https://www.halvorsen.blog

Creating Classes and Libraries with Arduino

Hans-Petter Halvorsen

Contents

- We will learn how we can create our own Arduino Libraries from Scratch
- Why create your own Libraries?
  - Better Code structure
  - Reuse your Code in different Applications
  - Distribute to others

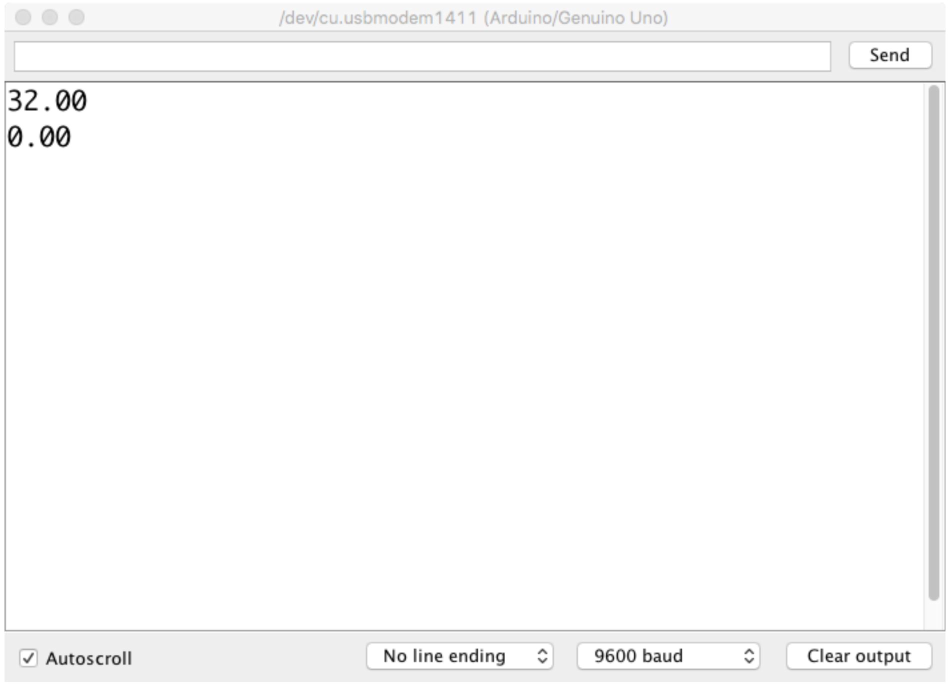
Fahrenheit Example

- We will create code that convert from degrees Celsius to degrees Fahrenheit (and the opposite)

```
void setup()
{
  float Tf;
  float Tc;

  Serial.begin(9600);

  Tc = 0;
  Tf = Tc * 9/5 + 32;
  Serial.println(Tf);

  Tf=32;
  Tc = (Tf-32)*((float)5/9);
  Serial.println(Tc);
}

void loop()
{

}
```

The Start

Serial Monitor:



/dev/cu.usbmodem1411 (Arduino/Genuino Uno)

```
32.00
0.00
```

Creating Functions

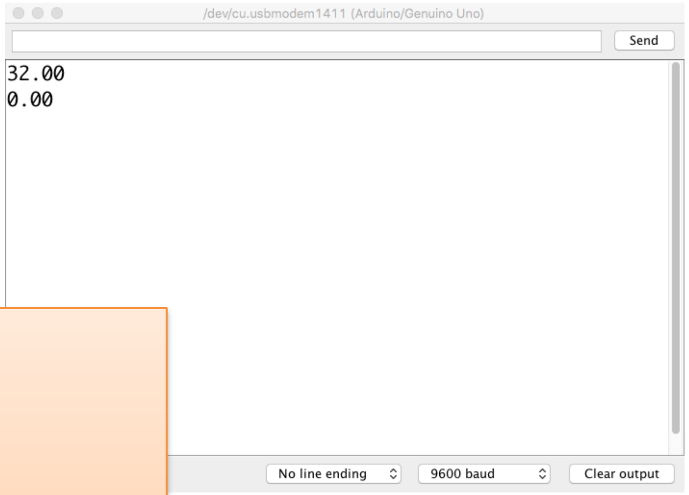
Why Creating Functions?

