Shejan Shuza

Building a precompiled library for Arduino

REFERENCES OF STUDY:

Some of these references are listed as hyperlinks in the document

https://arduino.github.io/arduino-cli/0.20/library-specification/

https://arduino.stackexchange.com/questions/63099/how-to-generate-a-and-so-files-to-add-in-arduino-project

https://www.arduino.cc/en/Hacking/libraryTutorial

https://forum.arduino.cc/t/how-to-generate-a-and-so-files-to-add-in-arduino-project/583552

WARNINGS

This only works on Arduino versions greater than 1.8.13 due to the `precompiled` tag being partially broken between versions. This was tested using the Win32 version of Arduino 1.8.19.

It is also extremely recommended that you use a complex text editor like Notepad++ (Windows only), Sublime Text (cross-platform), or Atom (cross-platform) instead of a simple text editor like Windows' Notepad or macOS' TextEditor or a complex IDE like Visual Studio, as simple editors or IDEs can cause annoyances that can be easily avoided.

CHAPTER 1 – Structuring the Library folder

First **you must convert your Arduino program (.ino) to that of a Header-organized C++ program (.h and .cpp)**, this is described here with a tutorial on how to restructure Arduino programs into C++.

This tutorial will be using an abstraction of common Printing commands that Arduino users use called `PrinterExtensionShejan` to simplify the printing process of their Arduino programs. As long as the program that you are converting works in Arduino, it should work in C++ with some reformatting.

The header file (.h) is as follows:

```
#ifndef PrinterExtensionShejan_h
#define PrinterExtensionShejan_h
#include "Arduino.h"
class PrinterExtensionShejan
{
    public:
        PrinterExtensionShejan(int baud); //set the output properly
        void printLog(String s);
        void printError(String s);
        void printLogTime(String s);
        void printErrorTime(String s);
};
#endif
```

It is important that the reference `#include "Arduino.h"` exists otherwise it may be impossible to link to a .ino file later.

The C++ file (.cpp) is as follows (this is the file that is being precompiled and the source code hidden):

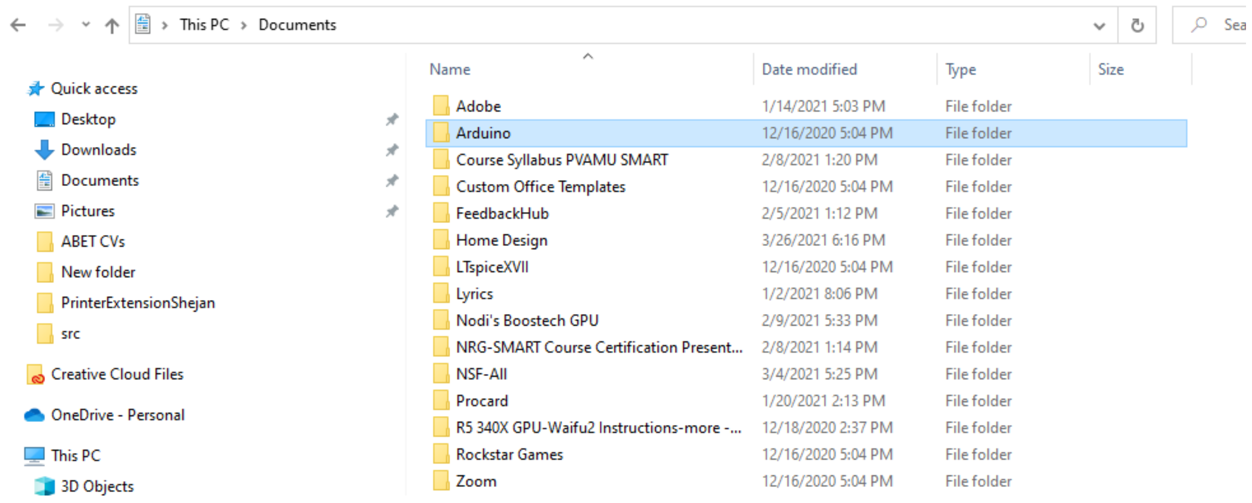
```
#include "PrinterExtensionShejan.h"
PrinterExtensionShejan::PrinterExtensionShejan(int baud)
{
    Serial.begin(baud);
    Serial.println("Initialized to "+baud);
}
void PrinterExtensionShejan::printLog(String s){
    Serial.println(s);
}
void PrinterExtensionShejan::printError(String s){
    Serial.print("ERROR: ");
    Serial.println(s);
}
void PrinterExtensionShejan::printLogTime(String s){
    Serial.print(millis());
    Serial.print(": ");
    Serial.println(s);
}
void PrinterExtensionShejan::printErrorTime(String s){
    Serial.print("ERROR: ");
    Serial.print(millis());
    Serial.print(": ");
    Serial.println(s);
}
```

