Guide for building API examples

Oct. 2020

Overview

This document is a guide to aid developers starting with the DWS3000 Arduino Shield. It includes an example to build one of the DW3000 API example using an IDE. It can be utilized for starting to build user applications on a target micro-controller board. It does not cover every details, please refer to the reference documents for more details.

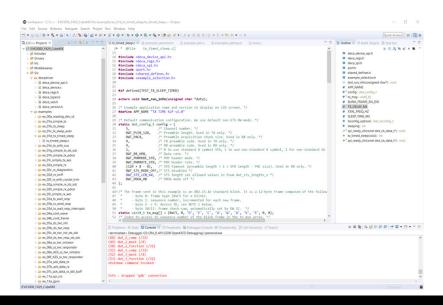
Building an API example code:

- Setup a AC6 tools
- Compile the example
- Debugging via OpenOCD and GDB on a target NUCLEO-F429ZI development board

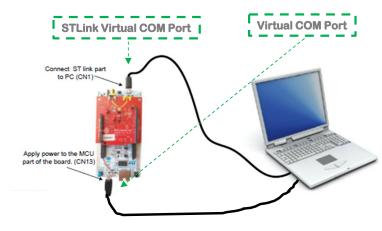
1: Build an API example code (1 of 5)

This example covers the NUCLEO-F429ZI from ST.

- a. Setup an AC6 tools
 - Install a Eclipse-based IDE (System Workbench for STM32)
 - Prepare another MicroUSB data cable and connect to top USB port in NUCLEO-F429 (install a driver if needed)
- b. Install GNU toolchain
 - Make an empty directory (example, C:/GnuToolsArmForEmbedded) then, install here
 - GNU Arm Embedded Toolchain
 - Q2 Update
- NOTE: Do not place space, special characters in the path of the source code + GNU toolchain folders







Both VCOM ports can be recognised from your PC

2: Build an API example code (2 of 5)

Place source code where there are no special char and spaces.

Import an existing project

Setup IDE for a project: import > General > Existing Projects into Workspace

Select root directory: path in your source code for API (example in the below)

Check 'NUCLEO_F429.cfg' file is under the project folder, then Build a project

NUCLEO_F429.cfg

Config for NUCLEO-F429ZI board with a single STM32F429ZITx chip generated by System Workbench for STM32

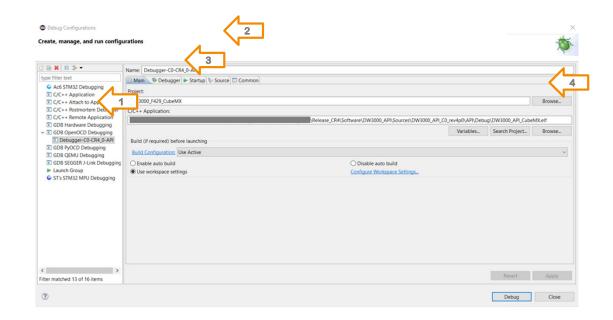
Example view: property of project



3: Build an API example code (3 of 5)

Create OpenOCD Debugging Pun > Debug Configurations > Create configurations >

Run > Debug Configurations > Create config: double-click GDB OpenOCD Debugging

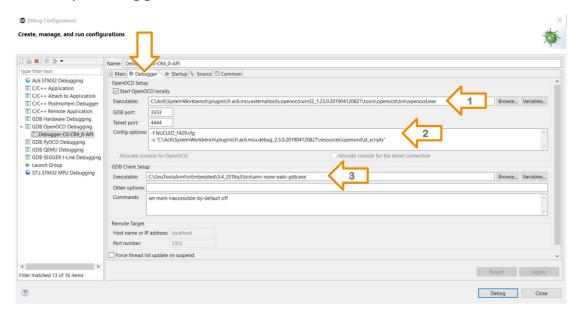


- If you don't have GDB Plug-in, install 'GNU ARM C/C++
 OpenOCD Debugging'
 - https://gnu-mcu-eclipse.github.io/debug/install/
- 2. Make name
- 3. Browse your project
- 4. Find application file (Debug folder in this project)

6

4: Build an API example code (4 of 5)

Setup Debugger



1. Find OpenOCD executable

 C:\Ac6\SystemWorkbench\plugins\fr.ac6.mcu.externaltools.o penocd.win32_1.23.0.201904120827\tools\openocd\bin\openocd\openocd\bin\openocd.exe

2. Config options

-f NUCLEO_F429.cfg -s
"C:\Ac6\SystemWorkbench\plugins\fr.ac6.mcu.debug_2.5.0.2
01904120827\resources\openocd\st_scripts"

3. Find GDB executable

 C:\GnuToolsArmForEmbedded\5.4_2016q3\bin\arm-noneeabi-gdb.exe

5: Build an API example code (5 of 5)