- In order to structure your code better
- You can reuse your Code

Creating Functions

```
void setup()
{
  float c;
  float f;

  Serial.begin(9600);

  c = 0;
  f = c2f(c);
  Serial.println(f);

  f = 32;
  c = f2c(f);
  Serial.println(c);
}

void loop()
{

}
```

```
float c2f(float Tc)
{
  float Tf;
  Tf = Tc * 9/5 + 32;
  return Tf;
}

float f2c(float Tf)
{
  float Tc;
  Tc = (Tf-32)*((float)5/9);
  return Tc;
}
```



```
32.00
0.00
```

Creating Classes

- Next, I will show how you can group your functions into a Class
- A class is simply a collection of functions and variables that are all kept together in one place

The functions and variables can be either private and public

Creating Classes

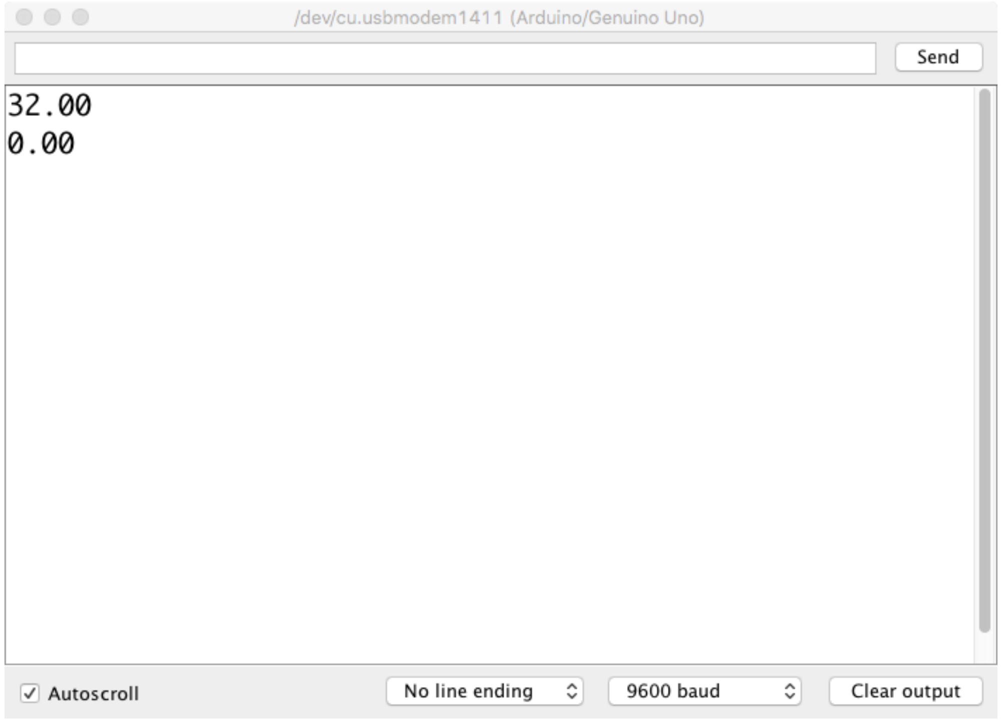
```
class Fahrenheit
{
  public:

  Fahrenheit()
  {

  };

  float c2f(float Tc){
    float Tf;
    Tf = Tc * 9/5 + 32;
    return Tf;
  }

  float f2c(float Tf){
    float Tc;
    Tc = (Tf-32)*((float)5/9);
    return Tc;
  }
};
```

```
void setup()
{
  float f;
  float c;

  Serial.begin(9600);

  Fahrenheit fahr;

  c = 0;
  f = fahr.c2f(c);
  Serial.println(f);

  f = 32;
  c = fahr.f2c(f);
  Serial.println(c);
}

void loop()
{

}
```

- **public**: they can be accessed by people using your library
- **private**: meaning they can only be accessed from within the class itself

- Each class has a special function known as a ***constructor***, which is used to create an *instance* of the class.
- The constructor has the same name as the class, and no return type.