And the `KEYWORDS` file (keywords.txt) is as follows (this is used by Arduino IDE to understand the parts and names of your library that are inside of the header)

```
PrinterExtensionShejan KEYWORD1
printLog KEYWORD2
printError KEYWORD2
printLogTime KEYWORD2
printErrorTime KEYWORD2
```

These files must be stored into an appropriate Arduino 1.5 library structure in order to be compiled to a precompiled binary or read as a precompiled library.

Find the User Arduino folder, this will be located in the Documents folder under a folder titled “Arduino”



This PC › Documents

Quick access
- Desktop
- Downloads
- Documents
- Pictures
- ABET CVs
- New folder
- PrinterExtensionShejan
- src

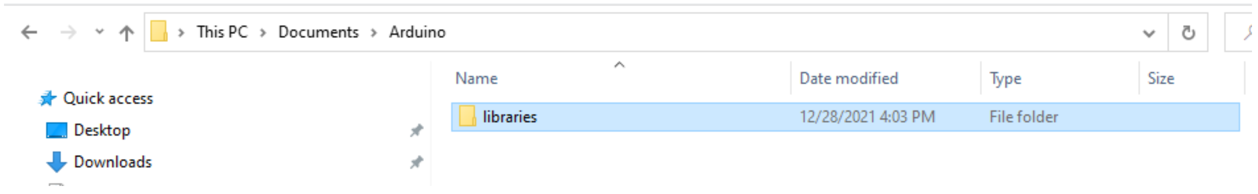
Creative Cloud Files

OneDrive - Personal

This PC
- 3D Objects

Name	Date modified	Type	Size
Adobe	1/14/2021 5:03 PM	File folder	
Arduino	12/16/2020 5:04 PM	File folder	
Course Syllabus PVAMU SMART	2/8/2021 1:20 PM	File folder	
Custom Office Templates	12/16/2020 5:04 PM	File folder	
FeedbackHub	2/5/2021 1:12 PM	File folder	
Home Design	3/26/2021 6:16 PM	File folder	
LTspiceXVII	12/16/2020 5:04 PM	File folder	
Lyrics	1/2/2021 8:06 PM	File folder	
Nodi's Boostech GPU	2/9/2021 5:33 PM	File folder	
NRG-SMART Course Certification Present...	2/8/2021 1:14 PM	File folder	
NSF-All	3/4/2021 5:25 PM	File folder	
Procard	1/20/2021 2:13 PM	File folder	
R5 340X GPU-Waifu2 Instructions-more -...	12/18/2020 2:37 PM	File folder	
Rockstar Games	12/16/2020 5:04 PM	File folder	
Zoom	12/16/2020 5:04 PM	File folder	

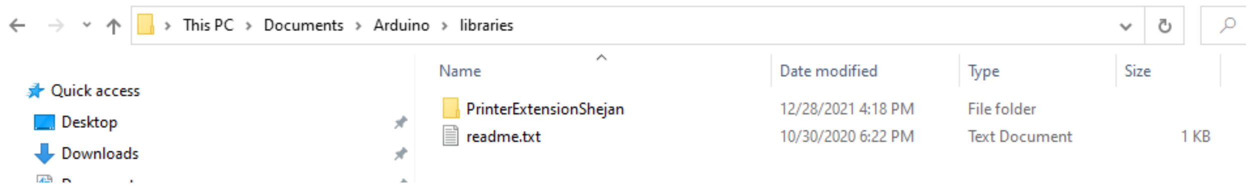
Inside of that folder should be a folder called "libraries", if it doesn't exist, then create it (it is case-sensitive, so ensure it is lowercase)



