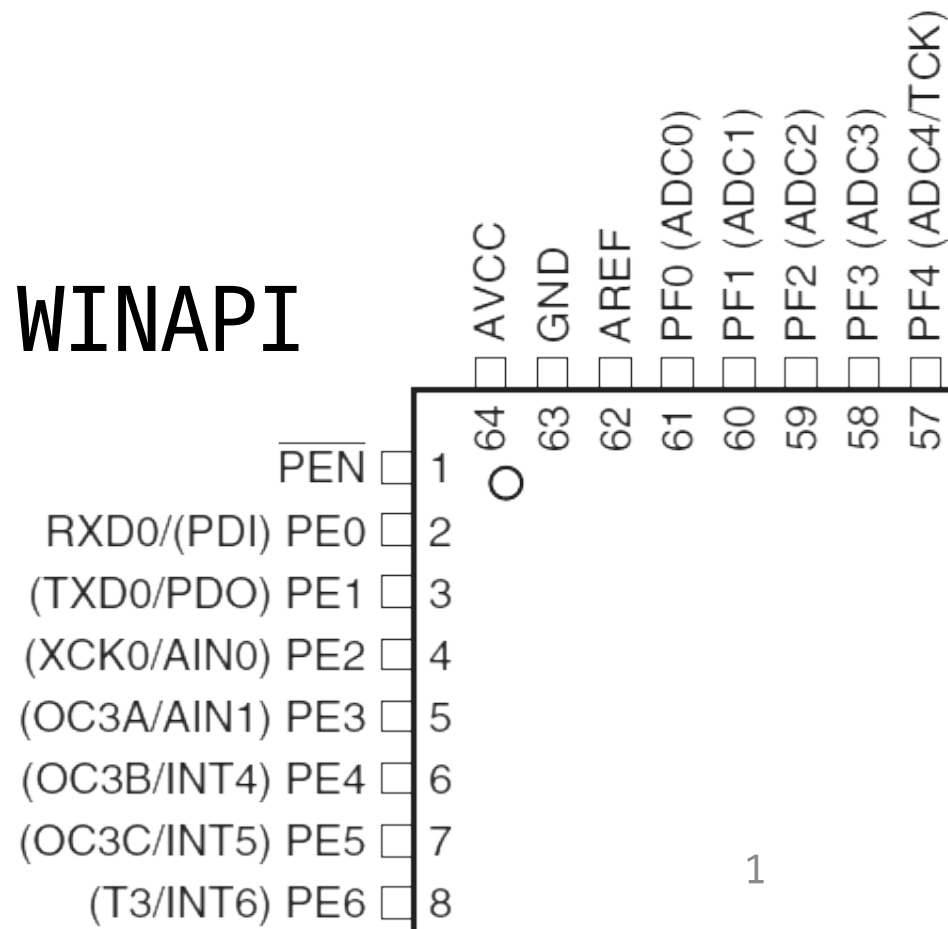


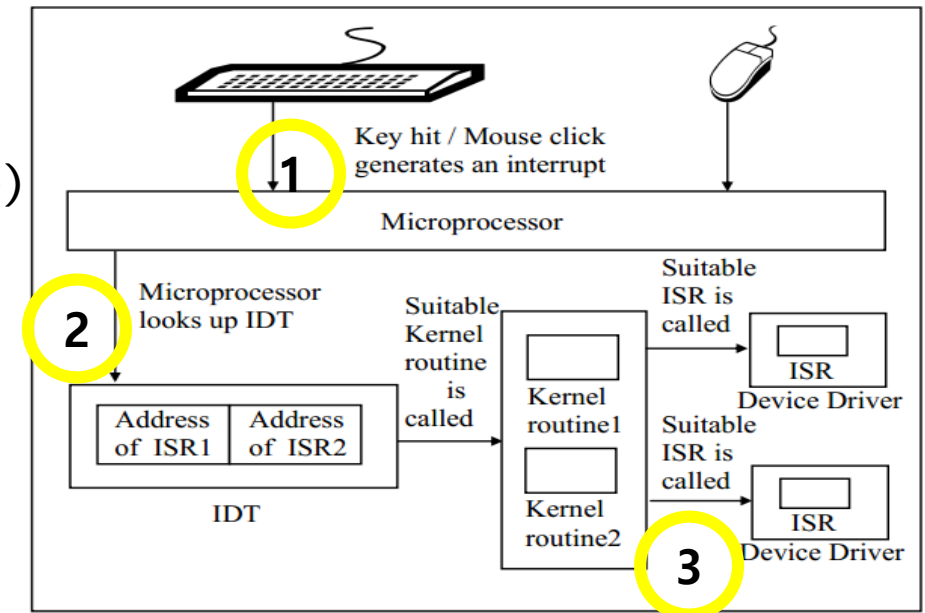
WINAPI

HW interaction with WINAPI



•Hardware 상호 작용

1. 윈도우 환경에서 외부 이벤트(external event, 키를 입력하거나, 타이머가 동작할 때 등...)는 하드웨어 인터럽트(hardware interrupt)가 발생했다는 신호를 생성한다.
2. 발생한 인터럽트에 대하여 IDT(Interrupt Descriptor Table)에서 해당 인터럽트에 대한 루틴이 호출된다.
 - IDT는 발생한 인터럽트에 대한 루틴이 저장된 메모리 주소를 저장하고 있다.
3. 이 루틴은 Windows OS의 일부이며, 커널 루틴이 호출되면, 커널 루틴은 다시 Device Driver내의 ISR(Interrupt Service Routine)을 호출한다. 이렇게 호출된 ISR이 HW와 상호 작용을 한다.



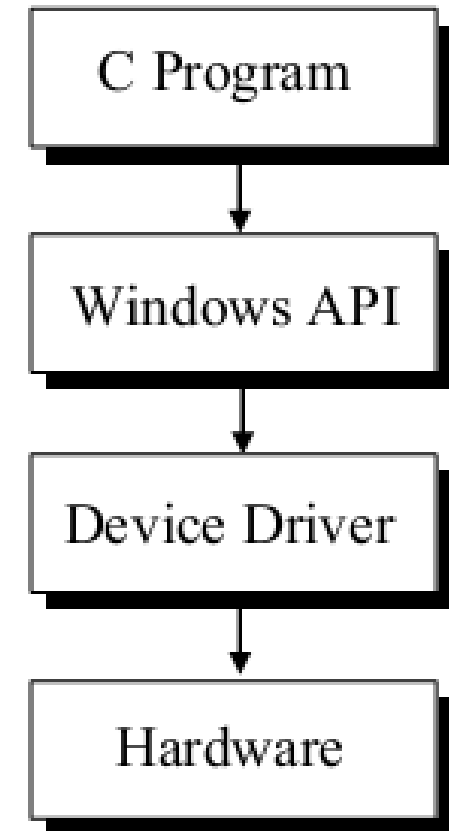
•Hardware 상호 작용

1. 윈도우는 응용 프로그램에게 디바이스에 직접 접근하는 것을 허가하지 않는다.

