

VR 시리얼 통신 기초

- VR 프로그램에서는 아두이노 보드와 같이 시리얼 통신이 가능한 장치와 시리얼 통신이 가능하다. 아두이노 보드를 사용할 경우, 아두이노 보드와 PC 간에는 USB 케이블로 연결되어 있어야 한다.



아두이노 <-> VR 시리얼 통신 코딩 절차

- 아두이노 보드와 VR 프로그램 간에 시리얼 통신을 하기 위해서는 다음과 같은 절차로 진행한다.

절차 1 아두이노 코드 작성 및 업로드

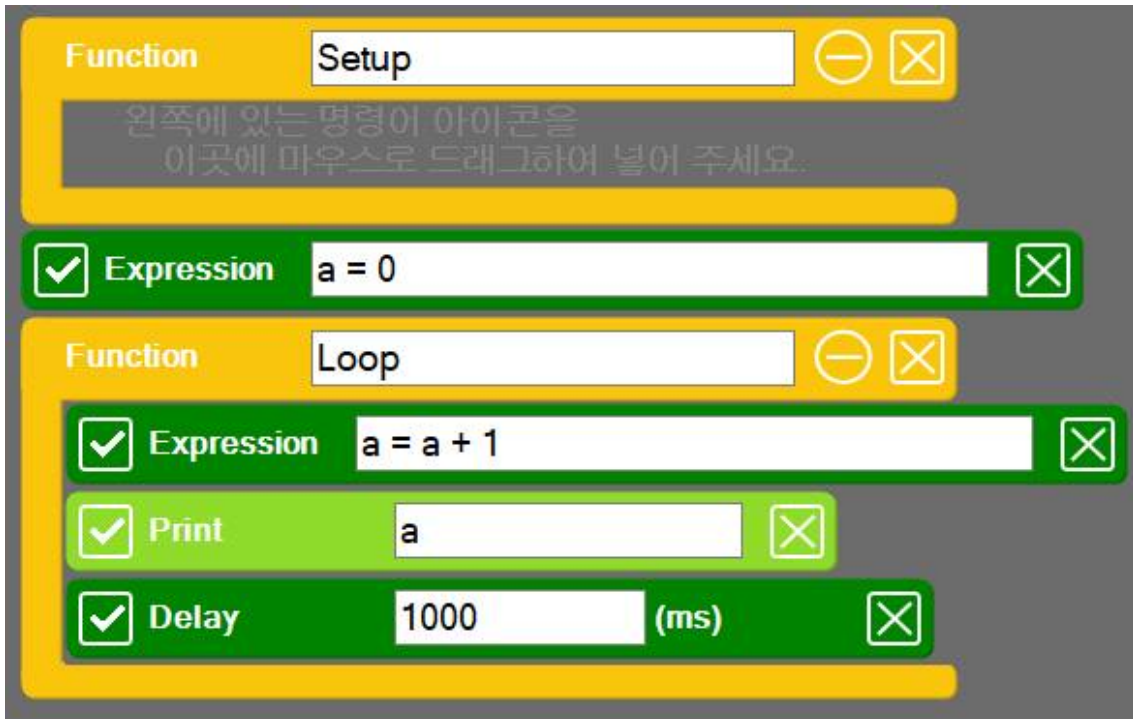
절차 2 시리얼 포트 확인 및 아두이노 코딩 편집기 종료

절차 3 USB 케이블 재연결 (연결 초기화)

절차 4 VR 코딩 편집기 실행 및 코드 작성

절차1) 아두이노 코드 작성 및 업로드

- 아두이노 코드 편집기를 실행한 후, 다음과 같이 간단하게 1씩 증가하는 값을 Print 명령어로 전송하는 기능을 작성해 본다.



```
void setup()
{
}

a = 0

void loop()
{
    a = a + 1
    Print(a)
    Delay(1000)
}
```

- 위의 예제에서 PrintLine 대신에 Print 명령어를 사용한 것에 대해 유의한다.
- 이제 위의 코드를 실행하여 아두이노 보드에 업로드 한다. 업로드가 정상적으로 진행되면, 콘솔창에 1초 마다 숫자값이 옆으로 이어서 출력되게 된다.
- 콘솔창을 종료한다.

