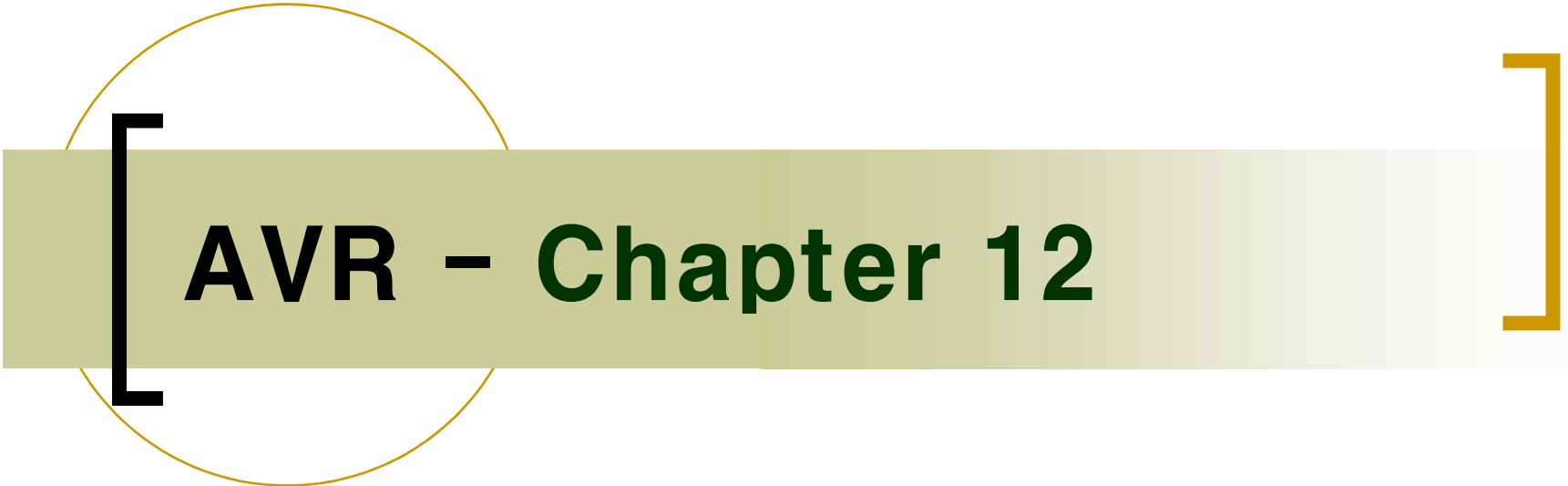




AVR - Chapter 12

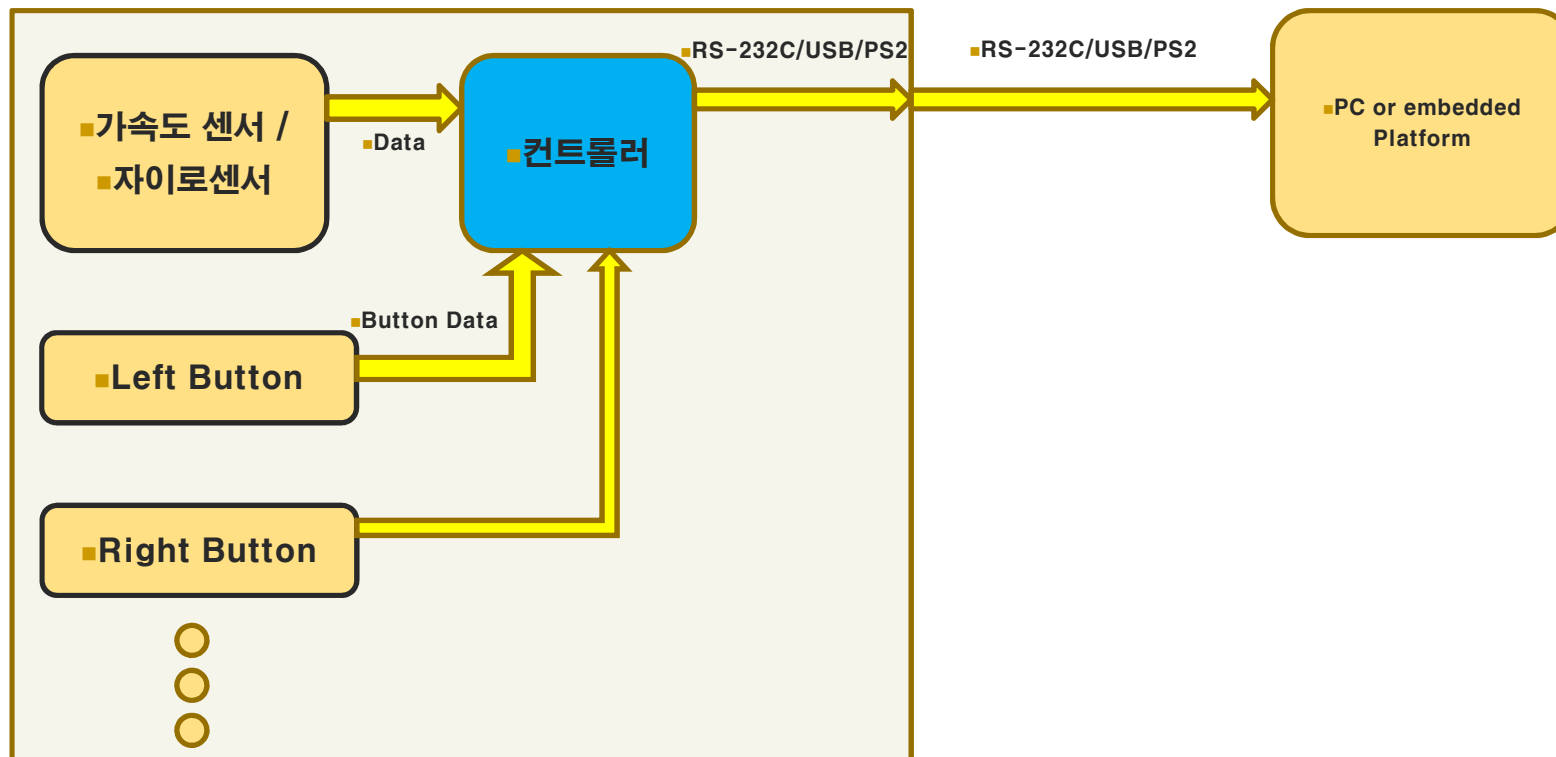
류대우

(주)뉴티씨(NEWTC) 대표이사

davidryu@newtc.co.kr

[Structure

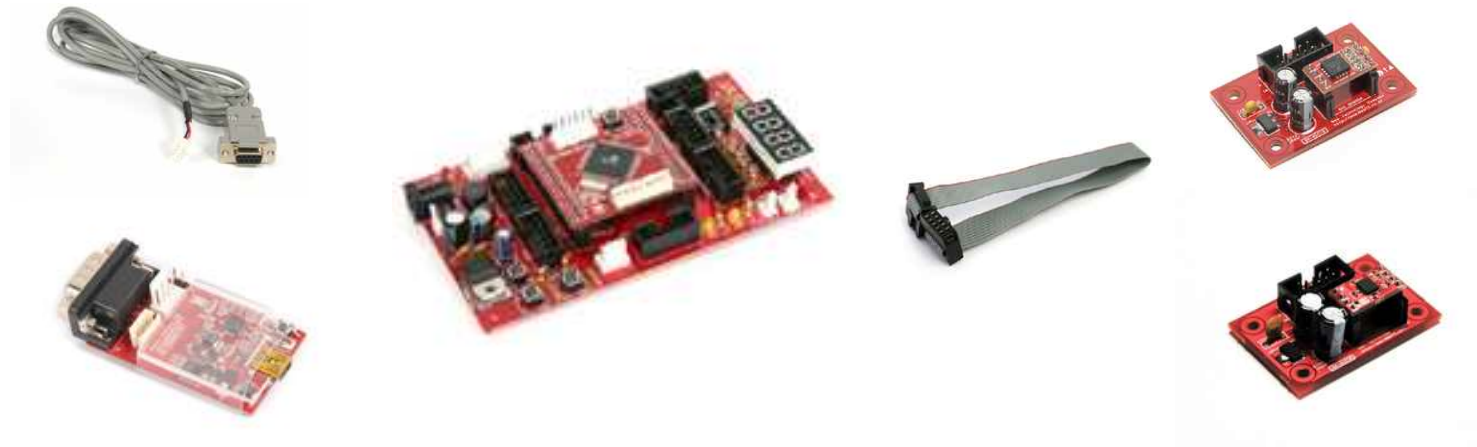
]



[준비물

]

- KD-128PRO-C 키트
- 3축 가속도 센서 플러스 모듈 (AM-3AXIS-P 모듈)
- 2축 자이로 센서 플러스 모듈 (AM-GYRO-P 모듈)
- 2x5 박스헤더 케이블 (SE-2X520)
- 시리얼 케이블 (SE-3T9)또는 USB to Serial 보드 (AD-USB SERIAL)



[Packet Data Format]

- 마우스 제어를 위하여 데이터를 주고 받을 수 있도록 그림 1과 같이 패킷 데이터 포맷을 정의한다.
- 본 패킷의 포맷은 하나의 예로 본 포맷은 경우에 따라 변경이 가능하다.

