



시리얼 통신(Serial Communication)

5.1 시리얼 장치

통신 분야에서 가장 널리 사용되고 있는 통신 중의 하나로 특히 임베디드 시스템에서는 놓을 수 없는 통신 방식 중에 하나입니다. RS232, RS485, UART 통신 등이 모두 시리얼 통신 방식을 사용합니다. UART, RS232, RS485 통신 모두 비동기 시리얼 통신 방식입니다. 즉 데이터 통신 라인에 클럭이 존재하지 않고, 데이터의 시작과 끝을 알리는 약속된 신호를 수신하는 쪽에서 찾아내어 데이터를 구분해야 합니다. 시리얼 통신은 데이터를 비트 단위로 전송하는 방식을 의미합니다. 그래서 시리얼 통신 전송 속도의 단위로 BPS(Bit Per Second)를 사용합니다. 라즈베리파이와 PC와의 시리얼 통신으로 대문자 'A'를 전송하는 예를 들어 보겠습니다.

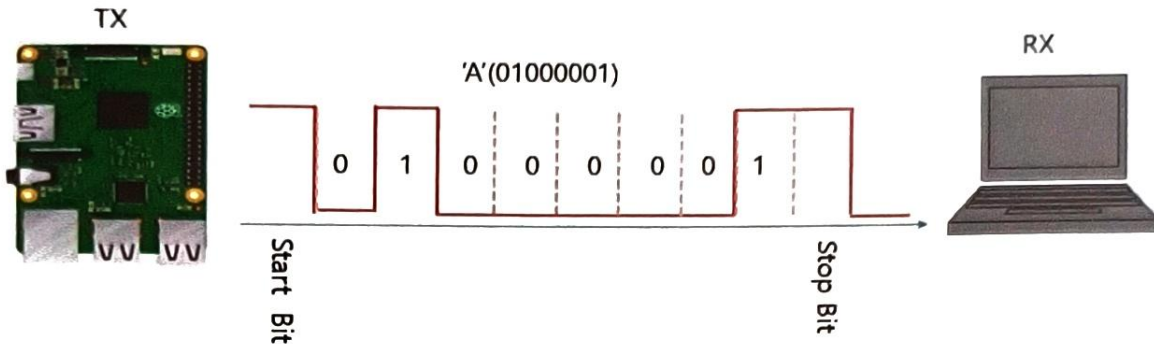
UART 비동기 시리얼 통신의 데이터 구조입니다.

비트	1	2	3	4	5	6	7	8	9	10	11
	Start Bit	데이터 비트(5~ 8비트)								Parity Bit	Stop Bit
Data	Start	D0	D1	D2	D3	D4	D5	D6	D7	Parity	Stop

데이터를 보내기 전에 항상 시작 비트(Start Bit)로 데이터를 전송한다는 신호를 먼저 보내야 합니다. 보내는 방식은 데이터 통신라인을 LOW → HIGH로 바뀌는 시점입니다. 데이터의

작을 알리고 데이터를 시간 순서에 따라 비트 단위로 전송하고, 1바이트 데이터를 전부 보냈으면 패리티 비트(Parity Bit: 약속된 형태로 사용하지 않을 수도 있음), 정지 비트(Stop Bit)를 보냅니다.

문자	10진수	16진수	2진수(비트단위)
'A'	65	0x41	1000001



시리얼 통신 개념도

'A'(1000001)를 비트 단위로 쪼개서 데이터를 전송하면 위의 그림과 같이 2진수 '1' 데이터 전송을 위해서 HIGH, 2진수 '0'을 전송하기 위해서 LOW의 전기적인 신호를 시간 순서에 따라 전송하게 됩니다. 위의 그림에서는 패리티 비트를 사용하지 않았을 때의 신호입니다. 이번 예제에서도 패리티 비트를 사용하지 않습니다. 데이터를 보내는 쪽을 TX, 받는 쪽을 RX라는 용어를 사용하고 전송 속도는 보레이트(Baudrate)로 표현합니다. 반드시 보내는 쪽과 받은 쪽은 동일한 보레이트로 통신해야 합니다. RS232 통신은 데이터를 보낼 때 클록과 데이터 라인을 별도로 사용하지 않는 대표적인 비동기 통신입니다.