➤대신 HW에 접근할 수 있는 WinAPI를 제공한다.

2. HW에 접근하기 위해 API 함수를 호출하면,
API 함수는 다시 Device Driver를 호출한다.

➤실질적으로 HW에 접근하는 것은 Device Driver다.



•API

➤ 큰 프로그램을 작성해봅시다.

- 화면에 그림을 띄우고
- 소리를 출력하고
- 통신장치를 통해 외부와 통신하고
- 마우스 / 키보드 입력을 받고
- 데이터 베이스에 많은 데이터를 저장하고
- 저장한 데이터에 대해 다양한 연산을 진행하고...
 - 이런 모든 기능을 다 구현하기 위해선 얼마나 많은 시간이 걸릴까?
 - 그래서 누군가가 이런 기능들을 미리 구현한다.
 - "라이브러리"

•API

➤그럼 라이브러리는 어떻게 사용하지?

- API를 통해 활용할 수 있다.
- "Application Programming Interface"
- 응용 프로그램을 작성할 때 활용할 수 있는 다양한 자료형, 구조체, 함수, 클래스 등에 대하여 정의.
 - "기능의 구현이 아니라 기능의 정의"
 - 경우에 따라서 같은 함수라도, 그 구현 방법이 달라질 수 있음.
 - printf함수의 경우, 일반적인 PC에선 모니터에 출력하지만,
 - 일반적인 임베디드 시스템에선 UART 통신을 통한 문자열 출력으로 구현될 수 있음.

•WinAPI를 통한 마우스 제어(1)

➤ 마우스 제어를 위해 WinAPI에서 제공하는 함수

Programming element	Description
<i>ClipCursor</i>	This function confines the cursor to a rectangular area on the screen.
<i>CreateCursor</i>	This function creates a cursor having the specified size, bit patterns, and hot spot.
<i>DestroyCursor</i>	This function destroys a cursor created by the CreateCursor function and frees any memory the cursor occupied.
<i>GetCapture</i>	This function retrieves the handle to the window, if any, that has captured the mouse or stylus input.
<i>GetClipCursor</i>	This function retrieves the screen coordinates of the rectangular area to which the cursor is confined.
<i>GetCursor</i>	This function retrieves the handle to the current cursor.
<i>GetCursorPos</i>	This function retrieves the cursor's position, in screen coordinates.
<i>GetDoubleClickTime</i>	This function retrieves the current double-click time for the mouse or stylus.
<i>GetMouseMovePoints</i>	This function retrieves points, associated with WM_LBUTTONDOWN and WM_MOUSEMOVE messages, that the OS typically discards when an application cannot process these messages as fast as the application receives these messages.
<i>LoadAnimatedCursor</i>	This function enables an application to define and load a customized, animated wait cursor.
<i>LoadCursor</i>	This function loads a cursor resource.
<i>mouse_event</i>	This function synthesizes mouse motion and button clicks.
<i>ReleaseCapture</i>	This function releases the mouse or stylus capture from a window in the current thread and restores normal processing of input.
<i>SetCapture</i>	This function sets the mouse or style capture to a specified window that belongs to the current thread.
<i>SetCursor</i>	This function establishes the cursor shape.
<i>SetCursorPos</i>	This function moves the cursor to the specified screen coordinates.
<i>ShowCursor</i>	This function displays or hides the cursor.

•예제 1, 2

예제 1

```
#include <Windows.h>
#include <stdio.h>

int main()
{
    int i;

    for (i = 0; i < 500; i++)
    {
        SetCursorPos(i, i);
        Sleep(5);
    }
    return 0;
}
```

예제 2

```
#include <Windows.h>
#include <stdio.h>

int main()
{
    POINT a;
    while (1)
    {
        GetCursorPos(&a);
        printf("X = %d , Y = %d\n", a.x, a.y);
    }
    return 0;
}
```

•WinAPI를 통한 마우스 제어(2)

➤INPUT_RECORD

- WinAPI에 정의된 구조체
- 콘솔 입력 버퍼에 있는 입력 이벤트에 대하여 기술한다
 - 입력 이벤트 : 키보드 입력, 마우스 클릭 등...
- ReadConsoleInput 또는 PeekConsoleInput 함수를 이용하여 입력 버퍼에서 입력 이벤트에 대한 값을 읽을 수 있다.

•WinAPI를 통한 마우스 제어(2)

```
typedef struct _INPUT_RECORD {  
    WORD EventType;  
    union {  
        KEY_EVENT_RECORD KeyEvent; //키 입력에 의해 발생하는 이벤트에 대한 기록  
        MOUSE_EVENT_RECORD MouseEvent; // 마우스에 의해 발생하는 이벤트에 대한 기록  
        WINDOW_BUFFER_SIZE_RECORD WindowBufferSizeEvent;  
        MENU_EVENT_RECORD MenuEvent;  
        FOCUS_EVENT_RECORD FocusEvent;  
    } Event;  
} INPUT_RECORD;
```

• 예제 3

```
#include <Windows.h>
#include <stdio.h>

int main(void)
{
    HANDLE hIn, hOut;
    INPUT_RECORD rec;
    DWORD dw;

    hIn = GetStdHandle(STD_INPUT_HANDLE);
    hOut = GetStdHandle(STD_OUTPUT_HANDLE);
    while (1)
    {
        ReadConsoleInput(hIn, &rec, 1, &dw);
        if (rec.EventType == MOUSE_EVENT)
        {
            if (rec.Event.MouseEvent.dwButtonState &
                FROM_LEFT_1ST_BUTTON_PRESSED)
            {
                SetConsoleCursorPosition(hOut,
                    rec.Event.MouseEvent.dwMousePosition);
                fputc('*', stdout);
                if (rec.Event.MouseEvent.dwEventFlags & DOUBLE_CLICK)
                    break;
            }
        }
    }
}
```

