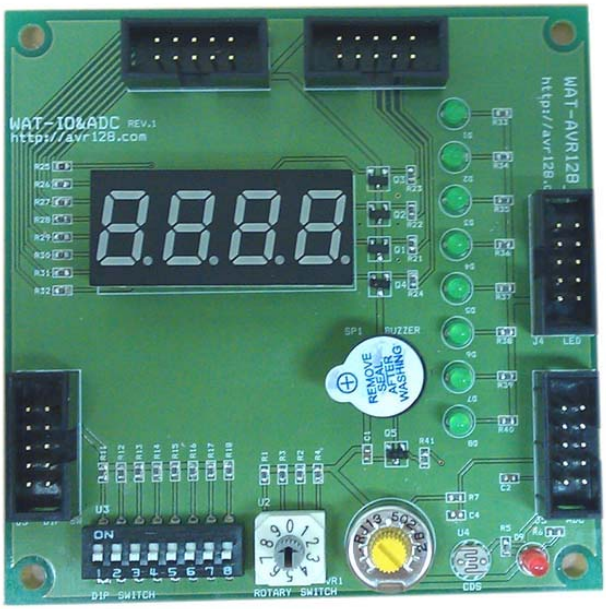
WAT-IO&ADC

IO실험 모듈



WhiteAT

HOMEPAGE: http://whiteat.com
E-MAIL: whiteat@whiteat.com
TEL: 070-4412-5754

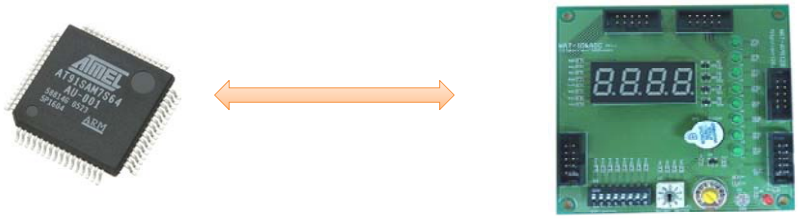
1. 제품 소개

WAT-IO&ADC 모듈은 AVR, Cortex-M3, PIC 등의 MCU와 10P Flat 케이블을 연결하여 Input/Output을 실험할 수 있는 모듈입니다. FND, LED, BUZZER 을 제어할 수 있으며 가변저항, CDS(빛의 밝기), 딥스위치, 로터리 스위치의 값을 읽을 수 있는 제품입니다.

1. 특징

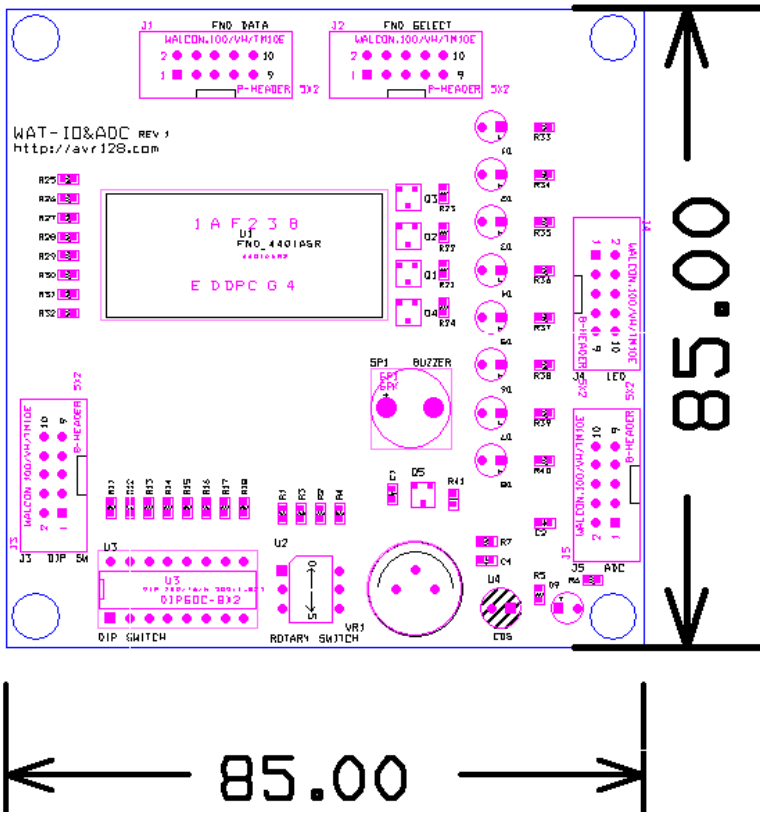
항목	내용
FND	4 digit
LED	8 port
BUZZER	1 port (5V)
가변저항	5K
CDS	CDS
딥 스위치	8 channel
로터리 스위치	10R
크기	85 mm x 85 mm
인터페이스	Digital Input/Output