참고

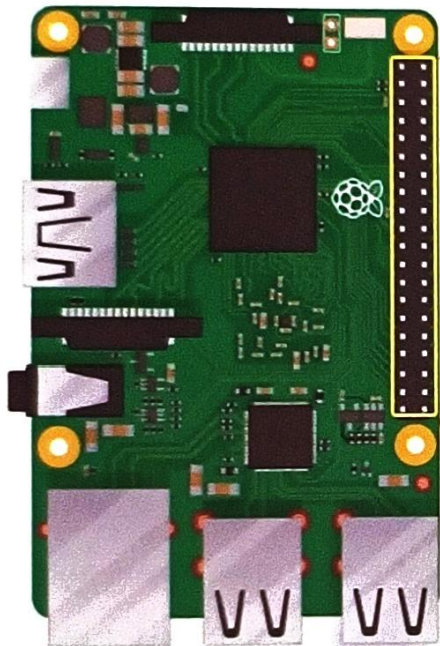
시리얼 통신 용어 정리

- 시작 비트: 통신의 시작을 의미하며 1비트를 전송하는 시간만큼 유지 1비트를 전송하는 시간은 사용하는 보레이트(Baudrate)에 따라서 달라지면 당연히 통신 속도가 빠를수록 시간도 짧아지게 됩니다.
- 데이터 비트: 데이터 비트는 5~8비트를 정해서 사용할 수 있지만 보통은 8비트를 가장 많이 사용합니다.

- 패리티 비트 : 오류 검증을 위한 패리티 값을 생성해서 보낼 수 있습니다. 사용 안 함으로 설정하면 패리티 비트 자체를 전송하지 않습니다.
- 정지 비트 : 데이터 전송을 종료합니다.

(1) 라즈베리파이 시리얼 장치

라즈베리파이에는 4개의 USB 채널과 2개의 UART 채널을 가지고 있습니다. 교재에서는 USB 포트는 사용하지 않고 GPIO 확장 포트에 연결된 UART 채널만 사용합니다.

