Rigado LLC
3950 Fairview Industrial Dr. SE Suite 100
Salem, Oregon 97302
866-6-RIGADO • modules@rigado.com
www.rigado.com/modules

BMD-200 Programming Guide

This document describes the programming process for the **BMD-200** System on Module (SoM) from Rigado.

The purpose of this guide to provide instructions for programming the **BMD-200** SoM using one of several supported development environments.

1. Overview

The **BMD-200** from Rigado is a powerful, highly flexible Bluetooth Smart module based on the nRF51822 SoC from Nordic Semiconductor. With an ARM® Cortex™ M0 CPU, embedded 2.4GHz transceiver, and on-module chip antenna, the BMD-200 provides a complete RF solution with no additional RF design, allowing faster time to market. The BMD-200 provides full use of the nRF51822's on-chip peripherals, allowing for a wide range of applications without the need for an external host microcontroller.

Software development for the **BMD-200** is supported with an extensive offering of tools, libraries, and example code. The software architecture, tools used for development, and the programming procedures for the **BMD-200** are described throughout the following sections of this document.

Table of Contents

Table of Figures

2. Software Architecture

A complete software solution for the nRF51822 consists of four key elements, including the SoftDevice, the Bluetooth Smart Profile(s), the application drivers, and the application code. These parts are illustrated in Figure 1.

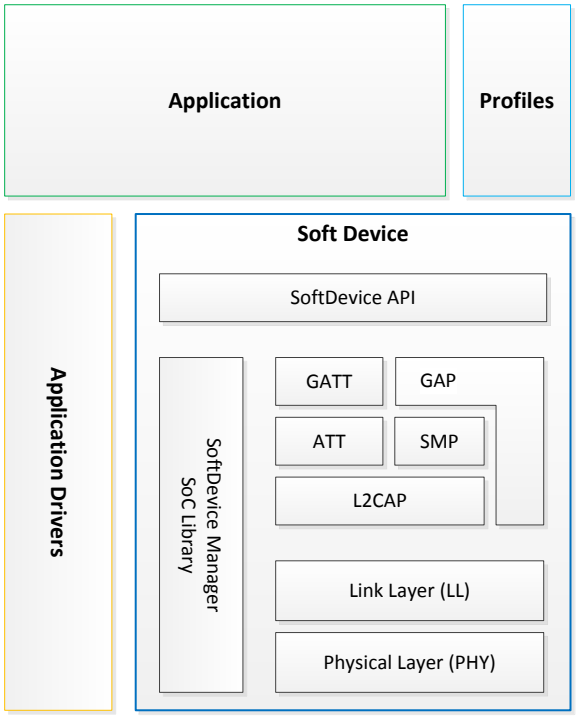

Figure 1- Software Architecture

As shown in the above figure, the SoftDevice is comprised of many different components that implement the wireless protocols for the Bluetooth Smart radio. Multiple versions of the SoftDevice are provided by Nordic Semiconductor as pre-compiled binaries on their website, as discussed in section 3.3.

On top of the SoftDevice, the Bluetooth Smart Profile(s) appropriate to the application are included. Profiles are high level definitions for how services are used to enable the application or product use case. Examples of profiles include Blood Pressure, Proximity, and Alert Notification.

The last two elements of the software solution are the application drivers, and the application itself. The application drivers enable abstraction of the SoC peripherals to simplify common tasks such as the UART, SPI, and I2C interfaces.

3. Software Tools

3.1. Software Development Kit

Software development begins with the nRF51 Software Development Kit (SDK) provided by Nordic Semiconductor. The nRF51 SDK contains C code for Bluetooth Smart Profiles, wireless communication application examples, and libraries for all of the peripherals on the SoC. The SDK can be downloaded through the Noric Semiconductor website: http://developer.nordicsemi.com/.

3.2. Development Toolchains

The nRF51 SDK is built on the ARM® CMSIS standard and is compatible with the following tool chains:

Keil MDK

The Keil MDK-ARM is an integrated development environment designed for microcontroller applications that enables development using the nRF51 SDK application and example files. MDK-ARM is available in four editions: MDK-Lite, MDK-Cortex-M, MDK-Standard, and MDK-Professional. All editions provide a C/C++ development environment and MDK-Professional includes extensive middleware libraries. More information is available at http://www.keil.com/arm/mdk.asp.

