





What is a Serial Terminal?

Basics:

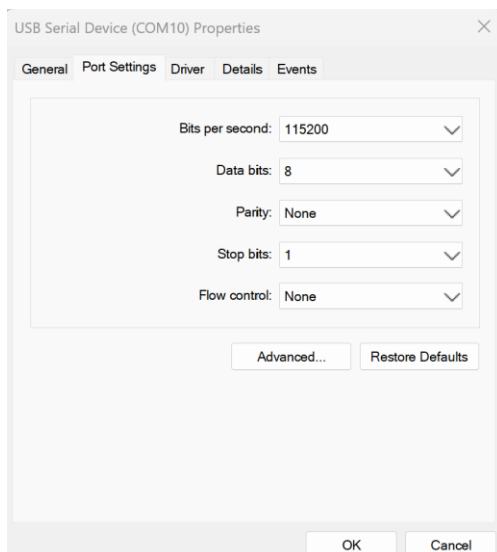
A serial terminal is a window that outputs data line by line of the device plugged into the computer you're reading from. This data comes in the form of strings.

```
Passthrough to Module 1, Serial Port 3 Active
Connected.
[,,,238,,,,,67]
[,,,238,,,,,67]
```

There are many ports that can be read from, the specific port associated with the device plugged into the computer is known as the COM port and can be identified in the device manager. As seen here, the ports plugged into the computer can be read from the ports tab

- ▼  Ports (COM & LPT)
 -  USB Serial Device (COM10)
 -  USB Serial Device (COM8)
 -  USB Serial Device (COM9)

Each serial connection is associated with certain parameters so that the data is streamed to the device correctly. These parameters are the COM port, the baud rate, data bits the stop bit, parity bit and flow control. Most likely the stop bit is always set to 1 and flow control is set to none.



Serial data is associated with a communication protocol known as UART and contains these parameters. It is a sequential “window” of data that gets fed into the port one at a time hence the same “serial,” or in “series.” This is analogous to a series circuit of resistors.

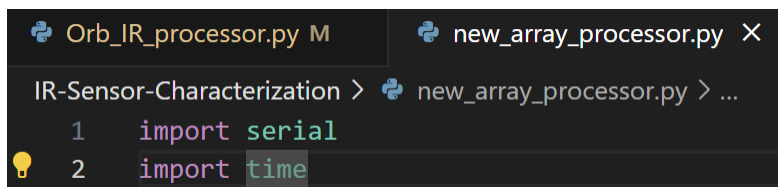
The amount of data in each sequence is denoted by the “data bits” And is stopped once the stop bit is reached and is continuously fed into the port used. The speed in which this occurs is known as the “baud rate.” If you would like to know more about this protocol, search up “UART communications protocol.”

Within each line, there will be coded information in the 8 bit binary string that comes in. This is used to communicate certain information within a digital system, but that is a more general definition. PuTTY is a great terminal interface to access serial data from a specific port.

Read the documentation for PuTTY is in the Git Hub repository for this project.

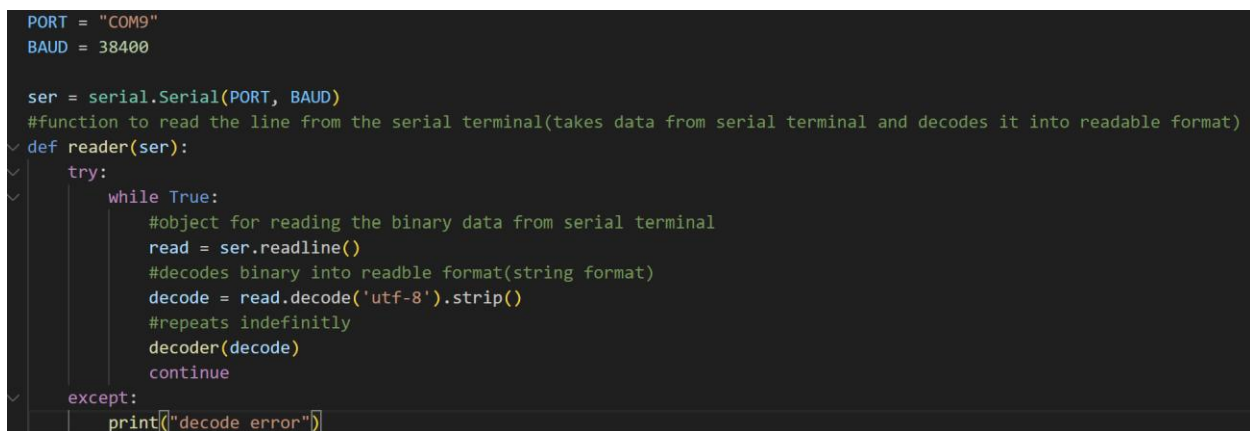
Once you have read that document, follow the instructions.

Make a script to read the data. Python installed with the “pyserial” library can be used to extract and parse this information.



```
Orb_IR_processor.py M  new_array_processor.py X
IR-Sensor-Characterization > new_array_processor.py > ...
1  import serial
2  import time
```

The practice for doing this is generally writing a function to extract and decode each line first.



```
PORT = "COM9"
BAUD = 38400

ser = serial.Serial(PORT, BAUD)
#function to read the line from the serial terminal(takes data from serial terminal and decodes it into readable format)
def reader(ser):
    try:
        while True:
            #object for reading the binary data from serial terminal
            read = ser.readline()
            #decodes binary into readable format(string format)
            decode = read.decode('utf-8').strip()
            #repeats indefinitely
            decoder(decode)
            continue
    except:
        print("decode error")
```

The object associated with the serial data is defined as “ser” and contains the port information such as the COM port and the baud rate. The parameters for “ser” are defined in all caps as constants.

```
PORT = "COM9"
BAUD = 38400

ser = serial.Serial(PORT, BAUD)
```

Make sure to copy this information from your device manager so the right information is being fed into the script.

Next, make a function to decode each line into a user readable string that can be further processed.

```
def reader(ser):
    try:
        while True:
            #object for reading the binary data from serial terminal
            read = ser.readline()
            #decodes binary into readable format(string format)
            decode = read.decode('utf-8').strip()
            #repeats indefinitely
            decoder(decode)
            continue
    except:
        print("decode error")
```

The parameter for this function would be the serial object, the functionality for each operation is commented in the function.

This script runs indefinitely, hence the while loop, and takes each line and decodes it on a new line.

Once you have this, take the string and process it as you would any other string given to you in a script.