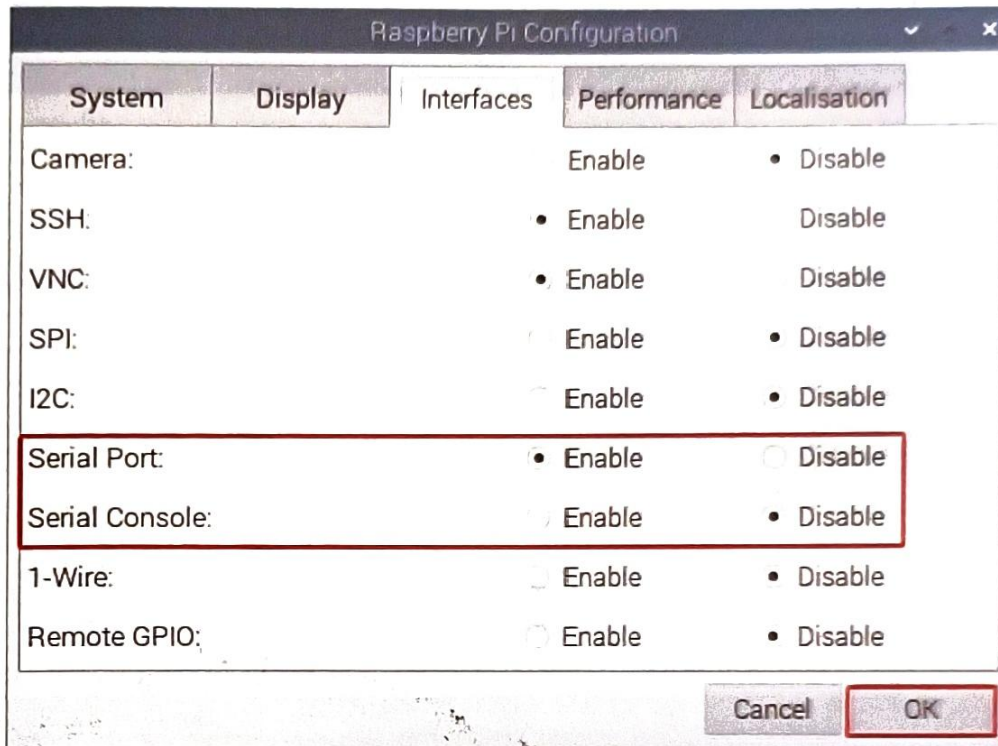


wPi	BCM	Pin	No	No	Pin	BCM	wPi
		3.3V	1	2	5V		
GPIO 08	GPIO 02	SDA1	3	4	5V		
GPIO 09	GPIO 03	SCL1	5	6	GND		
GPIO 07	GPIO 04		7	8	TXD	GPIO 14	GPIO 15
		GND	9	10	RXD	GPIO 15	GPIO 16
GPIO 00	GPIO 17		11	12		GPIO 18	GPIO 01
GPIO 02	GPIO 27		13	14	GND		
GPIO 03	GPIO 22		15	16		GPIO 23	GPIO 04
		3.3V	17	18		GPIO 24	GPIO 05
GPIO 12	GPIO 10	MOSI	19	20	GND		
GPIO 13	GPIO 09	MISO	21	22		GPIO 25	GPIO 06
GPIO 14	GPIO 11	SCLK	23	24	CE0	GPIO 08	GPIO 10
		GND	25	26	CE1	GPIO 07	GPIO 11
GPIO 30	GPIO 00	SDA0	27	28	SCL0	GPIO 01	GPIO 31
GPIO 21	GPIO 05		29	30	GND		
GPIO 22	GPIO 06		31	32		GPIO 12	GPIO 26
GPIO 23	GPIO 13		33	34	GND		
GPIO 24	GPIO 19		35	36		GPIO 16	GPIO 27
GPIO 25	GPIO 26		37	38		GPIO 20	GPIO 28
		GND	39	40		GPIO 21	GPIO 29

UART 채널	시리얼 포트	장치 이름	용도
UART0	serial0	/dev/ttyS0 or /dev/serial0	콘솔 or 시리얼 GPIO14(TXD) GPIO15(RXD)
UART1	serial1	/dev/ttyAMA0 or /dev/serial1	블루투스

(2) UART0 장치 활성화

라즈베리파이는 기본적으로 UART1 장치 사용이 Disable 되어 있습니다. 먼저 장치를 사용하기 전에 활성화를 해야 합니다. “/프로그램/기본 설정/Raspberry pi Configuration” 프로그램을 실행합니다.



Serial Port를 Enable, Serial Console은 Disable을 선택하여 저장하고 재부팅합니다. Serial Console이 Enable 되어 있으면 일반 UART 통신으로 사용할 수 없고 콘솔 로그인 창으로만 사용이 가능합니다.

Serial Port	Enable
Serial Console	Disable

Raspberry pi Configuration GUI 환경에서 수정을 해도 되고 직접 /boot/cmdline.txt 파일을 수정해도 됩니다.

```
pi@raspberrypi:~ $ sudo vi /boot/cmdline.txt
```


“console=serial0,115200” 문자열 대신에 “console=tty1”으로 수정한 다음, 반드시 재부팅을 해야 반영됩니다.

```
pi@raspberrypi: ~  
파일(F) 편집(E) 탭(T) 도움말(H)  
console=tty1 root=PARTUUID=4ca73f51-02 rootfstype=ext4 elevator=deadline fsck.re  
pair=yes rootwait quiet splash plymouth.ignore-serial-consoles
```