Build code, run/debug

1. Select an example: TX SLEEP TIMED

```
🔾 📦 🖯 😘 💸 💆 🗖 🖺 tx_timed_sleep.c 📓 example_selection.h 🖾 🗟 example_info.c 📓 examples_defines.h 🚨 main.c
C/C++ Projects 🛭 🤝

✓ 

EVK3000_F429_CubeMX

                                             ^ 2* * @file example_selection.h□
  > M Includes
                                                   13 #ifndef TEST_SELECTION_
  > 🐸 Drivers
                                                  14 #define TEST SELECTION
  > 4 Middlewares
                                                  16 #ifdef _cplusplus
  ∨ 🐸 Src
                                                      extern "C" {
    v 🗁 decadriver
      deca device api.h
                                                  20 //Enable the needed example/test. Please enable only one example/test!
      deca device.c
      > la deca_regs.h
                                                  220 //#define TEST READING DEV ID
      > 🖻 deca_types.h
                                                  23 //#define TEST_SIMPLE_TX
      deca_vals.h
                                                  24 //#define TEST_SIMPLE_TX_PDOA
25 //#define TEST_TX_SLEEP
      > 🖹 deca_version.h
                                                  26 //#define TEST_TX_SLEEP_IDLE_RC
27 //#define TEST_TX_SLEEP_AUTO

→ examples

      > ex_00a_reading_dev_id
                                                  28 #define TEST_TX_SLEEP_TIMED
      > ex 01a simple tx
                                                  29 //#define TEST TX WITH CCA
      > ex 01b tx sleep
      > @ ex_01c_tx_sleep_auto
                                                  310//#define TEST_SIMPLE_RX
32 //#define TEST_RX_DIAG

→ ex_01d_tx_timed_sleep

         > @ tx_timed_sleep.c
                                                  33 //#define TEST RX SNIFF
       > @ ex_01e_tx_with_cca
                                                  34 //#define TEST RX TRIM
       > @ ex_01g_simple_tx_sts_sdc
                                                  35 //#define TEST_SIMPLE_RX_PDOA
       ex 01h simple tx pdoa
```

2. Modify UWB config and blink interval: tx timed sleep.c

```
🖫 C/C++ Projects 🔞 🗘 🌣 降 🌡 👺 🤘 🗖 🔯 to_timed_sleep.c 🗯 🗎 example_selection.h 🔻 example_info.c 🕞 examples_defines.h 🔻 main
24 extern void test_run_info(unsigned char *data);
                                                                                                                      /* Example application name and version to display on LCD screen. */
      > 25 Drivers
                                                                                                                    #define APP_NAME "TX TIME SLP v1.0"
      > @ Middlewares
                                                                                                             29 /* Userfall Communication Configuration. We use userfall configuration.

**Static det_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_config_c_conf
       v @ decadriver
             deca device.c
               > R deca types.h
                                                                                                                                                                     /* 0 to use standard 8 symbol SFD. 1 to use non-standard 8 symbol. 2 for non-standard 16
                  deca_vals.h
                                                                                                                           > A deca_version.h

→ examples

               ex 00a reading dev id
                > ex_01a_simple_tx
                ex 01b tx sleep
                ex_01c_tx_sleep_auto
                                                                                                         ∨ ex 01d tx timed sleep
                   > @ tx_timed_sleep.c
               > @ ex_01e_tx_with_cca
                ex_01g_simple_tx_sts_sdc
                 ex 01h simple tx pdoa
                ex_01i_simple_tx_aes
                 ex_02a_simple_rx
                ≥ ex_02c_rx_diagnostics
                                                                                                                    #define BLINK FRAME SN IDX 1
                 ex_02d_rx_sniff
                ex_02f_rx_with_crystal_trim
                                                                                                      55 /* Inter-frame delay period in milliseconds. */
                 ex_02g_simple_rx_sts_sdc
                 ex_02h_simple_rx_pdoa
                  ex 02i simple rx aes
                                                                                                             59 #define XTAL FREQ HZ 38400000
                 ⊜ ex 03a tx wait resp
                    ex_03b_rx_send_resp
                                                                                                 610/* Sleep time, in milliseconds. See NOTE 2 below.
```

3. Run debug: It will build a project and download a firmware to the target

```
workspace - C/C++ - EVK3000_F429_CubeMX/Src/examples/ex_01d_tx_timed_sleep/tx_timed_sleep.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
1 Debugger-C0-CR4_0-API
© C/C++ Projects ☼ ⇔ ♀ @ □ 🕒 🕏 🔛 🗸 🗆 🖸 🔯 tx_timed_
                                                                          Debug As

→ 

EVK3000_F429_CubeMX

                                        24 exte
                                                                          char *data);
                                                  Debug Configurations...
                                         25
 > 🔊 Includes
                                                  Organize Favorites...
                                                                          rsion to display on LCD screen. */
 > 29 Drivers
                                        27 #define APP_NAME "TX TIME SLP v1.0
 > 🐸 Inc
  >  Middlewares
                                        29 /* Default communication configuration. We use default non-STS DW mode. */
                                         30 static dwt_config_t config = {

→ decadriver

                                         31
                                                              /* Channel number. */
                                         32
                                               DWT PLEN 128.
                                                              /* Preamble length. Used in TX only. */
     > la deca_device_api.h
                                                              /* Preamble acquisition chunk size. Used in RX onlv. */
```