

Summer Vacation Task

By Shreya Sandilya

17BEE1159

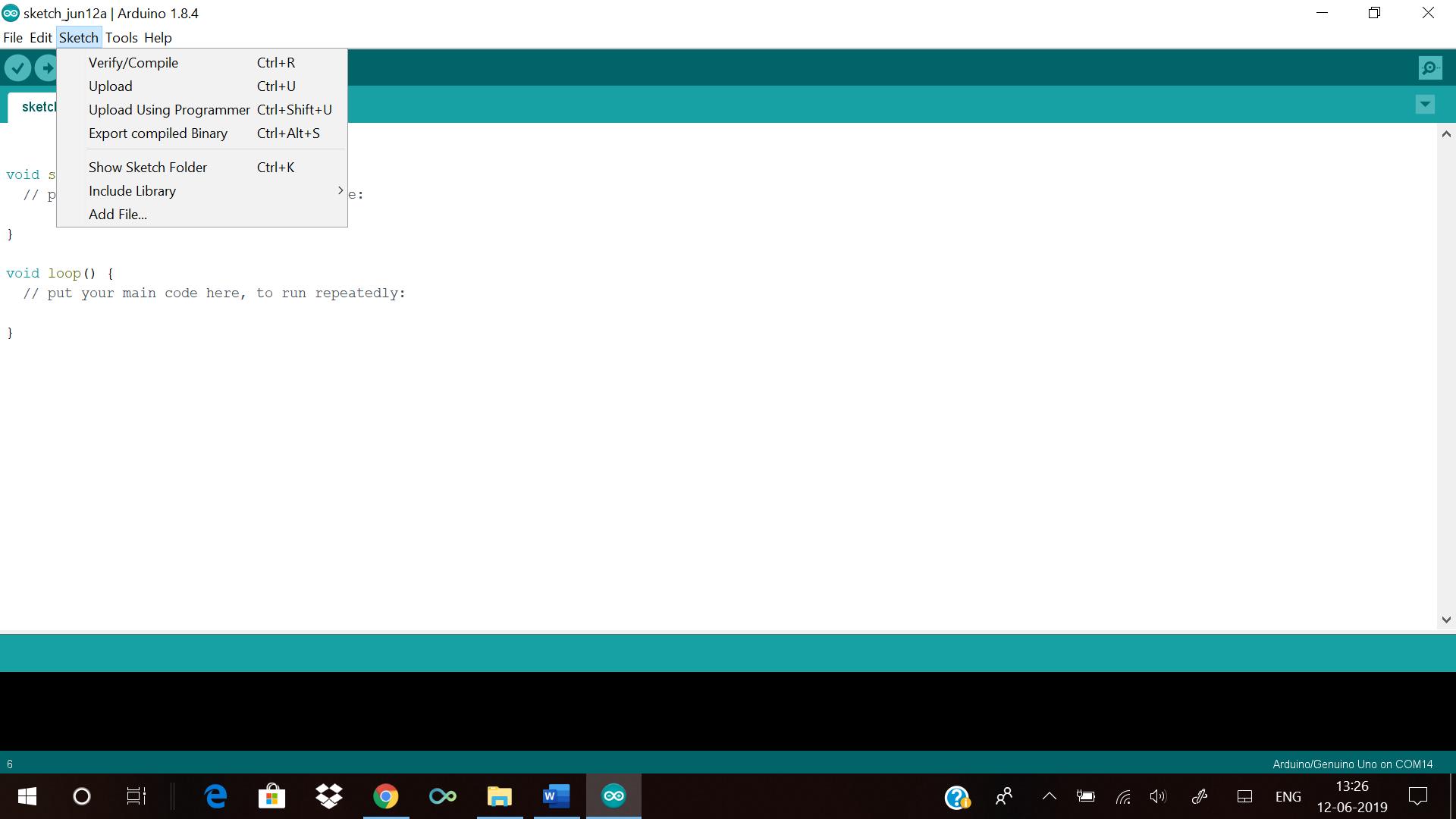
Arduino libraries are a convenient way to share code such as device drivers or commonly used utility functions. Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc.

Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library.