절차2) 시리얼 포트 확인 및 아두이노 코딩 편집기 종료하기

- 아두이노 코딩 편집기에서 아두이노 보드에 연결된 시리얼 포트 이름을 확인한다. 아래의 예에서는 COM3가 시리얼 컴포트 번호이다.



만약 COM 포트가 표시되어 있지 않다면 개발툴을 다시 실행하거나 USB 드라이버를 다시 설치해 주기 바랍니다.

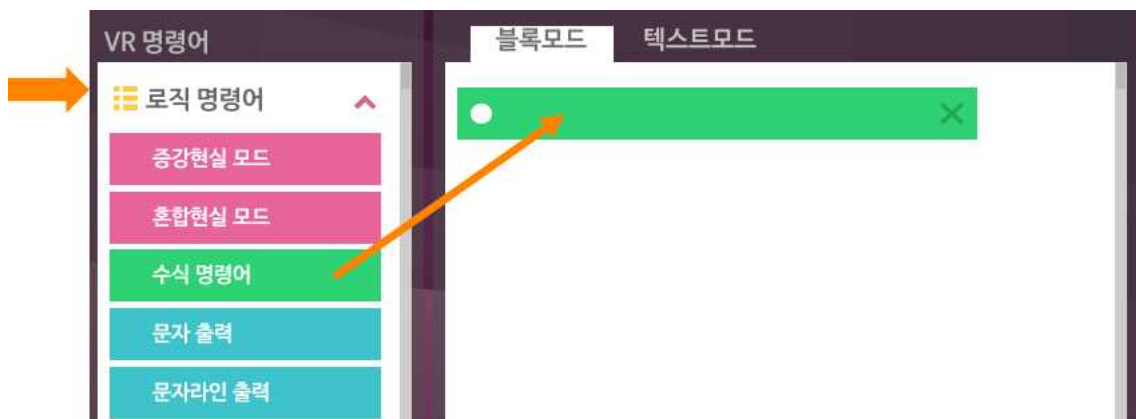
- 아두이노 코드 편집기에서 시리얼 포트를 사용하고 있으면 VR 코딩 편집기에서 해당 포트를 열 수 없다.
- 따라서 VR 코딩 전에 아두이노 코딩 편집기 프로그램을 종료하고, USB 케이블도 뺐다가 다시 연결해 주어야 한다.
- 일단 실행중인 아두이노 편집 프로그램을 모두 종료한다.

절차3) USB 케이블 재연결하기

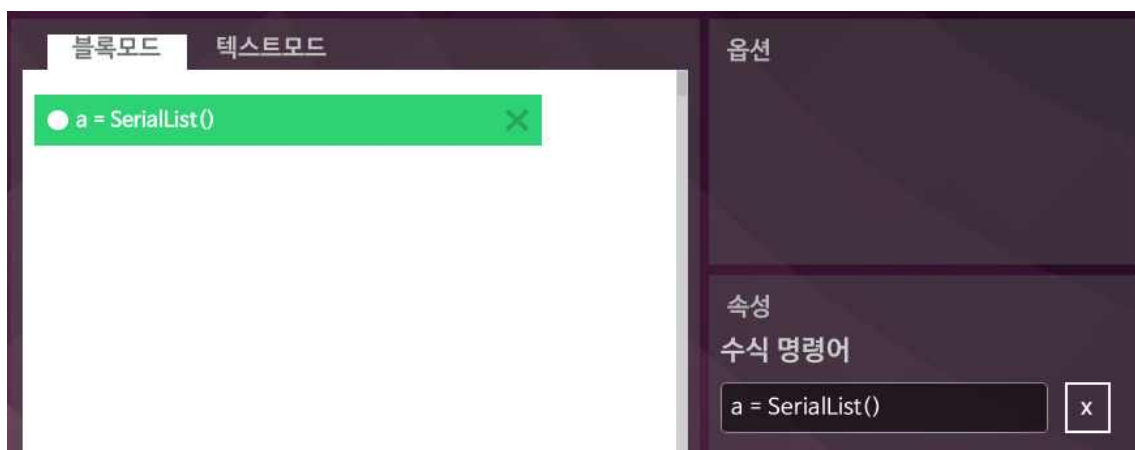
- 아두이노 보드와 PC간에 연결되어 있는 USB 케이블을 뽑다가 다시 연결해 준다.
- 이제 VR 코딩 편집기에서 아두이노 보드에 시리얼 통신 접속을 할 준비가 완료되었다.