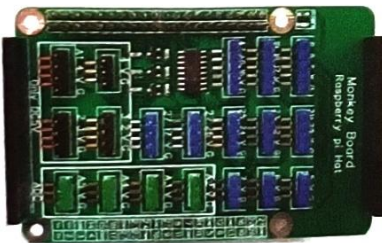
여기까지 수정을 완료하였다면, 라즈베리파이 터미널 창에서 다음과 같은 명령어를 실행하면 어떤 GPIO 포트에 시리얼 포트가 연결되었는지 확인할 수 있습니다.

```
pi@raspberrypi:~ $ raspi-gpio get 14-15
```

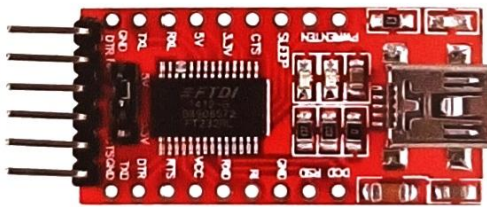
```
pi@raspberrypi: ~  
파일(F) 편집(E) 탭(T) 도움말(H)  
pi@raspberrypi:~ $ raspi-gpio get 14-15  
GPIO 14: level=1 fsel=2 alt=5 func=TXD1 pull=NONE  
GPIO 15: level=1 fsel=2 alt=5 func=RXD1 pull=UP
```

GPIO 14 : TXD, GPIO 15 : RXD로 매핑되어 있는 것을 확인할 수 있습니다.

