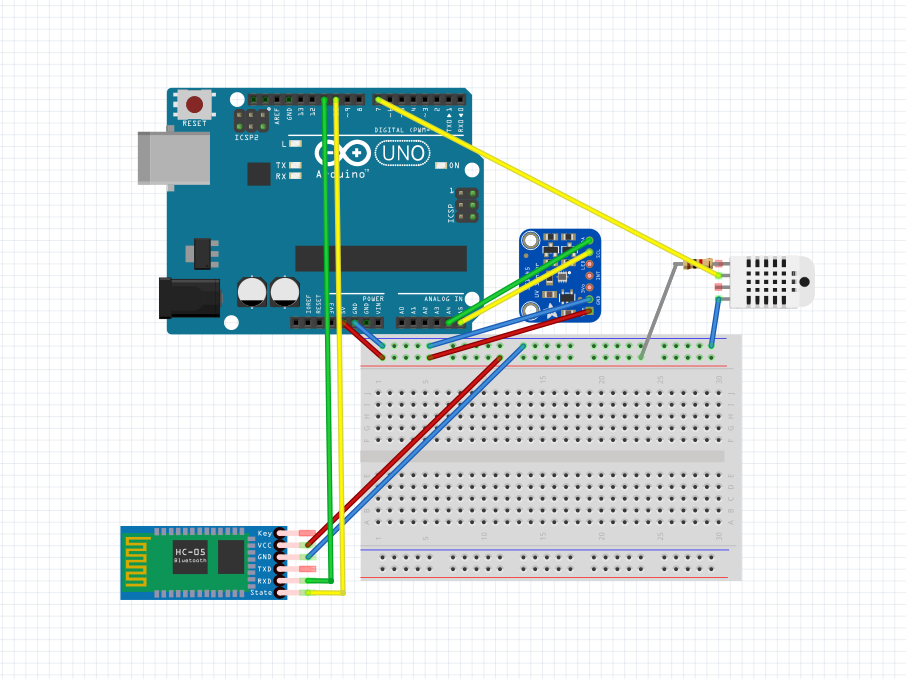
**Arduino Documentation**

1. **Hardware Design**