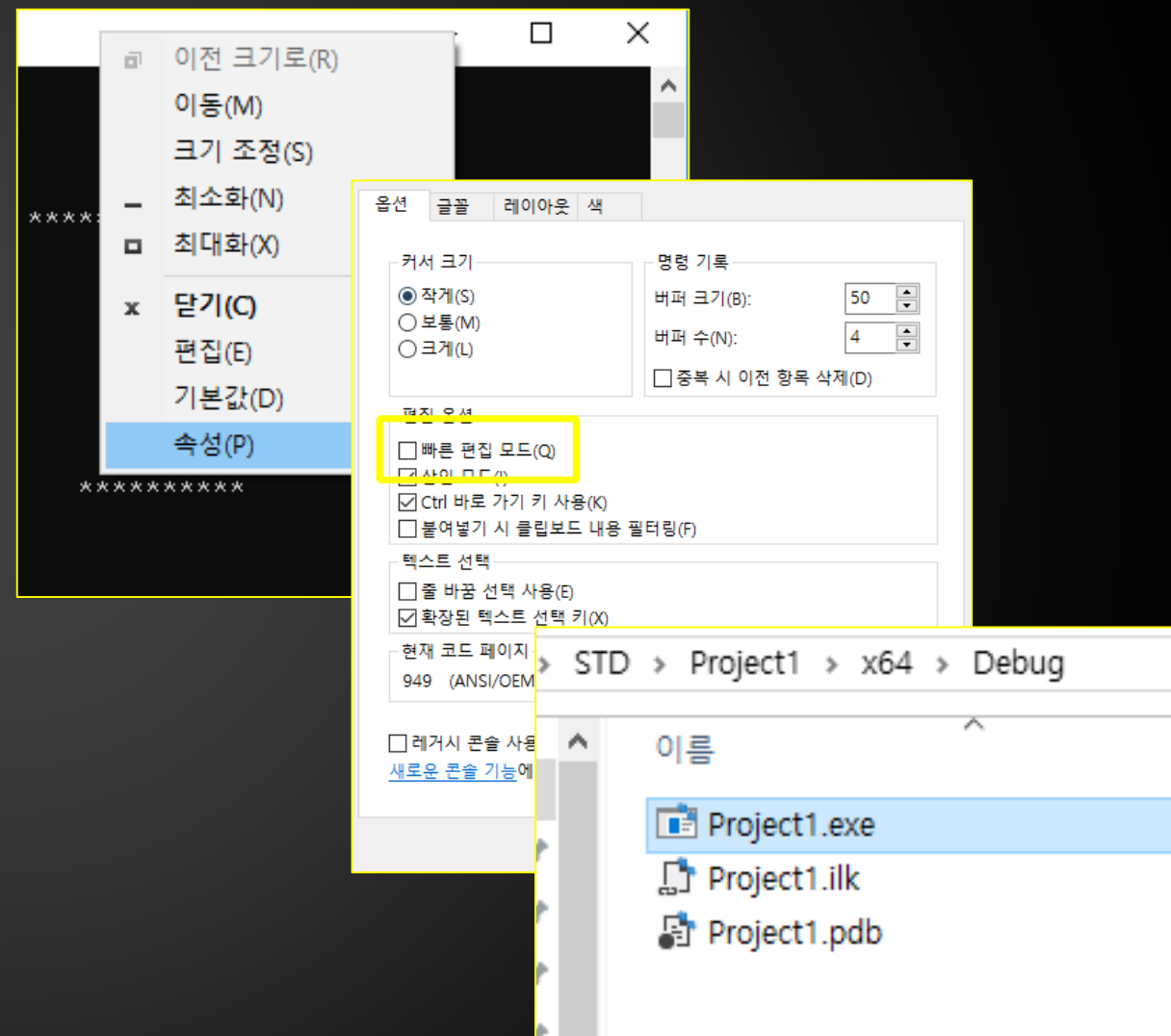
•예제 3 실행 결과



창에서 마우스를 클릭하면서 *이 찍히는지 확인해보자.

•예제 3 실행이 안되는 경우에...?

1. 프로그램 위쪽의 제목 표시줄에서 오른쪽 클릭
2. 속성 클릭
3. 빠른 편집 모드 해제
4. 그래도 실행이 안되는 경우에는, VS에서 F5 / Ctrl + F5를 눌러서 실행하지 말고, 프로그램을 직접 실행시켜본다.

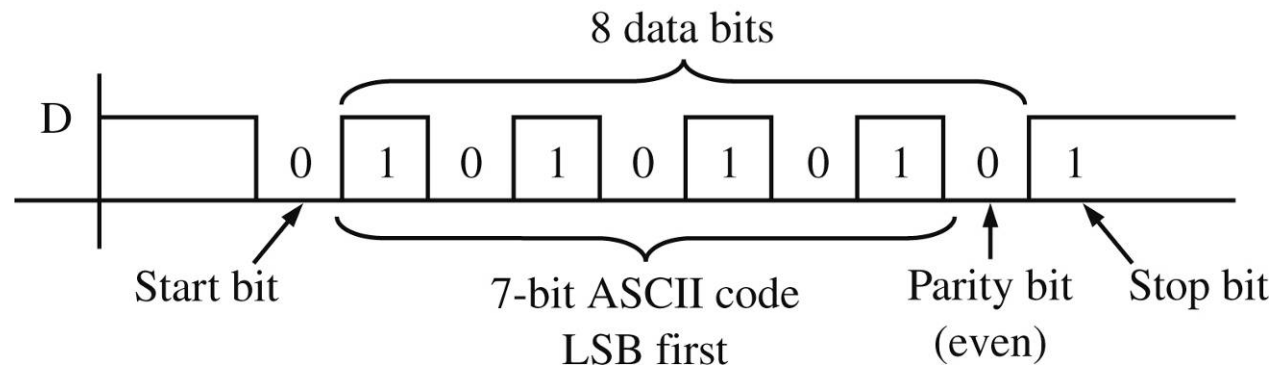
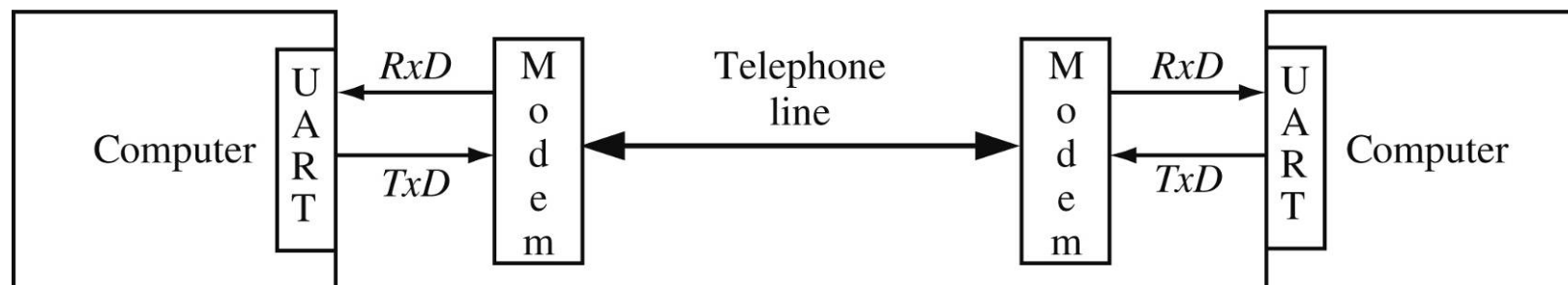


• WinAPI를 통한 UART 통신

➤ UART

➤ Universal Asynchronous Receiver Transmitter

➤ 직렬 데이터 통신



•WinAPI를 통한 UART 통신

➤UART 통신을 위한 WinAPI의 활용

1. Serial 통신을 위한 핸들 획득

- 핸들 : OS가 생성한 오브젝트의 고유 번호, 응용 프로그램은 OS가 생성한 오브젝트에 직접 접근하지 못하며, 대신 핸들을 통해서 접근한다. 오브젝트는 OS에서 사용할 수 있는 리소스를 의미한다.
- **CreateFile** 함수를 사용하여 핸들을 획득한다.

