Setting up Microbit Programming Environment

Before you start

The content of this guide has been tested on Ubuntu 20.04.

Setting up Eclipse IDE

Installing Eclipse IDE with C/C++ Development Tools (CDT)

Go to https://www.eclipse.org/downloads to download the latest version of Eclipse IDE. When installing, select `Eclipse IDE for C/C++ Developers`.

Installing Eclipse plugins

Within Eclipse IDE, go to `Help -> Install New Software`. In the drop-down menu `Work with`, select the URL that starts with CDT, then search and select the following plugins.

- C/C++ GCC Cross Compiler Support
- C/C++ GDB Hardware Debugging
- C/C++ Memory View Enhancements

Check on the lower left corner of the search box that you have 3 items selected, then click `Next` to finish the installation.

Repeat the above procedure, this time enter https://download.eclipse.org/embed-cdt/updates/v6/ in the `Work with` menu, click `Add` to the right of the drop-down menu to save it as a available site, and search for

- Embedded C/C++ OpenOCD Debugging

Similarly, add http://embsysregview.sourceforge.net/update as an available site and download

- Embedded Systems Register View (SFR)
- EmbSysRegView Data

Download OpenOCD and other required packages

Install required Ubuntu packages

```
$ sudo apt-get install gcc-arm-none-eabi
$ sudo apt-get install gdb-multiarch
```

(“`$`” indicates that the commands should be run in the terminal, and you don’t need to type “`$`” when running the commands yourself)

Install the latest OpenOCD version from souce code

1. Check if required packages are installed.

   These packages are required by OpenOCD

   - make
   - libtool
   - pkg-config >= 0.23 (or compatible)
   - libusb-1.0-0-dev

   These packages are required for building OpenOCD from source code:

   - autoconf >= 2.69
   - automake >= 1.14
   - texinfo >= 5.0

   To check whether a package is installed and/or what version it is, use the `--version` argument. For example, to check if `pkg-config` is installed, run `$ pkg-config --version`

2. Clone the OpenOCD repo `$ git clone git://git.code.sf.net/p/openocd/code openocd-code`

3. Build and install OpenOCD

   ```
   $ cd openocd-code
   $ ./bootstrap
   $ ./configure
   $ make
   $ sudo make install
   ```

   For more information regarding installing OpenOCD, refer to the `README` file in the `openocd-code` directory.

Setting up local project directory

Create an empty directory. This will be where your source code is located.

Download SDK for the nRF52 SoC

Download nRF5_SDK_17.0.2 from https://www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK/Download. Place it in a subfolder of your project directory called `nRF5_SDK_17.0.2_d674dde`.z (When you unzip the SDK, this folder will be created automatically)

Add custom board header

In the `nRF5_SDK_17.0.2_d674dde` directory, in subfolder `components/boards`, create custom_board.h. Its contents are in the appendix of this file (scroll below…)

Copy over example files

Place my modified blessed version in the project directory. The subdirectory should be called `blessed-devel`

Build the broadcaster example:

```
$ cd blessed-devel/examples/radio-broadcaster
$ make
```

Flash the example:

- in `blessed-devel/examples/radio-broadcaster`

```
$ chmod +x flash_openocd.sh
$ ./flash_openocd.sh
```

If this step fails with the warning "Error: unable to find a matching CMSIS-DAP device", try running

```
$ sudo ./flash_openocd.sh
```

first. If this succeeds, follow the steps described at https://forgge.github.io/theCore/guides/running-openocd-without-sudo.html to make it work without requiring root privileges.

After successfully flashing the example to the board you should see the microbit board wirelessly, e.g., by installing the "nRF connect for Mobile" app on your smartphone.

For debugging:

In Eclipse IDE: 1. Import the `radio-broadcaster` example as a makefile project 2. Click the in menu: `Run -> Debug Configurations` 3. Create a new `GDB`

`Openocd Debugging` configuration 1. in the "Main" tab, set "C/C++ Application" to `[PROJECTDIR]/blessed-devel/examples/microbit_leds/build/timing-example.out` (replace `[PROJECTDIR]` by an absolute path to your project) 2. in the "Debugger" tab, set the executable path to `/usr/local/bin/openocd` and the actual executable to `/usr/local/bin/openocd`. 3. in the "Config options" of the "Debugger" tab, insert `-f interface/cmsis-dap.cfg -f target/nrf52.cfg` 4. under "GDB client setup", set "Executable name" and "Actual executable" both to `/usr/bin/gdb-multiarch` 4. chose 5. enjoy :)

```
---  custom_board.h ---
/**
 * Copyright (c) 2012 - 2020, Nordic Semiconductor ASA
 *
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice, this
 *    list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form, except as embedded into a Nordic
 *    Semiconductor ASA integrated circuit in a product or a software update for
 *    such product, must reproduce the above copyright notice, this list of
 *    conditions and the following disclaimer in the documentation and/or other
 *    materials provided with the distribution.
 *
 * 3. Neither the name of Nordic Semiconductor ASA nor the names of its
 *    contributors may be used to endorse or promote products derived from this
 *    software without specific prior written permission.
 *
 * 4. This software, with or without modification, must only be used with a
 *    Nordic Semiconductor ASA integrated circuit.
 *
 * 5. Any software provided in binary form under this license must not be reverse
 *    engineered, decompiled, modified and/or disassembled.
 *
 * THIS SOFTWARE IS PROVIDED BY NORDIC SEMICONDUCTOR ASA "AS IS" AND ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL NORDIC SEMICONDUCTOR ASA OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
 * GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */
#ifndef PCA10000_H
#define PCA10000_H

#ifdef __cplusplus
extern "C"
#endif

#include "nrf_gpio.h"

// Definitions for PCA10000 v2.0.0 or higher

#define LEDS_NUMBER    0


#define LEDS_ACTIVE_STATE 0

#define BUTTONS_LIST
#define LEDS_LIST

#define LEDS_INV_MASK  LEDS_MASK

// there are no buttons on this board
#define BUTTONS_NUMBER 0

// UART pins connected to J-Link

#define TX_PIN_NUMBER   NRF_GPIO_PIN_MAP(0,6)
#define RX_PIN_NUMBER  NRF_GPIO_PIN_MAP(1,8)
#define CTS_PIN_NUMBER 0
#define RTS_PIN_NUMBER 0
#define HWFC           false

#ifdef __cplusplus

#endif

#endif
```