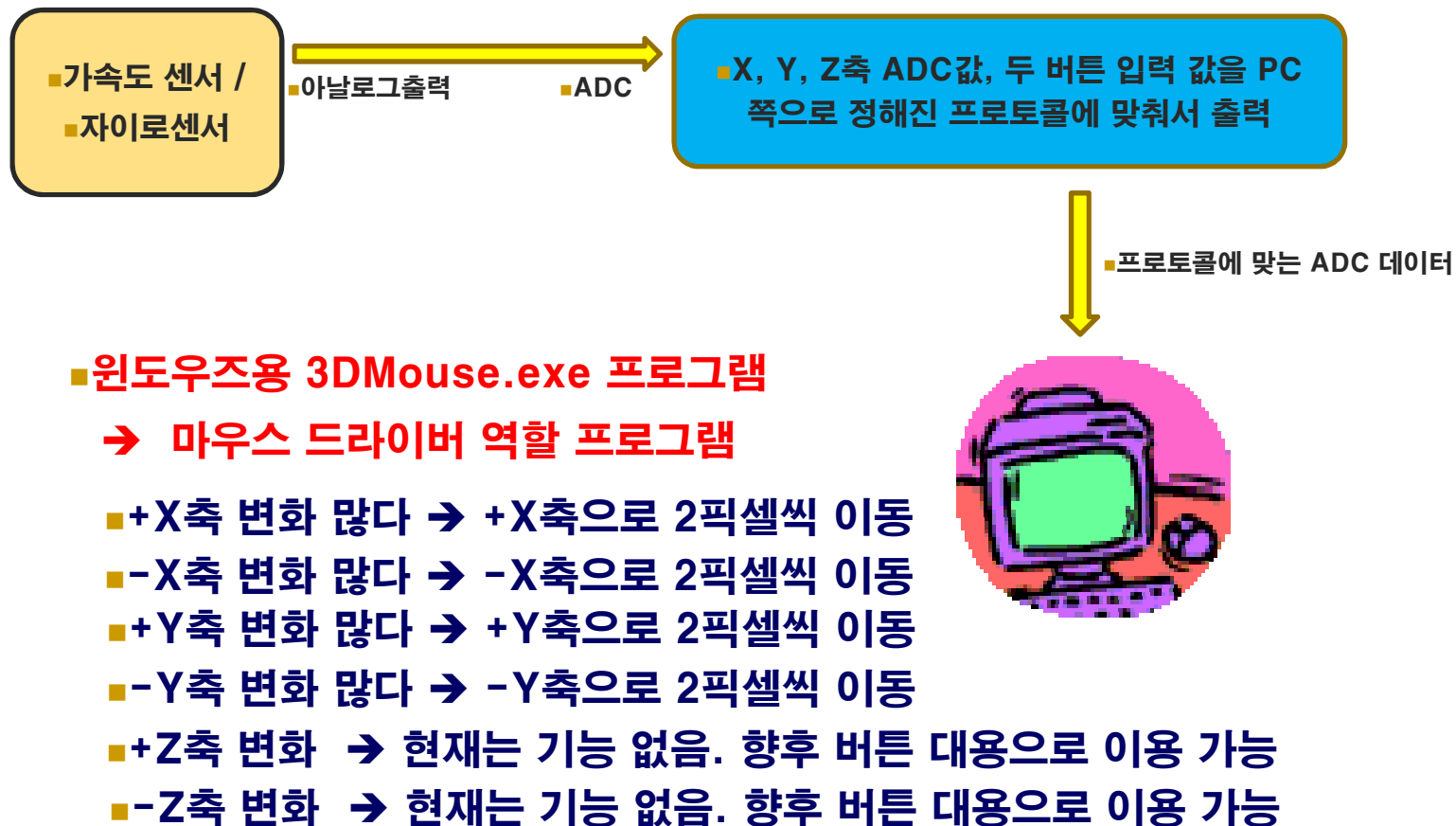
■Start ■(2 byte)	■X(2 byte)	■Y(2 byte)	■Z(2 byte)	■LEFT ■(1byte)	■RIGHT ■(1byte)
---------------------	------------	------------	------------	-------------------	--------------------

(그림 1)