•WinAPI를 통한 UART 통신

➤UART 통신을 위한 WinAPI의 활용

2. 파라미터 세팅

- GetCommState, SetCommState, SetCommTimeouts 등을 사용해 세팅
- Baudrate, Stop bits등 다양한 파라미터에 대하여 세팅 진행
- DCB, COMMTIMEOUTS 등의 구조체를 사용함.

3. WriteFile, ReadFile등의 함수를 사용하여 UART 통신 진행

•예제 4를 실행하기 전에...

➤여러분 PC에 UART 통신 장치/포트가 있나요?

➤요즘 나오는 대부분 PC는 없습니다...

➤있어도 외부로 빼놓는 경우가 적습니다.

➤있다고 해도, 실습에서 통신을 주고 받기 위해선 추가 장치가 필요합니다...

➤그러면 실습은 어떻게 하나요?

➤가상의 serial 통신 장치를 만들어 줘야 합니다! -> VSPE

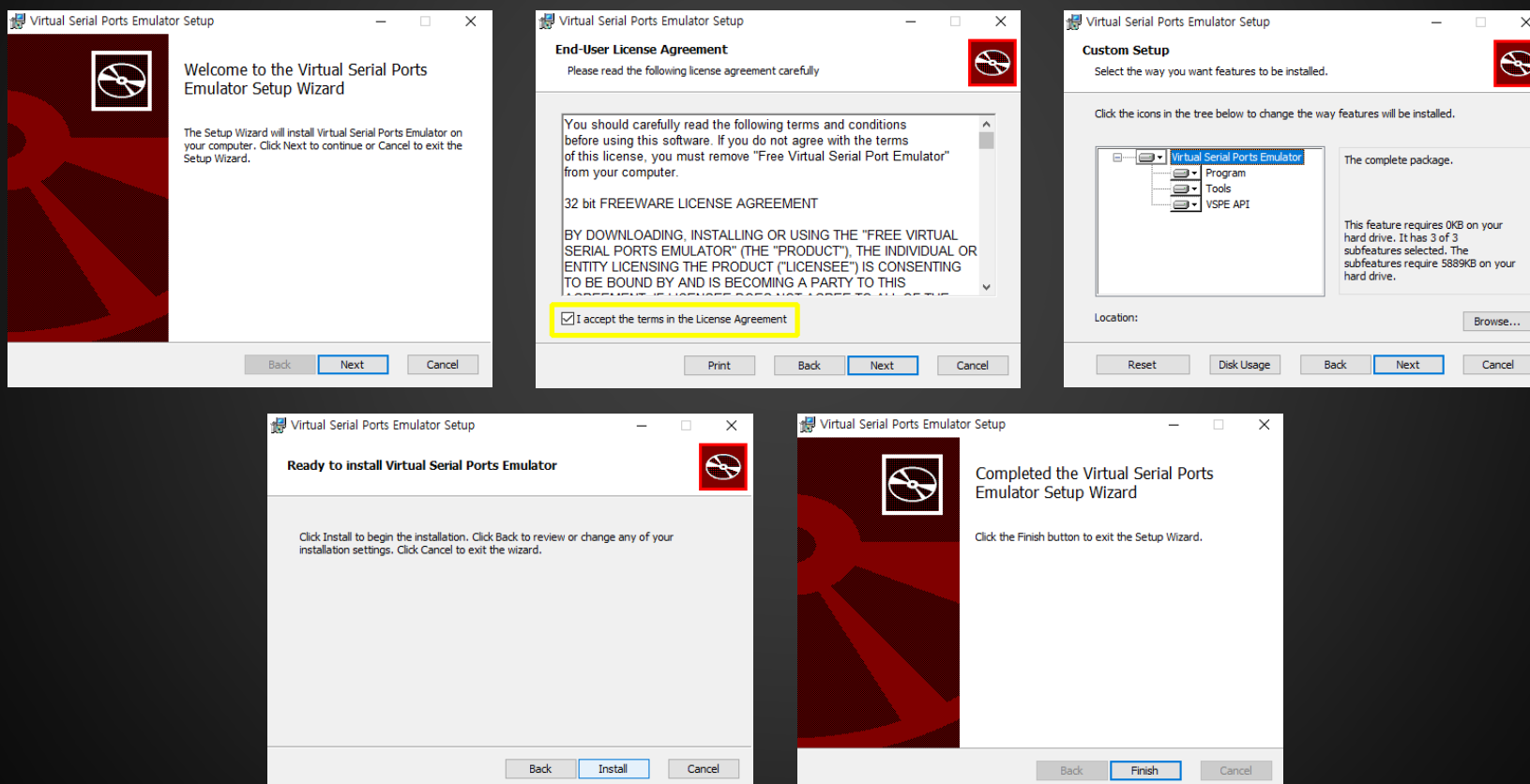
➤그리고 serial 통신을 위한 터미널 프로그램도 같이 설치해야 합니다. -> PuTTY



•예제 4를 실행하기 전에...

1. VSPE 설치

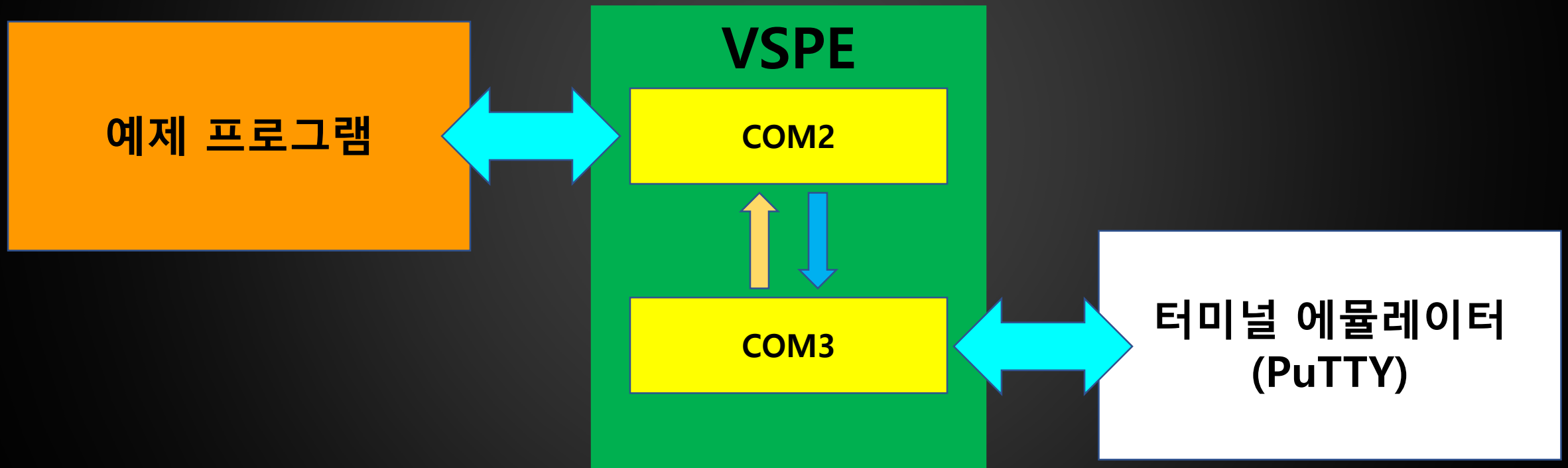
1. 긍정적인 버튼(Next, Install...)을 계속 클릭해서 설치



