Куда добавить новый файл .h и .cpp в проект STM32CubeIDE

Basically you can place source file in any source folder. You can use existing folders or create a new one, however you like.

Header files have to be placed inside a folder that is in the include path (e.g. Core/Inc) or you have to tell the compiler that a specific file should be included. If you want to create a new folder for your header files you have to add the folder or files inside this folder to the include path via Project -> Properties -> C/C++ Build -> Tool Setting (tab) -> MCU GCC Compiler or MCU G++ Compiler -> Include paths. Note that you might have to add it twice, once for C files (MCU GCC Compiler) and once for C++ files (MCU G++ Compiler).

To create a file you can simply click on the folder you want to create it in and add a new <whatever>file. You can also just create files and directories on your local filesystem and refresh the project (right click -> Refresh or F5) to show changes in STM32CubeIDE.

For small projects (like a blinky project) it is probably enough to just use the Core folder.

For bigger projects I personally like to seperate all the autogenerated code from my own files. Especially for C++ projects it can become very annoying that STM32CubeIDE always renames main.cpp to main.c and deletes code segments you accidentally put outside their comment blocks every time you run the code generation.

I usually create a Project source folder for project specific files and a Lib source folder for libraries. Then I create a cppmain.h file that simply declares a void cppMain() function and a cppmain.cpp file that I use to write my main program. The only change to the autogenerated code is that you have to include cppmain.h and call cppMain() before the main loop in main.c.

Directory tree:

Project name

+ Core

| + ...

+ Drivers

| + ...

+ Libs

| + someLib

| | + Inc

| | | + someLib.h or hpp

| | + Src

| | + someLib.c or cpp

| + ...

+ Program

| + Inc

| | + cppmain.h

| | + ...

| + Src

| + cppmain.cpp

| + ...

+ ...

You will have to add Program/Inc and any Lib/Inc folder to the include path. The changes to main.c then look like this:

...

/\* Includes ------------------------------------------------------------------\*/

#include "main.h"

/\* Private includes ----------------------------------------------------------\*/

/\* USER CODE BEGIN Includes \*/

#include "cppmain.h" // this goes in user includes

/\* USER CODE END Includes \*/

...

/\* USER CODE BEGIN 2 \*/

cppMain(); // this goes before the main loop

// anything after this point will never be reaced

// when you create a main loop inside cppMain()

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

while (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

}

...

Inside the cppmain.h file I usually #include "main.h" to get all the STM32 HAL stuff and declare all hardware resource handles as extern (e.g. extern UART\_HandleTypeDef huart1;)

The full cppmain.h file:

/\*\* @file cppmain.h

\*

\* @author some\_author

\* @date some\_date

\*/

#ifndef PROGRAM\_INC\_CPPMAIN\_H\_

#define PROGRAM\_INC\_CPPMAIN\_H\_

#ifdef \_\_cplusplus

extern "C" {

#endif

#include "main.h"

void cppMain(void);

#ifdef \_\_cplusplus

}

#endif

#endif /\* PROGRAM\_INC\_CPPMAIN\_H\_ \*/

Some random cppmain.cpp file:

/\*\* @file cppmain.cpp

\*

\* @author some\_author

\* @date some\_date

\*/

#include "cppmain.h"

#include "pins.h"

// pins is useful when you connect some things to a dev board

// and want to have a quick reference where you put stuff

// this just includes things like (for STM32F103xB "blue pill" in this case)

// #define LED\_PIN\_PORT GPIOC

// #define LED\_PIN\_PIN GPIO\_PIN\_13

// #define LED\_PIN LED\_PIN\_PORT, LED\_PIN\_PIN

extern UART\_HandleTypeDef huart1;

int some\_variable = 0;

void foo(int input){

// do something

}

void cppMain(){

// do stuff

// main loop

while(1){

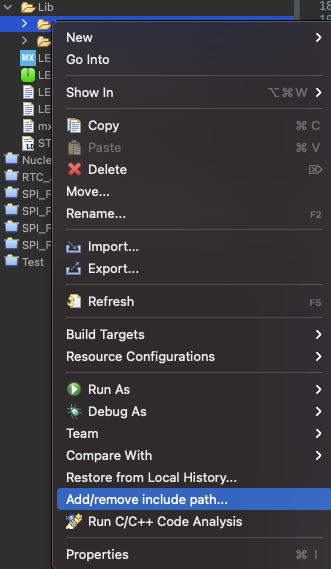
// do stuff regularly

}

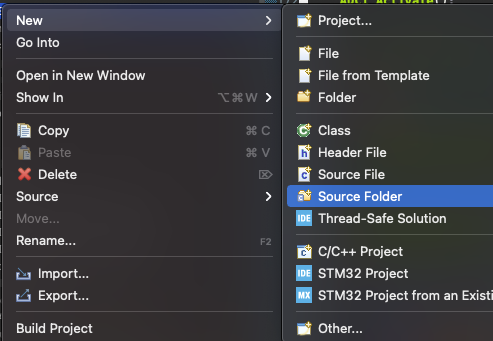
}

In my Lib folder I usually link to libraries I want to change for all projects while working on them: right click on Libs -> New -> Folder -> Advanced >> -> Link to alternate location. Note that this will link to an absolute path on your system and you can't simply import your project on a different system this way. If you work in a team or across different devices you might want to solve this a different way.

I am using STM32CubeIDE version 1.8, it can let you add include folder directly from the project menu -> Add/remove include path:

[](https://i.stack.imgur.com/JoK5b.png)

To add source folder, you can create source folder from the project menu -> New -> Source Folder:

[](https://i.stack.imgur.com/cWcGv.png)

If you want to add include and source folders manually:

Project Properties-> C/C++ General->Paths and Symbols:

Select Includes tab to add header folder

Select Source Location to add source folder