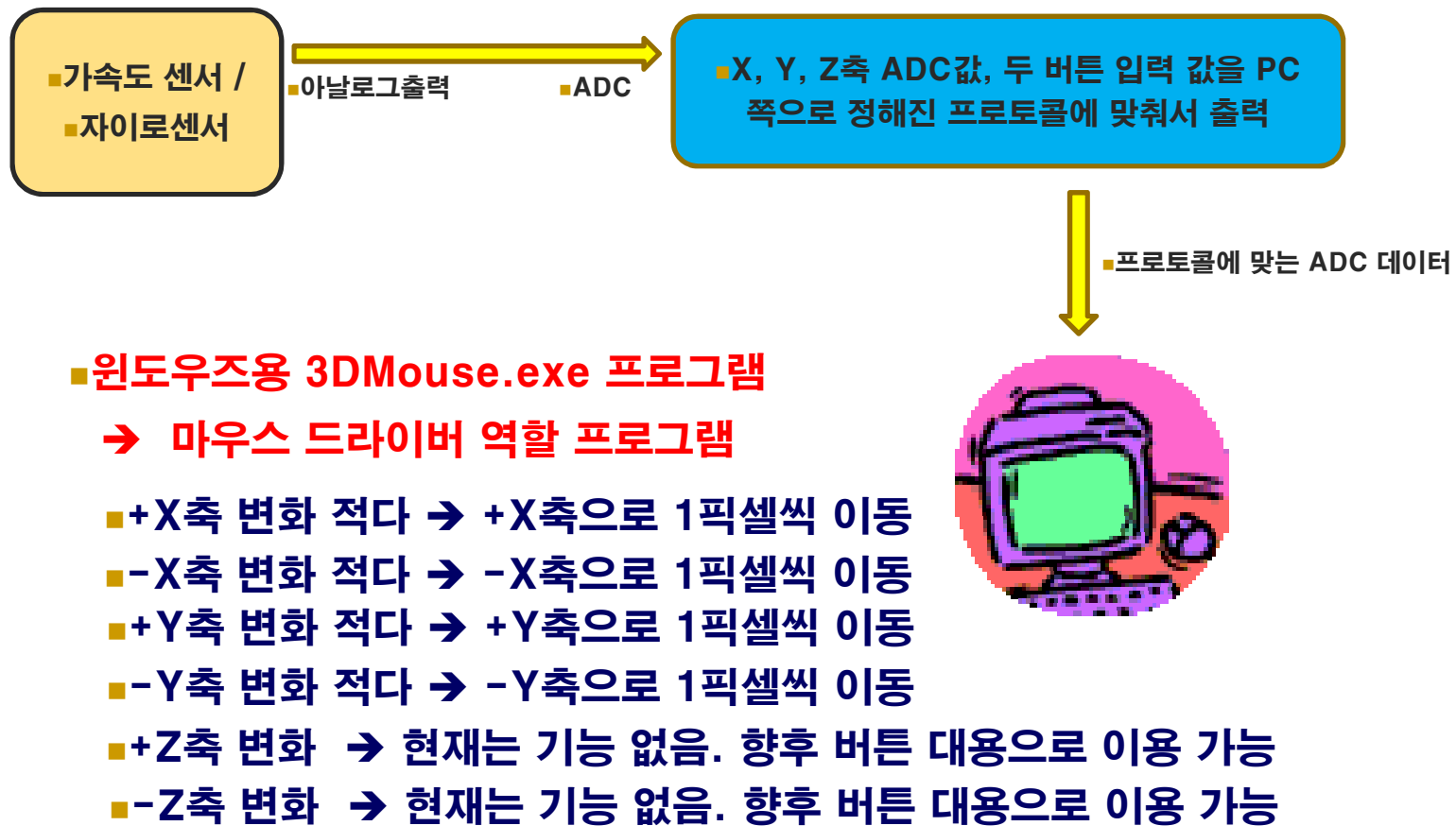
[실제 패킷 작성의 예]

- **Start code**
 - 0xff 0xff
- **X, Y, Z 각각 2 바이트 (총 10비트 데이터)**
 - HIGH Byte : D1, D0 비트만 사용
 - LOW Byte : D7~D0 비트 사용
- **버튼**
 - 눌렀을 때 : 0xf0
 - 안 눌렀을 때 : 0x00

[동작 원리]



[동작 원리]



]



[하드웨어의 단순화 과정]

- 공간마우스 부 하드웨어 구성

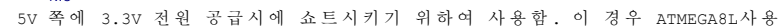
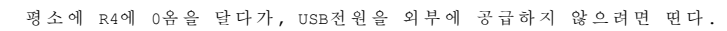
- ATMEGA8A-AU
- 가속도(또는 자이로) 센서
- 송신부 무선 모듈

- 수신부 하드웨어 구성

- 수신부 무선 모듈
- USB to Serial

- PC용 소프트웨어

- 3DMouse.exe



[소스코드(1/4)]

```
■ //ICC-AVR application builder : 2009-04-02 오전 2:05:56
■ // Target : M8
■ // Crystal: 7.3728Mhz
```

```
■ #include <iom8v.h>
■ #include <macros.h>
■ #include <stdio.h>
```

```
■ #define BUFSIZE          10
■ #define STX              2

■ #define LEFT_BTN        (PIND>>3)&0x01
■ #define RIGHT_BTN       (PIND>>2)&0x01
■ #define OK              1
■ #define LOW             0
■ #define HIGH            1
```

```
■ volatile static unsigned char buf[BUFSIZE];//
■ volatile unsigned int ch0, ch2, ch4;
■ volatile unsigned char Start[2], X[2], Y[2], Z[2], flexLeft,
flexRight;
```

```
■ void port_init(void)
```

```
■ {
■     PORTB = 0x00;
■     DDRB = 0x03;
■     PORTC = 0x00;
■     DDRC = 0x00;
■     PORTD = 0x00;
■     DDRD = 0x00;
■ }
```

```
■ void sys_init(void){
```

```
■     Start[0] = 0xff;
■     Start[1] = 0xff;
■     X[0]=X[1]=Y[0]=Y[1]=Z[0]=Z[1]=0;
■     flexLeft=flexRight=0;
■ }
```