This PC › Documents › Arduino

Name	Date modified	Type	Size
libraries	12/28/2021 4:03 PM	File folder	

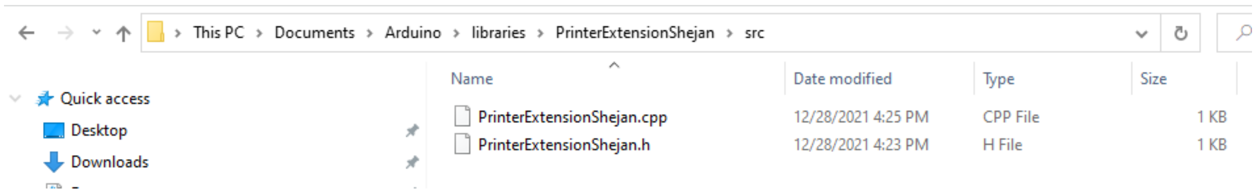
Inside the "libraries" folder create a folder that you intend to store your final library in.



This PC › Documents › Arduino › libraries

Name	Date modified	Type	Size
PrinterExtensionShejan	12/28/2021 4:18 PM	File folder	
readme.txt	10/30/2020 6:22 PM	Text Document	1 KB

Inside of your library folder, create a `src` folder and move your header and C++ files into the `src` folder.

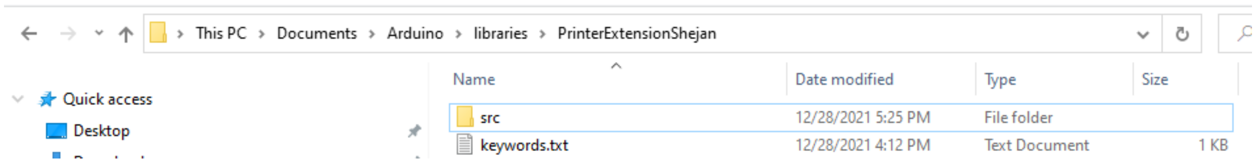


This PC › Documents › Arduino › libraries › PrinterExtensionShejan › src

Name	Date modified	Type	Size
PrinterExtensionShejan.cpp	12/28/2021 4:25 PM	CPP File	1 KB
PrinterExtensionShejan.h	12/28/2021 4:23 PM	H File	1 KB

If you intend to have any example Arduino files (.ino), you can make a `examples` folder and create your example programs, these will show up in the Arduino IDE under the Files>Examples section.