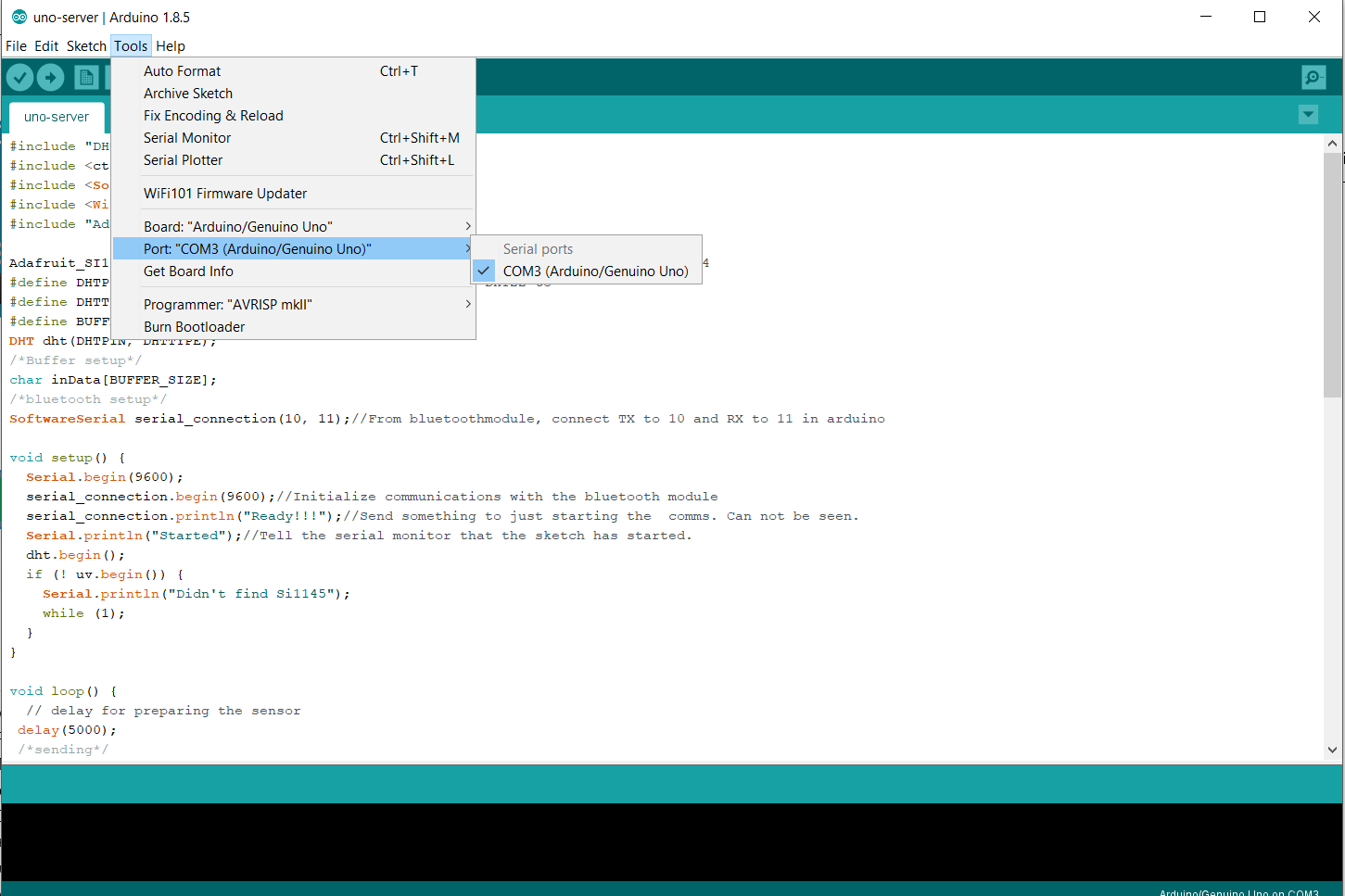
This part will demonstrate the way Arduino and the sensor connect to each other.