[소스코드 (2/4)]

```
■ //UART0 initialize
■ // desired baud rate: 115200
■ // actual: baud rate:115200 (0.0%)
■ // char size: 8 bit
■ // parity: Disabled
■ void uart0_init(void)
■ {
■     UCSRB = 0x00; //disable while setting baud rate
■     UCSRA = 0x00;
■     UCSRC = BIT(URSEL) | 0x06;
■     UBRRL = 0x03; //set baud rate lo
■     UBRRH = 0x00; //set baud rate hi
■     UCSRB = 0x18;
■ }
```

```
■ void init_devices(void)
■ {
■     //stop errant interrupts until set up
■     CLI(); //disable all interrupts
■     port_init();
■     timer0_init();
■     uart0_init();

■     MCUCR = 0x00;
■     GICR = 0x00;
■     TIMSK = 0x00; //timer interrupt sources
■     SEI(); //re-enable interrupts
■     //all peripherals are now initialized
■ }
```

[소스코드 (3/4)]

```
void delay(int n)
{
    volatile int i,j;
    for(i=1;i<n;i++)
    {
        for(j=1;j<600;j++);
    }
}

// printf 함수 사용시 추가할 것.
int putchar(char c)
{
    while (((UCSRA>>UDRE)&0x01) == 0) ;
    // UDRE, data register empty
    UDR = c;
    return c;
}

// scanf 함수 사용시 추가할 것.
int getchar(void)
{
    while ((UCSRA & 0x80) == 0);
    return UDR;
}
```

```
void get_adc_data(void){
    unsigned int i;
    for(i=0;i<16;i++){
        ADCSRA &= 0x7f;
        ADMUX = 0x40;
        ADCSRA |= 0xC0;

        Start[1] = Start[0] = buf[1] = buf[0] = 0xff;

        while((ADCSRA & 0x10) != 0x10);
        X[LOW] = buf[3] = ADCL;
        X[HIGH] = buf[2] = ADCH & 0x03;
        ch0 = buf[0]+buf[1]*256;

        ADCSRA &= 0x7f;
        ADMUX = 0x41;
        ADCSRA |= 0xC0;

        while((ADCSRA & 0x10) != 0x10);
        Y[0] = buf[5] = ADCL;
        Y[1] = buf[4] = ADCH & 0x03;

        ADCSRA &= 0x7f;
        ADMUX = 0x42;
        ADCSRA |= 0xC0;

        while((ADCSRA & 0x10) != 0x10);
        Z[0] = buf[7] = ADCL;
        Z[1] = buf[6] = ADCH & 0x03;
    }
}
```

[소스코드 (4/4)]

```
void init_buf(void){
    buf[8] = flexLeft = 0x00;
    buf[9] = flexRight = 0x00;
}
void main(){
    int i;
    unsigned char send;
    init_devices();
    printf("\r\n\r\n3AXIS Test Program V01\r\n");
    ADMUX = 0x00;           // single-ended ADC3
    ADCSRA = 0x86;          // ADC enable, 125kHz
    while(1){
        get_adc_data();
        if(RIGHT_BTN == OK){
            buf[9] = flexRight = 0xF0;
        }else{
            buf[9] = flexRight = 0x00;
        }
        if(LEFT_BTN == OK){
            buf[8] = flexLeft = 0xF0;
        }else{
            buf[8] = flexLeft = 0x00;
        }
        for(i=0; i<(BUFSIZE); i++){
            send = buf[i];
            putchar(send);
        }
    }
}
```

[활용 방안 (1/2)]

- 장갑형으로 제작하여, 특수 목적 공간 마우스로 활용
- 장애인 또는 특정 목적(정밀실험, 비행체, 우주 공간 등)에서 사용 가능
- 팔찌 형태로 제작하여 사용도 가능
 - 이 경우 버튼은 Z 축으로 이용
 - 예 : 아래위 방향으로 심하게 흔들(왼쪽 클릭)
 - 양 옆으로 심하게 흔들(오른쪽 클릭)
- 손가락 근육 장애인들이 쓸 수 있도록 만들기 위한 입
으로 혹은 불면 동작하는 버튼
- 자동차 운전 게임 등의 모션 게임용으로 사용 가능

[활용 방안 (2/2)



- 눈의 깜박임을 인식하도록 PC용 카메라를 통하여 왼쪽 오른쪽 눌림을 인식하는 버튼
- 특정 움직임을 특정버튼으로 인식하는 기능(예를 들어서, L자로 마우스를 들어서 움직이면 왼쪽 버튼으로 인식)
- 손이 없는 장애인들이 쓸 수 있도록 보드를 머리 쪽에 장착할 수도 있다. 이 경우, 머리 끄덕끄덕거림이나 좌우로 흔들기 나 눈의 깜빡임 등을 버튼으로 인식할 수도 있다.
- 손가락에 Flex 센서 등을 넣어서, 구부러짐을 감지하여 버튼 눌림으로 인식하는 버튼
- 보드를 손 대신, 발에도 장착이 가능하며, 이 경우 반대쪽 발이나, 위에서 쓴 버튼으로 여러 가지 방법이 활용 가능하다.