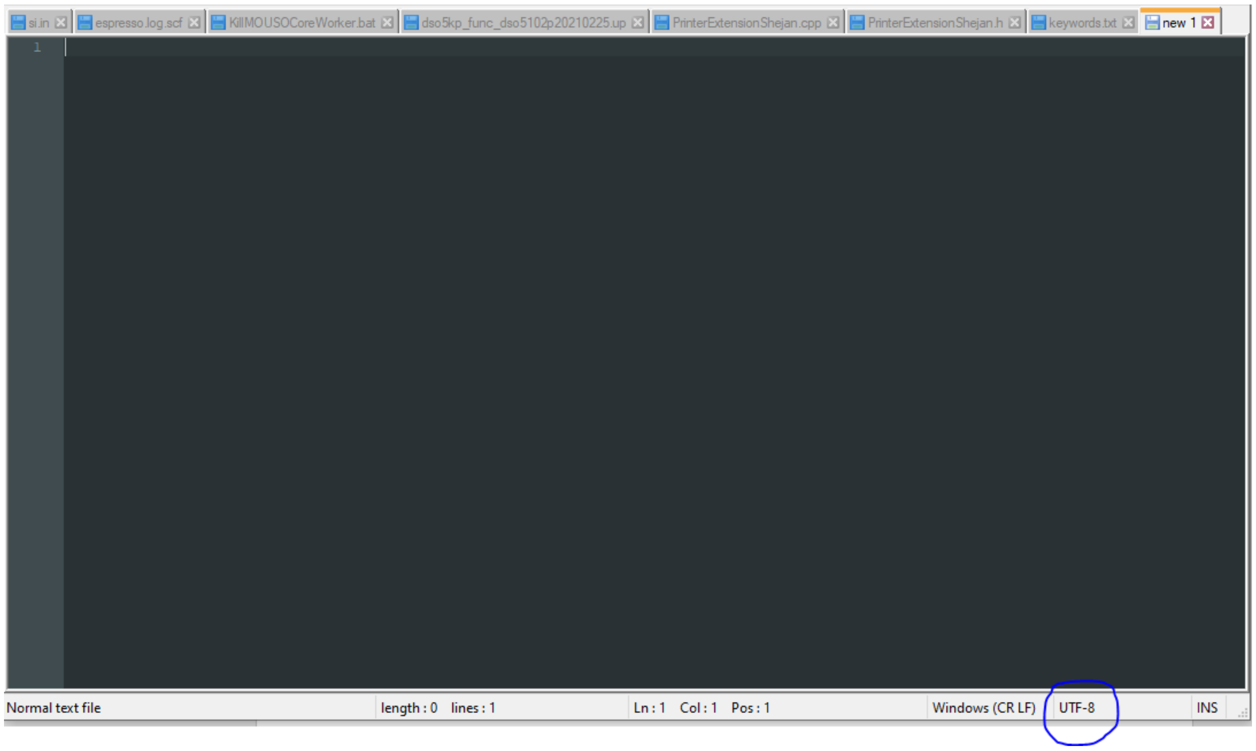
Go back to the root of your library folder, and copy your keywords.txt file to the root (you should have made this when converting your Arduino program to a C++ program)


This PC › Documents › Arduino › libraries › PrinterExtensionShejan

Name	Date modified	Type	Size
src	12/28/2021 5:25 PM	File folder	
keywords.txt	12/28/2021 4:12 PM	Text Document	1 KB

Now, create a file titled `library.properties`, this is used for describing to the Arduino IDE how to handle this library, and open it with a complex text editor

Before writing anything, ensure that the editor is set to write **UTF-8 bytes** for your file, this is a mandatory standard of the properties file for 1.5 compliance. (Notepad++ is depicted)



Normal text file   length : 0   lines : 1   Ln : 1   Col : 1   Pos : 1   Windows (CR LF)   UTF-8   INS

The tags for **name, version, author, maintainer, sentence, paragraph, category, url,** and **architectures** are **required** in order for your library to bind properly. A more in-depth explanation for what each of these tags mean can be found here.

This tutorial's properties file is as follows:

```
name=PrinterExtensionShejan
version=0.0.1
author=Shejan Shuza
maintainer=Shejan Shuza <shejan0@gmail.com>
sentence=A basic library to handle common print statements for Arduino
paragraph=Includes logging and erroring including versions for time
category=Display
```

```
url=https://github.com/shejan0
architectures=*
```

(The architectures is set to * as that means it compiles onto all architectures, if you intend to limit the architectures, or your code uses architecture-specific functions, it can be typed as a list of architectures separated by commas)