•예제 4를 실행하기 전에...

2. VSPE 설정

- VSPE를 사용하여 2개의 가상 Serial 장치를 만들고, 둘을 Redirection 합니다.



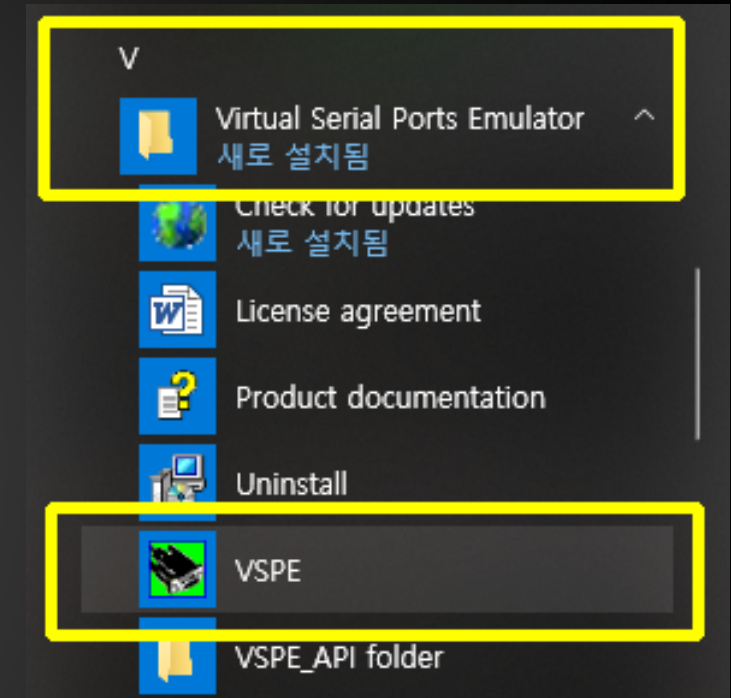
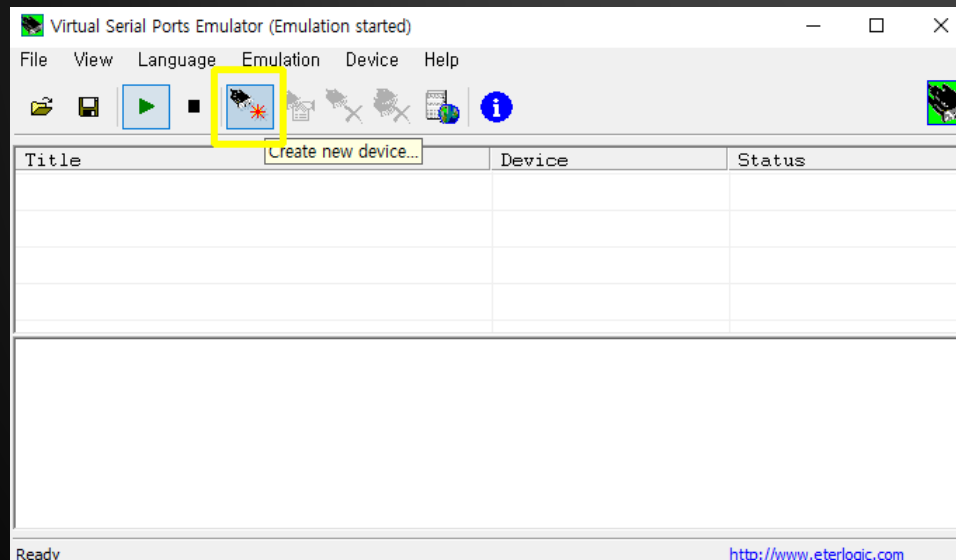
•예제 4를 실행하기 전에...

2. VSPE 설정

1. VSPE를 실행합니다.

- 시작 - Virtual Serial Ports Emulator - VSPE 클릭
- 실행할 때 오류가 생기면 오른쪽 클릭해서 관리자 권한으로 실행

2. Create New device 클릭



•예제 4를 실행하기 전에...

2. VSPE 설정

3. Device type을 Connector로 설정 후 '다음' 클릭
4. Virtual serial port의 이름을 COM2로 설정 후 '마침' 클릭
5. 같은 방법으로 COM3를 생성해주자

Specify device characteristics

Virtual serial port

COM2

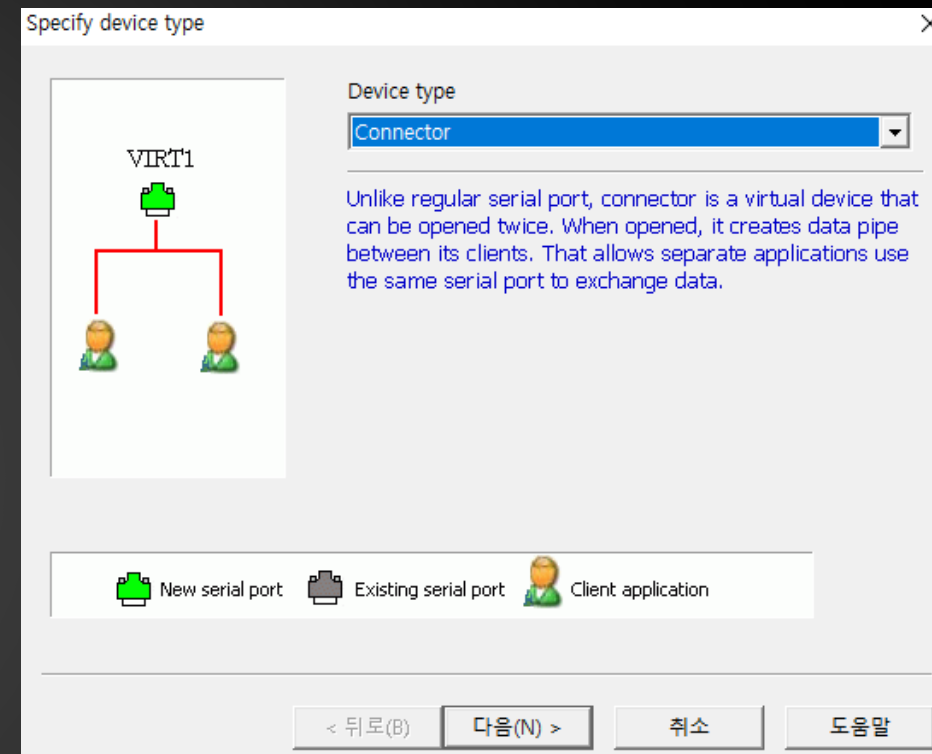
☐ Emulate baud rate (optional)