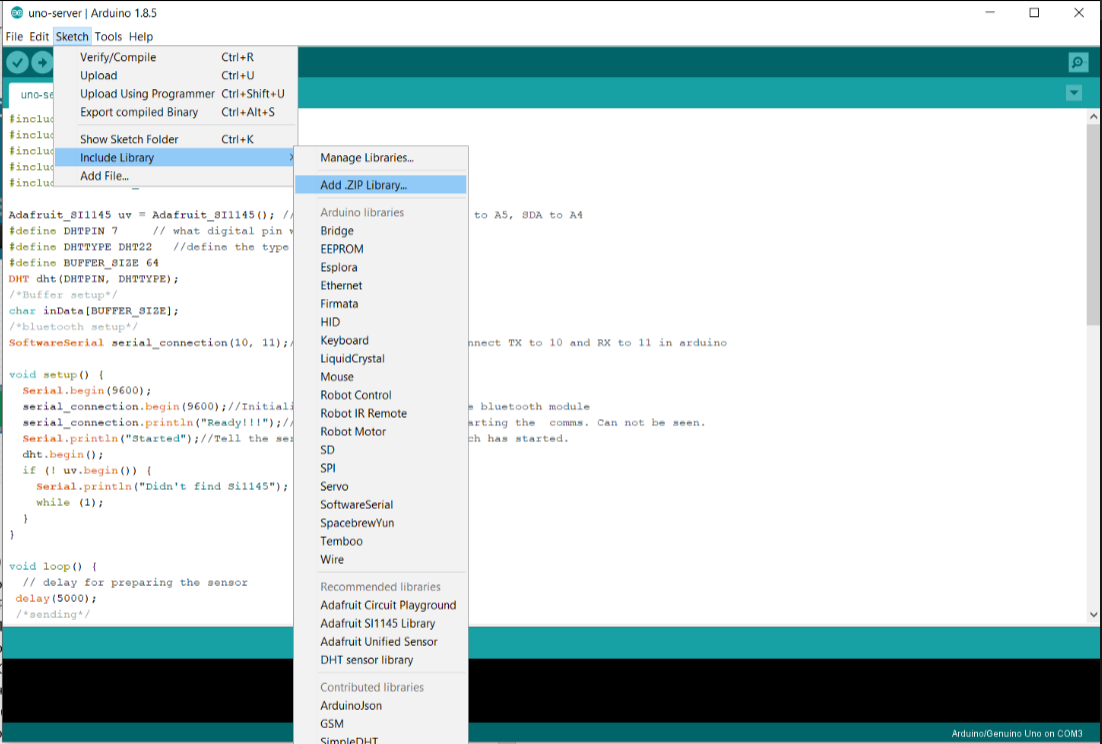
* Connection scheme:
  + Bluetooth HC-05 module:
    - Digital pin:
      * PX is connected to Pin 11 of digital pin in the Arduino board.
      * TX is connected to Pin 10 of digital pin in the Arduino board.
    - Power pin:
      * GND is connected to GND of power pin in the Arduino board.
      * +5V (Vcc) is connected to the 5V line in breadboard.
  + Temperature Humidity DHT22 sensor:
    - Digital pin:
      * VOut is connected to Pin 7 of digital pin in the Arduino board.
    - Power pin:
      * GND (-) is connected to GND line in breadboard.
      * Vcc (+) is connected to the 5V line onboard.
  + SI 1145 UV sensor:
    - Analog pin:
      * Connect SCL to A5
      * Connect SDA to A4
    - Power pin:
      * GND is connected to GND of power pin in the Arduino board.
      * +5V (Vcc) is connected to the 5V line in breadboard.

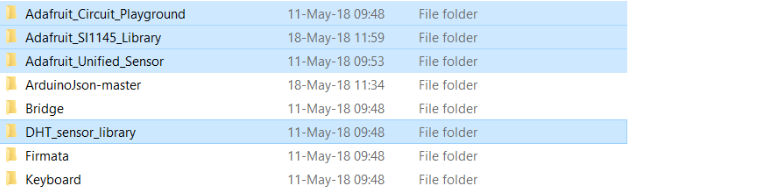
1. **Software Design**

1. **Setup environment**
   1. \_ Install Arduino: <https://www.arduino.cc/en/Main/Software>
   2. Setting up the Board, Port and programmer type on Arduino IDE



* 1. Adding needed lib to Arduino IDE



4 libs need to include (highlighted) 