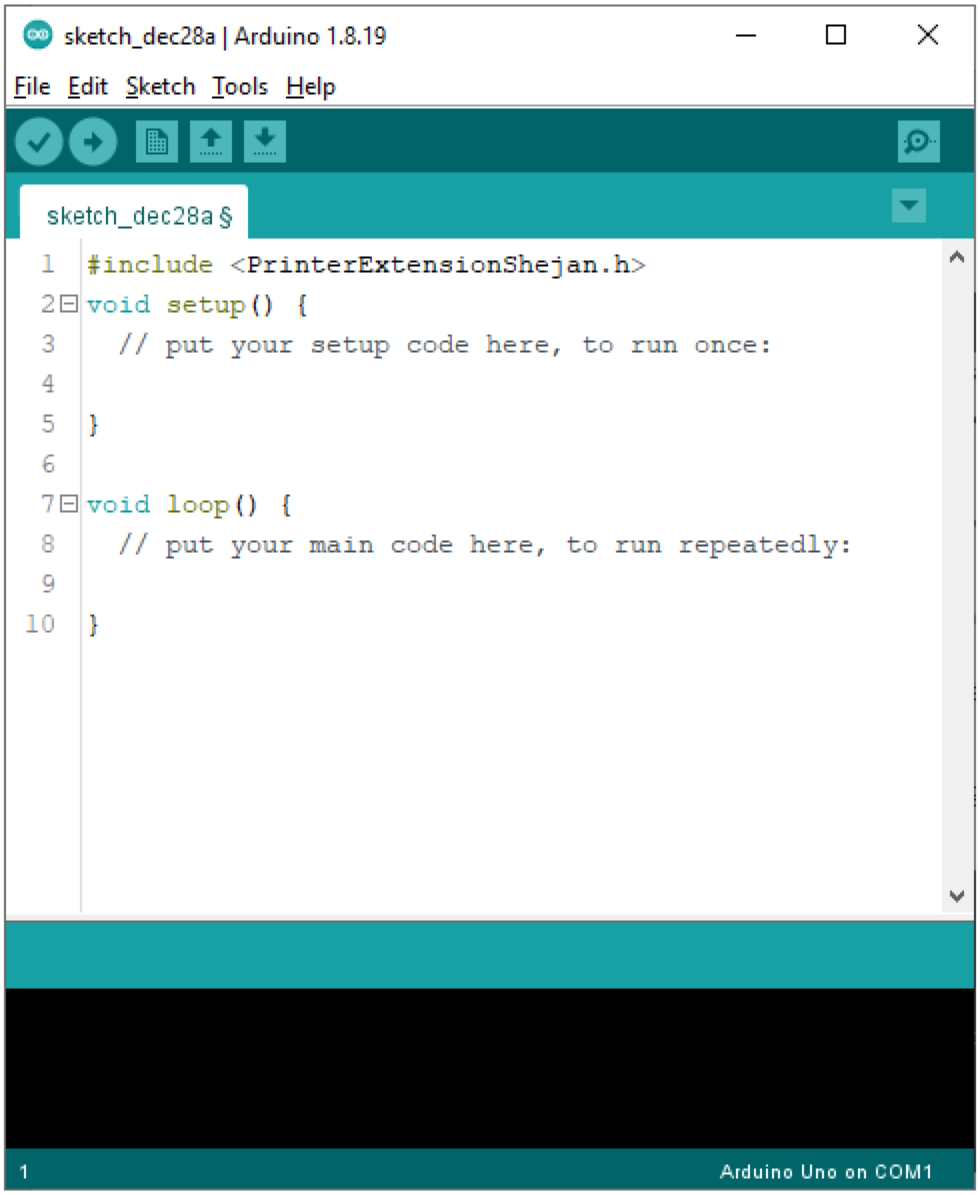
CHAPTER 2 – Precompiling/Updating the precompiled Library

By default, Arduino assumes that all libraries have source code (C++/C and Header files) inside of all libraries, and these are to be compiled on every single run of the compiler unless a temporary version from a previous compile exists. We want to explicitly compile a version that can be stored and used directly inside of the library folder without having to be compiled on a previous compile of a Arduino program.

Inside of your `library.properties` file, add a `dot_a_linkage` tag to the end and set it to `true` (ensure that there are no references to `precompiled` or `ldflags`)

```
name=PrinterExtensionShejan
version=0.0.1
author=Shejan Shuza
maintainer=Shejan Shuza <shejan0@gmail.com>
sentence=A basic library to handle common print statements for Arduino
paragraph=Includes logging and erroring including versions for time
category=Display
url=https://github.com/shejan0
architectures=*
dot_a_linkage=true
```

Now open the Arduino IDE, and simply add a reference to the header file of your library (or open a example that references your library), and compile this program. In this case, I am compiling for the Arduino Uno, which is a ATMega328p processor architecture, depending on what board you compile to, you will have to keep note of what architecture it uses since the binary will be architecture-specific (this is one big reason to limit the architectures your library can compile to)



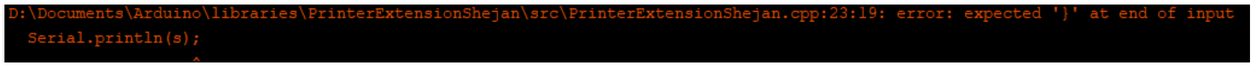
```
#include <PrinterExtensionShejan.h>
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

At this time, if your library has syntax errors inside of it, these will appear as compiler errors for the current Arduino program. Commonly the compiler will be able to tell you which lines your error exists on and in which file, and usually what the error is.



