

Sky Wave Design Template Project

1 Overview

This document outlines setting up a Linux virtual machine for developing nRF51 applications using GCC and Qt as the debugger

1.1 Virtual Machine

The standard development environment used at Sky Wave Design is a VirtualBox virtual machine running **Ubuntu 15.04 64 bit**. VirtualBox can be downloaded from <https://www.virtualbox.org/wiki/Downloads>. Ubuntu can be downloaded here: <http://www.ubuntu.com/download/desktop>

1.2 Compiler

The compiler is GCC. ARM supports a branch of GCC specifically for Cortex M processors. Pre-built binaries are distributed at this site: <https://launchpad.net/gcc-arm-embedded>. The setup scripts described later on download releases from this site.

1.3 Python

In order to use the debugger, you need to have 32-bit Python 2.7.x installed. It is contained in the setup scripts but if you attempt to set up a different system than a clean Linux VM you may encounter issues.

Sky Wave Design does not use Python 2.x except for this debugger

2 Linux Setup

2.1 Setup Scripts

Sky Wave Design has a project in github that has setup scripts to get the necessary programs installed in your development machine. Currently the only OS described is Linux, but eventually Windows will be added and perhaps someday Mac OS X. Nothing in principle prevents you from using these other operating systems however the only way to be absolutely sure that this will work the first time is to use a virtual Linux machine.

Once your virtual machine is booted up obtain the setup scripts by running:

```
git clone https://github.com/skywavedesign/swd\_env\_setup.git
```

Then once you have cloned this repository, run the setup scripts starting with the script for Qt. Run all of the scripts as super user (starting with sudo).