Specify device characteristics

Virtual serial port

COM3

☐ Emulate baud rate (optional)



•예제 4를 실행하기 전에...

2. VSPE 설정

➤ 2개의 COM port 생성 확인

Title	Device
COM2	Connector
COM3	Connector

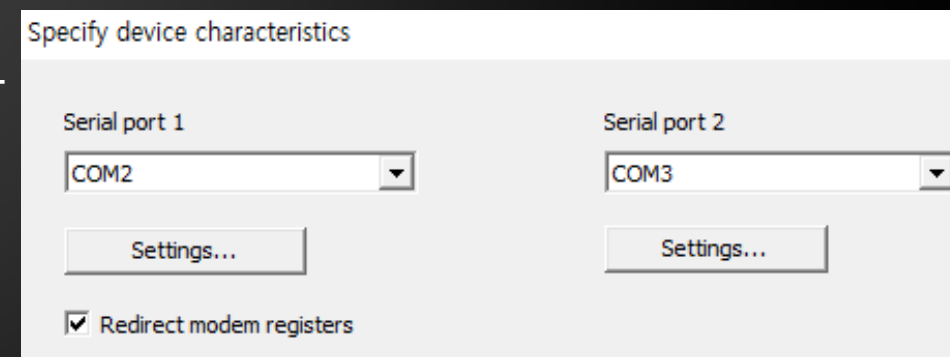
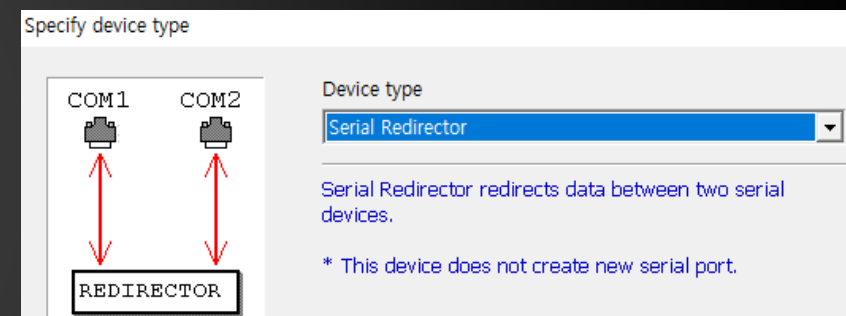
6. Create New device 클릭

7. 이번에는 Device type을 Serial Redirector로 설정하고 '다음'클릭

8. Serial Port 1과 2에 각각 COM2, COM3를 설정

➤ 앞에서 다른 번호의 COM을 생성했으면, 그 번호를 입력한다.

➤ 아직 '다음'클릭하지 마세요



•예제 4를 실행하기 전에...

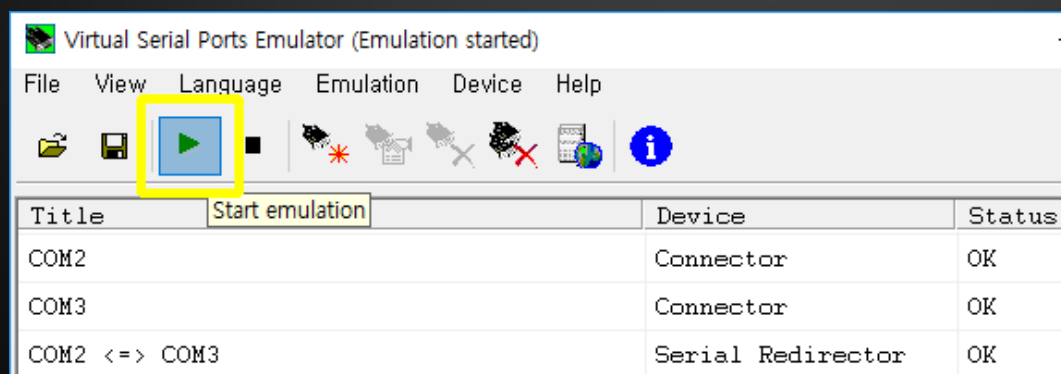
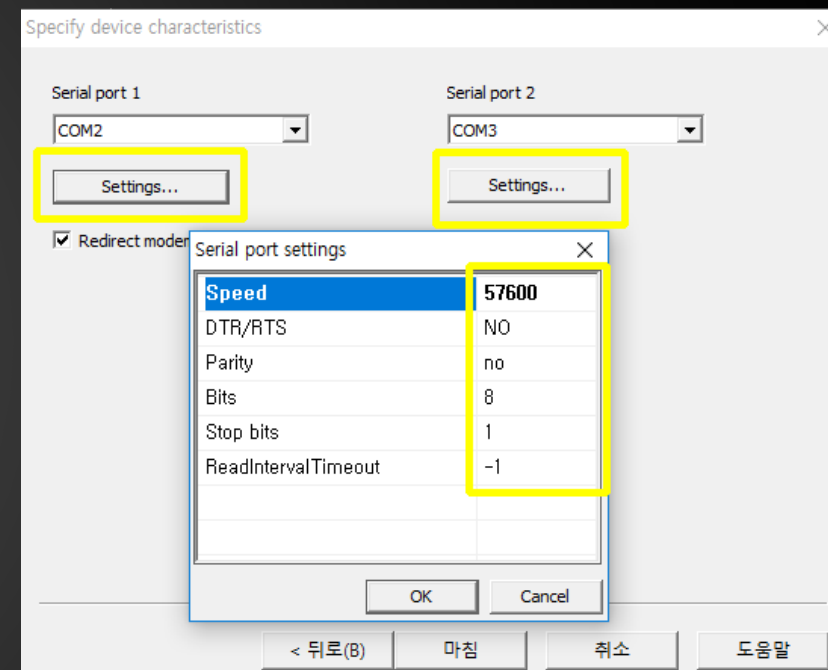
2. VSPE 설정

9. Settings...를 클릭하여 통신속도, 제어 신호 여부, 패리티 비트, 비트 수, 정지 비트 등의 파라미터를 오른쪽 그림과 같이 설정합니다.

➤ Serial Port1, 2 모두 세팅해줍니다.