```
D:\Documents\Arduino\libraries\PrinterExtensionShejan\src\PrinterExtensionShejan.cpp:23:19: error: expected '}' at end of input
  Serial.println(s);
                   ^
```

If your compile was successful then your compiled library will exist inside of a System temporary folder with the rest of your Arduino build.

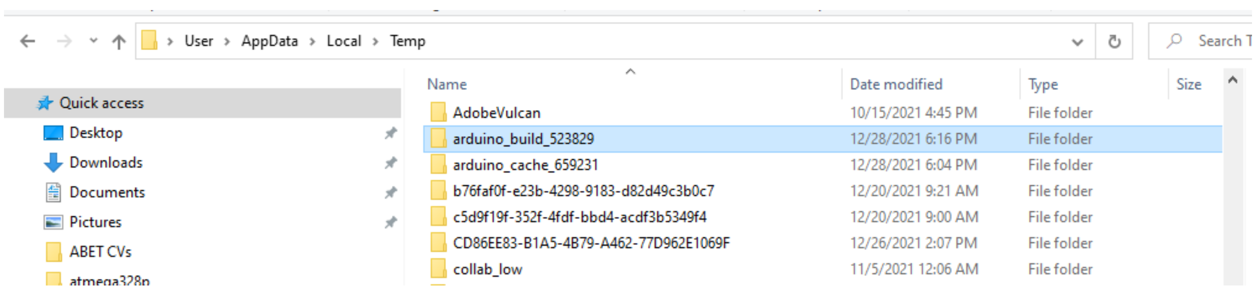
On Windows:

```
%localappdata%\Temp\arduino_build_######
```

On Linux:

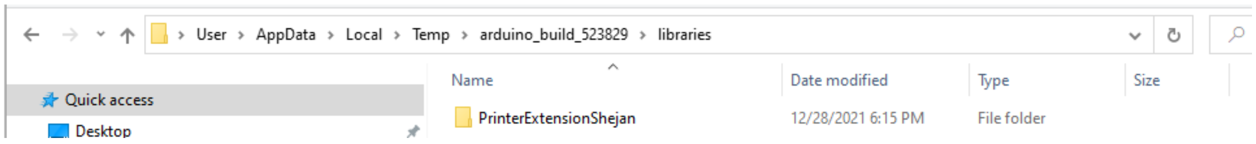
`/tmp/arduino_build_######`

On macOS the folder is placed within a randomly generated folder in the `/var/folders` directory, an easy way of finding this folder is to use to compiler output and find where this is by either looking at the most recently modified folder within that folder or looking in the compiler debug output for where the final binary was compiled to (this can be enabled in the Arduino preferences for compilation verbose output).



Name	Date modified	Type	Size
AdobeVulcan	10/15/2021 4:45 PM	File folder	
arduino_build_523829	12/28/2021 6:16 PM	File folder	
arduino_cache_659231	12/28/2021 6:04 PM	File folder	
b76faf0f-e23b-4298-9183-d82d49c3b0c7	12/20/2021 9:21 AM	File folder	
c5d9f19f-352f-4fdf-bbd4-acdf3b5349f4	12/20/2021 9:00 AM	File folder	
CD86EE83-B1A5-4B79-A462-77D962E1069F	12/26/2021 2:07 PM	File folder	
collab_low	11/5/2021 12:06 AM	File folder	

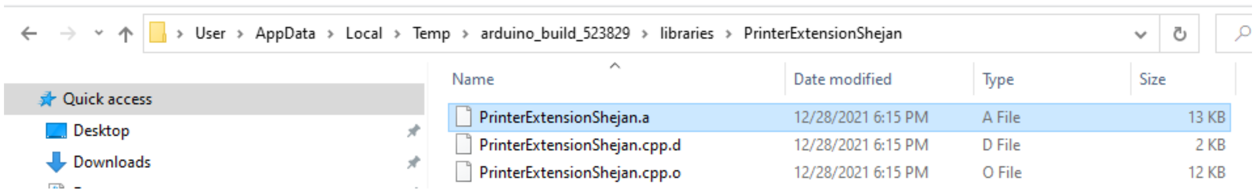
Within this folder, find the “libraries” folder, and within that folder, a folder titled with your library



User › AppData › Local › Temp › arduino_build_523829 › libraries
Quick access
Desktop

Name	Date modified	Type	Size
PrinterExtensionShejan	12/28/2021 6:15 PM	File folder	

Inside this folder should exist **`.a`**, **`.cpp.d`**, and **`.cpp.o`** file versions of your library, the **`.a`** file is the complete compile of your library in archive format (it can be used for future compilations)



User › AppData › Local › Temp › arduino_build_523829 › libraries › PrinterExtensionShejan
Quick access
Desktop
Downloads

Name	Date modified	Type	Size
PrinterExtensionShejan.a	12/28/2021 6:15 PM	A File	13 KB
PrinterExtensionShejan.cpp.d	12/28/2021 6:15 PM	D File	2 KB
PrinterExtensionShejan.cpp.o	12/28/2021 6:15 PM	O File	12 KB

CHAPTER 3 – Restructuring library for Precompiled sharing only

Once you have a **`.a`** file that contains the compiled binary of your library. It can be moved back into your library folder, this can allow your library to be shared without sharing the explicit inner workings of the library, however, this means that for every architecture your library is intended to be used on, there must be a binary archive for that architecture.

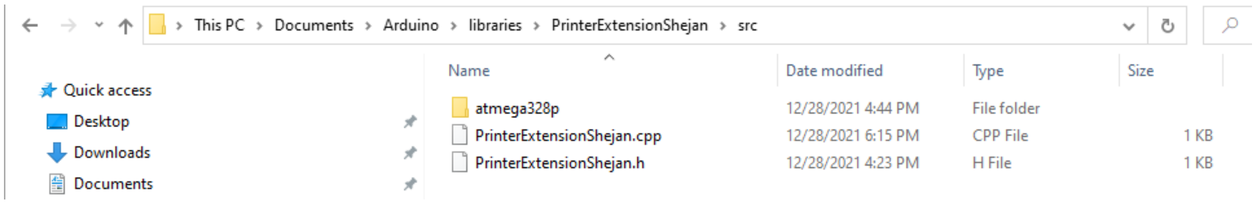
Inside of your `library.properties` file, add a `precompiled` tag to the end and set it to `full`, and `ldflags` tag and set it to `“-l”` with the name of the **`.a`** file, this is used by the linker to ensure that the library is attached properly (ensure that there are no references to `dot_a_linkage`)

```
name=PrinterExtensionShejan
version=0.0.1
author=Shejan Shuza
maintainer=Shejan Shuza <shejan0@gmail.com>
sentence=A basic library to handle common print statements for Arduino
paragraph=Includes logging and erroring including versions for time
category=Display
url=https://github.com/shejan0
```