GCC in Eclipse

Eclipse is an open source, integrated development environment. Combined with the GNU C Compiler (GCC), it creates a free and powerful C/C++ development toolchain. Instructions for development with GCC in Eclipse are available in the nAN-29 application note from Nordic Semiconductor: http://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822.

3.3. SoftDevice

SoftDevices integrate a Bluetooth Smart controller and host, and provide an API for building Bluetooth Smart solutions with the nRF51822 SoC. Nordic Semiconductor offers three versions of the SoftDevice:

- **S110** SoftDevice is a Bluetooth® Smart Peripheral protocol stack solution
- **S120** SoftDevice is a Bluetooth® Smart Central protocol stack solution supporting up to eight simultaneous Central role connections.
- **S130** SoftDevice is a Bluetooth® Smart concurrent multi-link protocol stack solution supporting simultaneous Central/Peripheral role connections

SoftDevices can be downloaded from the Nordic Semiconductor website: www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822.

3.4. Additional Software Tools

Below is a list of additional tools that aid in development with the BMD-200 Bluetooth module. Not all tools will be required depending on which software suite is used.

Tool	*Description*
Nordic nRFgo Studio	The Nordic software suite is used to program and configure Nordic nRF devices. It supports programming of nRF51 application, bootloader, and SoftDevice.
nRF Master Control Panel for Android	A Nordic tool for Android devices to allow active scanning for Bluetooth low energy devices and communication.

4. Hardware Tools

The recommended programmer/debugger used in this document is the Segger J-Link. Several versions of this described on the Segger website (https://www.segger.com/jlink-debug-probes.html) and are available from various distributors.

The latest software and drivers for the J-Link can be downloaded directly from the Segger website. The J-Link drivers are also part of the nRF Tools package that is bundled with nRFgo Studio (see section 3.4). In addition, the nRF Tools package is available as a separate download from the Nordic Semiconductor nRF51822 page under the 'Downloads' tab.

5. Program the BMD-200 with Segger J-Link in Keil

This section walks through the process for set up and programing the BMD-200 using the Segger J-Link and the Keil MDK-ARM development kit.

5.1. Set up the Hardware

The BMD-200 supports the two pin Serial Wire Debug (SWD) interface. As shown in Figure 2, these two pins along with power and ground are the only connections needed to program the board.