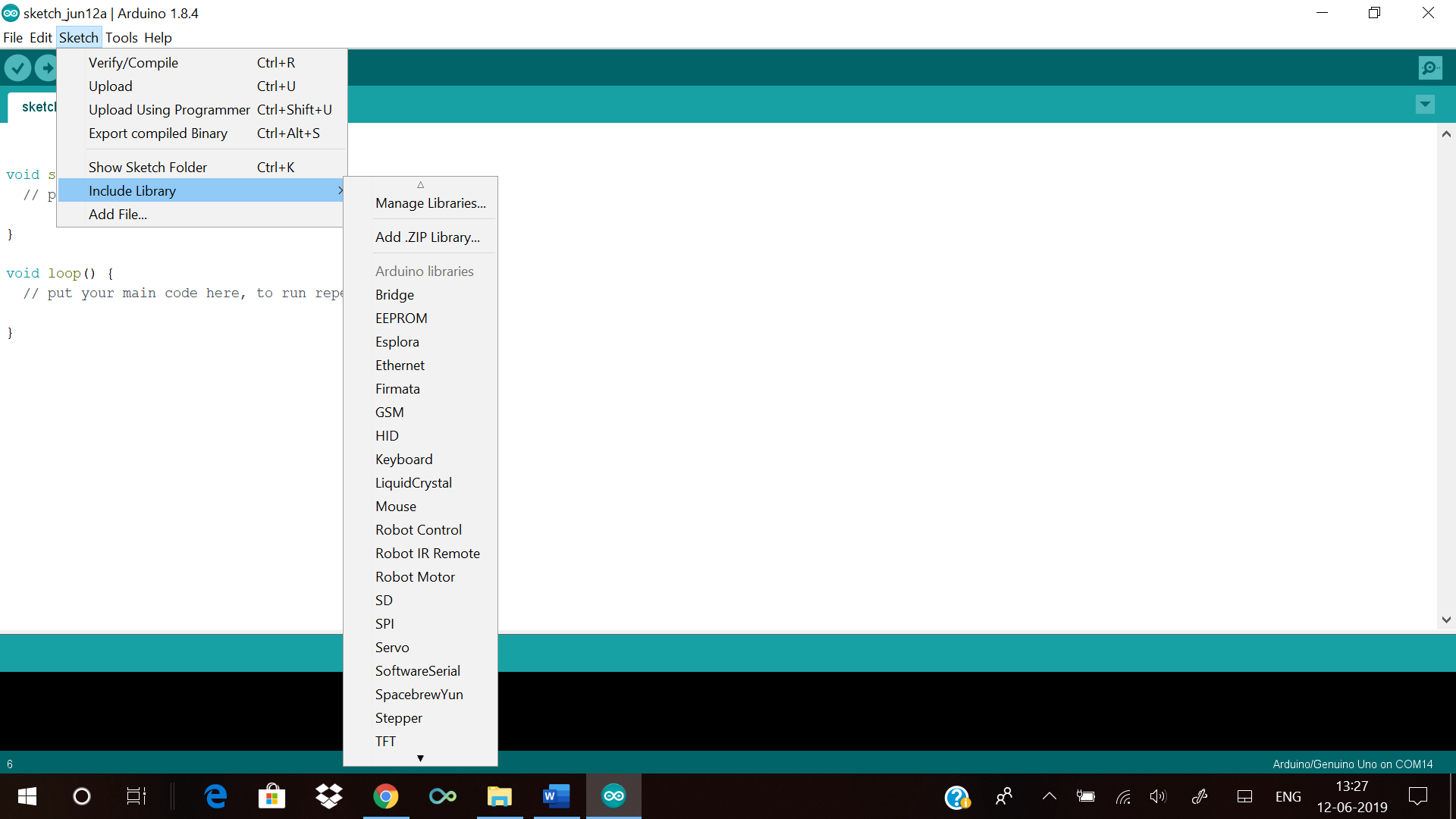
There are hundreds of additional libraries available on the Internet for download. To use these additional files, we have to first install them. The Arduino IDE has a Library Manager which facilitates installing third-party libraries submitted to Arduino for use.