2. 구조

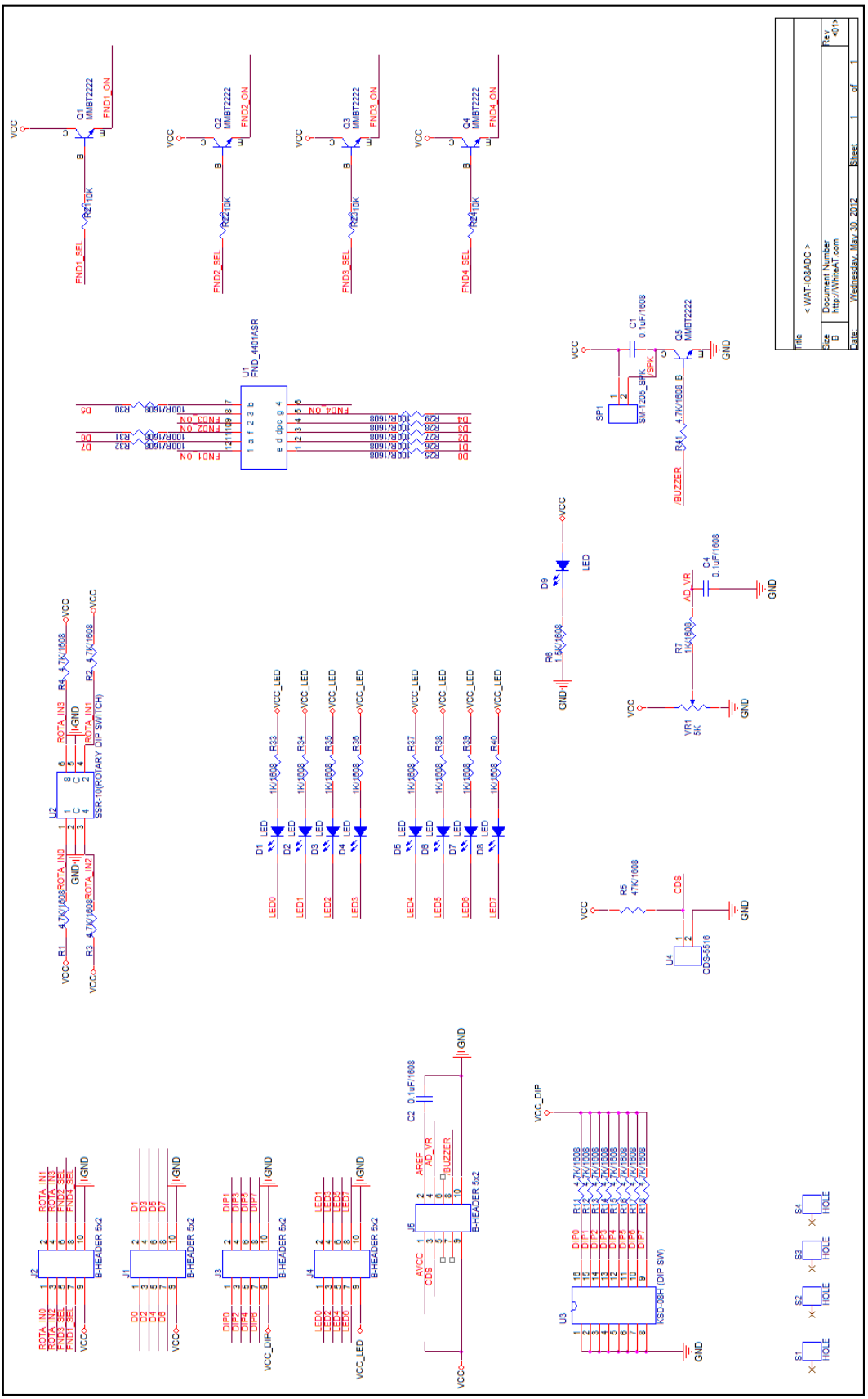


MCU 의 Digital 신호(일반 IO)와 ADC로 연결할 수 있습니다.

3. **외형**

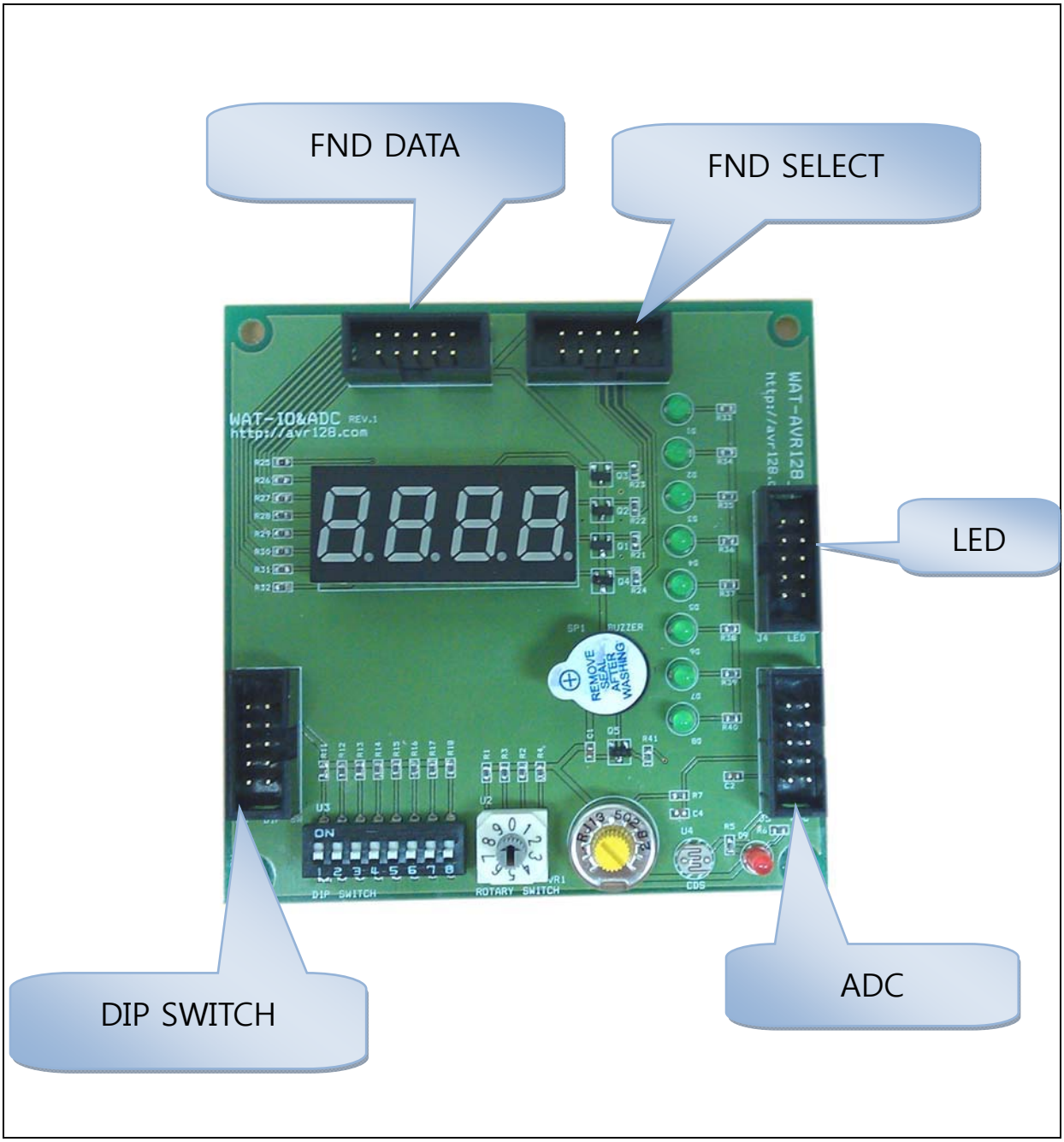

WAT-IO&ADC REV 1
http://avr128.com
FND_DATA
FND_SELECT
WALCON.100/VH/TM10E
B-HEADER
FND_4401ASR
SP1 BUZZER
LED
J3 DIP SW
DIP SWITCH
ROTARY SWITCH
VR1
CDS
J5 ADC
85.00
85.00

4. 회로도



2. 외부 인터페이스

1. 커넥터



번호	핀명	설명
1	D0	DATA (bit 0)
2	D1	DATA (bit 1)
3	D2	DATA (bit 2)
4	D3	DATA (bit 3)
5	D4	DATA (bit 4)
6	D5	DATA (bit 5)
7	D6	DATA (bit 6)
8	D7	DATA (bit 7)
9	VCC	전원 (DC 5V)
10	GND	그라운드

< FND DATA >

번호	핀명	설명
1	RO0	로터리 스위치 핀 0
2	RO1	로터리 스위치 핀 1
3	RO2	로터리 스위치 핀 2
4	RO3	로터리 스위치 핀 3
5	FND0	FND 0 선택 핀
6	FND1	FND 1 선택 핀
7	FND2	FND 2 선택 핀
8	FND3	FND 3 선택 핀
9	VCC	전원 (DC 5V)
10	GND	그라운드

< FND SELECT, ROTARY SWITCH >

번호	핀명	설명
1	LED0	LED 0 제어 핀
2	LED1	LED 1 제어 핀
3	LED2	LED 2 제어 핀
4	LED3	LED 3 제어 핀
5	LED4	LED 4 제어 핀
6	LED5	LED 5 제어 핀
7	LED6	LED 6 제어 핀
8	LED7	LED 7 제어 핀
9	VCC	전원 (DC 5V)
10	GND	그라운드

< LED >

번호	핀명	설명
1	AVCC	ADC 컨버터용 전압
2	AREF	ADC 컨버터용 기준 전압
3	CDS	CDS 센서 연결 핀
4	VR	가변저항 연결 핀
5		
6		
7		
8	BUZ	부저 연결 핀
9	VCC	전원 (DC 5V)
10	GND	그라운드

< ADC >

번호	핀명	설명
1	DIP0	딥스위치 1번 핀
2	DIP1	딥스위치 2번 핀
3	DIP2	딥스위치 3번 핀
4	DIP3	딥스위치 4번 핀
5	DIP4	딥스위치 5번 핀
6	DIP5	딥스위치 6번 핀
7	DIP6	딥스위치 7번 핀
8	DIP7	딥스위치 8번 핀
9	VCC	전원 (DC 5V)
10	GND	그라운드

< DIP SWITCH >

2. ATMEGA128 모듈에 연결

WAT-AVR128 모듈의 PORTA는 LED, PORTB는 DIP SWITCH, PORTC는 FND SELECT, PORTE는 FND DATA, PORTE는 ADC 에 연결하여 PC 프로그램에서 제어 및 상태를 실시간으로 모니터링 하는 예제입니다.

WAT-AVR128 모듈	WAT-IO&ADC 모듈
PORTA	FND DATA
PORTB	
PORTC	FND SELECT, ROTARY SWITCH
PORTD	DIP SWITCH
PORTE	LED
PORTF	ADC

ATEMGA128 펌웨어 코드

```
  /*
        EX_09_03.c

        USART0 로보드의상태PC로전송
        PC에서LED, BUZZER, FND 제어
        AVRStudio  4.18
*/

#include <avr/io.h>
#include "WAT128.h"

BYTE g_FNDData[4]={1,2,3,4};
BYTE g_BUZZER = 0;
BYTE g_LED = 0;

UINT16 g_adcCDS;          // CDS 값보관
UINT16 g_adcVR;// 가변저항값보관

void OperDisplayFND()
{
```

```
        DisplayFND4(g_FNDData[0],g_FNDData[1],g_FNDData[2],g_FNDData[3]);
}

INT16 g_byteOperPCTXTimer = 0;
void OperPCTX()
{
        if(--g_byteOperPCTXTimer>0)
                return;

        PutChar0(0x02);                                         // 0 시작신호
        PutChar0(PIND);                                         // 1 DIP SWITCH
        PutChar0(GetRotaryInt());                       // 2 ROTARY
        PutChar0(g_adcCDS>>8);                          // 3 CDS 상위값
        PutChar0(g_adcCDS);                             // 4 CDS 하위값
        PutChar0((g_adcVR)>>8 &0xFF); // 5 가변저항상위값
        PutChar0(g_adcVR&0xFF);                         // 6 가변저항하위값
        PutChar0(0x03); // 7
        PutChar0(0xCC);                                         // 8 체크섬
        PutChar0(0x03);                                         // 9 끝신호

        g_byteOperPCTXTimer = 20;

}

// 가변저항, CDS 값을ADC로읽기
void OperReadADC()
{
        INT16 uiTemp; // 임시변수

        g_adcCDS = 0;
        // 노이즈를생각해서값을번읽어평균을낸다.

        for(uiTemp = 0; uiTemp<16;uiTemp++)
        {
                ADMUX=0x40 | 0x00;
                ADCSRA = 0xD7;
                while((ADCSRA & 0x10) != 0X10);
                g_adcCDS += ADCL + (ADCH*256);
        }
```

```
	g_adcCDS>>=4;

	g_adcVR = 0;
	// 노이즈를생각해서값을번읽어평균을낸다.

	for(uiTemp = 0; uiTemp<16;uiTemp++)
	{
		ADMUX=0x40 | 0x01;
		ADCSRA = 0xD7;
		while((ADCSRA & 0x10) != 0X10);
		g_adcVR += ADCL + (ADCH*256);
	}

	g_adcVR>>=4;
}

int main()
{

	BUZZER_INIT; // BUZZER 초기화
	OpenSCI0(115200);		// USART 0 열기

	InitFND4(); // FND 초기화

	InitADC();		// ADC 초기화

	DisplayFND4(3,4,5,6);
	InitRotary();

	DDRD = 0x00; // 딥스위치를입력으로설정
	DDRE = 0xFE; // LED


	while(1)
	{
		BUZZER_OFF;
		OperDisplayFND();
		OperReadADC();
```

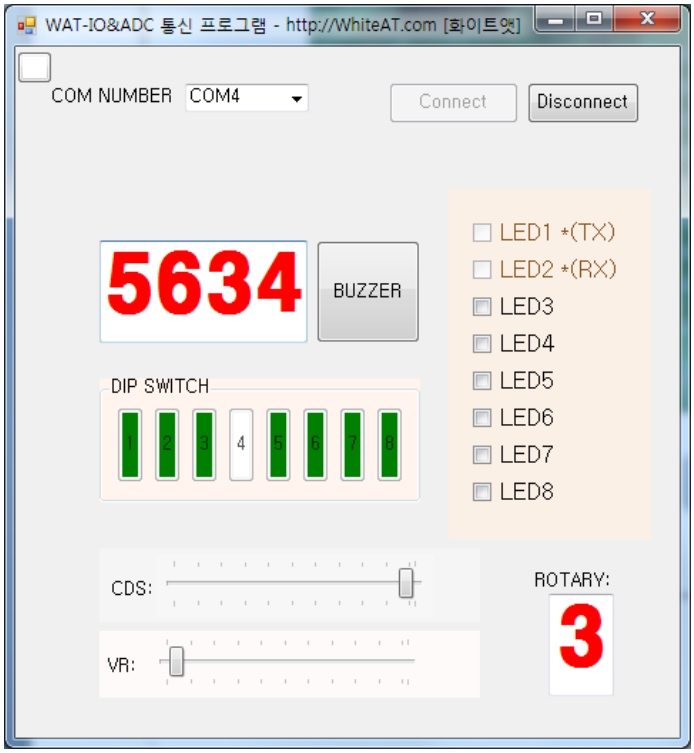
```
                OperPCTX();

                if(0x02 ==GetByte0())
                {
                        UINT16 uiData = 0;
                        uiData = GetByte0(); //1
                        uiData<<=8;
                        uiData += GetByte0(); //2

                        //buzzer
                        g_BUZZER = GetByte0();


                        g_LED = GetByte0(); //4
                        GetByte0(); //5
                        GetByte0(); //6
                        GetByte0(); //7
                        if(0xCC == GetByte0() && 0x03 == GetByte0() )
                        {
                                if( g_BUZZER)
                                {
                                        BUZZER_ON;
                                        DelayMS(2);
                                }

                                // PC에서받은LED 값을출력
                                PORTE =~((g_LED)&0xFC);

                                // FND 표시
                                g_FNDData[0]=((uiData/1000)%10);
                                g_FNDData[1]=((uiData/100)%10);
                                g_FNDData[2]=((uiData/10)%10);
                                g_FNDData[3]=(uiData%10);
                        }
                }
        }
}
```

윈도우 코드(통신 처리 부분)

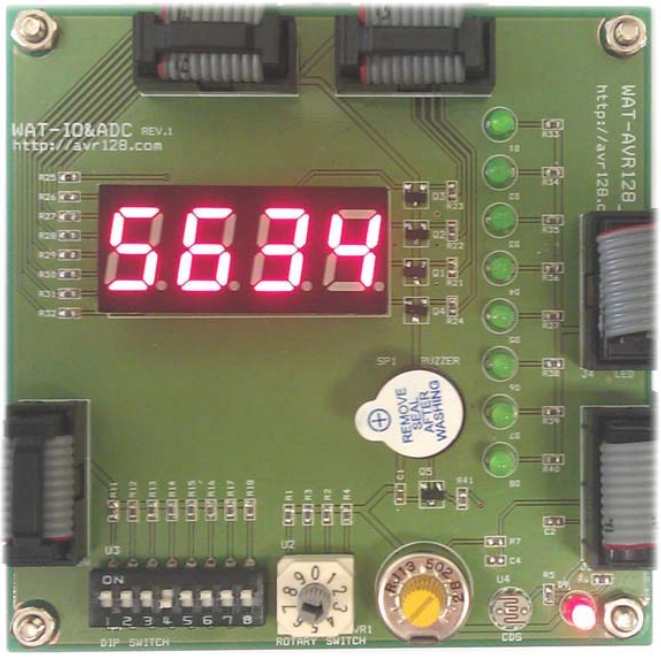
```
private void tmrRxData_Tick(object sender, EventArgs e)
        {
            btnRxSignal.BackColor = Color.White;


            if (null == m_serialPort) return;
            if (!m_serialPort.IsOpen) return;

            int iRecSize = m_serialPort.BytesToRead; // 수신된 데이터 갯수


            // AVR에서 10바이트식 보내는데 10바이트 이상이 들어 왔는지 체크
            if (iRecSize >= 10)
            {
                byte[] buff = new byte[iRecSize];    // 임시 변수

                // 시리얼 포트에서 데이터를 가져오자.
                m_serialPort.Read(buff, 0, iRecSize);

                // 맨앞의 값이 0x02 인지 체크
                if (buff[0] !=0x02   )
                {
                    for(int i=0;i<iRecSize-1;i++)
                    {
                        buff[i] = buff[i+1];
                    }
                }


                if (buff[0] == 0x02 && buff[8] == 0xCC && buff[9] == 0x03)
                {
                    this.txbRotaryValue.Text = buff[2].ToString();

                    this.btnDIP1.BackColor = ((buff[1] & 0x01) == 0x01) ? Color.White : Color.Green;
                    this.btnDIP2.BackColor = ((buff[1] & 0x02) == 0x02) ? Color.White : Color.Green;
                    this.btnDIP3.BackColor = ((buff[1] & 0x04) == 0x04) ? Color.White : Color.Green;
                    this.btnDIP4.BackColor = ((buff[1] & 0x08) == 0x08) ? Color.White : Color.Green;
                    this.btnDIP5.BackColor = ((buff[1] & 0x10) == 0x10) ? Color.White : Color.Green;
                    this.btnDIP6.BackColor = ((buff[1] & 0x20) == 0x20) ? Color.White : Color.Green;
                    this.btnDIP7.BackColor = ((buff[1] & 0x40) == 0x40) ? Color.White : Color.Green;
                    this.btnDIP8.BackColor = ((buff[1] & 0x80) == 0x80) ? Color.White : Color.Green;
```

```
                    this.trbCDS.Value = (Convert.ToInt32(buff[3]) << 8) + (Convert.ToInt32(buff[4]) << 0);
                    this.trbCDS.BackColor = Color.FromArgb((1024-this.trbCDS.Value) / 4, (1024-
this.trbCDS.Value) / 4, (1024-this.trbCDS.Value) / 4);
                    this.trbVR.Value = (Convert.ToInt32(buff[5]) << 8) + (Convert.ToInt32(buff[6]) << 0);



                    btnRxSignal.BackColor = Color.Green;
                }
            }

            // AVR 모드로 명령 전송
            this.SendToBoard();


        }
```