절차4) VR 코딩 시작하기

- VR 코딩에서 시리얼 통신은 PC 모드에서만 실행 가능하다. 시리얼 통신을 하기 위해서는 COM 포트 번호와 Baudrate (통신속도) 값이 필요하다.
- 시리얼 통신 명령어는 수식 명령어를 사용하여 입력한다.



- 수식 명령어에 `a = SerialList()` 명령어를 추가한다.



- 아래의 예는 사용 가능한 통신 포트 목록을 가져온 후, 목록을 출력하는 예제이다.
- SerialList() 함수는 현재 PC에 연결되어 있는 시리얼 COM 포트 목록을 문자열 배열로 넘겨주는 함수이다.

```

a = SerialList()
for (i = 0; i < a.length; i++)
    문자라인 출력(a[i])

```

```

a = SerialList()
for (int i = 0; i < a.length; i++)
{
    PrintLine(a[i])
}

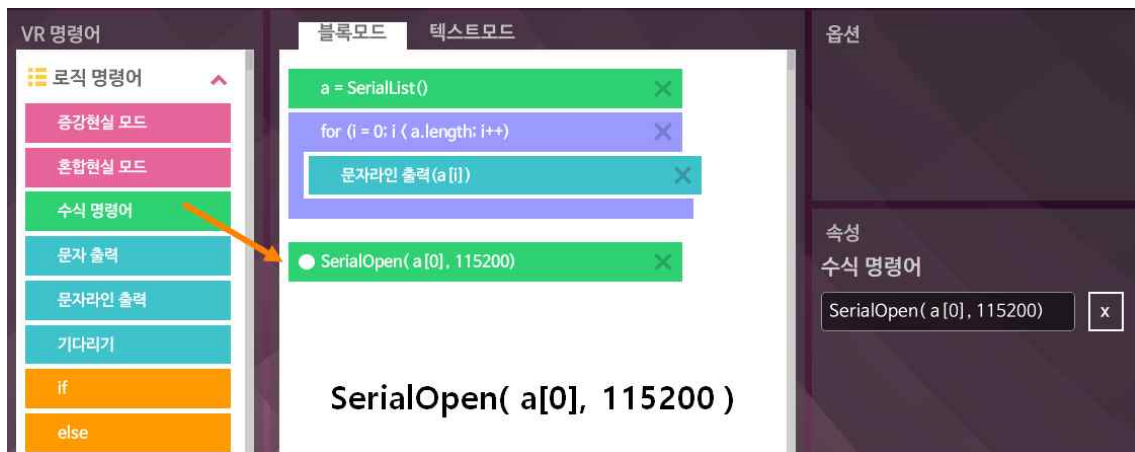
```

- 만약, 아두이노 보드와 USB 케이블로 잘 연결되어 있다면, COM 포트 이름이 화면에 표시될 것이다.

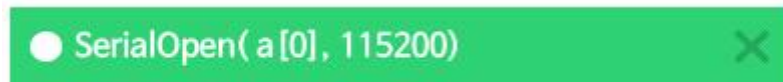


첫 번째 COM 포트 오픈하기

- 여러 개의 COM 포트 중에서 첫 번째 COM 포트를 이용하여 시리얼 통신을 오픈하려면 새로운 수식 명령어를 추가한 후, 다음과 같이 VR 코드를 작성한다.