To install these, we have to follow the following set of instructions:

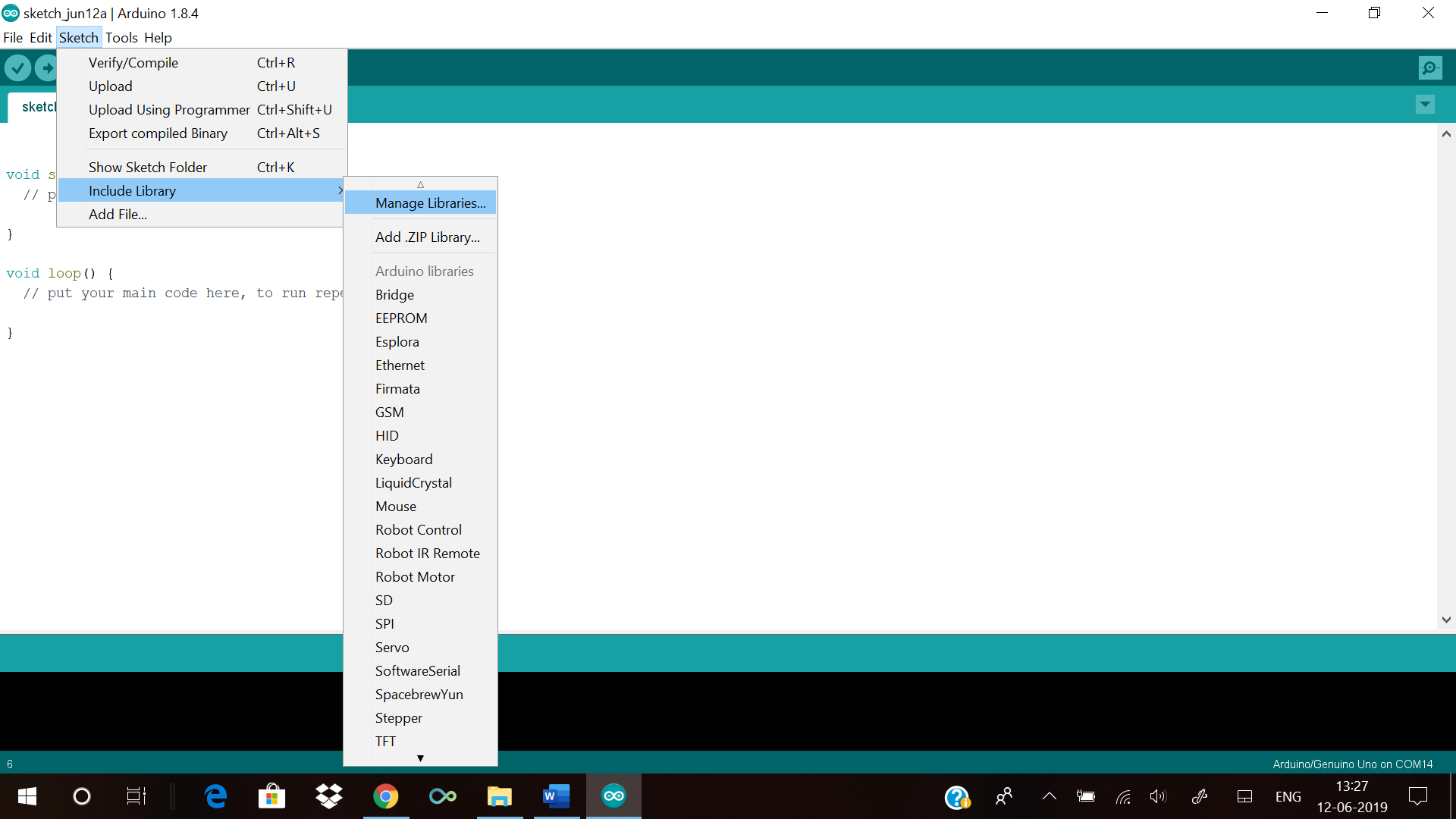
1. first open a new sketch and then go to the ‘sketch’ menu.