Running the Program



```
32.00
0.00
```

☑ Autoscroll — No line ending — 9600 baud

Arduino Libraries

- Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc.
- There are hundreds of additional libraries available on the Internet for download.
- You can also create your own Libraries from scratch – Thats what we will show her

Arduino Libraries

Why create your own Libraries?

- Better Code structure
- Reuse your Code in different Applications
- Distribute to others

Arduino Libraries

You need at least two files for a library:

- Header file (.h) - The header file has definitions for the library
- Source file (.cpp) – The Functions within the Class

Note the Library Name, Folder name, .h and .cpp files all need to have the same name

Arduino Libraries

Location:

- Windows: C:\Users\hansha\Documents\Arduino\libraries
- macOS: /Users/hansha/Documents/Arduino

Creating Libraries

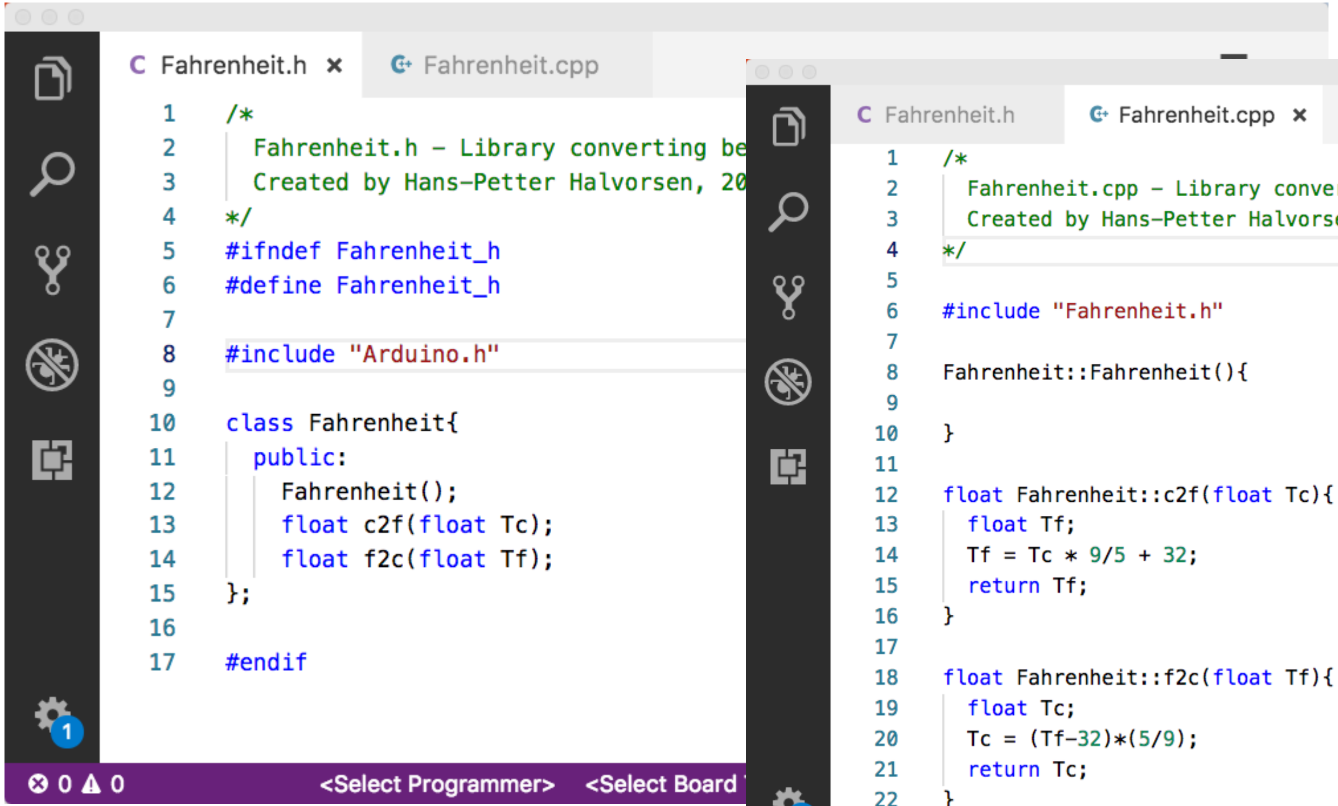
Fahrenheit.h

```
/*
Fahrenheit.h - Library converting
between Celsius and Fahrenheit.
Created by Hans-Petter Halvorsen, 2018
*/
#ifndef Fahrenheit_h
#define Fahrenheit_h

#include "Arduino.h"

class Fahrenheit{
public:
Fahrenheit();
float c2f(float Tc);
float f2c(float Tf);
};

#endif
```