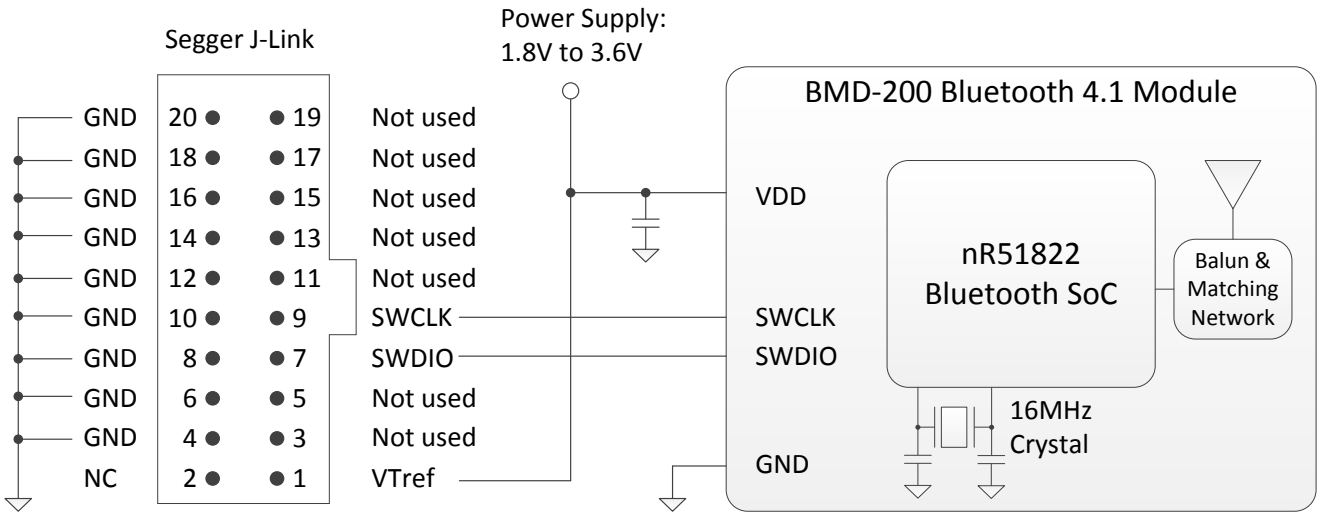

Figure 2- Programmer Connection

5.2. Set up the Toolchain

1. Download and install nRFgo Studio from www.nordicsemi.com/eng/Products/2.4GHz-RF/nRFgo-Studio
2. Download and extract the latest SoftDevice (SD110-SD-v8 is used for this guide) from the 'Downloads' tab at www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822.
3. Download and install the latest Keil MDK-ARM development kit from www.keil.com/arm/mdk.asp
4. Once the Keil MDK-ARM installation is complete, Keil will launch the Pack Installer. The Keil Pack Installer manages software components for the various microcontrollers supported by Keil. In the pane on the far right, select Nordic Semiconductor. Install all packs as shown in Figure 3 with a green diamond and "Up to date" in the action column

   Note: If the Pack Installer does not run, it can be opened in the 'Project' menu under 'Manage -> Pack Installer….'

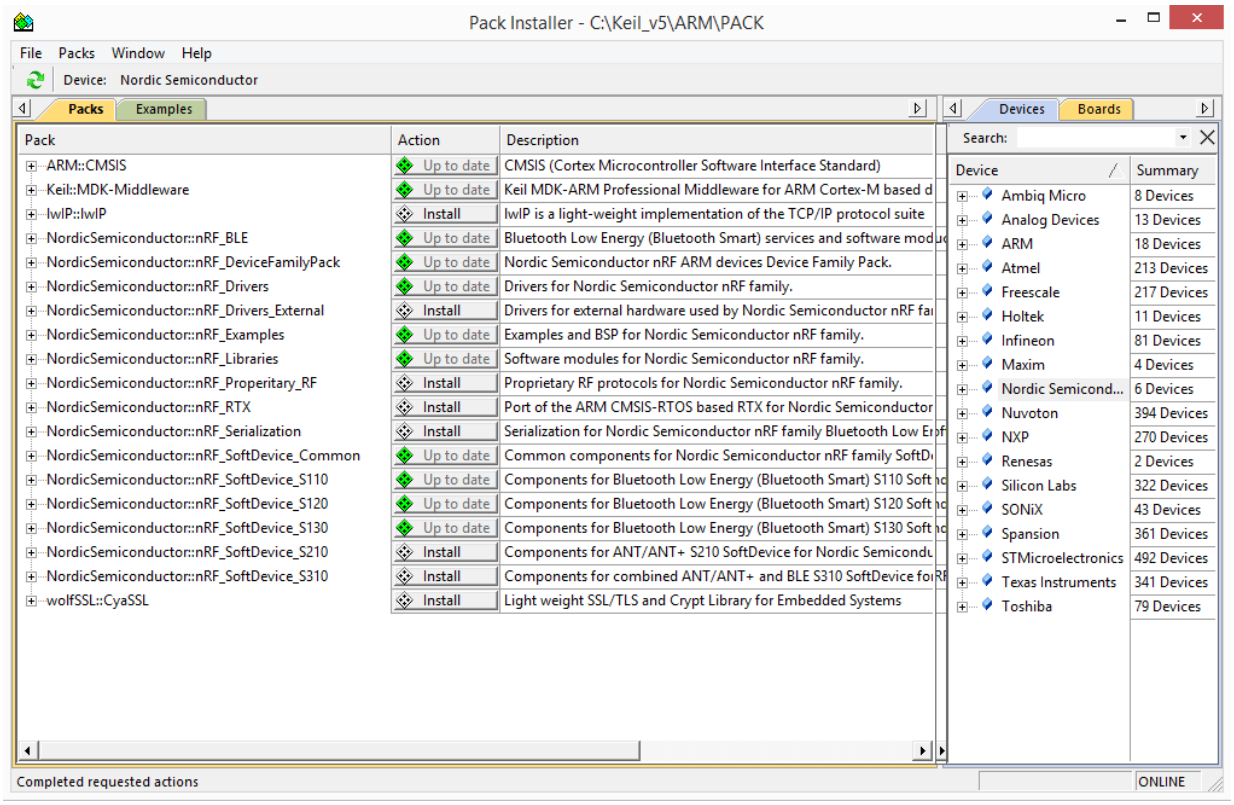

Figure 3- Keil MDK-ARM Pack Installer

5. The installed packs should include:
   - nRF_BLE
   - nRF_DeviceFamilyPack
   - nRF_Drivers
   - nRF_Example
   - nRF_Libraries
   - nRF_SoftDevice_Common
   - nRF_SoftDevice_S110
   - nRF_SoftDevice_S120
   - nRF_SoftDevice_S130