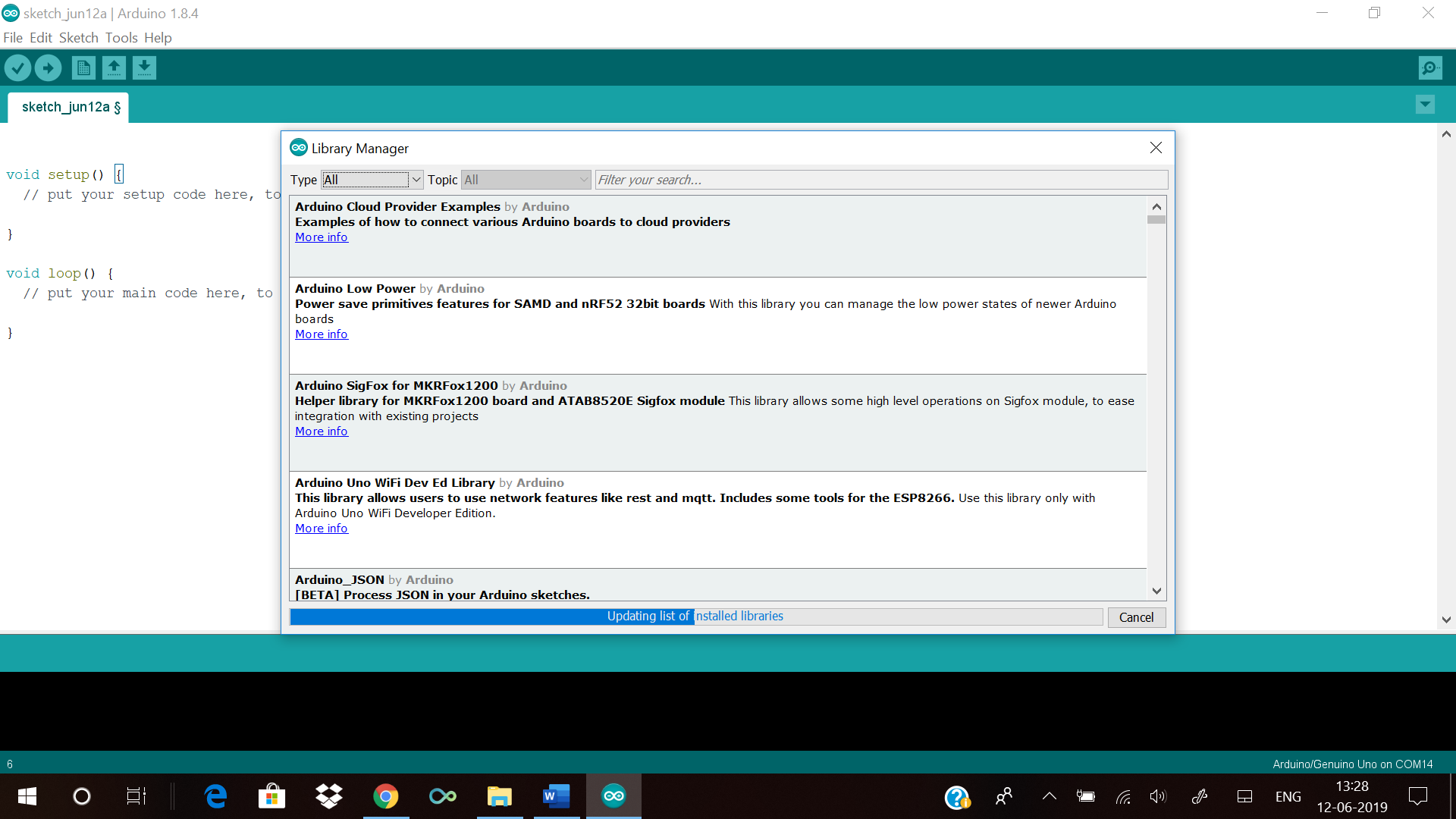
2. then go to the ‘Include libraries’. If you see the name of the library you want to include then click on that library and the library will be added to your sketch.



3. if the name is not available there then open the ‘Library manager’ and then a dialog box will open and the names of the already installed or ready to install libraries will be available there.