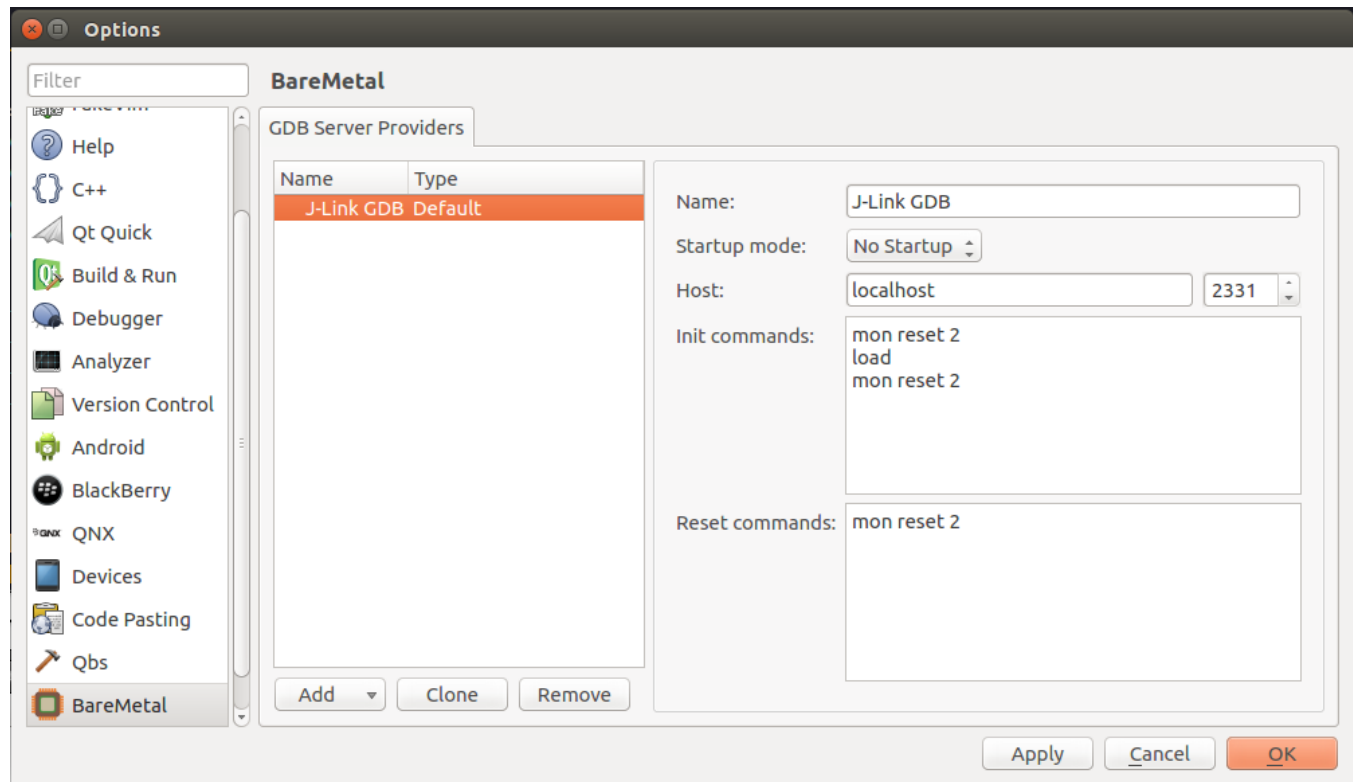
3 Qt Setup

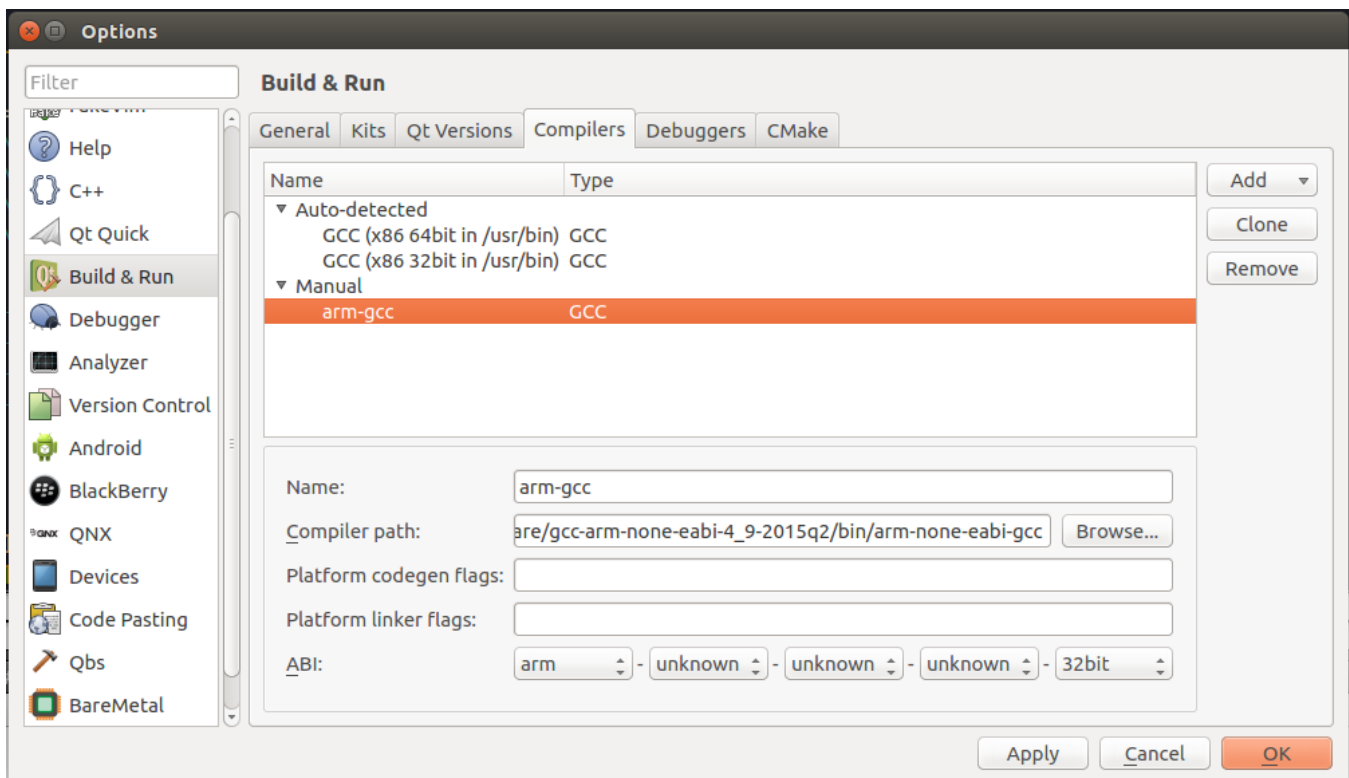
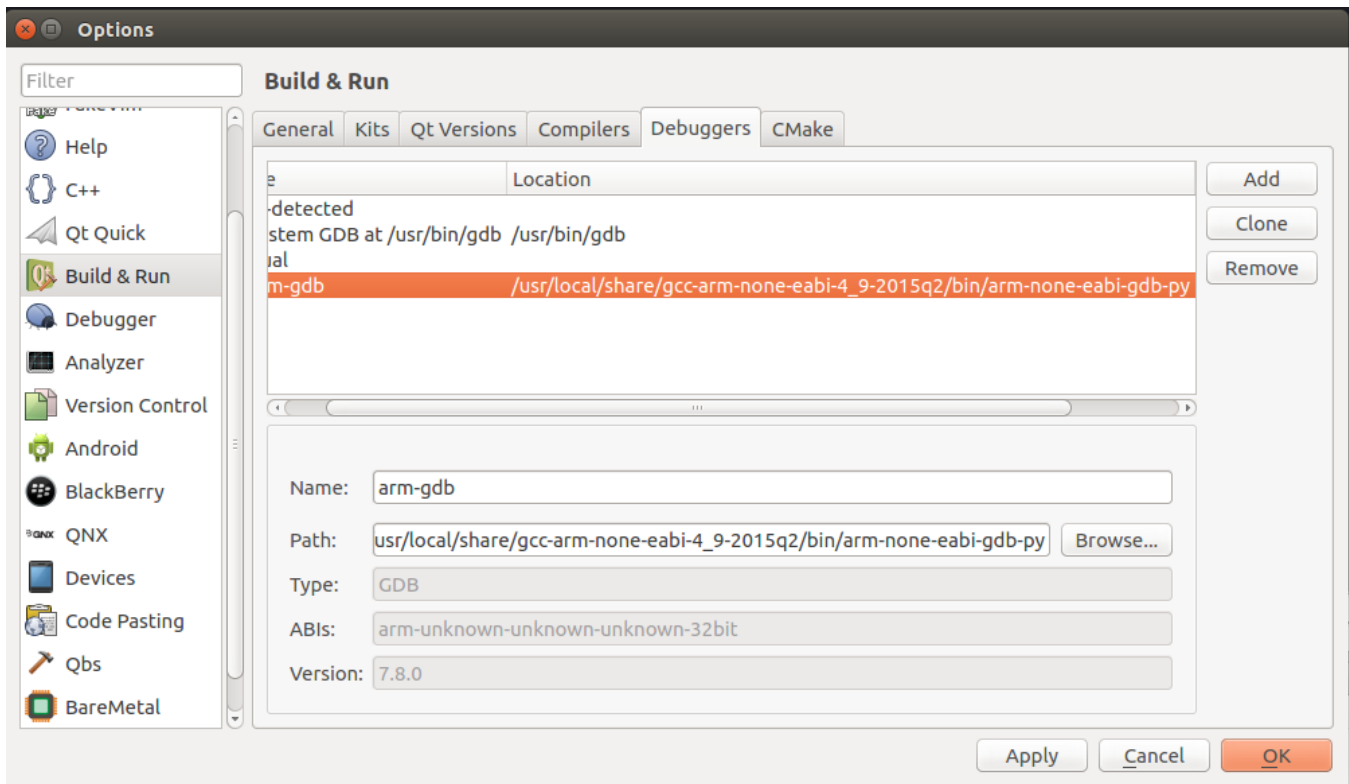
Once you have run all of the setup scripts, you have to set up Qt with the proper build settings. Unfortunately Qt does not offer a way to script this setup at this time.