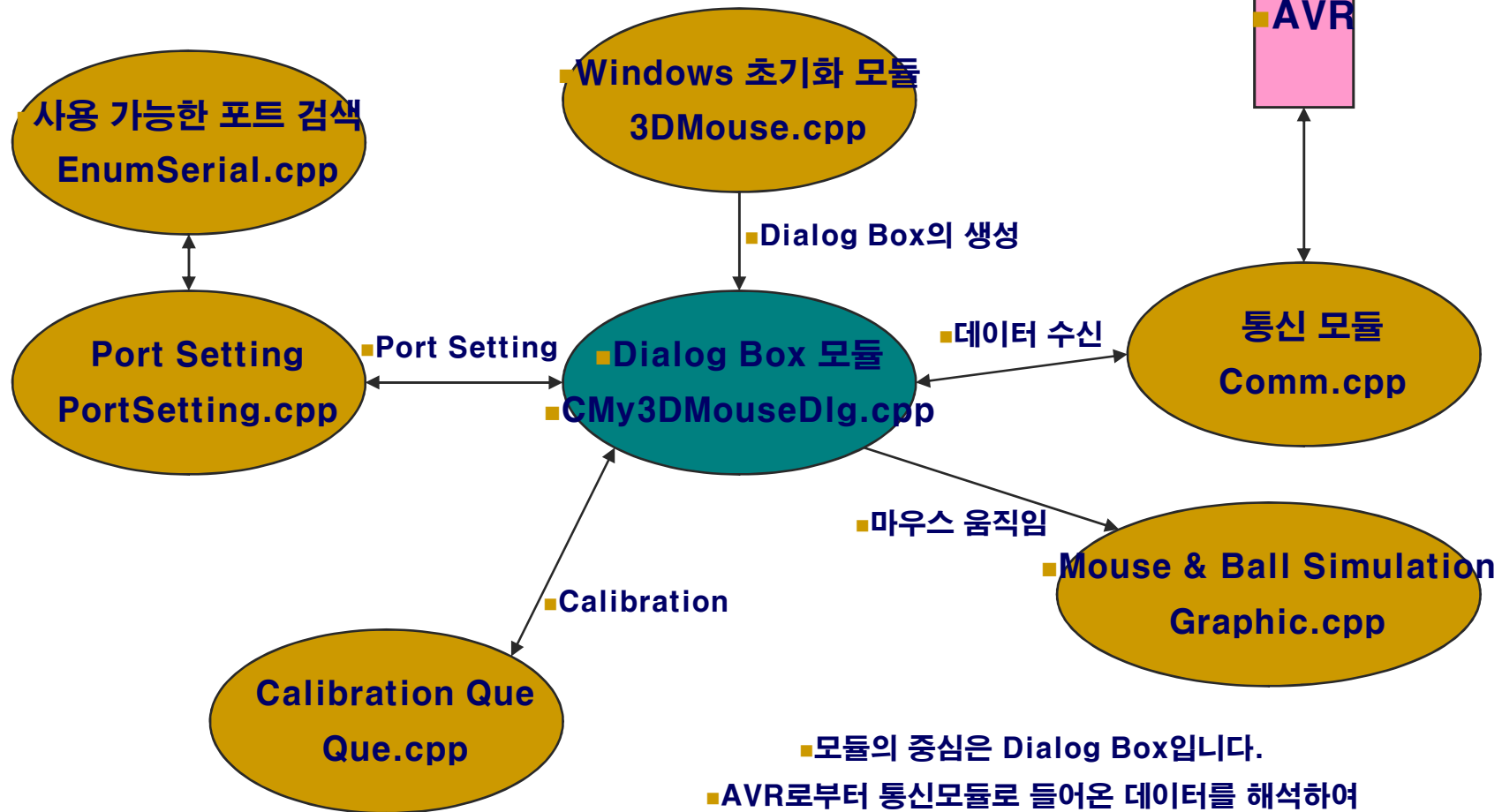
[공간 마우스 장착 위치 제안]

- 장갑 형태로 장갑 위에 보드를 장착한다.
- 반지 형태로 손가락 위에 보드를 장착한다.
- 모자나 두건 형태로 보드를 장착한다.
- 헤드폰 형태로 보드를 장착한다.
- 손목이나 발에 장착한다.
- 기타 필요에 따라 어떤 곳이든 사람이 움직일 수 있는 곳에 장착이 가능하다.
- 원격으로도 모터 장착 등으로 마우스를 움직일수 있다.
이 경우, 특정 IP로 직접 통신하거나, PC를 통하여 통신할 수 있다.

[프로그램 설명]

- 본 프로그램은 AM-AXIS의 기울기 값을 AVR로부터 Serial 통신으로 전송 받아, 마우스를 움직이고, 공의 위치를 표시하는 데모입니다.
- AVR에서 버튼이 눌리면 마우스의 좌우측 버튼이 동작 됩니다.
- AVR로부터 전송 받는 데이터는 X,Y,Z축 값과 2개의 버튼 상태입니다.