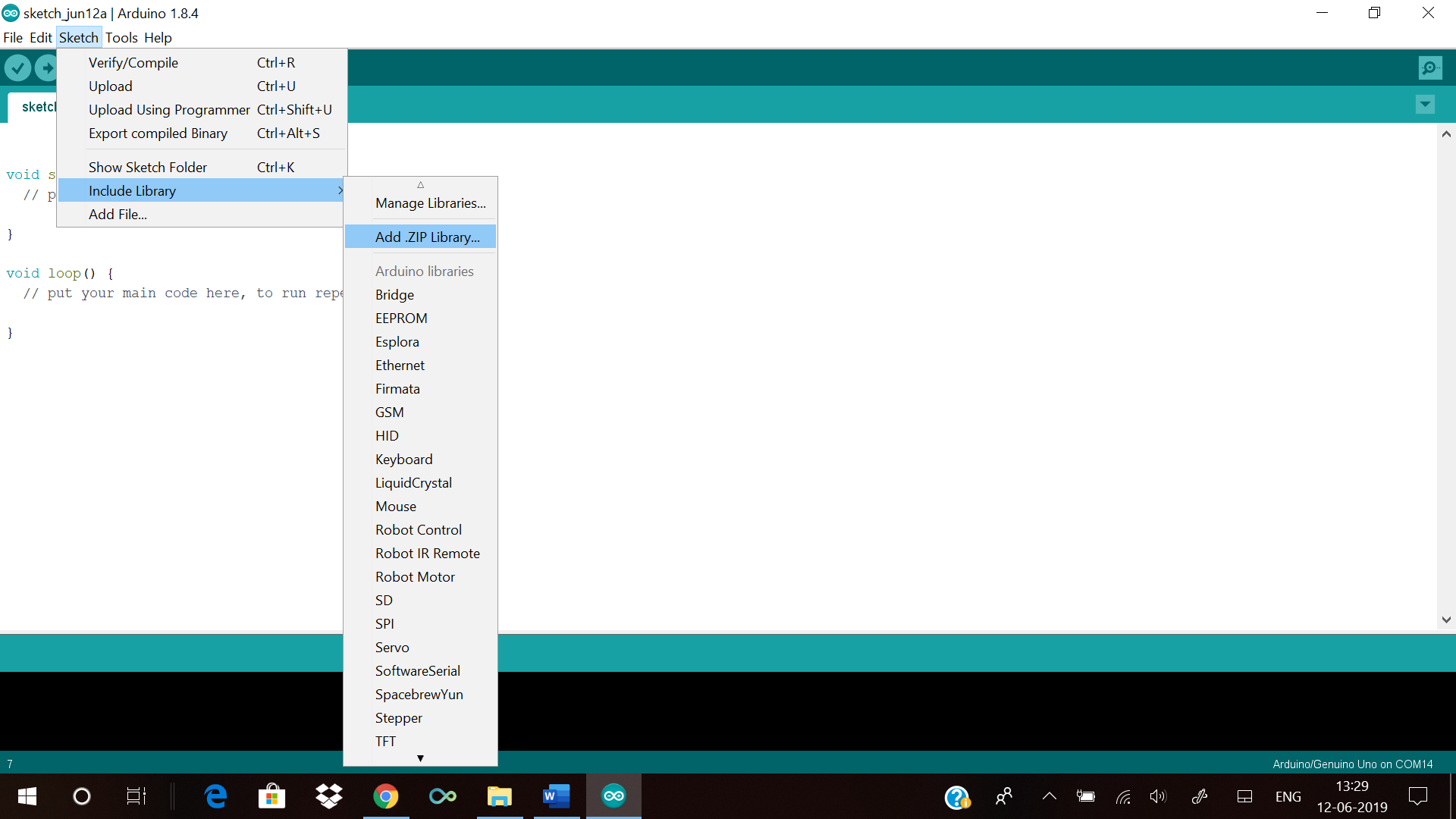


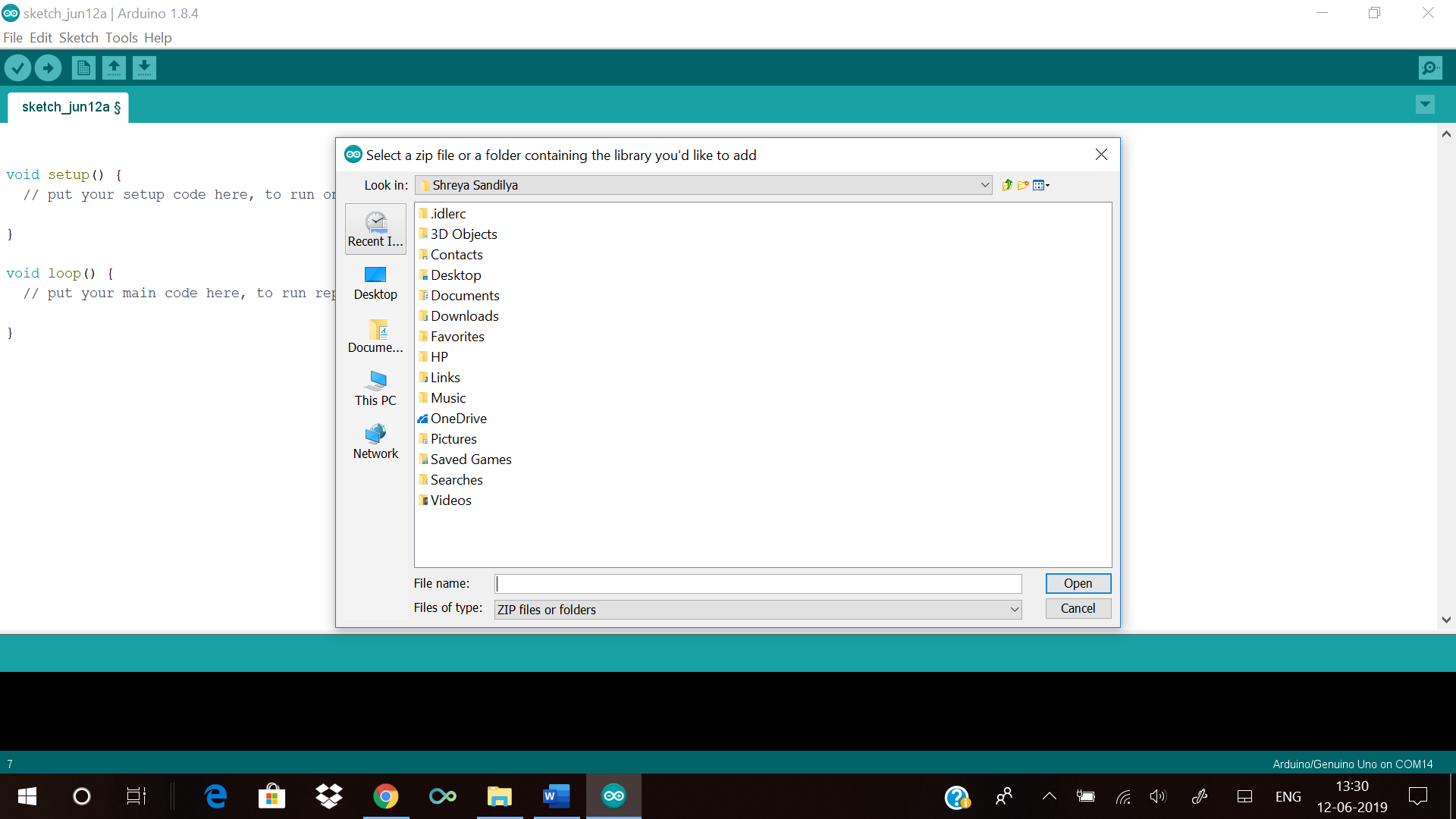
4. find the library you want to include in your sketch and then install it if not already installed.