Fahrenheit.cpp

```
/*
  Fahrenheit.cpp - Library converting between
Celsius and Fahrenheit.
  Created by Hans-Petter Halvorsen, 2018
*/

#include "Fahrenheit.h"

Fahrenheit::Fahrenheit(){

}

float Fahrenheit::c2f(float Tc){
  float Tf;
  Tf = Tc * 9/5 + 32;
  return Tf;
}

float Fahrenheit::f2c(float Tf){
  float Tc;
  Tc = (Tf-32)*((float)5/9);
  return Tc;
}
```

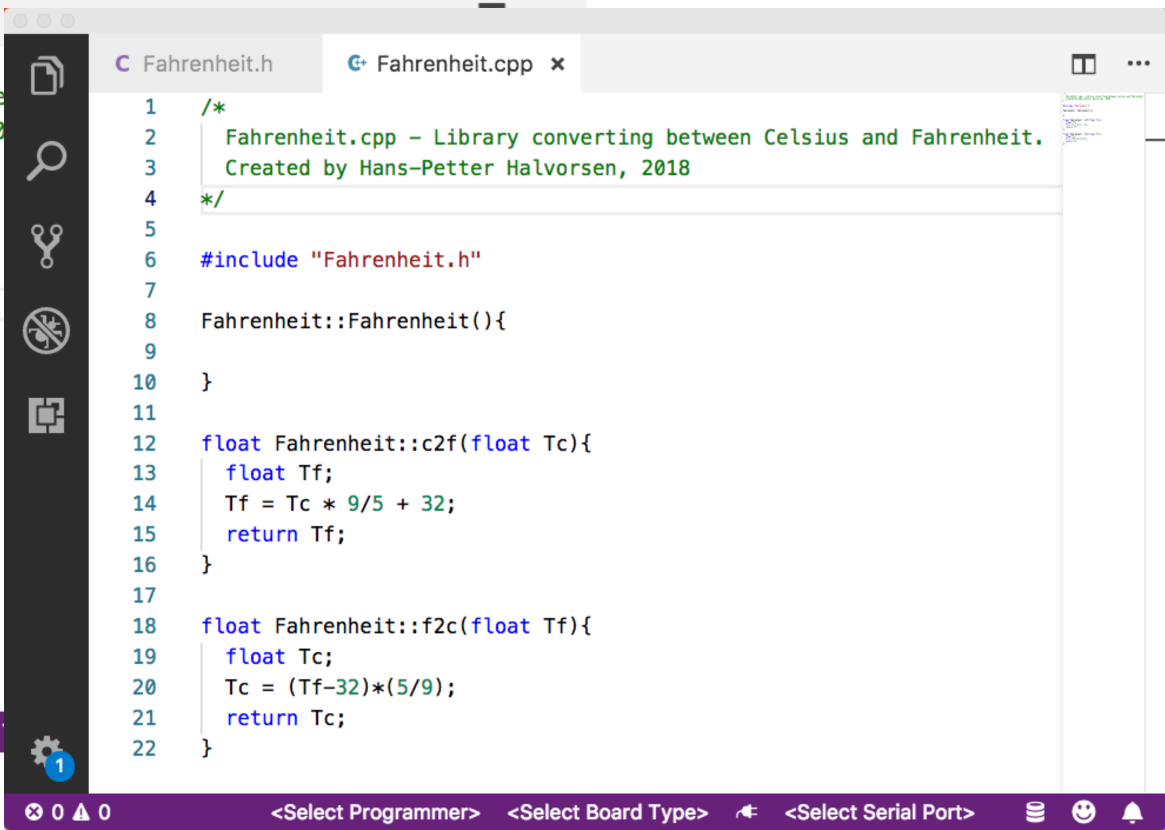
Creating Libraries



Fahrenheit.h × Fahrenheit.cpp

```
/*
  Fahrenheit.h - Library converting be
  Created by Hans-Petter Halvorsen, 20
*/
#ifndef Fahrenheit_h
#define Fahrenheit_h

#include "Arduino.h"

class Fahrenheit{
  public:
    Fahrenheit();
    float c2f(float Tc);
    float f2c(float Tf);
};

#endif
```

⊗ 0 ⚠ 0 <Select Programmer> <Select Board