[1. 모듈 구성도



- 모듈의 중심은 Dialog Box입니다.
- AVR로부터 통신모듈로 들어온 데이터를 해석하여
- 마우스나 볼의 움직임으로 표현하고,
- Tray Icon 관리도 합니다.

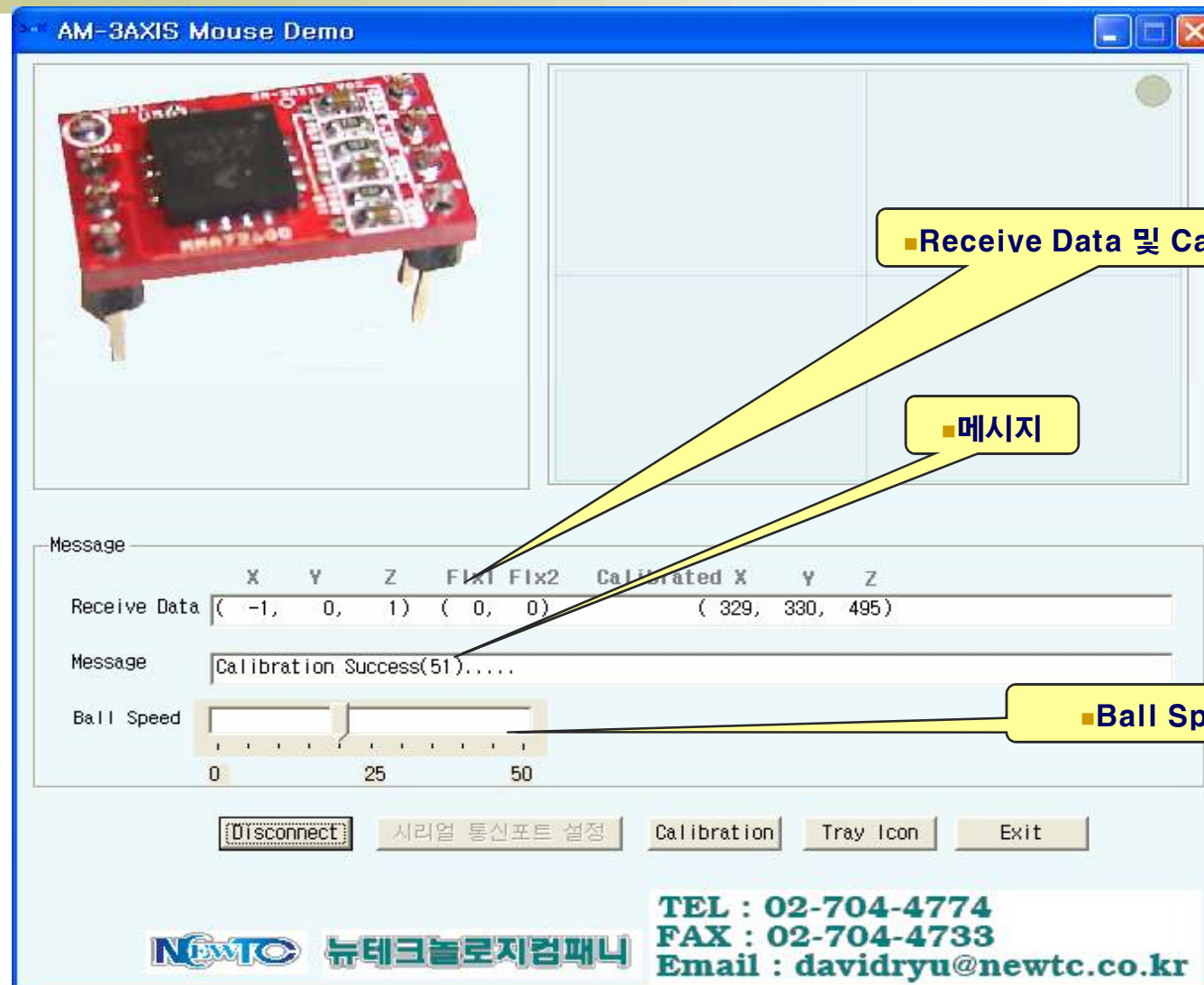
[2. 주요 화면 출력]

■ 연결이 되었을 때의 Tray Icon



■ Double Click하면 본 화면이 나온다.