<https://github.com/adafruit/Adafruit_CircuitPlayground>

<https://github.com/adafruit/Adafruit_SI1145_Library>

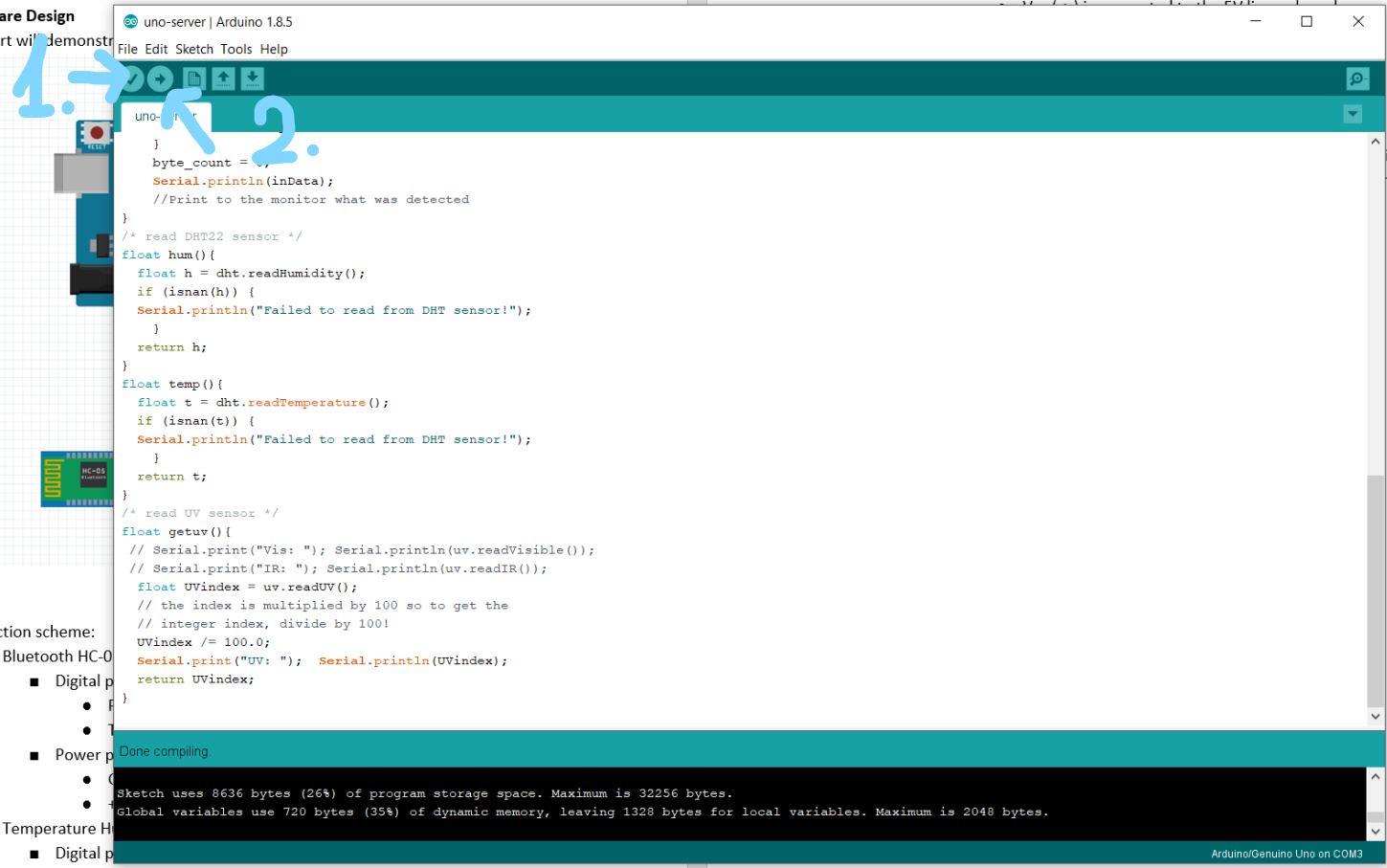
<https://github.com/adafruit/Adafruit_Sensor>