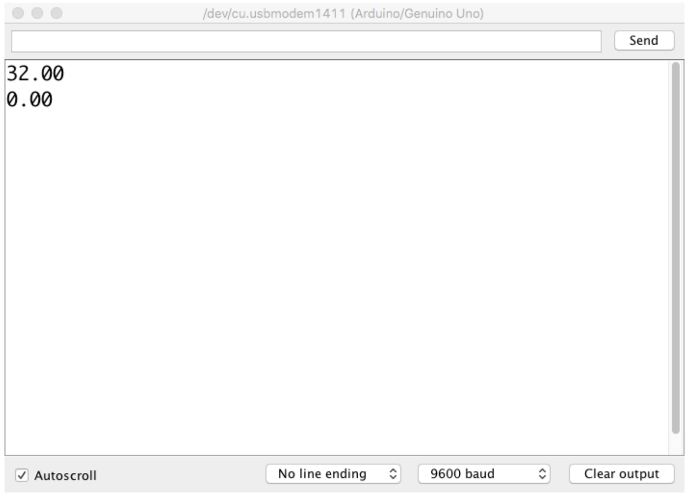
Fahrenheit.h Fahrenheit.cpp ×

```
/*
  Fahrenheit.cpp - Library converting between Celsius and Fahrenheit.
  Created by Hans-Petter Halvorsen, 2018
*/

#include "Fahrenheit.h"

Fahrenheit::Fahrenheit(){

}

float Fahrenheit::c2f(float Tc){
  float Tf;
  Tf = Tc * 9/5 + 32;
  return Tf;
}

float Fahrenheit::f2c(float Tf){
  float Tc;
  Tc = (Tf-32)*(5/9);
  return Tc;
}
```

⊗ 0 ⚠ 0 <Select Programmer> <Select Board Type> <Select Serial Port>

Testing the Library

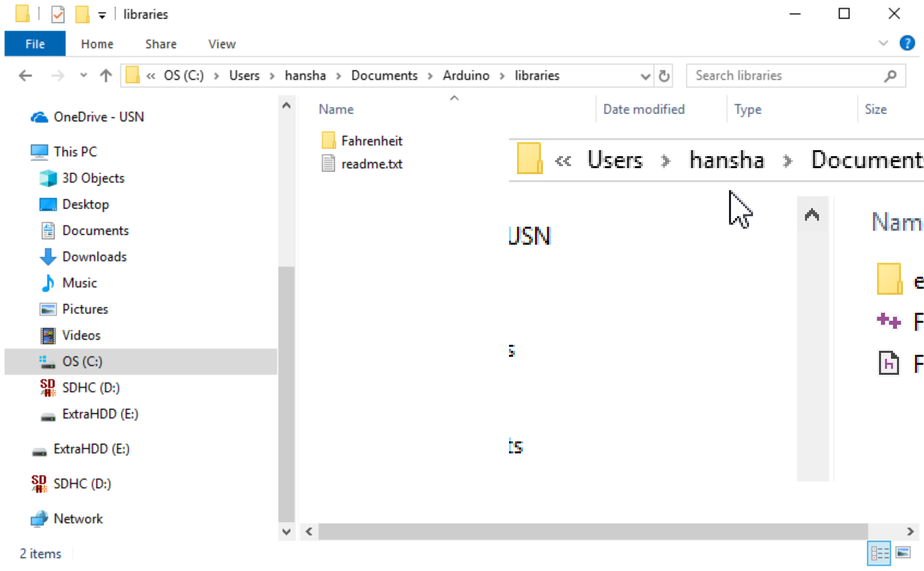
```
#include <Fahrenheit.h>

Fahrenheit fahr;

void setup()
{
  float f;
  float c;

  Serial.begin(9600);

  c = 0;
  f = fahr.c2f(c);
  Serial.println(f);

  f = 32;
  c = fahr.f2c(f);
  Serial.println(c);
}

void loop()
{

}
```



```
32.00
0.00
```