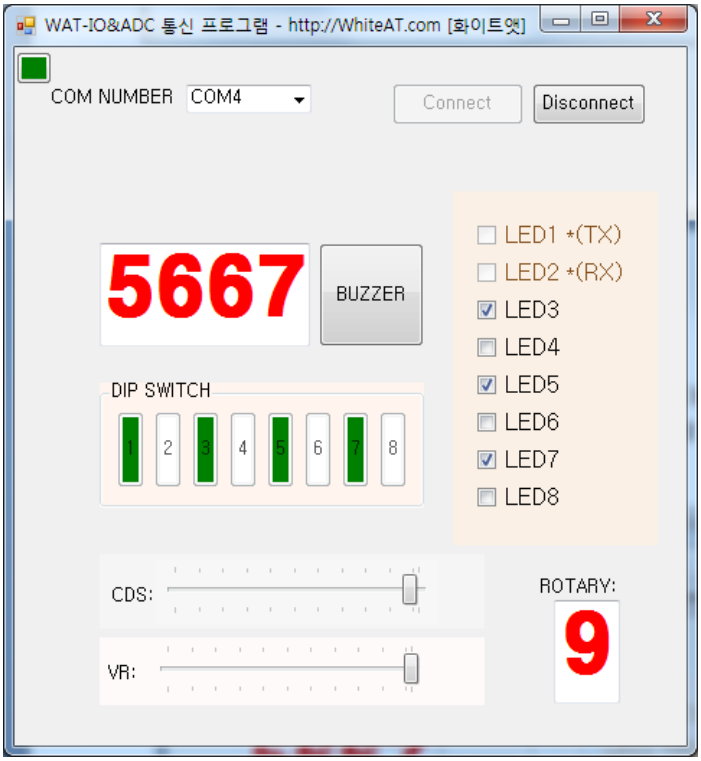
윈도우 프로그램에서 FND 에 5634, LED 모두 OFF 값을 전송하며 ATMEGA128 보드에서 딥 스위치 4 번 OFF, CDS 값 최대, 가변저항 값 최소, 로터리 스위치 값 3 을 전송되면 아래와 같이 됩니다.

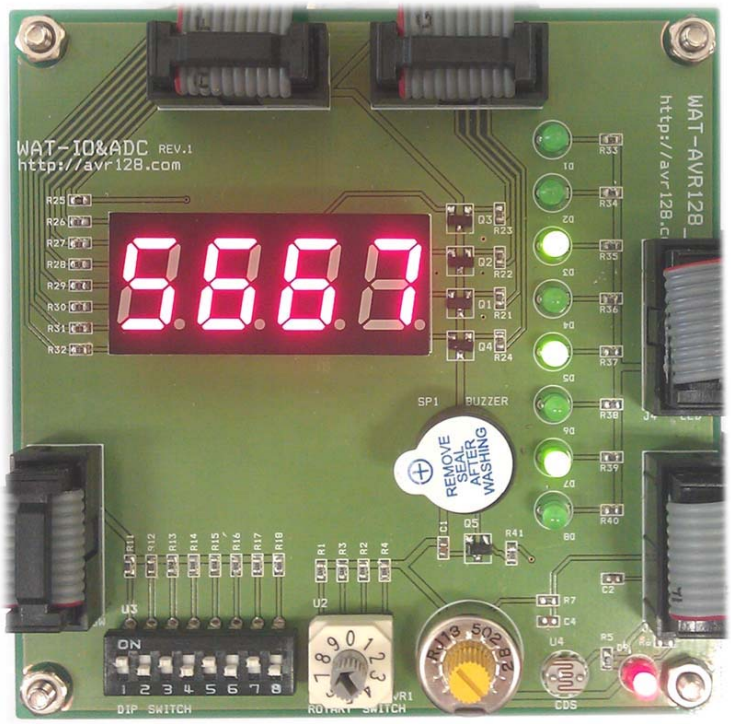


< WAT-IO&ADC에 연결한 윈도우 프로그램 >



윈도우 프로그램에서 FND 에 5667, LED 3,5,7 번 ON 값을 전송하며 ATMEGA128 보드에서 딥 스위치 1,3,5,7,번 ON, CDS 값 최대, 가변저항 값 최대, 로터리 스위치 값 9 을 전송되면 아래와 같이 됩니다.





모듈 구매하기: http://kit128.com/goods/view.php?seq=36

본 문서는 제품의 품질향상을 위해 사전 예고 없이 업데이트 될 수 있으며

제공되는 회로도와 소스 코드는 홈페이지 (http://WhiteAT.com) 에서 확인하실 수 있습니다.