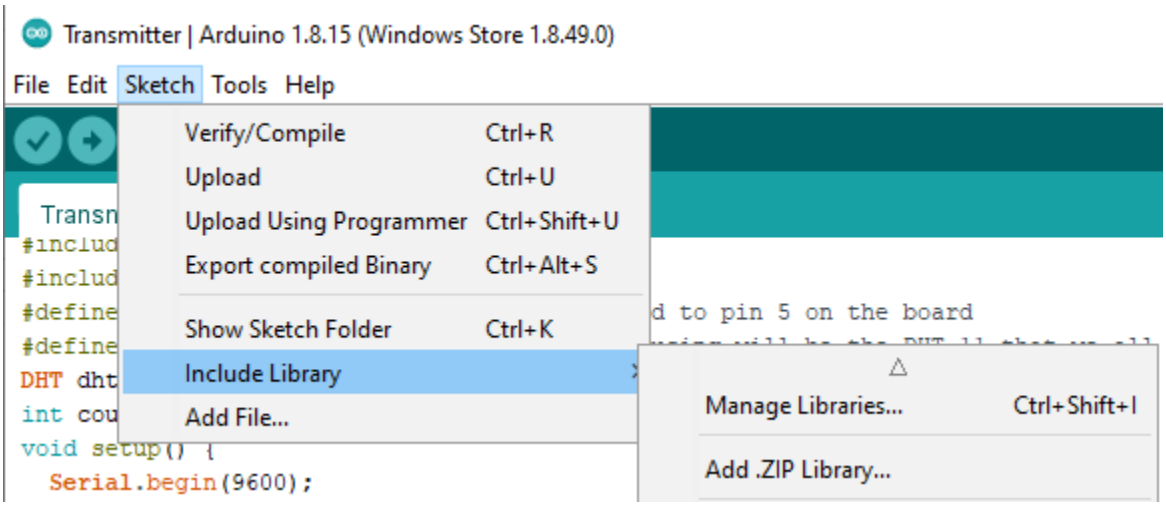
All code can be found via github: https://github.com/JCarter19999/UAV-UAS

Data Transferring:

In order to get the data from the sensor of choice to the ground station, we will utilize two microcontrollers/processors that are paired with LoRa shields to act as a receiver and transmitter.

Set-Up:

1) *Install the Arduino LoRa Library*
    a) Open the Arduino IDE and select the **Sketch** tab
    b) Scroll down to **Include Library** and select **Manage Libraries…**



    c) Type **LoRa** into the library manager search bar and select the library by **Sandeep Mistry** (Version 0.8.0 as of 5/22/2021) and select **Install**

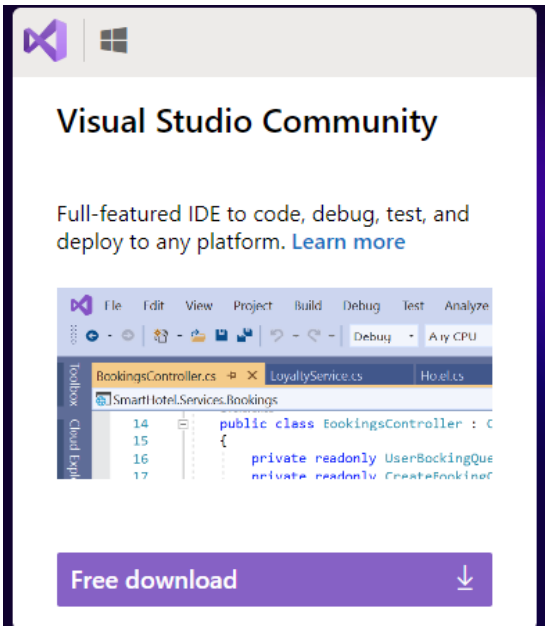
    d) Once the library is installed then the set up for the transmitter and receiver is complete

NOTE: Code inspired by https://www.youtube.com/watch?v=3yoQXtaN3wE. It may prove useful to watch this video if you are having any trouble following along with the written instructions.