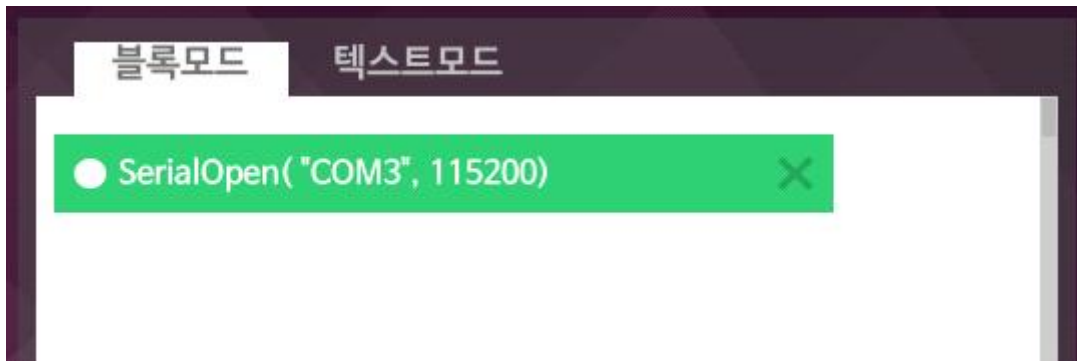
SerialOpen(COM포트 이름, 통신속도)



- 위의 예는 a[0] 즉, 배열 a 항목에서 첫 번째 항목에 있는 COM 포트 이름을 사용하고, 통신속도는 115200으로 설정하겠다는 의미이다.

COM 포트 이름을 지정하여 오픈하기

- 만약, COM 포트 이름을 이미 알고 있는 상태라면, 아래와 같이 COM 포트 이름을 직접 지정하여 Open해 주면 된다. 예를 들어 아두이노 보드와 연결되어 있는 시리얼 포트 이름이 COM3라면 다음과 같이 VR 코드에서 지정해 줄 수 있다.



SerialOpen("COM3", 115200)

- 시리얼 포트 이름을 알고 있는 경우는 SerialList로 콤포트 목록을 가져올 필요없이 위의 명령어 하나만 사용하면 시리얼 포트가 오픈된다.

VR에서 시리얼 통신으로 문자열 읽기

- SerialOpen() 명령어 이후에는 대략 200밀리초 정도 기다리는 명령어를 추가해 준다.
- 이제 loop 함수를 추가하여 반복적으로 시리얼 통신을 문자를 읽을 준비를 해 보자. 다음과 같이 loop 함수를 추가해 준다.



```
SerialOpen( "COM3", 115200)
delay(200)

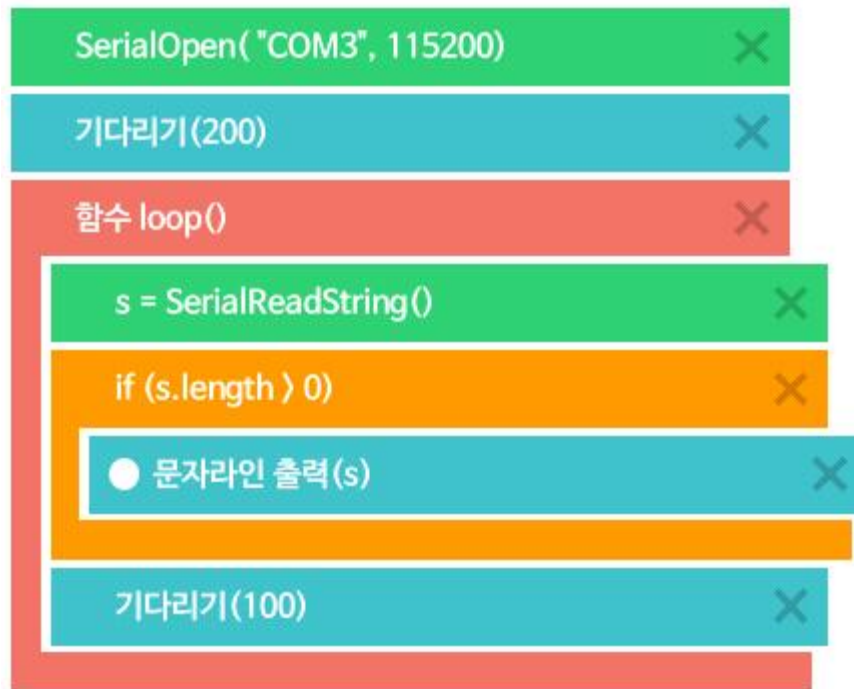
void loop()
{
}
```