🍷 실험에 필요한 준비물들



라즈베리파이 GPIO HAT



FT232 USB to Serial



미니 USB 케이블

배선도 및 회로

PC

5V

VCC

3.3V

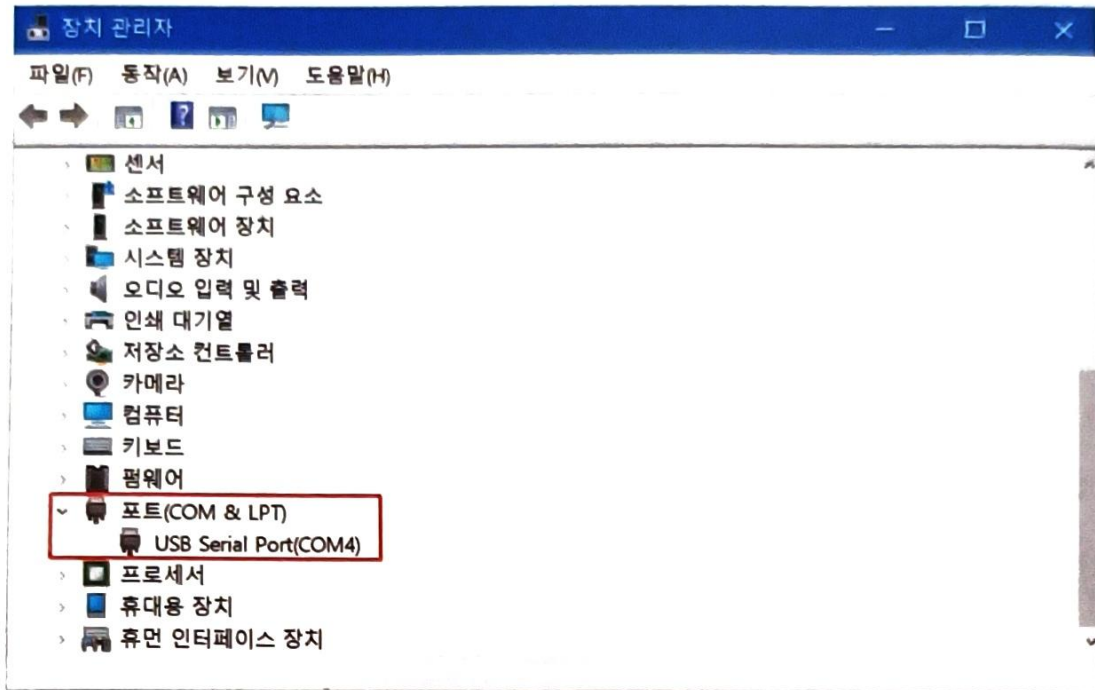
3.3V로 점퍼 설정

DTR
RXD
TXD
VCC
CTS
GND

GND
BCM.14
BCM.15


Monkey Board
Raspberry pi Hat

5.1 시리얼 장치 199



PC에 USB to Serial 장치를 연결하면 위의 그림과 같이 USB 장치 드라이버가 설정되고 가상 COM 포트가 추가됩니다. 만약 USB장치가 잘 올라오지 않는다면 아래 URL에서 FT232RL USB 드라이버를 다운로드하고 설치합니다.

<https://www.ftdichip.com/Drivers/VCP.htm>

ftdichip.com/Drivers/VCP.htm				
	Operating System	Release Date	x86 (32-bit)	x64 (64-bit)
	Windows*	2017-08-30	2.12.28	2.12.28
	Linux	-	-	-

운영체제에 맞는 드라이버를 설치하면 됩니다.

(3) PY-SERIAL 라이브러리 설치

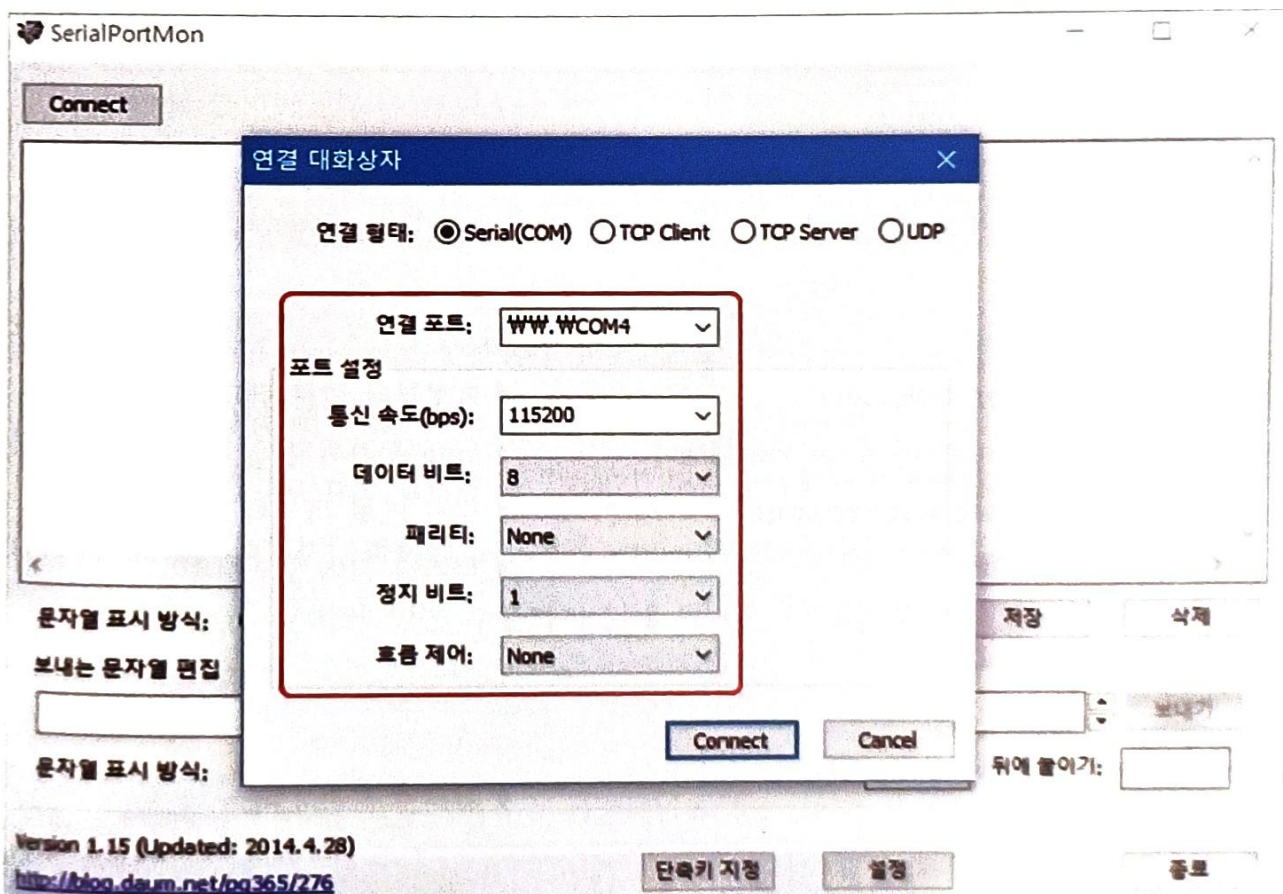
py-serial 라이브러리가 설치가 되어 있지 않다면 파이썬에서 serial 포트를 간단하게 제어하게 해주는 py-serial 라이브러리를 설치합니다. 일반적으로는 이미 설치되어 있습니다.

```
pi@raspberrypi:~ $ sudo apt-get install python-serial
```