6. Once these packs are up to date, close the Pack Installer

5.3. Set up the Application Project

5.3.1. Copy the Example Project

1. Open the Keil uVision5 IDE. Open the pack installer with the button in the right side of the bottom row of the toolbar ( ). Select the examples tab at the top of the left pane of the pack installer. Make sure that Nordic Semiconductors is selected on the right hand pane so that only the relevant examples are shown

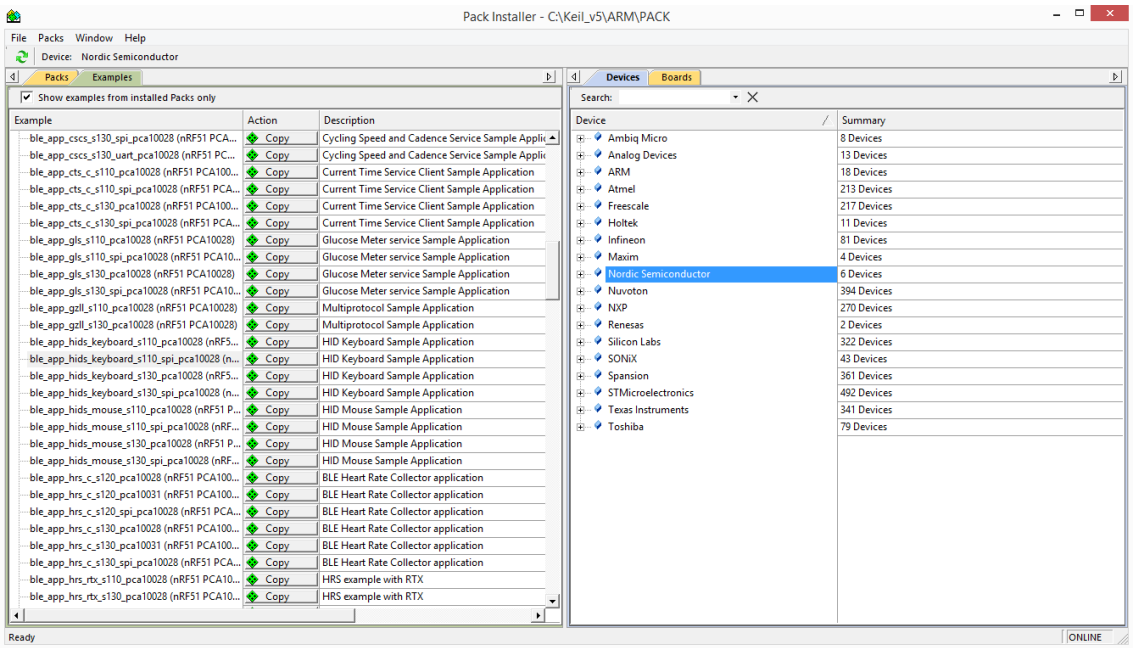

Figure 4- Keil MDK-ARM Example Browser

2. Select one of the examples that is labeled with (nRF51 PCA10028). The Heart Rate Service Sample Application is a good place to start. If trying to make a custom application, it is recommended to begin with the template project. **NOTE: This guide only covers the installation and use of SoftDevice S110. Make sure to select an example that is also labeled for use with S110.** Click the copy button and select the desired path for the project. Keil uVision will open the copied project from the selected directory

5.3.2. Add the Board Support File

The application requires a board support file that contains specific definitions for this application. A simple file for this Beacon project is available on the Rigado website.

Go to www.rigado.com/modules, browse to the BMD-200 page and download the BMD-200 Board Support File for Keil under the 'Downloads' tab. Save the file into the following directory: C:\Keil_v5\ARM\Pack\NordicSemiconductor\nRF_Examples\X.X.X\bsp\

Note: Typically, this file is modified or replaced with one that includes definitions specific to the board hosting the BMD-200 module.

5.3.3. Modify the Code As Needed

If the hardware used for this example **does not** have a 32KHz crystal connected to the BMD-200, it is necessary to edit the 'main.c' file to use the internal Ring Oscillator slow clock in the nRF51822 for this project. If the board hosting the BMD-200 module **does** have a 32KHz clock connected to the BMD-200, please skip to section 5.4.

1. Open Keil and the example project. Note that you may have to change the 'File Type' filter in the 'Open File' window to 'Project Files.'
2. Double click on the 'main.c' file under the 'Application' folder in the 'Project' window to open it, and find the function named, "ble_stack_init."
3. Comment out the line:
   SOFTDEVICE_HANDLER_INIT(NRF_CLOCK_LFCLKSRC_XTAL_20_PPM, false);
4. Add the line:
   SOFTDEVICE_HANDLER_INIT(NRF_CLOCK_LFCLKSRC_RC_250_PPM_TEMP_4000MS_CALIBRATION, false);

   The code should look like Figure 4.

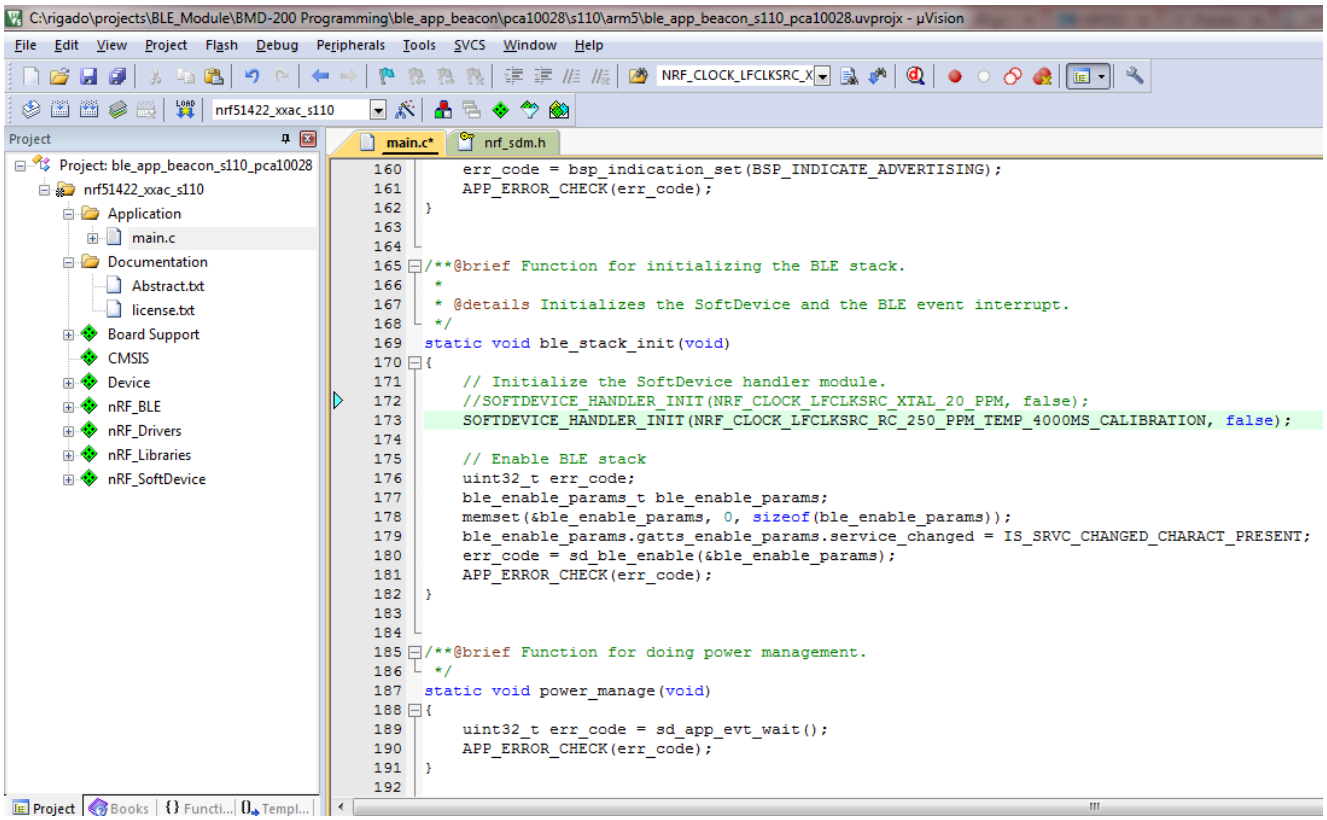

Figure 4- Edit for NO External 32KHz Clock

5.4. Connect and Program the BMD-200

5.4.1. Connect Segger J-Link to computer

1. Connect the Segger J-Link to the computer
2. After a few seconds the computer will recognize the J-Link

5.4.2. Program the BMD-200 with the SoftDevice

