

- 날짜 : 2014년 3월 3일
- 제목 : UART를 이용한 Register 및 변수설정 Interface
- 문서번호 : KR_ES_0215
- 작성자 : 신기원 (kenny.shin@microchip.com , FAE)

< Engineering Issue 내용 >

1. 관련 Devices : All MCU

2. 개요

- UART를 이용한 MCU내부의 Variable 이나 Register를 변경하는 Interface를 소개한다. Pickit3나 ICD3와 같은 개발 Tool을 이용하여 Register나 변수를 변경할 수도 있지만 실시간으로 변경하여 변화에 따른 동작을 확인하기에 어려움이 있다. 특히 Power와 같은 Application에서 MCU의 동작이 멈추게 되면 주변 H/W의 오동작이 발생할 할 수 있다.

3. Engineering Issue

3.1. Issue

- UART Interface를 이용한 Register / Variable 값 변경 Interface를 소개한다.

4. 해결방안

예제 코드는 첨부파일로 첨부한다.아래와 같은 예로 extern 변수를 선언할 수 있다.

예제 프로그램에는

- main.c
- uart.ch
- user.c

파일 3개를 사용하여 예제 코드를 소개한다.

- UART Init

// User.c

////////////////////////////////////

// UART

////////////////////////////////////

RXPPS = 0b01101; //UART RX as RB5 pin

RB6PPS = 0b10100; //UART TX as RB6 pin

SP1BRG = (32000000L/(115200*4)) - 1;

TX1STAbits.BRGH = 1;

BAUD1CONbits.BRG16 = 1;

```
TX1STAbits.TXEN = 1;  
RC1STAbits.CREN = 1;  
RC1STAbits.SPEN = 1;
```

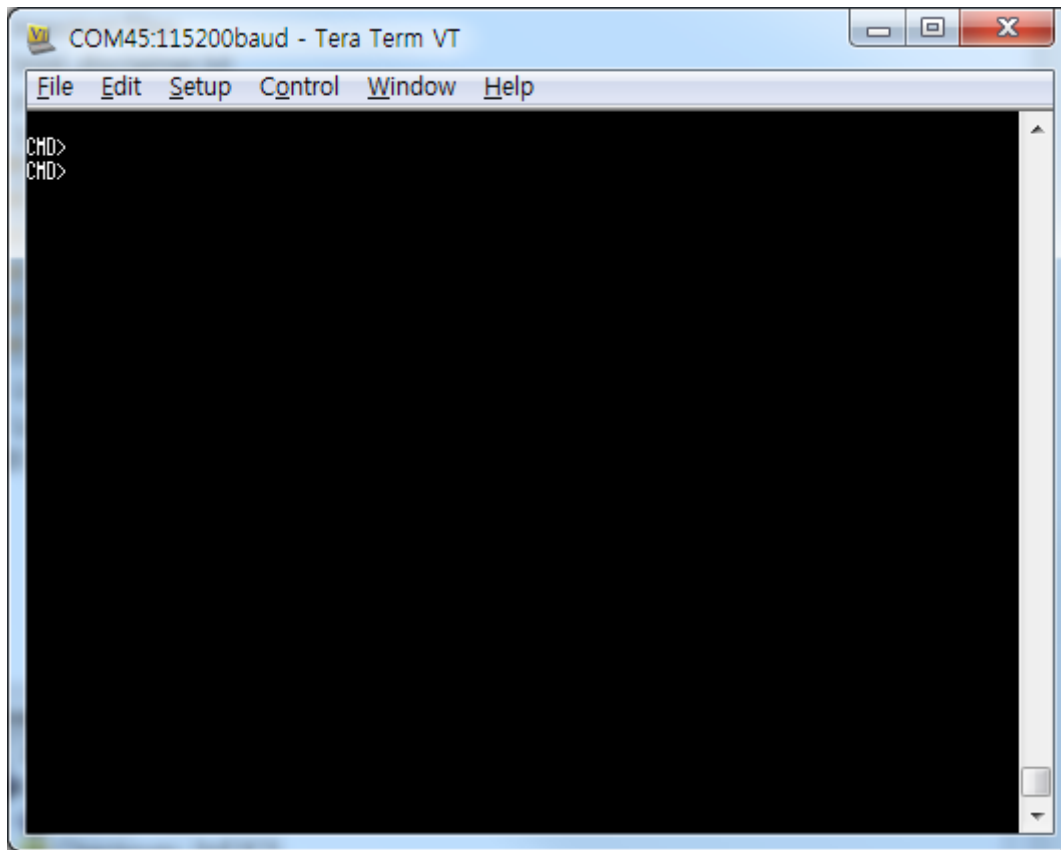
- Main문 사용 예제

```
// main.c  
  
while(1)  
{  
    /* TODO <INSERT USER APPLICATION CODE HERE> */  
    UartReceiveFuncion();  
}
```

Main문에서 UartReceiveFunction(); 함수가 주기적으로 실행되도록 한다. 이 함수의 내용에서 사용하는 함수는 uart.c에 모두 포함한다.

예제 코드는 PIC16F1708로 구성되어 있다. 하지만 간단한 UAR Peripheral 포트 설정만으로 모든 MCU에서 사용할 수 있다. Baudrate는 115200을 사용한다.

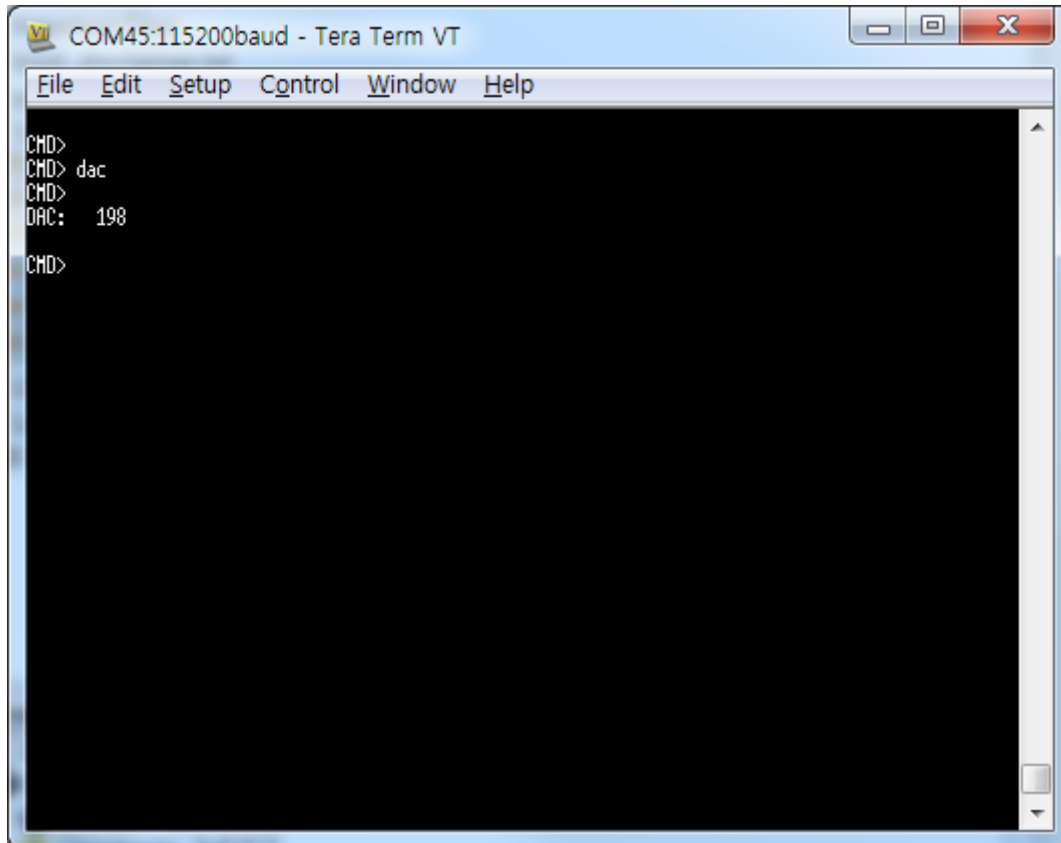
MCU에 Firmware가 download되고 UART가 정상적으로 PC와 연결된 후 Enter키를 입력하면 아래와 같은 화면을 확인 할 수 있다. 명령어를 입력 받기 위해 “CMD> “와 같은 프롬프트형태가 나타난다.



예제 코드에서는 MCU내부의 DAC1CON1값을 변경하고 읽는 방법에 대해 소개한다.

- DAC값 읽기

dac를 입력 후 Enter키를 입력하면 현재의 dac값을 읽을 수 있다.