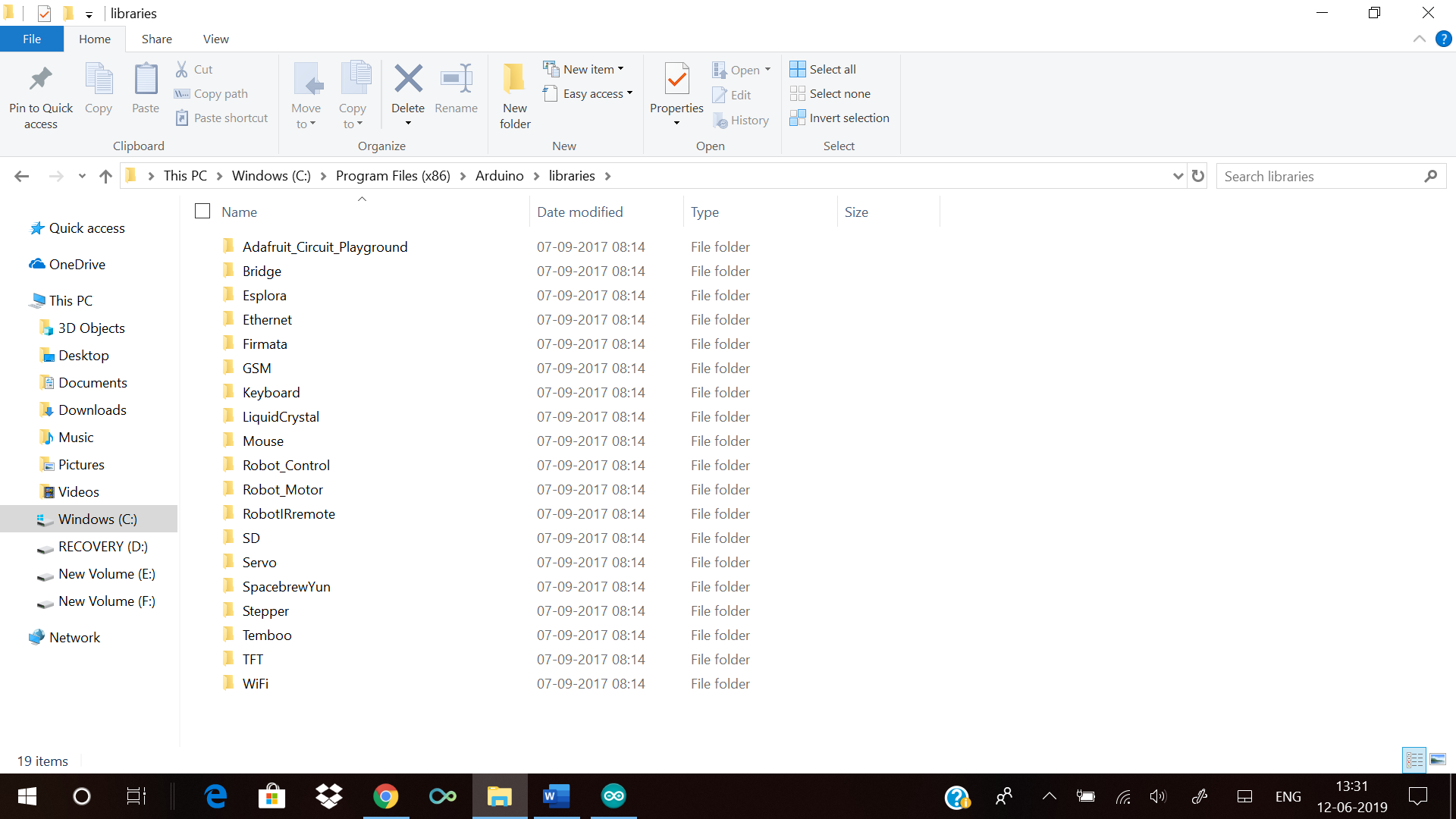
5. after the library is installed then go to the ‘Include libraries’ and add the required library.

6. if you want to install your own library or third-party library can be then you have to again the ‘Include libraries’ and then go to the ‘Add .ZIP library’ menu and then a dialog box will open then navigate to the location the library is stored and then install the library, by clicking ‘Open’.





7. the addition of your own library or third-party library can be also done manually. First extract the zip file of the libraries and then copy that in the libraries folder inside ‘Arduino IDE’.



How to create your own library?

In an Arduino library file we have a header file (w/ the extension .h) and the source file (w/ extension .cpp). The header file has definitions for the library, that is listing of everything that's inside; while the source file has the actual code.

The core of the header file consists of a line for each function in the library, wrapped up in a class along with any variables that is needed. The header files start with:

#ifndef name\_of\_the\_library\_h

#define name\_of\_the\_library\_h

#include “Arduino.h”

~the core of the header file~

#endif

The core of the header file looks something like:

class name\_of\_the\_library{

public:

private:

protected:

}

The source code starts with include statements:

#include “Arduino.h”

#include “name\_of\_the\_library.h”

Then comes the constructor and after that the whole code comes which you want to turn into a library.

Code for driving a 4 wheel driver in forward, backward, sideways, and breaking:

int pinA = 10;

int pinB = 11;

int pinA1 = 3;

int pinA2= 4;

int pinB1 = 5;

int pinB2 = 6;

void setup(){

pinmode(pinA, OUTPUT);

pinmode(pinB, OUTPUT);

pinMode(pinA1, OUTPUT);

pinMode(pinA2, OUTPUT);

pinMode(pinB1, OUTPUT);

pinMode(pinB2, OUTPUT);

}

Void loop(){

Forward();

Delay(200);

Backward();

Delay(200);

Left();

Delay(200);

Right();

Delay(200);

Breeak();

}

Void Forward(){

analogwrite( pinA, 210);

analogwrite( pinB, 210);

Digitalwrite( pinA1, HIGH);

Digitalwrite( pinA2, LOW);

Digitalwrite( pinB1, HIGH);

Digitalwrite( pinB2, LOW);