Deploying the Library

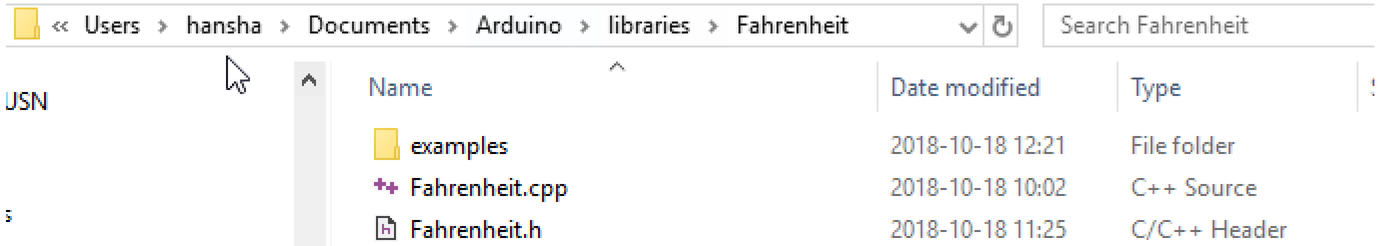
The Arduino Libraries need to be in the following folder (but can be changed from File-Preferences):



« OS (C:) › Users › hansha › Documents › Arduino › libraries

Name	Date modified	Type	Size
Fahrenheit			
readme.txt			

2 items



« Users › hansha › Documents › Arduino › libraries › Fahrenheit

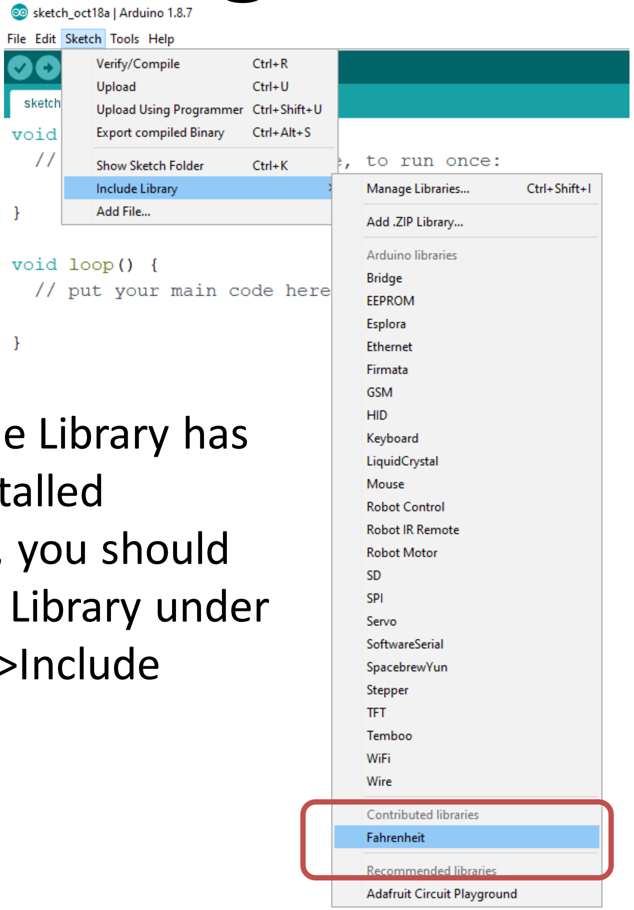
Name	Date modified	Type
examples	2018-10-18 12:21	File folder
Fahrenheit.cpp	2018-10-18 10:02	C++ Source
Fahrenheit.h	2018-10-18 11:25	C/C++ Header

C:\Users\hansha\Documents\Arduino\libraries

Here you need to put your .h and .cpp files.