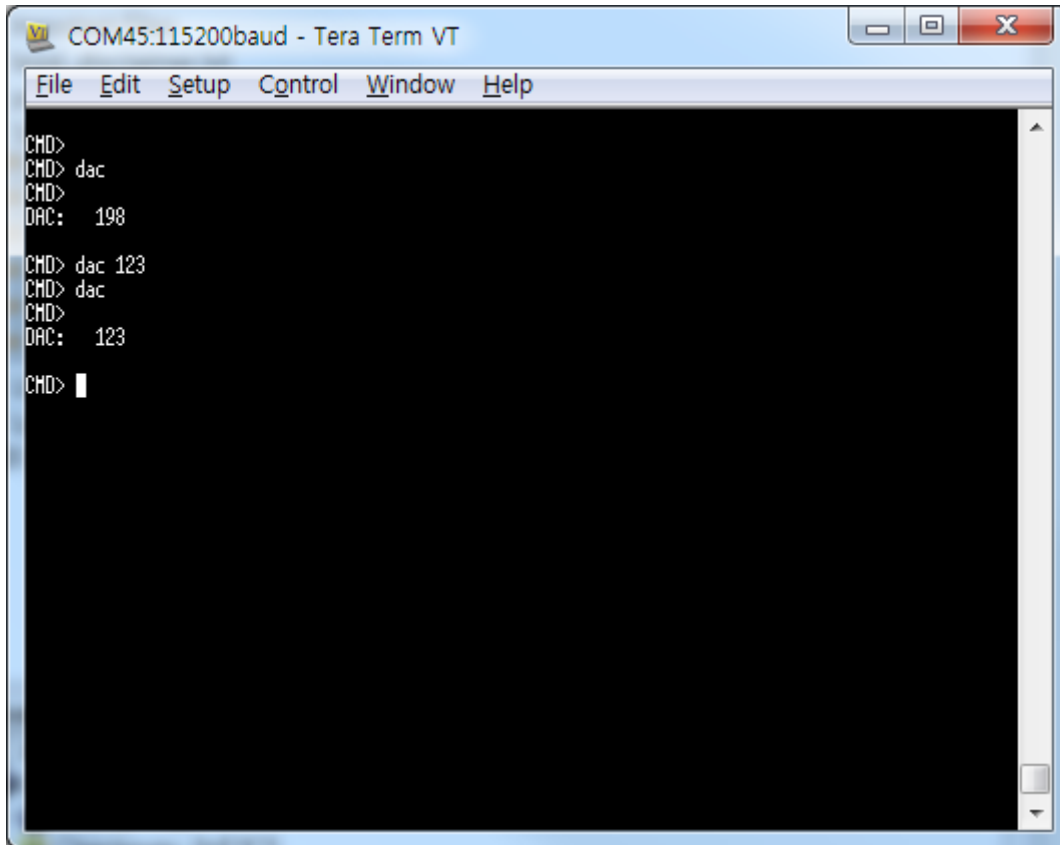


The screenshot shows a Tera Term VT terminal window titled "COM45:115200baud - Tera Term VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The terminal text shows a command prompt "CMD>" followed by the input "dac", which results in the output "DAC: 198".

```
CMD>  
CMD> dac  
CMD>  
DAC: 198  
CMD>
```

- DAC값 쓰기(변경)

dac를 입력 후 “Space bar”를 입력 후 원하는 값을 10진수로 입력한 다음 Enter키를 입력하면 User가 원하는 값이 Register에 설정된다. 원하는 값이 정상적으로 입력되었는지 확인하는 방법은 dac입력 후 Enter키를 입력하면 확인할 수 있다.



- Register값의 읽기 쓰기 확장

아래 내용은 dac 읽기/쓰기에 대한 코드이며 uart.c “void Command Decoder()”에 정의되어 있다. 명령어는 Space를 구분하며 각 space 단위로 cmd[0], cmd[1] 과 같이 space 를 구분하여 각각 저장된다. 이 cmd를 단위로 명령어를 비교하여 처리한다.

```

////////////////////////////////////
// WRITE [addr] [value]
else if( !Strcmp(cmd[0], (unsigned char *)"dac") )
{
    if(Strlen(cmd[1]) == 0){
        UART_puts("\r\nDAC: ");
        UART_putdec( DAC1CON1, 5 );
        UART_puts("\r\n");
        UART_puts("\r\nCMD> ");
    }else{

```

```
DAC1CON1 = StrToDec(cmd[1]);
```

```
}
```

```
}
```

압출 파일명: PIC16F1708_UartInterface.zip