10. '마침'을 클릭하여 빠져나옵니다.

11. Start emulation을 클릭하여 에뮬레이션 시작.



•예제 4를 실행하기 전에...

3. PuTTY 다운로드

- <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- 자신의 환경에 적합한 msi 파일을 다운받아 실행

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ('Windows Installer')

32-bit: [putty-0.71-installer.msi](#) [\(or by FTP\)](#) [\(signature\)](#)

64-bit: [putty-64bit-0.71-installer.msi](#) [\(or by FTP\)](#) [\(signature\)](#)

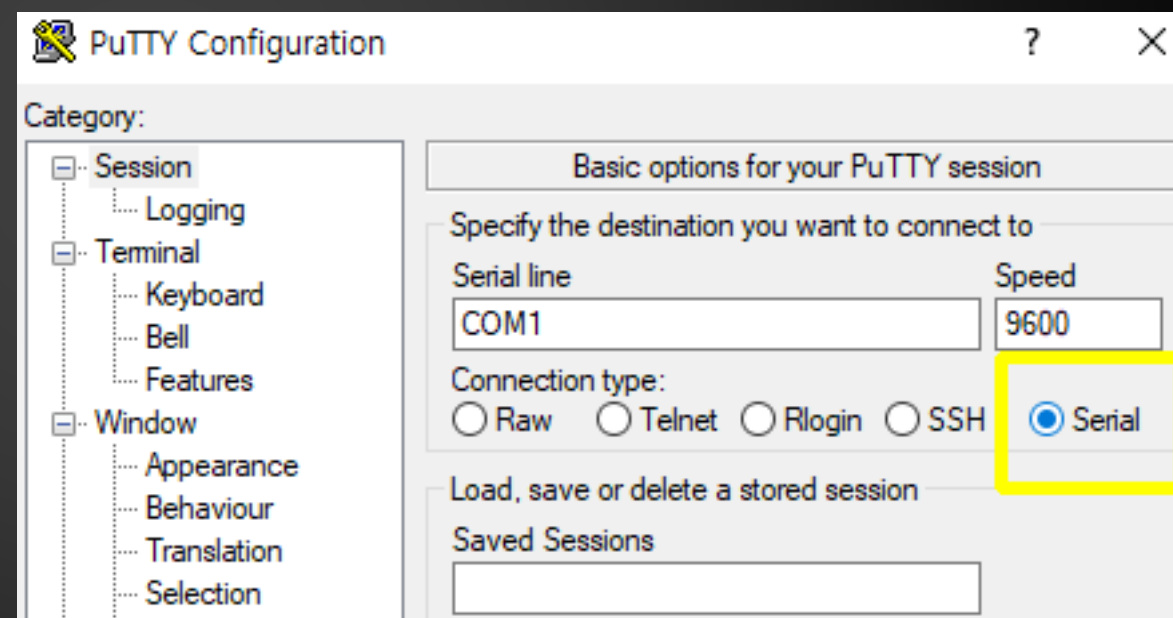
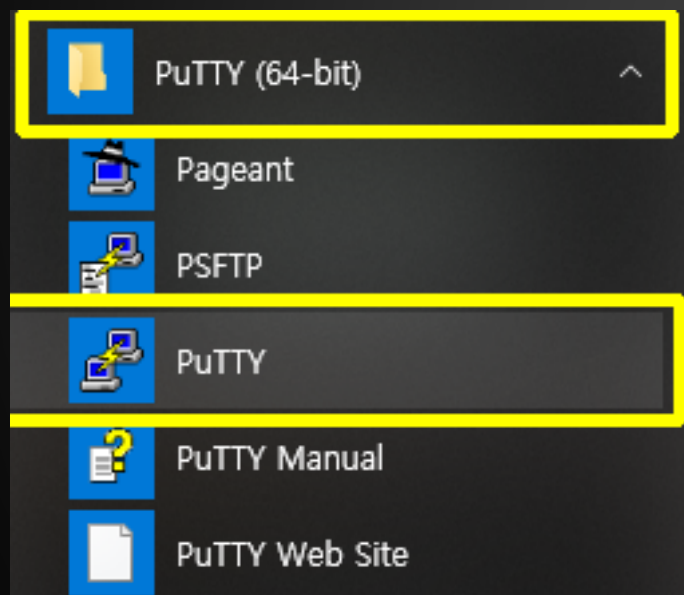
Unix source archive

.tar.gz: [putty-0.71.tar.gz](#) [\(or by FTP\)](#) [\(signature\)](#)

•예제 4를 실행하기 전에...

3. PuTTY 실행 및 설정

1. 시작 - PuTTY - PuTTY 클릭
2. Session에서 Serial 체크



•예제 4를 실행하기 전에...

