STM32CubeIDE & C++

3 (simple?) solutions...

Assumptions

- STM32CubeIDE… is this the best possible solution? The decision is yours.. but…
- I want to use STM32CubeIDE because:
  - I am implementing R&D project…
  - I am learning how to work with a hardware platform…
  - The environment has the tools I want…
  - I need easy platform configuration in GUI mode…
  - …
- I accept the tool’s inconveniences

Requirements

- Easy switch Hal <-> LL ...
- Genetate perypherial code to other *.c/*.h files
- Switch public/private initialize perypherial functions
- Simplicity…
- …

Base project – demo

Solution 1

- Copy main.c and main.h to main.cpp and main.hpp
- Exclude main.c and main.h from project build for all configurations (Debug, Release, …)
- Change:
  #include ”main.h” to
  #include ”main.hpp”
  in main.cpp

- **Pros:**
  - Simple
  - Understandable
- **Cons:**
  - Requires copying from main.c/h to main.cpp/hpp every time the platform configuration changes
  - In practice, it prevents the generation of separate *.c/*.h modules from starting

Solution 1 – demo

I’m Sorry Munch…

Solution 2

- Unchanged main.c/h
- Add entry.cpp and entry.hpp
- Define entry(void) wrapped extern ”C” {.. } in entry.cpp
- Add #include ”entry.hpp” in main.c
- Add entry() call in main() (main.c file)
- Move while(1) to entry() in entry.cpp
- Add #include ”main.h” in entry.cpp

- **Pros:**
  - Partial C / C++ separation
  - One entry point
  - Working all C calls…
  - Less coupling when generating separate *.c/*.h modules
- **Cons:**
  - Laborious
  - Unresolved problem of generating separate modules by the code generator
  - Copying connections to entry.cpp file in case of changes in the separated modules generator

Solution 2 – demo

Probably.. OK!

Solution 3

- Disable main() generation
- Move:
  void entry(void) function to
  int main(void)
- Include main.h in entry.cpp
- Initialize platform in main()
- Switch visibility (static)
- Wrap macro - initialize HAL/LL

- **Pros:**
  - Full separation C/C++
  - Natural entry point (main())
  - Easy switch HAL < - > LL
  - Easy switch perypherial
    *.c / *.h
- **Cons:**
  - Complicated (first) configuration
  - Manual add init added perypherial

Solution 3 – demo

½ Empty

½ Full

Side effects

The End

Images: https://www.pexels.com/