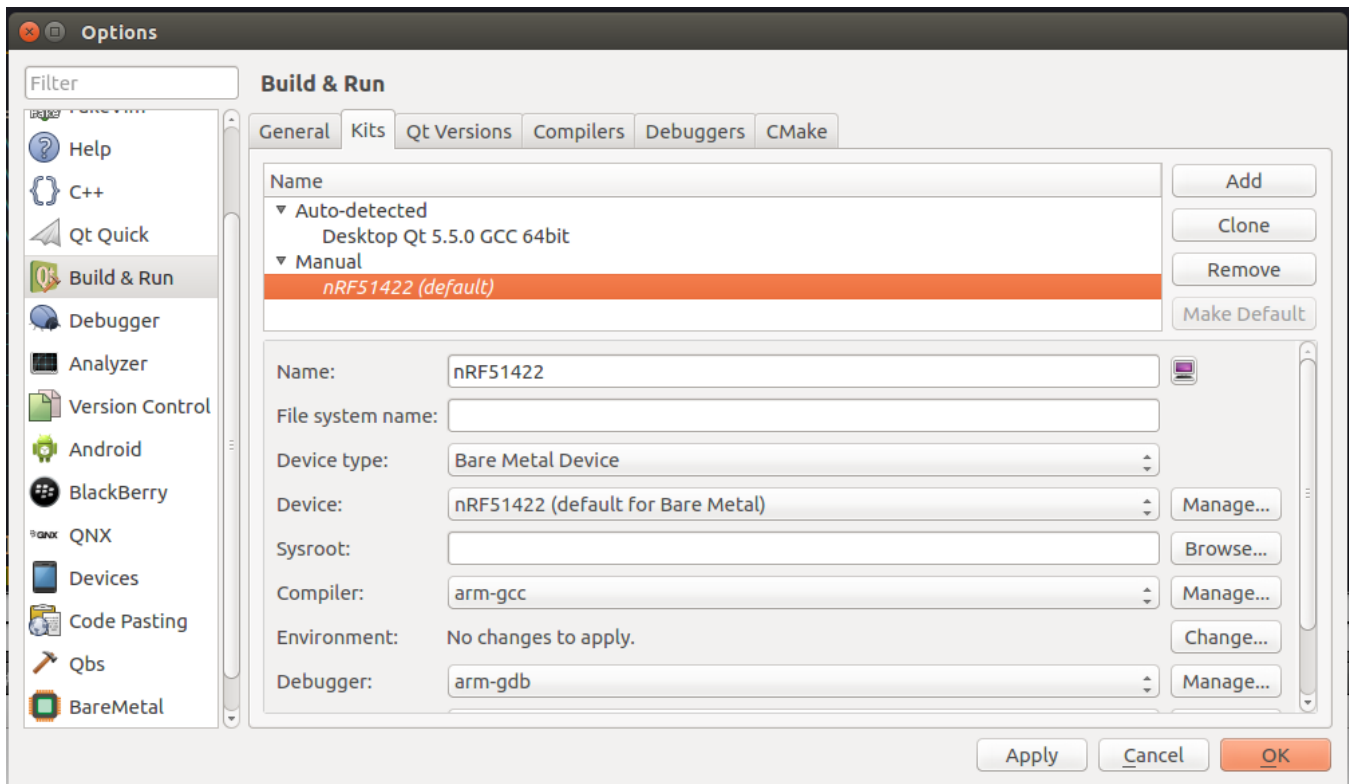
First of all, you need to enable the bare metal plugin:

- 1) Help -> About Plugins
- 2) Enable the Bare Metal plugin
- 3) Restart Qt Creator

Once you have done this, you can then proceed to set up the rest of the environment. Your Options window should look like the following screenshots.