(4) PC용 시리얼 프로그램 설치

PC에서도 시리얼 장치를 열고 데이터를 수신하거나 송신이 가능한 시리얼 프로그램이 있어야 합니다. 어떤 것을 사용해도 무방합니다. 본 교재에서는 SerialPortMon.exe 라는 공개 프로그램을 사용했습니다.

<https://cafe.naver.com/codingblock/48> URL에서 다운로드 가능합니다.



- 연결 형태 Serial
- 연결 포트 PC의 장치관리에 추가된 가상 시리얼 COM 포트 번호를 선택
- 통신 속도 115200

프로그램을 실행하고 위와 같이 설정한 다음 'Connect' 합니다.

다음 파일을 examples\serial\3\serial_1.py

```

1  import RPi.GPIO as GPIO          # RPi GPIO 패키지 사용
2  import serial                     # pyserial 패키지 사용
3
4  ser = serial.Serial( dev=serial.DEV1, "115200" )  # 115200bps 통신 속도로 설정
5  ser.close()                       # 포트를 먼저 Close
6  ser.open()                        # 시리얼 포트 Open
7
8  str = b'Python Serial Example\n'  # 포트 전송할 바이트 데이터
9  n = ser.write(str)                # 바이트 데이터 전송
10
11  while True:
12      if ser.readable():             # PC로부터 받은 데이터가 있으면
13          response = ser.readline()  # '\n' 문자까지 읽고
14          ser.write(response)        # 다시 포트 데이터 전송
15          print(response)            # 터미널 창에 받은 데이터 출력
16
17  except KeyboardInterrupt:
18      pass
19  finally:
20      ser.close()                   # 시리얼 포트 Close

```

(실행 결과)