}

Void Backward(){

analogwrite( pinA, 210);

analogwrite( pinB, 210);

Digitalwrite( pinA1, LOW);

Digitalwrite( pinA2, HIGH);

Digitalwrite( pinB1, LOW);

Digitalwrite( pinB2, HIGH);

}

Void Right(){

analogwrite( pinA, 210);

analogwrite( pinB, 210);

Digitalwrite( pinA1, HIGH);

Digitalwrite( pinA2, LOW);

Digitalwrite( pinB1, LOW);

Digitalwrite( pinB2, HIGH);

}

Void LEFT(){

analogwrite( pinA, 210);

analogwrite( pinB, 210);

Digitalwrite( pinA1, LOW);

Digitalwrite( pinA2, HIGH);

Digitalwrite( pinB1, HIGH);

Digitalwrite( pinB2, LOW);

}

Void Breeak(){

analogwrite( pinA, 0);

analogwrite( pinB, 0);

Digitalwrite( pinA1, LOW);

Digitalwrite( pinA2, LOW);

Digitalwrite( pinB1, LOW);

Digitalwrite( pinB2, LOW);

}

The header file for the above code:

#ifndef 4WDcontrol\_h

#define 4WDcontrol\_h

#include “Arduino.h”

class 4WDcontrol{

public:

4WDcontrol( int pinA, int pinB, int pinA1, int pinA2, int pinB1, int pinB2);

Void Forward();

Void Backward();

Void Right();

Void Left();

Void Breeak();

private:

int \_pinA;

int \_pinA1;

int \_pinA2;

int \_pinB;

int \_pinB1;

int \_pinB2;

#endif

The course file for the above code will be:

#include “4WDcontrol.h”

#include “Arduino.h”

4WDcontrol :: 4WDcontrol( int pinA, int pinB, int pinA1, int pinA2, int pinB1, int pinB2)

pinmode(pinA, OUTPUT);

pinmode(pinB, OUTPUT);

pinMode(pinA1, OUTPUT);

pinMode(pinA2, OUTPUT);

pinMode(pinB1, OUTPUT);

pinMode(pinB2, OUTPUT);

\_pinA = pinA;

\_pinA1 = pinA1;

\_pinA2 = pinA2;

\_pinB = pinB;

\_pinB1 = pinB1;

\_pinB2 = pinB2;

}

Void 4WDcontrol::Forward(){

analogwrite( \_pinA, 210);

analogwrite( \_pinB, 210);

Digitalwrite( \_pinA1, HIGH);

Digitalwrite( \_pinA2, LOW);

Digitalwrite( \_pinB1, HIGH);

Digitalwrite( \_pinB2, LOW);

}

Void 4WDcontrol::Backward(){

analogwrite( \_pinA, 210);

analogwrite( \_pinB, 210);

Digitalwrite( \_pinA1, LOW);

Digitalwrite( \_pinA2, HIGH);

Digitalwrite( \_pinB1, LOW);

Digitalwrite( \_pinB2, HIGH);

}

Void 4WDcontrol::Right(){

analogwrite( \_pinA, 210);

analogwrite( \_pinB, 210);

Digitalwrite( \_pinA1, HIGH);

Digitalwrite( \_pinA2, LOW);

Digitalwrite( \_pinB1, LOW);

Digitalwrite( \_pinB2, HIGH);

}

Void 4WDcontrol::LEFT(){

analogwrite( \_pinA, 210);

analogwrite( \_pinB, 210);

Digitalwrite( \_pinA1, LOW);

Digitalwrite( \_pinA2, HIGH);

Digitalwrite( \_pinB1, HIGH);

Digitalwrite( \_pinB2, LOW);

}

Void 4WDcontrol::Breeak(){

analogwrite( \_pinA, 0);

analogwrite( \_pinB, 0);

Digitalwrite( \_pinA1, LOW);

Digitalwrite( \_pinA2, LOW);

Digitalwrite( \_pinB1, LOW);

Digitalwrite( \_pinB2, LOW);

}

So after saving this in the Arduino libraries file, we can use this library.

After using the library the code will look like:

#include 4WDcontrol.h

4Wdcontrol robot(10,11,3,4,5,6);

void setup(){

}

void loop(){

robot.Forward();

delay(200);

robot.Backward();

delay(200);

robot.Left();

delay(200);

robot.Right();

delay(200);

robot.Breeak();

}