4 Debugging

4.1 J-Link Debugger Software

Download and install the J-Link debugger software from J-Link. I was unable to figure out how to script this process.

4.2 Running the Debugger

Use the following command to run the GDB server:

```
JLinkGDBServer -device nrf51822 -if swd -speed 1000 -port 2331
```

You should then be able to hit the play button in Qt Creator or the debug button and debug away!

5 SDK Mods

We at Sky Wave Design start fresh with our board config so we don't use the board support package. Because of this you will have to comment out a few lines in the 8.1 SDK in order to get this project to build. Just comment out the `#include "bsp.h"` line and any subsequent lines with not found errors.

We have our own version of the SDK in a private repository where we make bug fixes and other necessary modifications, but obviously Nordic would not appreciate us sharing that code.

6 Template Project Description

6.1 Custom Profile

The template project uses a custom profile, which is a collection of services. It is not a true profile as we do not have any profile descriptors in the GATT server attribute table, however we called it profile for lack of a better term.

The profile contains the custom service, device information service, and battery service.

6.2 Custom Service

Many BLE applications require a custom service with 128-bit custom UUIDs. This custom service can be used as a template with all of the necessary code to immediately get started. Note that included in this repository is a Python script that will generate a random UUID to use as your base UUID. Use this function to ensure that you do not end up with the same UUID as everyone else who cloned this repository 😊.

The custom service contains a command characteristic and a status characteristic. Eventually we will add the ability to send back a notification using the buttons on the PCA10028 board.

The only command we have implemented is the command to turn the LEDs on/off. The first byte written to the command characteristic is the command number, in this case 01. The second byte 4 lsbs are a bitmask for the LEDs.



Sky Wave Design, LLC
7083 Hollywood Blvd
Los Angeles, CA 90028
Tel: 310.564.6144
www.skywavedesign.com