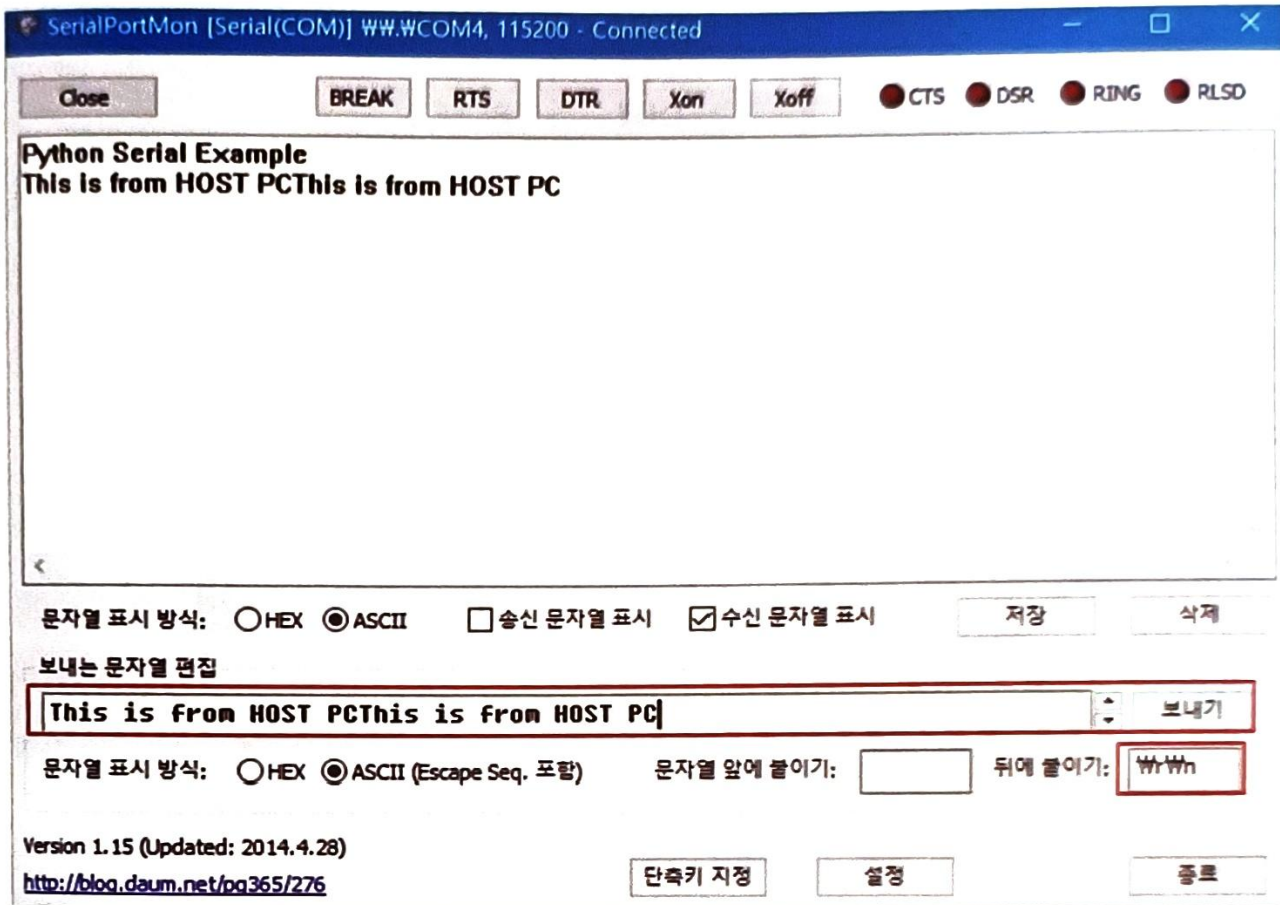
```

# PC로부터 전송받은 데이터를 출력
b'This is from HOST PC\nThis is from HOST PC\n'

```

(5) PC와 시리얼 통신

이제 파이썬 프로그램을 실행하면 PC 시리얼 프로그램에서 다음과 같이 표시됩니다.



