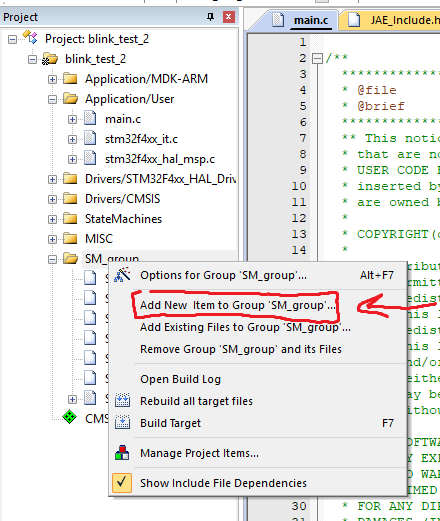
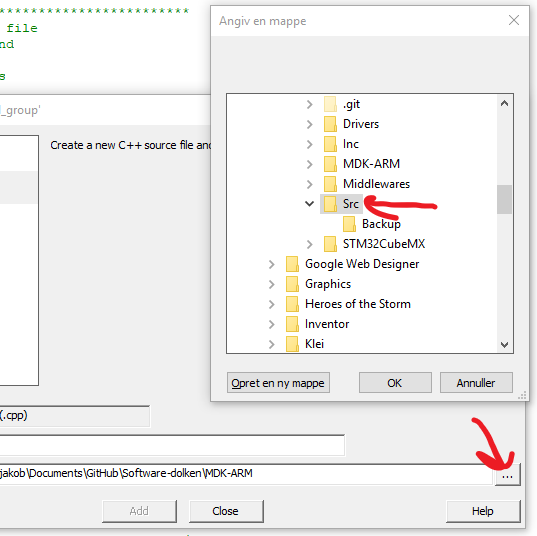
# Keil overview

## Creating a new h, cpp or c file

Click on the group (it’s not a folder) that you need to create the file in:



We use SM\_group for every StateMachine we use, MISC is for functions or alike and User are for essential files.

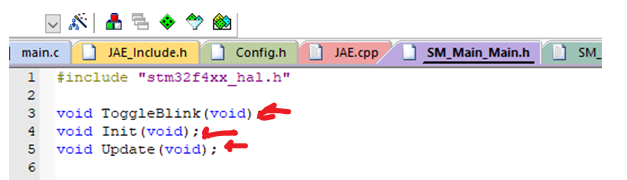


Make sure every new file is created in the “Src” folder.

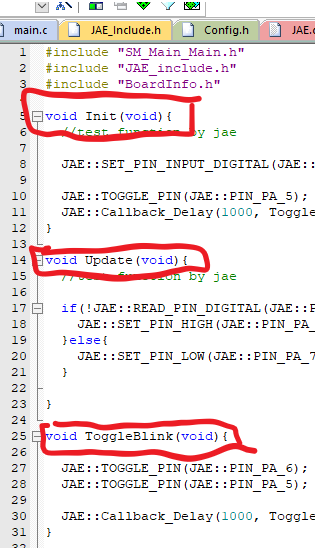
I recommend to use cpp even if you have never used it before since cpp can do all c can.

When creating the object file always create a header file and a source file (.cpp or .c).

Then declare all functions that will be used outside the object in the header file:



And define them in the cpp/c file:



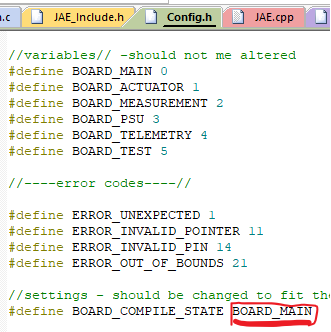
## Naming to files

Filesnames should be easy to understand and all stateMachines files should be named “SM\_YOURBOARDHERE\_YOURSTATEMACHINEHERE”.

## Notes

For some reason keil would like you to use #ifndef #define xxxxxx #ifend instead of #pragma once

When compiling for a board edit the BOARD\_COMEPILE\_STATE in config.h to the board that you need to compile to.



If in need of a vector or arraylist class use JAE::vector

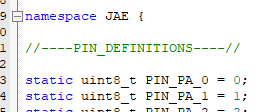
Also a lot of functions like SET\_PIN\_INPUT\_DIGITAL can be found in JAE namespace (use JAE::)

## **Coming from** c to c++

Use “extern “C” { #include “thingyouwanttoinclude.h”}” when including a c file (like stdclib witch is a std lib for c).

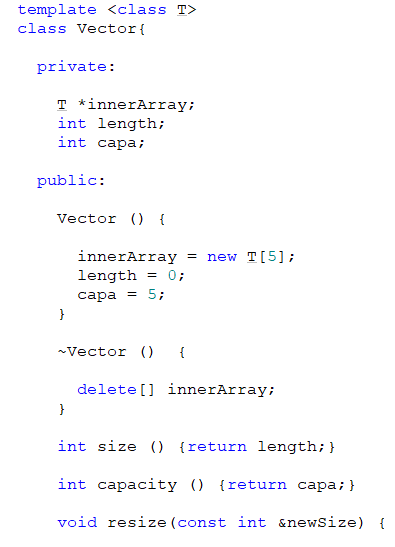
### Namespace

In c++ there are namespaces witch can be used to make it easer to see where functions belong, so as an example you can use a namespace to group functions, variabels or classes together, to use access the variables use the :: operator.



### Templates

You can use a template class for functions or classes that need to take in an unknown type af variable, like an vector class can store any type of variable.



### Classes