Master Control Panel

File Help

Master emulator

COM8 - 680707471 680707471 connected Reset

Device info

Device address: CA161362DFC2 Bonded: False

Actions

Disconnect Bond Update Discover services Disable services DFU

Service Discovery

- PrimaryService, Value: 00-18, Generic Access (0x1800)
 - CharacteristicDeclaration, Value: 0A-03-00-00-2A, Properties: Read, Write, Characteristic UUID: 0x2A00
 - DeviceName, Value: 54-65-6D-70-6C-61-74-65-20-41-70-70-20-42-75-69-6C-64-20-30-30-30, DeviceName: Template App Build 000
 - CharacteristicDeclaration, Value: 02-05-00-01-2A, Properties: Read, Characteristic UUID: 0x2A01
 - Appearance, Value: 00-00, Appearance: 0x0000
 - CharacteristicDeclaration, Value: 02-07-00-04-2A, Properties: Read, Characteristic UUID: 0x2A04
 - SlavePreferredConnectionParameters, Value: 10-00-20-00-00-00-90-01, MinConnInterval: 0x0010, MaxConnInterval: 0x0020, SlaveLatency: 0x0000, SupervisionTimeoutMultiplier: 0x0190
- PrimaryService, Value: 01-18, Generic Attribute (0x1801)
 - CharacteristicDeclaration, Value: 20-0A-00-05-2A, Properties: Indicate, Characteristic UUID: 0x2A05
 - ServiceChanged, (No values read)
 - ClientCharacteristicConfiguration, Value: 02-00, CharacteristicConfigurationBits: Indication (0x0002)
- PrimaryService, Value: B7-48-30-2C-8A-DE-B4-B7-77-4B-BD-40-00-00-6F-7A, 0x7A6F0000-40BD-4B77-B7B4-DE8A2C3048B7
 - CharacteristicDeclaration, Value: 1C-0E-00-B7-48-30-2C-8A-DE-B4-B7-77-4B-BD-40-01-00-6F-7A, Properties: WriteWithoutResponse, Write, Notify, Characteristic UUID: 0x7A6F0001-40BD-4B77-B7B4-DE8A2C3048B7
 - UUID: 7A6F0001-40BD-4B77-B7B4-DE8A2C3048B7, Value: 01-00
 - CharacteristicUserDescription, Value: 43-6F-6D-6D-61-6E-64, UserDescription: Command
 - ClientCharacteristicConfiguration, Value: 01-00, CharacteristicConfigurationBits: Notification (0x0001)
 - CharacteristicDeclaration, Value: 12-12-00-B7-48-30-2C-8A-DE-B4-B7-77-4B-BD-40-02-00-6F-7A, Properties: Read, Notify, Characteristic UUID: 0x7A6F0002-40BD-4B77-B7B4-DE8A2C3048B7
 - UUID: 7A6F0002-40BD-4B77-B7B4-DE8A2C3048B7, Value: 00
 - CharacteristicUserDescription, Value: 53-74-61-74-75-73, UserDescription: Status
 - ClientCharacteristicConfiguration, Value: 01-00, CharacteristicConfigurationBits: Notification (0x0001)
- PrimaryService, Value: 0F-18, BatteryService (0x180F)
 - CharacteristicDeclaration, Value: 12-17-00-19-2A, Properties: Read, Notify, Characteristic UUID: 0x2A19
 - Battery Level, Value: 64
 - ClientCharacteristicConfiguration, Value: 01-00, CharacteristicConfigurationBits: Notification (0x0001)
- PrimaryService, Value: 0A-18, Device Information (0x180A)
 - CharacteristicDeclaration, Value: 02-1B-00-29-2A, Properties: Read, Characteristic UUID: 0x2A29
 - Manufacturer Name String, Value: 41-43-4D-45-20-42-4C-45-20-44-65-73-69-67-6E
 - CharacteristicDeclaration, Value: 02-1D-00-24-2A, Properties: Read, Characteristic UUID: 0x2A24
 - Model Number String, Value: 50-43-41-30-30-30-32-38
 - CharacteristicDeclaration, Value: 02-1F-00-25-2A, Properties: Read, Characteristic UUID: 0x2A25
 - Serial Number String, Value: 30-30-30-30-2D-30-30-30-31
 - CharacteristicDeclaration, Value: 02-21-00-27-2A, Properties: Read, Characteristic UUID: 0x2A27
 - Hardware Revision String, Value: 52-65-76-20-31
 - CharacteristicDeclaration, Value: 02-23-00-26-2A, Properties: Read, Characteristic UUID: 0x2A26
 - Firmware Revision String, Value: 30-2E-30-2E-31-20-42-75-69-6C-64-20-30-30-33

Attribute value

UUID (0x): 0001 Handle (0x): 000E ☐ Display as UTF8

Value: ☒ hex ☐ text 01-0F

Read long Read

Write long Write

Back

Log

```
[15:46:45.6] Connection Latency: 0
[15:46:45.6] Supervision Timeout: 3000ms
[15:46:45.6] Clock Accuracy: (1)
[15:46:45.6] -----
[15:46:45.6] Connected to CA161362DFC2
[15:46:46.5] EnableServices([0x000B:2,0x0010:1,0x0014:1,0x0018:1,])
[15:46:46.6] Updated handle 0010 with value [1, 0]
[15:46:46.6] Successfully updated the store value of CCCD
[15:46:46.7] Updated handle 0018 with value [1, 0]
[15:46:46.7] Successfully updated the store value of CCCD
[15:46:46.7] Updated handle 000B with value [2, 0]
[15:46:46.7] Successfully updated the store value of CCCD
[15:46:46.7] Updated handle 0014 with value [1, 0]
[15:46:46.7] Successfully updated the store value of CCCD
```



Sky Wave Design, LLC
7083 Hollywood Blvd
Los Angeles, CA 90028
Tel: 310.564.6144
www.skywavedesign.com

7 TODO List

We have a lot of things we want to add to this project:

- 1) Easy way to switch back and forth between DFU mode and regular mode
- 2) Button presses to send back status updates via notifications
- 3) Ability to flash the bootloader and stack to the board

If you would like to contribute, please make the modifications and submit pull requests!