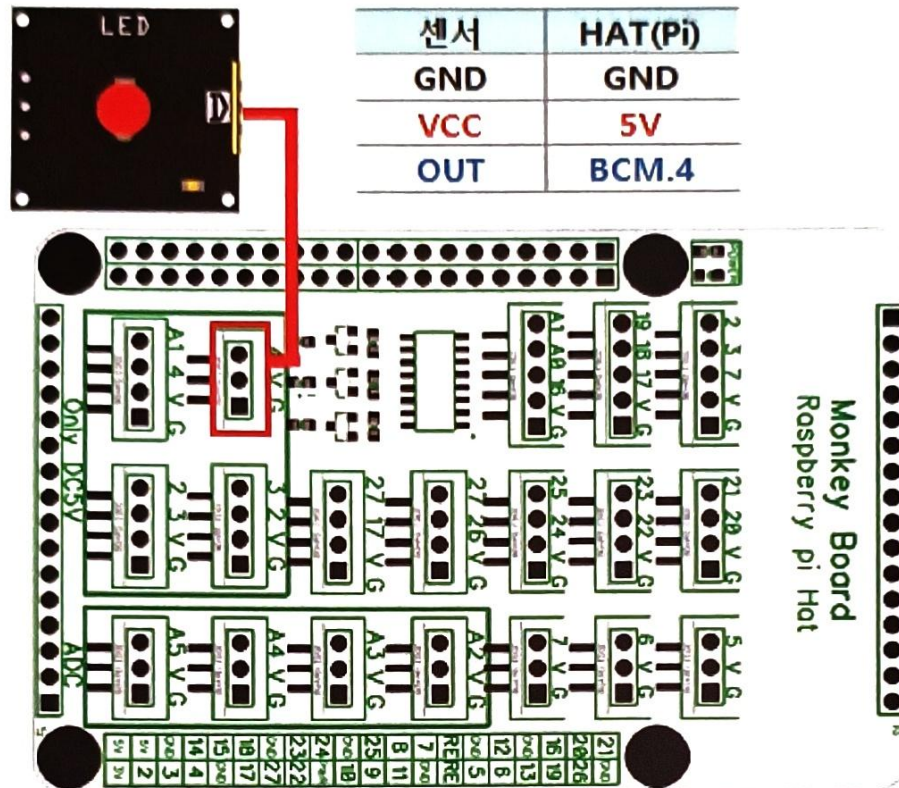
“Python Serial Example” 문자열이 시리얼 프로그램 창에 나타나고

- 보내는 문자열 편집 창: This is from HOST PThis is from HOST PC

- 뒤에 붙이기: \r\n을 입력한 후 “보내기” 버튼을 누르면 라즈베리파이 파이썬 실행 창에 문자가 표시됩니다. 파이썬 코드에서 ser.readline() 함수를 사용했기 때문에 데이터를 읽을 때 “\r\n” 문자가 있는 부분까지 읽게 됩니다. 한 가지 더 재미있는 실험을 해보도록 하겠습니다.

윈도 PC에서 터미널 프로그램으로 라즈베리파이에 시리얼로 접속을 해서 라즈베리파이에 연결된 LED를 ON/OFF 하는 실습을 해보도록 하겠습니다.

배선도 및 회로



라즈베리파이 GPIO HAT 연결

- "ON\r\n" 문자열 수신 : LED 켜기
- "OFF\r\n" 문자열 수신 : LED 끄기

실습 파일 : examples/serial/5.1_serial_2.py

```

1  import RPi.GPIO as GPIO                # RPi.GPIO 패키지 사용
2  import serial                          # py-serial 패키지 사용
3
4  LED=4 # LED포트 정의
5  GPIO.setmode(GPIO.BCM)                # BCM 모드 사용
6  GPIO.setup(LED, GPIO.OUT)              # LED를 출력으로 설정
7
8  ser = serial.Serial('/dev/serial0', 115200) # 115200bps 통신 속도로 설정
9  ser.close()                            # 포트를 먼저 Close

```

```

10  ser.open()                                # 시리얼 포트 Open
11
12  str = b'Serial LED Control\r\n'          # PC로 전송할 바이트 데이터
13  n = ser.write(str)                        # 바이트 데이터 전송
14  try:
15      while True:
16          if ser.readable():                # PC로부터 받은 데이터가 있다
17              response = ser.readline()     # \r\n 문자까지 읽기
18              if response == b'ON\r\n':     # "ON" 데이터 수신
19                  GPIO.output(LED, True)    # LED 켜기
20              elif response == b'OFF\r\n':  # OFF 데이터 수신
21                  GPIO.output(LED, False)   # LED 끄기
22
23              print(response)
24  except KeyboardInterrupt:
25      pass
26  finally:
27      ser.close()

```

〈실행 결과〉

PC로부터 전송받은 데이터를 출력

b'OFF\r\n'

b'ON\r\n'

PC 시리얼 프로그램에서 "ON"과 "OFF" 문자열을 입력해서 LED가 제어가 되는지 확인합니다.

