# Instructions

1. Dependencies

* Pyserial

1. Hardware

* Arduino uno
* DS18b20 Digital thermometer
* USB type A to USB type mini B connector (laptop to Arduino connection)
* About 5 Male to male jumper wires

1. Tools
   1. Arduino IDE
   2. SPYDER IDE
2. Steps to follow
   1. Wire the DS18b20 thermometer sensor to the Arduino board as directed in *Lesson 9 DS18B20 DIGITAL TEMPERATURE SENSOR MODULE* of the Elegoo 37 sensor kit guide which can be found at [Elegoo 37 sensor kit guide pdf](https://images-eu.ssl-images-amazon.com/images/I/C1lrpIfADaS.pdf) (The power cord, ground wire and S port of DS18B20 are connected to +5v, GND and D2 port of the Arduino uno, other digital ports can also be used instead of D2.
   2. Upload the arduino\_send.ino sketch to the Arduino board using the Arduino IDE. At the time of this experiment, I have used the Arduino Uno.
   3. After connecting the Arduino uno to the PC, make sure that the sketch works by using print statements.
   4. By going to “device manager” in your PC’s control panel, find out the name which is associated with the Arduino and save it for future use in the Python sketch. Refer to Figure 1. The name of my device is “USB-SERIAL CH340”. This name is the unique identifier which will be used later in the Python script on the receiving side to identify and connect to a particular serial device irrespective of the USB port which it is connected to.

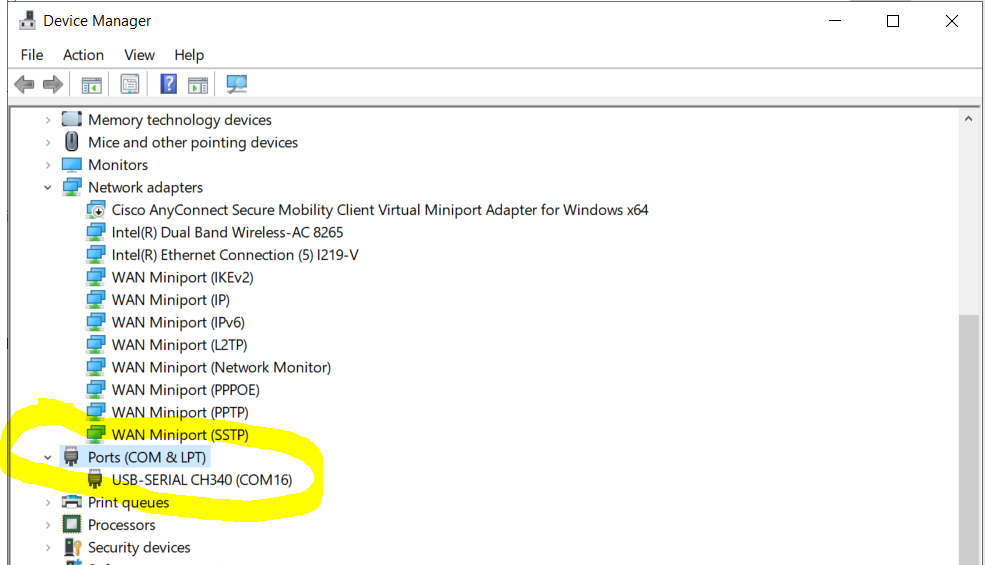


Figure 1:Identifying the name of your USB device using Device manager.

* 1. Open Anaconda and launch SPYDER. Open the python\_receive.py script in the SPYDER IDE. Use the name which we found in point 4.d on page 1 of this document to replace the name in line 51 in the python\_receive.py script.
  2. Run the python script from the SPYDER IDE and observe the progress in the command window. The execution can be terminated by pressing ctrl+c which is the keyboard interrupt. The interrupt will work only while the command window is the last actively selected window.

## Common issues and their solutions

1. While executing the python\_receive.py script, there is an exception error stating that the serial device is unavailable and the code execution stops.

Caused due to the serial port not being closed during the execution of the script on a previous run. The serial port will not be closed if the code execution raises an exception during the run causing the code to halt at the line raising an exception. The ser.close() method does not get executed. When encountered with this error, one can close the serial port by physically removing the USB connection. Removing the physical connection will close the port. After this has been done, the USB cable can be attached again to start over with the communication.

1. The Arduino IDE is unable to upload the sketch on to the arduino board.

Check that the right COM port has been selected in the Arduino IDE before initiating sketch upload.