- loop 함수를 안에 새로운 수식 명령어를 하나 추가한 후, 아래와 같이 문자열을 읽어오는 명령어를 추가해 준다.

```
s = SerialReadString()
```



- 읽은 문자열의 길이를 비교하여 0보다 크면 화면에 한 줄씩 출력하는 기능을 추가한다.



```
SerialOpen( "COM3", 115200)
delay(200)
void loop()
{
    s = SerialReadString()
    if (s.length > 0)
    {
        println(s)
    }

    delay(100)
}
```

시리얼 포트 목록으로 통신 열기

- 만약 시리얼 통신 포트 이름을 잘 모를 경우에는 다음과 같이 목록을 출력하는 기능을 추가한 후, 배열에서 필요한 위치에 있는 포트 이름을 활용하여 수신된 데이터를 읽을 수 있다.

```
블록모드 텍스트모드
● a = SerialList()
for (int i = 0; i < a.length; i++)
    문자라인 출력(a[i])
SerialOpen(a[0], 115200)
기다리기(200)
함수 loop()
    s = SerialReadString()
    if (s.length > 0)
        문자라인 출력(s)
    기다리기(100)
```

```
a = SerialList()
for (int i = 0; i < a.length; i++)
{
    println(a[i])
}
```

```
SerialOpen(a[0], 115200)
delay(200)
```

```
void loop()
{
    s = SerialReadString()
    if (s.length > 0)
    {
        Println(s)
    }
    Delay(100)
}
```

VR에서의 시리얼 통신 명령어

명령어	설명
SerialOpen("COM3", 115200)	통신 포트를 오픈한다.
IsSerialOpened()	현재 시리얼 통신이 오픈중이면 True, 그렇지 않으면 False 리턴
a = SerialList()	시리얼 통신 포트 목록을 문자열 배열로 넘겨준다.
SerialClose()	시리얼 통신을 종료한다.
SerialPrint(a) SerialPrint(123) SerialPrint("Hello")	괄호 안의 값을 문자열로 전송한다.
SerialPrintln(a) SerialPrintln(123) SerialPrintln("Hello")	괄호 안의 값에 줄바꿈 문자를 추가한 후 문자열로 전송한다.
SerialWrite(1) SerialWrite(1, 2) SerialWrite(1, 2, 3)	괄호 안의 값들을 바이트 값으로 변경하여 바이트 배열로 전송한다. 괄호안의 값이 1개만 있으면 1바이트 값만 전송된다.
SerialWriteBytes(byte[] a)	괄호 안의 바이트 배열을 전송한다.
a = SerialRead()	한 바이트 값을 읽어서 int 타입으로 저장한다. 만약 수신된 값이 없으면 -1 값이 저장된다.
byte[] a = SerialReadBytes()	수신된 값을 바이트 배열로 넘겨준다.
byte[] a = SerialReadBytesUntil(1024, ";")	특정한 문자 값까지의 값을 바이트 배열로 읽어서 리턴한다.
s = SerialReadLine()	줄바꿈 문자가 있는 한 라인을 읽어서 리턴한다.
s = SerialReadString()	수신된 전체 데이터를 문자열로 넘겨준다.

명령어	설명
<pre>s = SerialReadStringUntil(";")</pre>	<p>특정 문자가 있는 위치까지 값을 읽어서 문자열로 넘겨준다. 왼쪽의 예는 ";" 문자 앞까지 수신된 데이터를 문자열로 넘겨준다.</p>
<pre>float[] a = SerialGetValues(",")</pre> <pre>float[] a = SerialGetValues(" ")</pre>	<p>지정된 문자로 수신 데이터를 분리한 후, 실수형 배열로 리턴한다.</p> <p>예를 들어, 12,34,56 값이 수신된 경우 실수형 배열 a에는 12, 34, 56의 값이 저장된다.</p>
<pre>s = SerialGetStrings(",")</pre> <pre>s = SerialGetStrings(" ")</pre>	<p>지정된 문자로 수신 데이터를 분리한 후, 문자열 배열로 리턴한다.</p> <p>예를 들어, 12,34,56 값이 수신된 경우 문자열 배열 a에는 "12", "34", "56"의 값이 저장된다.</p>