You should also create an “examples” folder where you include one or more examples showing how to use your Library.

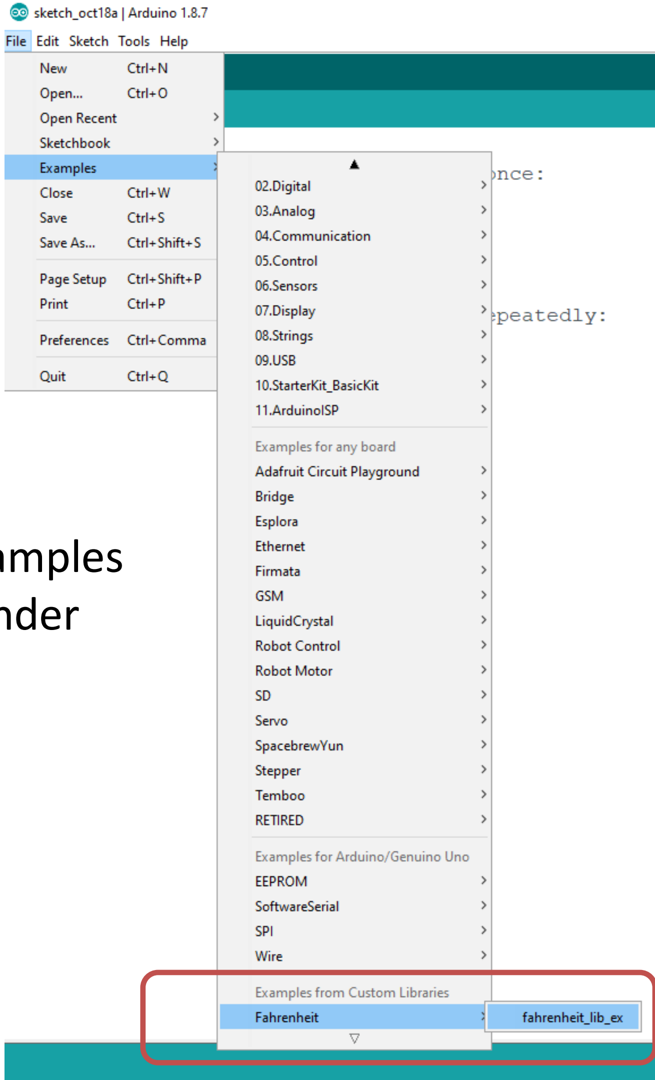
Using the Library



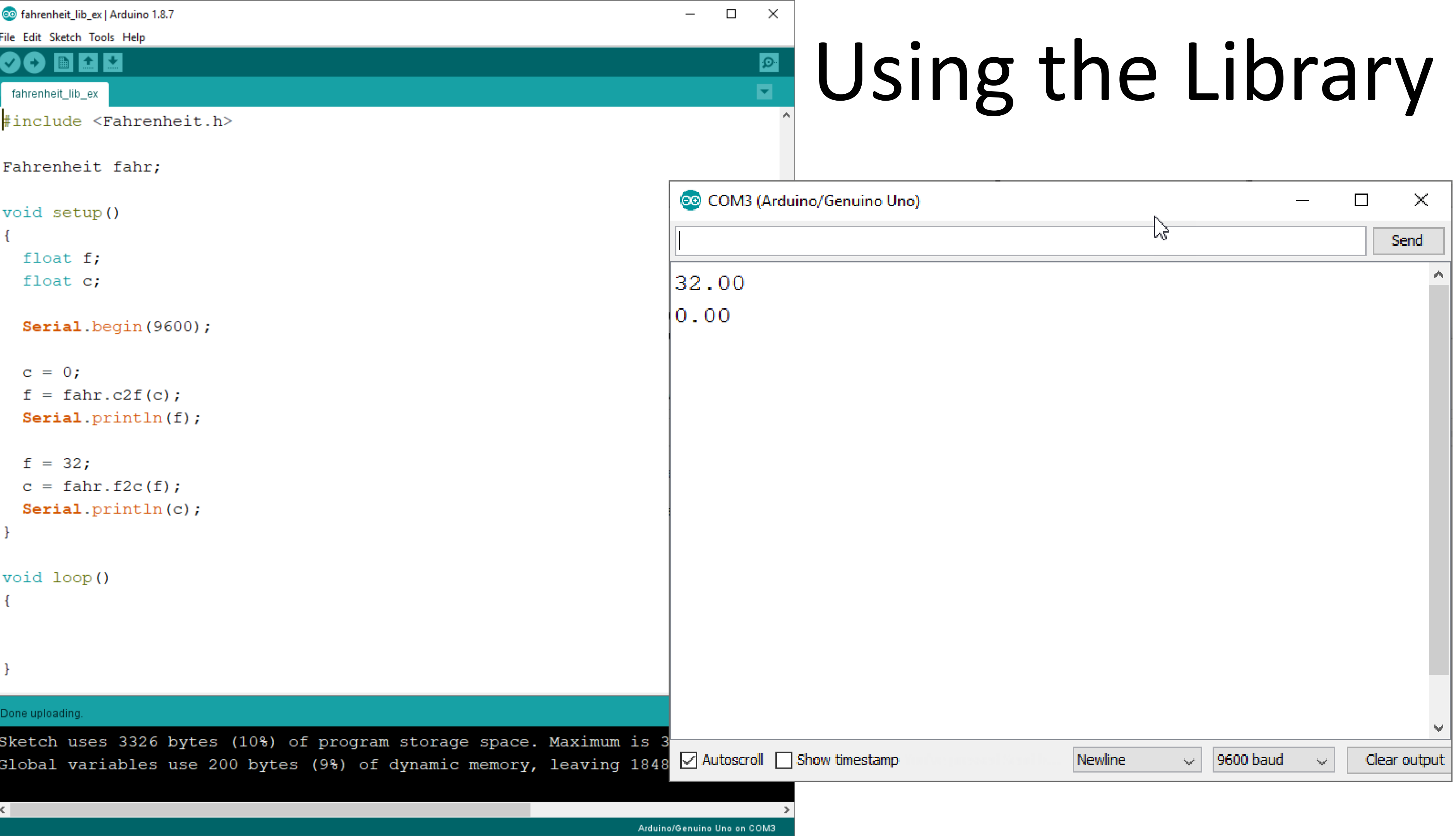
```
void
  //                    e, to run once:

}

void loop() {
  // put your main code here
}
```

When the Library has been installed properly, you should see your Library under “Sketch->Include Library”

Your Library Examples can be found under File->Examples



Using the Library


fahrenheit_lib_ex | Arduino 1.8.7
File Edit Sketch Tools Help
fahrenheit_lib_ex

```
#include <Fahrenheit.h>

Fahrenheit fahr;

void setup()
{
  float f;
  float c;

  Serial.begin(9600);

  c = 0;
  f = fahr.c2f(c);
  Serial.println(f);

  f = 32;
  c = fahr.f2c(f);
  Serial.println(c);
}

void loop()
{


}
```

Done uploading.

```
Sketch uses 3326 bytes (10%) of program storage space. Maximum is 3
Global variables use 200 bytes (9%) of dynamic memory, leaving 1848
```

Arduino/Genuino Uno on COM3
COM3 (Arduino/Genuino Uno)
Send

```
32.00
0.00
```

Autoscroll
Show timestamp
Newline
9600 baud
Clear output

References

- Installing Additional Arduino Libraries: https://www.arduino.cc/en/Guide/Libraries
- Writing a Library for Arduino: https://www.arduino.cc/en/Hacking/LibraryTutorial
- How to write libraries for the Arduino? http://playground.arduino.cc/Code/Library

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog