README FreeRTOS Cortex M4/M7

The board we are using to verify the software is a Nucleo L476RG (CM4) and XM112 (CM7). Just send an e-mail to info@acconeer.com if you want a step-by-step bringup guide on how to connect the GPIOS and flash it.

To view release notes use the following link: https://developer.acconeer.com/sw-release-notes/

1 Development environment

The instructions are verified for Debian-based Linux distributions (such as Ubuntu). It is mandatory to use Ubuntu 18 if you are building your own applications with CM7. We have seen issues with the Ubuntu 16 and CM7 r0p1.

Make sure that the following packages are installed: make Use "apt-get install [package]" if needed.

Install compiler package for Cortex M

Go to [https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads] and download GNU Arm Embedded Toolchain 7-2018-q2-update June 27, 2018. you can also download the Linux64 version directly by running the command:

```
wget https://armkeil.blob.core.windows.net/developer/Files/downloads/gnu-rm/7-2018q2/gcc-arm
```

```
tar xjf gcc-arm-none-eabi-7-2018-q2-update-linux.tar.bz2
```

To build for Cortex M based targets you must also set GNU_INSTALL_ROOT to point to the toolchain's bin folder. Add the following to your .bashrc file: "' export GNU_INSTALL_ROOT=/gcc-arm-none-eabi-7-2018-q2-update/bin

2 Distributed files

- makefile and rule/ contain all makefiles to build the example programs.
- lib/*.a are pre-built Acconeer software.
- include/*.h are interface descriptions used by applications.
- source/example_*.c are applications to use the Acconeer API to communicate with the sensor.
- source/acc_board_*.c are board support files to handle target hardware differences.
- source/acc_driver_*.c are hardware drivers supposed to be customized to match target hardware.
- source/acc_driver_os_*.c is the operating system support module. It is not meant to be modified, but is provided for reference.

- doc/ contains HTML documentation for all source files. Open doc/rss_api.html .
- out/ contains pre-built applications (same as executing “make” again).

3 Building the software

- Change directory to the directory containing this file.
- To build the example programs, type “make” (the ZIP file already contains pre-built versions of them).
- All files created during build are stored in the out/ directory.
- “make clean” will delete the out/ directory.

4 Hardware adaptations

The provided example includes FreeRTOS, CMSIS, ST Standard Peripheral Library and example drivers for GPIO, SPI and UART. They have successfully been tested and verified on an STM32L476 (L476RG) and on a XM112, but they can be adapted to match your specific hardware needs. If an STM32F4xx MCU is used, these files can probably be used without change to get something up and running.

IMPORTANT:

The important file to change is source/acc_board_*.c. This file defines GPIO pin mapping for the specific hardware. The function acc_board_init() configures GPIO, SPI and UART drivers as well as initializes a UART for logging.

5 Executing the software

The provided rule/makefile_build_example_*.inc contain examples of how to program a device using OpenOCD. Feel free to adapt to your own hardware.