■ Tray Icon을 더블 클릭 했을 때
전체 화면



■ Receive Data 및 Calibration 값

■ 메시지

■ Ball Speed



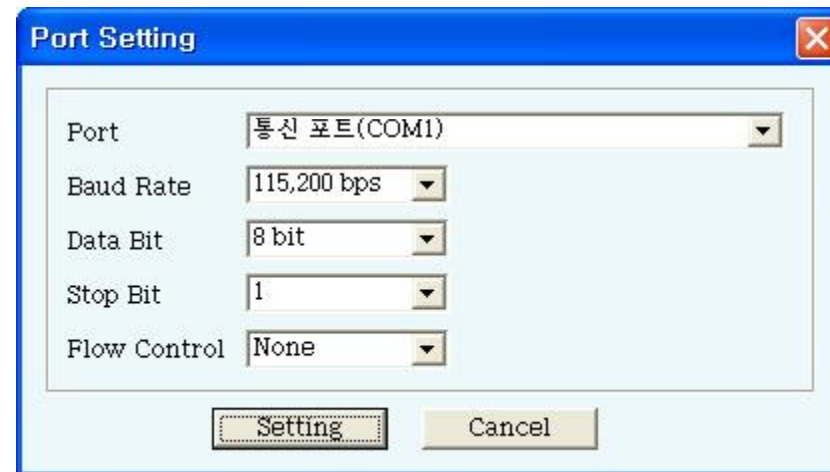
뉴테크놀로지컴패니

TEL : 02-704-4774
FAX : 02-704-4733
Email : davidryu@newtc.co.kr

[

]

■Port setting



The image shows a 'Port Setting' dialog box with a blue title bar and a close button (X) in the top right corner. The dialog contains five settings, each with a label and a dropdown menu:

- Port: 통신 포트(COM1)
- Baud Rate: 115,200 bps
- Data Bit: 8 bit
- Stop Bit: 1
- Flow Control: None

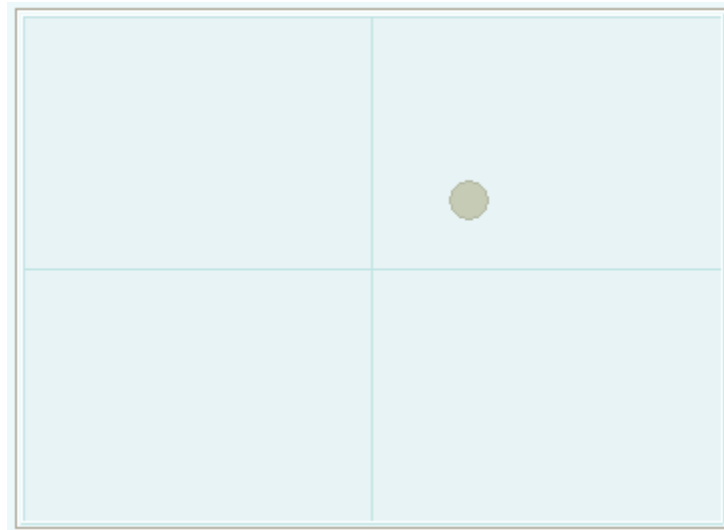
At the bottom of the dialog, there are two buttons: 'Setting' and 'Cancel'.

- 사용할 수 있는 통신 포트와 그 외 정보를
- 맞추어 주어야 한다.

[

]

■ Ball Simulation



- AM-AXIS의 기울기 값에 따라 볼의 위치가
- 좌표에 나타난다.

[3. Tray Icon



1. Dialog Box 초기화 시 SetDefaultIconInTray()가 실행된다.

```
void CMy3DMouseDlg::SetDefaultIconInTray()
{
.... 이전 생략.....
    //NOTIFYICONDATA structure
    NOTIFYICONDATA tnid;

    //set the values tray Icon의 환경 설정
    tnid.cbSize = sizeof(NOTIFYICONDATA);

    //call the Shell_NotifyIcon
    Shell_NotifyIcon(NIM_ADD, &tnid);

    //now the icon is shown in the tray and we can free the icon
}
```

2. 다이얼로그 박스에서 Tray Icon을 누르면 OnBnClickedTrayicon() 함수가 실행된다.
=> ON_BN_CLICKED(IDC_TRAYICON,
&CMy3DMouseDlg::OnBnClickedTrayicon)
3. 단지 자신의 윈도우를 숨기는 기능만 포함된다.
=> ShowWindow(SW_HIDE);

[PC쪽 수신 데이터 구조체]

- `typedef struct _HandData`
- `{`
- `//BYTE        stx1;`
- `//BYTE        stx2;`
- `unsigned short    x;`
- `unsigned short    y;`
- `unsigned short    z;`
- `BYTE flex1;`
- `BYTE flex2;`
- `}HANDDATA;`

[

]

■ 마우스 움직이게 하는 이벤트

- `mouse_event(MOUSEEVENTF_MOVE,x/(6+nBallSpeed*2),-1*y/(3+nBallSpeed*2), 0, GetMessageExtraInfo());`

■ 마우스 버튼 관련 이벤트

- `mouse_event(MOUSEEVENTF_LEFTDOWN,0,0,0,GetMessageExtraInfo());`
- `mouse_event(MOUSEEVENTF_LEFTUP,0,0,0,GetMessageExtraInfo());`
- `mouse_event(MOUSEEVENTF_RIGHTDOWN,0,0,0,GetMessageExtraInfo());`
- `mouse_event(MOUSEEVENTF_RIGHTUP,0,0,0,GetMessageExtraInfo());`