, 0);

0x261 장치의 출력포트 1번을 0.2초 주기로 on/off를 반복한다.

- blink 명령(on/off 50% 및 시간동기)

출력 포트를 cmd 숫자값*0.2초의 주기로 점멸시키는데, 동시에 점멸하거나 반전되는 점멸을 하고 싶을 때 PortNo 4개까지 동기가 가능하다.

cmd=1~15

Ex)usb_io_output(0x261, 5, 1, 2, 0, 0);

0x261 장치의 출력포트 1번과 2번이 1.0초 주기로 on/off를 반복하는데, 1번과 2번이 동시에 동작한다.

Ex)usb_io_output(0x261, 5, 1, -2, 0, 0);

0x261 장치의 출력포트 1번과 2번이 1.0초 주기로 on/off를 반복하는데, 1번과 2번의 on/off가 서로 반전되도록 동작한다.

- blink 명령(on/off 시간을 다르게)

출력 포트를 cmd 상위4비트 숫자값*0.1초는 High

cmd 하위4비트 숫자값*0.1초는 Low를 반복한다.

cmd=0x11~0xFF

Ex)usb_io_output(0x261, 0xA5, 1, 0, 0, 0);

0x261 장치의 출력포트 1번을 1.0초 on, 0.5초 off 주기로 on/off를 반복한다.

- 1회 펄스 명령

출력 포트를 cmd 상위4비트 숫자값*0.1초 동안 1회만 High 또는 Low 신호를 줄 때 편리하게 사용할 수 있다.

cmd=0x10~0xF0 (하위 4비트는 반드시 0)

Ex)usb_io_output(0x26, 0x50, 1, 0, 0, 0);

0x26 장치의 출력포트 1번을 0.5초 동안 High로 후에 Low 상태를 유지

Ex)usb_io_output(0x261, 0xA0, -1, 0, 0, 0);

0x261 장치의 출력포트 1번을 1.0초 동안 Low로 후에 High 상태를 유지

> 입력포트 상태값 가져오기

입력 값에 변화가 생기면 WM_INPUT 메시지를 통해서 입력 포트의 상태 값을 얻을 수 있습니다. 입력 포트를 사용할 경우 윈도우XP이상의 운영체제에서 실행해야 됩니다.

1. 윈도우 메시지를 얻기 위해서는 프로그램 실행 초기에 메시지가 발생하도록 설정하는 함수 `set_usb_events();`를 호출하고
2. 이후에는 WM_INPUT 이벤트가 왔을 때 그 값을 분석 해주는 함수 `get_usb_input();`를 호출하면
3. 버튼 상태 값이 포함된 배열변수 `usb_input` 에서 확인이 가능합니다.

```
typedef struct usb_input
{
    int ProductID;           // USB 장치번호
    unsigned char Status;   // 0= 입력변화에 의해, 1=재전송 요구에 의해
    unsigned char Button;   // I/O포트의 입력값(입력포트만 표시)
    unsigned char Output;   // I/O포트의 출력값(출력포트만 표시)
    unsigned char Mask;     // I/O포트의 설정상태(비트0=출력, 비트1=입력)
} USB_INPUT;
```