3. PuTTY 실행 및 설정

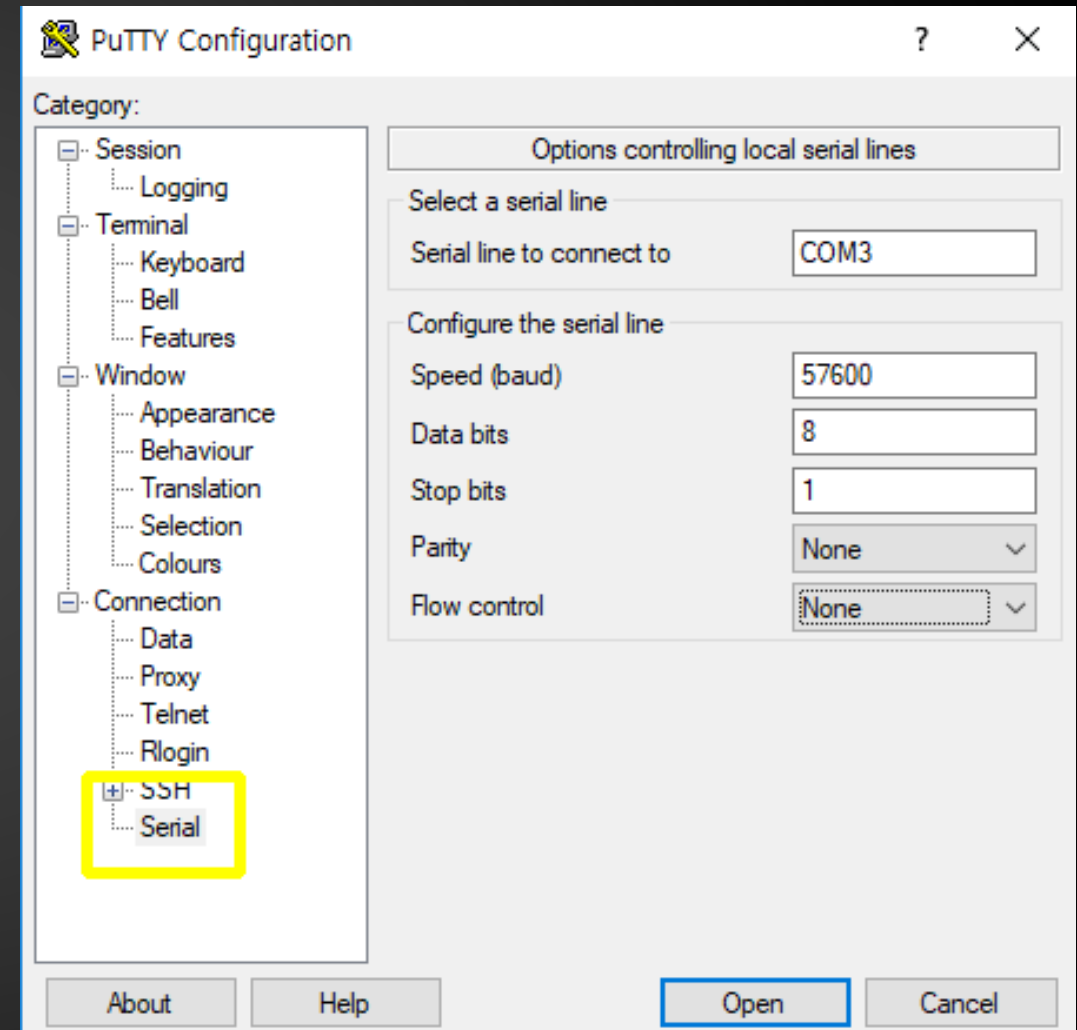
3. 우측 하단의 Serial 클릭

4. 옵션을 오른쪽 그림과 같이 설정

➤ COM 번호는 앞의 VSPE에서 두 번째로 만든 Port의 번호를 사용한다.

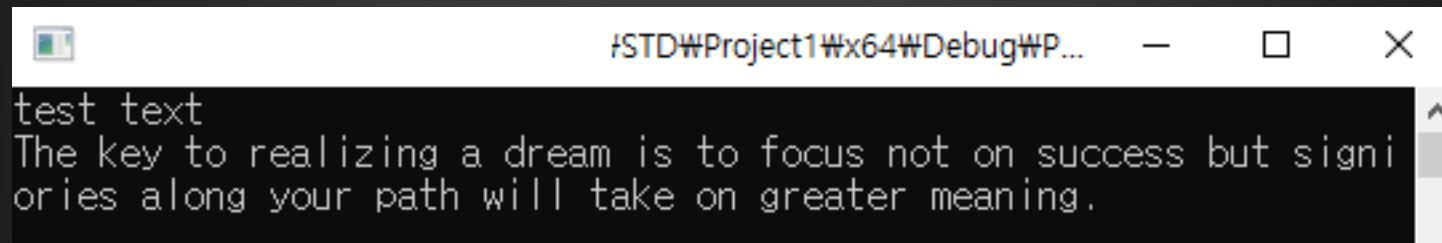
5. Open 클릭

6. 비주얼 스튜디오에서 예제 빌드 및 실행하여 결과 확인

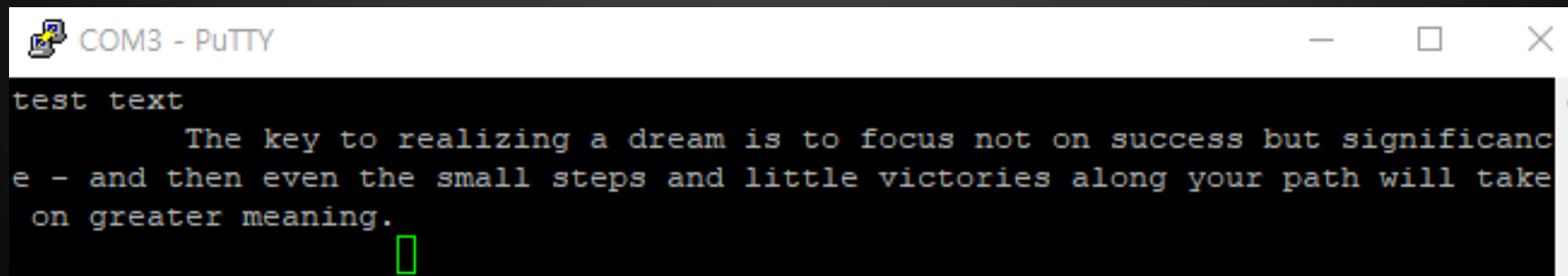


•예제 4 실행 결과

- 프로그램을 실행시키고, 프로그램에 문자를 입력할 때, PuTTY가 UART 통신을 통해 문자열을 받아서 표시하는 것을 확인한다.
- 소스코드는 랩 홈페이지에서 다운



```
#STDWProject1Wx64WDebugWP...
test text
The key to realizing a dream is to focus not on success but signi
ories along your path will take on greater meaning.
```



```
COM3 - PuTTY
test text
    The key to realizing a dream is to focus not on success but significanc
e - and then even the small steps and little victories along your path will take
on greater meaning.
```

•예제 5

- 프로그램을 실행시키고, PuTTY에 문자를 입력할 때 UART 통신을 통해 프로그램이 문자열을 받아서 현시하는 것을 확인한다.
- 소스코드는 랩 홈페이지에서 다운
- PuTTY 창에 아무런 글자가 뜨지 않을 수 있다.

• 과제

- 마우스의 위치를 실시간으로 UART로 출력한다.
- 마우스에서 더블클릭이 발생하면, UART를 통해 좌표 값을 수신 받고, 그 좌표로 마우스 커서를 이동시킨다.

COM3 - PuTTY

```
Xpos : 0619 Ypos : 0438
Xpos : 0647 Ypos : 0435
Xpos : 0672 Ypos : 0425
Xpos : 0697 Ypos : 0418
Xpos : 0720 Ypos : 0407
Xpos : 0741 Ypos : 0396
Xpos : 0755 Ypos : 0388
Xpos : 0762 Ypos : 0381
Xpos : 0768 Ypos : 0376
Xpos : 0771 Ypos : 0372
Xpos : 0772 Ypos : 0370
Xpos : 0772 Ypos : 0369
Xpos : 0773 Ypos : 0368
Xpos : 0773 Ypos : 0366
Xpos : 0773 Ypos : 0366
Xpos : 0773 Ypos : 0366
Xpos : 0773 Ypos : 0366
DoubleClick!!
Xpos : 123
Xpos : 0123 Ypos : 0456
Xpos : 0123 Ypos : 0456
Xpos : 0123 Ypos : 0456
Xpos : 0123 Ypos : 0456
```