<https://github.com/adafruit/DHT-sensor-library>

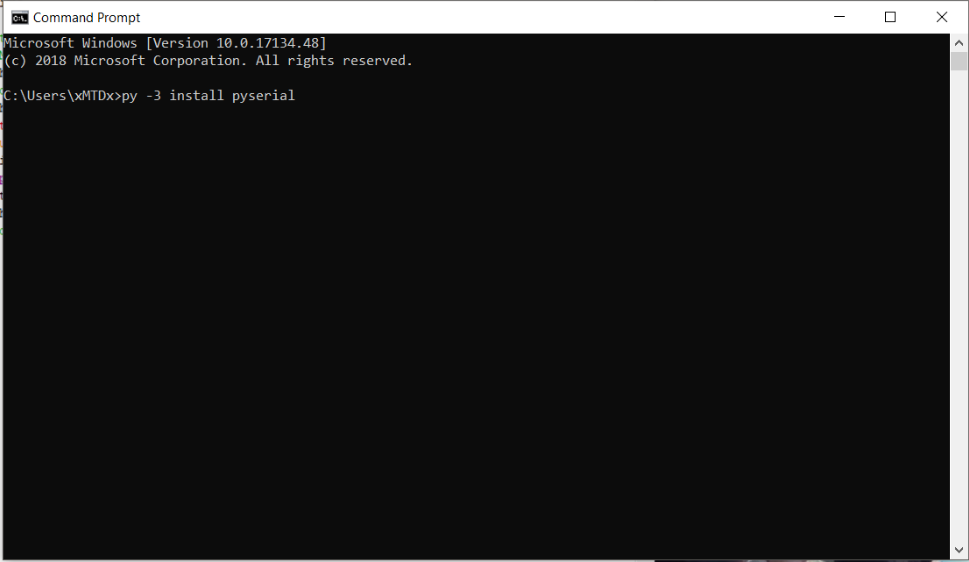
* 1. Verified and upload the code to Uno R3 board

1.Verified code to found errors

2.Upload the code to the Uno board



* 1. Install python3: <https://www.python.org/downloads/> & pyserial



1. **Code structure**

The Arduino now can send the data from attached sensor to server write on python3 using Pyserial (which can run on Windows, Linux or OSX system)

**Arduino code:**

#include "DHT.h"

#include <ctype.h>

#include <SoftwareSerial.h>

#include <Wire.h>

#include "Adafruit\_SI1145.h"

Adafruit\_SI1145 uv = Adafruit\_SI1145(); //connect the UV sensor SCL to A5, SDA to A4

#define DHTPIN 7 // what digital pin we're connected DHT22 to

#define DHTTYPE DHT22 //define the type of DHT sensor

#define BUFFER\_SIZE 64

DHT dht(DHTPIN, DHTTYPE);

/\*Buffer setup\*/

char inData[BUFFER\_SIZE];

/\*bluetooth setup\*/

SoftwareSerial serial\_connection(10, 11);//From bluetoothmodule, connect TX to 10 and RX to 11 in arduino

void setup() {

Serial.begin(9600);

serial\_connection.begin(9600);//Initialize communications with the bluetooth module

serial\_connection.println("Ready!!!");//Send something to just starting the comms. Can not be seen.

Serial.println("Started");//Tell the serial monitor that the sketch has started.

dht.begin();

if (! uv.begin()) {

Serial.println("Didn't find Si1145");

while (1);

}

}

void loop() {

// delay for preparing the sensor

delay(5000);

/\*sending\*/

serial\_connection.print("{ humiditi: ");/\*sending humidity data\*/

serial\_connection.println(hum());

serial\_connection.print("temprature: ");/\*sending temp data\*/

serial\_connection.println(temp());

serial\_connection.print("UVindex: ");/\*sending UV data\*/

serial\_connection.println(getuv());

serial\_connection.println("}");

/\*receiving\*/

int i=0;

byte byte\_count=serial\_connection.available();//This gets the number of bytes that were sent by the python script

for(i=0;i<byte\_count;i++)//Handle the number of incoming bytes

{

inData[i]=serial\_connection.read();

}

byte\_count = 0;

Serial.println(inData);

//Print to the monitor what was detected

}

/\* read DHT22 sensor \*/

float hum(){

float h = dht.readHumidity();

if (isnan(h)) {

Serial.println("Failed to read from DHT sensor!");

}

return h;

}

float temp(){

float t = dht.readTemperature();

if (isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

}

return t;

}

/\* read UV sensor \*/

float getuv(){

// Serial.print("Vis: "); Serial.println(uv.readVisible());//read the Vis

// Serial.print("IR: "); Serial.println(uv.readIR()); // read the IR

float UVindex = uv.readUV(); //read the UV

// the index is multiplied by 100 so to get the

// integer index, divide by 100!

