STM32CubeIDE+MX part 3

STM32 Sequencer

- Integration cost of a real time operating system can be penalizing for a simple application:
  - required RTOS skills
  - increased application complexity
  - impact on RAM/ROM size
- Sequencer utility:
  - simple alternative to using a real-time operating system for less complex application cases
  - it does not cover all the services provided by an operating system
- See https://wiki.st.com/stm32mcu/wiki/Utility:Sequencer

Sequencer

Note that

- Sequencer is NOT an operating system
- goal:
  - not to compete with standard operating systems
  - but rather with standard bare metal implementation
- it is an optimized packaging of a while loop bare metal classic implementation
- it helps to avoid race conditions
  - often encountered in bare metal implementations
  - particularly when low power modes are implemented

Sequencer

Provided features :

- Register up to 32 tasks
- Request a task to be executed
- Pause and resume a task
- Wait for a specific event (which might be not blocking)
- Set priority on tasks
- Allow to manage an IDLE task

Example #1: execute a task every 400 ms

- Put the *sequencer* folder somewhere
- Create a project
- Add file *stm32_seq.c* and *utilities_conf.h* to the project
- Add the sequencer path …/*firmware/utilities/sequencer* to the include list /firmware/utilities/sequencer
  - *Project… Properties… C/C++ Build… Settings… MCU GCC Compiler… Include paths… Add...*
- Test *sequencer1.c*

Example #2: low power

- In MX:
  - Timers… LPTIM1 (we need a low power timer):
    - Mode: Counts internal clock events
    - Clock Prescaler: Div32
    - NVIC: Enable
  - Clock Configuration: set the LSI for LPTIM1
- In IDE
  - Set the project as in #1
  - Test *sequencer2.c*