GUI

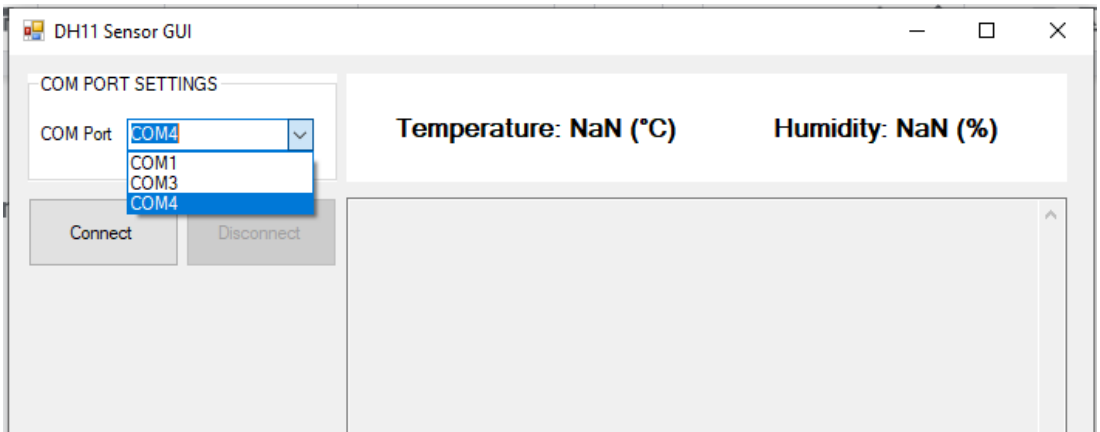
Initial Set-Up:

1) *Installing Microsoft Visual Studio*
    a) Go to **https://visualstudio.microsoft.com/** and select **Free Visual Studio**, then from there select and install the **Visual Studio Community Edition**.



    b) Once everything is installed, go to the github repository and download and open the '**DH11Sensor.sln**' file using Visual Studio Community (Do not open with Visual Studio Code), to begin editing it.

2) *Connecting to a COM Port*
    a) Connecting to the receiving Arduino is a matter of figuring out which COM Port it's plugged into and then selecting it from the drop down menu and pressing the connect button.



3) *Data Parsing*

a) The incoming data being transmitted to the receiver was structured in this way:

```
@25.10A39.00B
@25.10A39.00B
@25.10A39.00B
```

b) In order to add more arguments in the case of more data parameters, then you want to change the data being transmitted from the Arduino to follow the trend "@data1Adata2Bdata3Cdata4D[...]\n' where the \n signals to the GUI that the line of data has ended (observe the ReadTo command)

```
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    string dataIn = serialPort1.ReadTo("\n");
    Data_Parsing(dataIn);
```

c) From there you need to edit the data parsing section to accommodate any increase in parameters by adding more variables while maintaining the **sbyte IndexOf(letter) = (sbyte)data.IndexOf('letter')** format. The video linked above goes more in depth into this process.

```
1 reference
private void Data_Parsing(string data)
{
    sbyte IndexOf_startDataCharacter = (sbyte)data.IndexOf("@");
    sbyte IndexOfB = (sbyte)data.IndexOf("B");
    sbyte IndexOfA = (sbyte)data.IndexOf("A");



    if (IndexOfA!=-1 && IndexOfB!=-1 && IndexOf_startDataCharacter != -1)
    {
        try
        {
            string str_temp = data.Substring(IndexOf_startDataCharacter + 1, (IndexOfA - IndexOf_startDataCharacter) - 1);
            string str_humid = data.Substring(IndexOfA + 1, (IndexOfB - IndexOfA) - 1);

            temp = Convert.ToDouble(str_temp);
            humid = Convert.ToDouble(str_humid);
            dataUpdate = true;

        }
        catch (Exception)
        {

        }
    }
    else
    {
        dataUpdate = false;
    }
```

SQL Database

For this project I chose to use Microsoft SQL Server Management Studio 18, since it allows for a very easy connection to the Microsoft Visual Studio GUI.

Set-Up:

1) Install SQL Server Management Studio (https://www.youtube.com/watch?v=QsXWszvjMBM)
2) Once installed follow this tutorial to set up the database and table with the proper columns. (https://youtu.be/VT76ruswmHo)
    a) Once your database is constructed and the connection is made, replace my database string with whatever value is in your connection string box.

```
1 reference
82  private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
83  {
84      string dataIn = serialPort1.ReadTo("\n");
85      Data_Parsing(dataIn);
86      this.BeginInvoke(new EventHandler(show_data));
87      SqlConnection con = new SqlConnection("Data Source=JOEYS-DESKTOP-C\\TEW_SQLEXPRESS;Initial Catalog=DH11Sensor;Integrated Security=True");
88      // Replace the inner line of code with whatever is in your connection string box in the Server Explorer.
89      con.Open();
90      SqlCommand cmd = new SqlCommand("insert into dbo.weather(timeCollected,temperature, humidity) values ('" + dt.ToString("yyyy-MM-dd HH:mm:s
91      int i = cmd.ExecuteNonQuery();
92      con.Close();
93  }
94
```

    b) From there you want to change your SqlCommand function to insert into the name of your database (mine in this case was dbo.weather) alongside the column names in parenthesis, and then you want to add the values command where you will choose what sensor data to input into each column.