UVindex /= 100.0;

Serial.print("UV: ");

Serial.println(UVindex);

return UVindex;

}

**Server code:**

import serial

import time

print("Start")

port="COM4"#This will be different for various devices and on windows it will probably be a COM port.

bluetooth=serial.Serial(port, 9600, timeout=1)#Start communications with the bluetooth unit

print("Connected")

bluetooth.flushInput() #This gives the bluetooth a little kick

#bluetooth.write(b"data test") #sending back a string/command to Arduino

while True :

input\_data=bluetooth.readline()#This reads the incoming data. In this particular example it will be the "Hello from Blue" line

print(input\_data.decode())

time.sleep(1) #A pause between bursts

bluetooth.close() #Otherwise the connection will remain open until a timeout which ties up the /dev/thingamabob

print("Done")

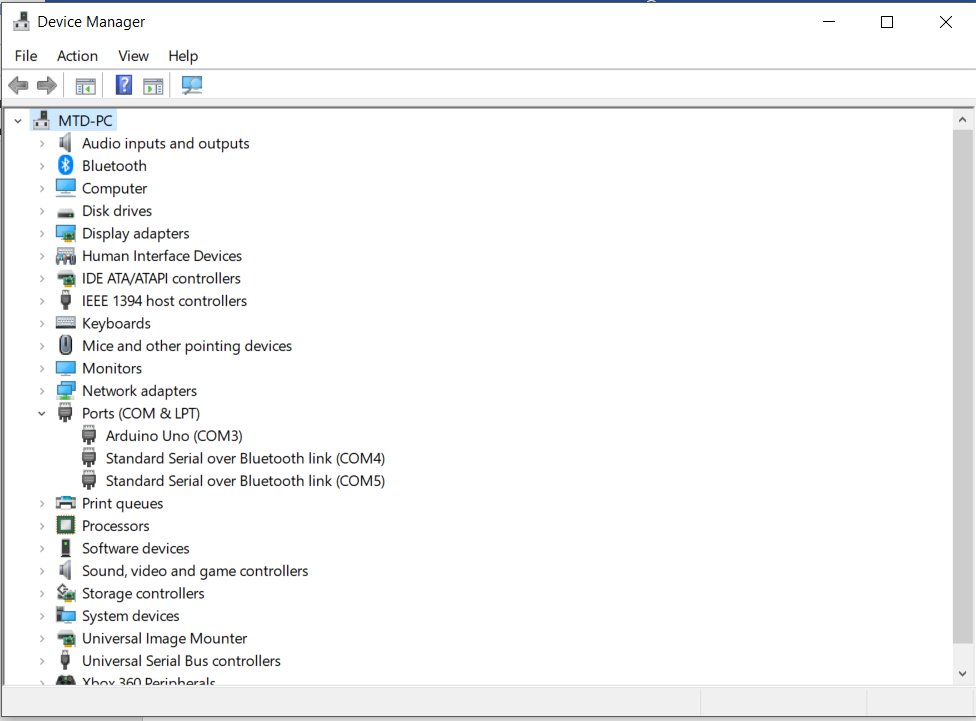
1. **Running the code**

After getting the all the Arduino module wired and code uploaded, pair the Bluetooth of the Arduino with the PC or Ev3 (

default password for Bluetooth module when paring: 1234

After paired, the python server need to know the Serial Bluetooth Port in order get the data

Example: In Windows, when Bluetooth paired, user can find the port in Device manager

like the Bluetooth serial at COM4 => changing the port in the python server code (at line 5) to port="COM4"

In Linux system the port look like: /dev/rfcomm0 (after paired, using blueman to active serial device )

In OSX system the port look like: /dev/tty.HC-05-DevB (use ls -lart /dev command to find the right port)