```
architectures=*
precompiled=full
ldflags=-lPrinterExtensionShejan
```

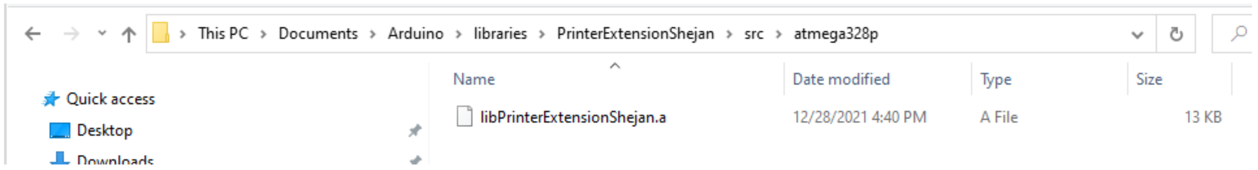
(ensure to not copy the `.a` part of the library file name to `ldflags`, this is because the Arduino linker is only looking for the name, it doesn't explicitly care what kind of file it is).

Inside the `src` folder of your library, make a folder with the name of the architecture for the `.a` architecture-specific file, since this tutorial compiled into Uno, the folder will be called `atmega328p`. If you intend to compile to more than one architecture, than each `.a` architecture-specific file should be in separate folders for each architecture.


This PC > Documents > Arduino > libraries > PrinterExtensionShejan > src
Quick access
Desktop
Downloads
Documents

Name	Date modified	Type	Size
atmega328p	12/28/2021 4:44 PM	File folder	
PrinterExtensionShejan.cpp	12/28/2021 6:15 PM	CPP File	1 KB
PrinterExtensionShejan.h	12/28/2021 4:23 PM	H File	1 KB

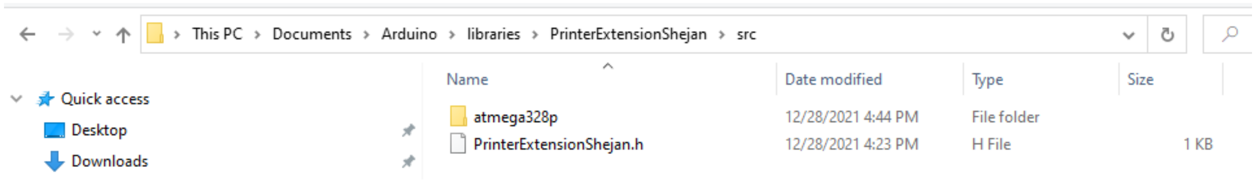
Inside of your architecture-specific folder, place your `.a` file inside and rename the file to include "`lib`" at the beginning, this ensures that only it is considered a library, and the Arduino linker can easily find it when specified as a "`-l`", if you intend to compile more than one architecture, this must be done in every architecture folder.


This PC > Documents > Arduino > libraries > PrinterExtensionShejan > src > atmega328p
Quick access
Desktop
Downloads

Name	Date modified	Type	Size
libPrinterExtensionShejan.a	12/28/2021 4:40 PM	A File	13 KB

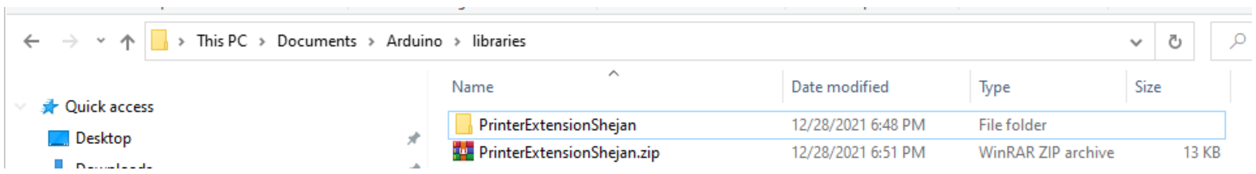
You can now remove the C++ file (.cpp) from the `src` folder (do not have to delete it, you can simply move it to somewhere else), even when Arduino is reopened, the compiler will always succeed compiling, even without the C++ source code, as long as the archive file for the specific architecture the Arduino program being compiled to exists.

Ensure that your header file (.h) is still in the `src` folder since this is used to include into Arduino programs.


This PC > Documents > Arduino > libraries > PrinterExtensionShejan > src
Quick access
Desktop
Downloads

Name	Date modified	Type	Size
atmega328p	12/28/2021 4:44 PM	File folder	
PrinterExtensionShejan.h	12/28/2021 4:23 PM	H File	1 KB

You can now zip or copy the final Library folder (within the Arduino user library folder inside of Documents) without having to copy the source code.



Name	Date modified	Type	Size
PrinterExtensionShejan	12/28/2021 6:48 PM	File folder	
PrinterExtensionShejan.zip	12/28/2021 6:51 PM	WinRAR ZIP archive	13 KB