Texas Instruments CC3200

This guide explains how to build applications for Texas Instruments CC3200 LaunchPad (further, CC3200) based on the Kaa C SDK.

Installing necessary components for Linux (Ubuntu)

Before building the C SDK for the CC3200 platform on Linux, you need to perform the following installation.

1. Install the GNU toolchain for ARM: `gcc-arm-none-eabi`

   ```
   wget
   https://launchpad.net/gcc-arm-embedded/4.9/4.9-2015-q2-update/+download/gcc-arm-none-eabi-4_9-2015q2-20150609-linux.tar.bz2
   tar -xjf gcc-arm-none-eabi-4_9-2015q2-20150609-linux.tar.bz2
   sudo mkdir /opt/kaa
   sudo chown <username>:<usergroup> /opt/kaa
   mv gcc-arm-none-eabi-4_9-2015q2 /opt/kaa/gcc-arm-none-eabi
   ```

2. Install and configure OpenOCD, which is needed for running and debugging applications.
   a. Install OpenOCD.

      ```
      sudo apt-get install openocd
      ```

   b. Add a rule file.

      ```
      cd /etc/udev/rules.d/
      sudo nano 98-usbftdi.rules
      ```

   c. Write the following code into the added rule file and then save the file.

      ```
      SUBSYSTEM=="usb", ATTRS{idVendor}=="0451", ATTRS{idProduct}=="c32a",
      MODE="0660", GROUP="dialout",
      RUN+="/sbin/modprobe ftdi-sio" RUN+="/bin/sh -c '/bin/echo 0451 c32a >
      /sys/bus/usb-serial/drivers/ftdi_sio/new_id'"
      ```

   d. Reload the rules.

      ```
      sudo udevadm control --reload-rules
      ```

   e. To use Openocd as a regular user, add yourself to the dialout group.

      ```
      sudo usermod -a -G dialout <username>
      ```

   f. Log out and log in to finish the process.

      NOTE: The board should be enumerated as `/dev/ttyUSB{0,1}`. Use `ttyUSB1` for UART.

3. Install CC3200SDK.
    a. Install Wine.

    ```
    sudo apt-get install wine
    ```

    b. Download CC3200 SDK.
    c. Unpack CC3200 SDK.

    ```
    wine /home/<user>/Downloads/CC3200SDK-1.2.0-windows-installer.exe
    cp -R <where_sdk_is_installed>/cc3200-sdk /opt/kaa
    ```

    d. Change the configuration file for the debug interface.

    In `/opt/kaa/cc3200-sdk/tools/gcc_scripts/cc3200.cfg` replace the following few lines

    ```
    interface ft2232
    ft2232_layout luminary_icdi
    ft2232_device_desc "USB <-> JTAG/SWD"
    ft2232_vid_pid 0x0451 0xc32a
    ```

    with

    ```
    interface ftdi
    ftdi_device_desc "USB <-> JTAG/SWD"
    ftdi_vid_pid 0x0451 0xc32a
    ftdi_layout_init 0x00a8 0x00eb
    ftdi_layout_signal nSRST -noe 0x0020
    ```

    e. Edit GDB configuration.

    ```
    sed -i 's#-f cc3200.cfg#-f
    /opt/kaa/cc3200-sdk/tools/gcc_scripts/cc3200.cfg#g'
    /opt/kaa/cc3200-sdk/tools/gcc_scripts/gdbinit
    ```

Installing necessary components for Windows

Before building the C SDK for the CC3200 platform on Windows, you need to perform the following installation.

1. Install cygwin with the additional following packages:
    - Archive/unzip
    - Archive/zip
    - Devel/autoconf
    - Devel/automake
    - Devel/libtool
    - Devel/subversion (Note: if using TortoiseSVN/Windows7, skip this file)
    - Devel/make
    - Devel/gcc-core
    - Devel/gcc-g++
    - Devel/mingw-gcc-core
    - Devel/mingw-gcc-g++
    - Devel/mingw-runtime

2. Install the GNU toolchain for ARM gcc-arm-none-eabi to the opt\kaa directory under the cygwin root directory (default: c:\cygwin)

3. Install CC3200SDK to the opt\kaa(if this folder doesn't exist create it) folder under the cygwin root directory.
4. Install cmake and add it bin folder to the system environment.

To enable debugging for your CC3200 applications, you will also need to build OpenOCD as described in the official *CC3200-Getting_Started_Guide (item 3.3.3)*

For more information, please refer to the official *CC3200-Getting_Started_Guide*.

Creating applications based on C SDK

Building C SDK for Linux

Before creating applications based on the C SDK, you should obtain the C SDK and build a static library from it. To do so, generate the C SDK in Admin UI, then extract the archive, `cd` to it and execute the following for Linux:

```
mkdir -p build
cd build
cmake -DKAA_PLATFORM=cc32xx -DCMAKE_TOOLCHAIN_FILE=../toolchains/cc32xx.cmake ..
make
```

For more details on building the C SDK for Linux, please refer to this page.

Building C SDK for Windows

Open the cygwin terminal and execute the following:

```
mkdir -p build
cd build
cmake.exe -G "Unix Makefiles" -DKAA_PLATFORM=cc32xx
-DKAA_TOOLCHAIN_PATH=c:/cygwin/opt/kaa
-DCMAKE_TOOLCHAIN_FILE=../toolchains/cc32xx.cmake ..
make
```

Example

To quickly start with the Kaa IoT platform, you can download one of the Kaa demo applications from the Kaa Sandbox and run it on the CC3200 board. We recommend that you start with the ConfigurationDemo.

Connect CC3200 LaunchPad to your PC through a micro-USB connector and execute the following in your terminal:

```
tar -xzf configuration_demo.tar.gz
cd CConfigurationDemo
./build.sh build
```

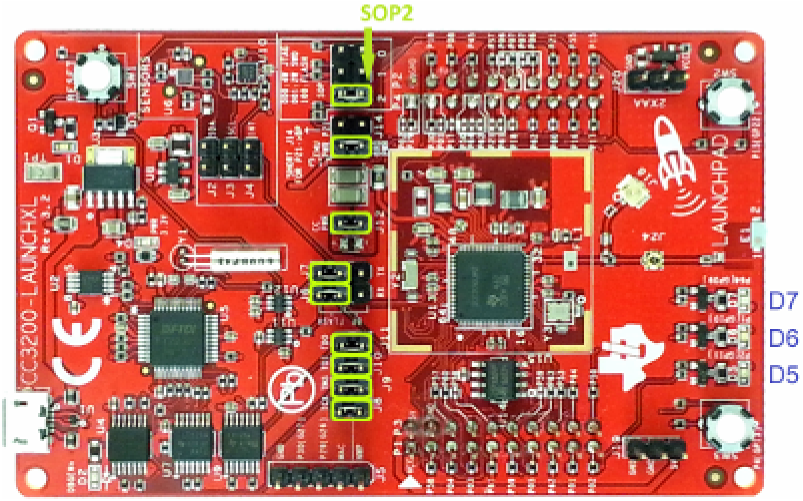
To launch the application, execute the following:

```
/opt/kaa/gcc-arm-none-eabi/bin/arm-none-eabi-gdb -x
/opt/kaa/cc3200-sdk/tools/gcc_scripts/gdbinit build/demo_client.afx
```

NOTE: If you want to see the debug output in the terminal, make sure to get connected to `/dev/ttyUSB{0,1}`.

Flashing (Windows only)

Jumpers on the CC3200 board should be connected as shown below.



To run an application, remove SOP2 and J8 jumpers, and then connect J2 and J3 jumpers.

For more information, see the official UniFlash Quick Start Guide.