1. Open the Nordic nRFgo Studio application. In the 'Device Manager' window, select 'nRF51 Programming.' There will be a device named "Segger," followed by a serial number and a programming interface in the main window (see Figure 5)
2. Click 'Erase All' to clear the device of all current programmed software
3. Select the 'Program SoftDevice' tab in the pane on the right
4. Click 'Browse…' in the 'Program SoftDevice' tab and locate the SoftDevice *.hex file downloaded in Step 2 of section 5.2.
5. Click program
6. After programming, nRFgo Studio should look similar to Figure 5.

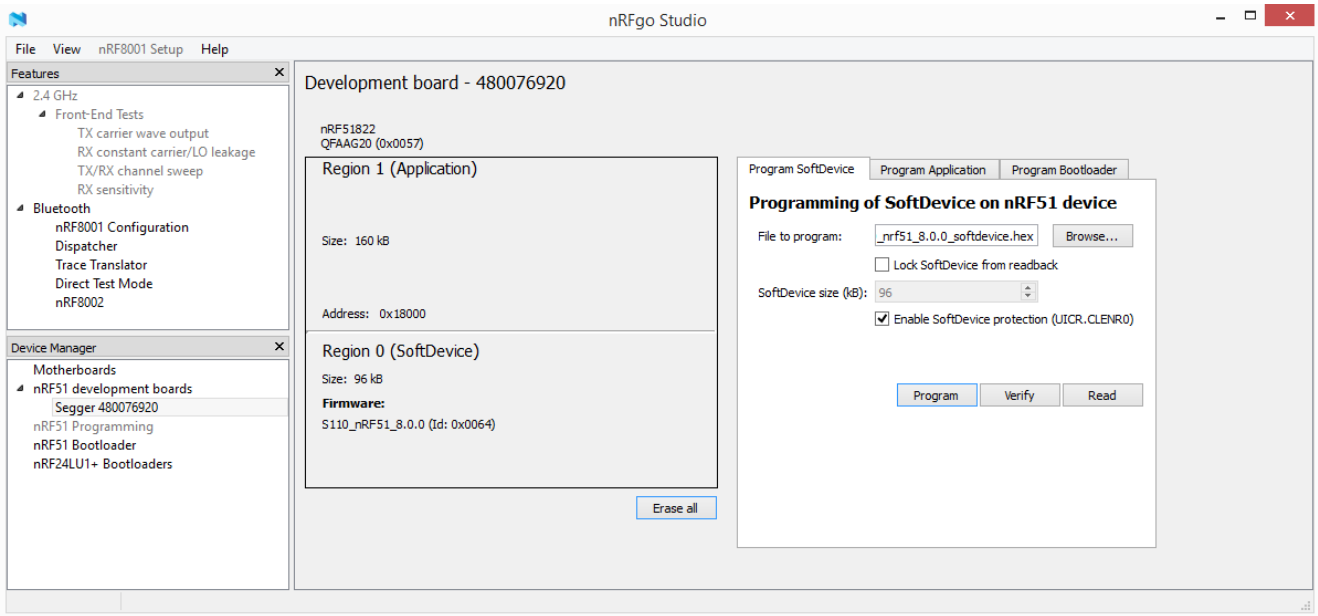

Figure 5- nRFgo Studio SoftDevice window

7. Close nRFgo studio.

5.4.3. Program the BMD-200 with the Application

1. Open Keil and the desired example project (created in section 5.3). Note that you may have to change the 'File Type' filter in the 'Open File' window to 'Project Files'
2. In the 'Flash' drop down menu click on 'Configure Flash Tools.' In this window go to the 'Device' tab and change the target device in the bottom left field to **nRF51822_xxAA**.

   Note: Any time a different device is selected, the values in the 'Target' tab will change

3. In the same 'Configure Flash Tools' window, go to the 'Target' tab and change the settings to match Figure 6 below

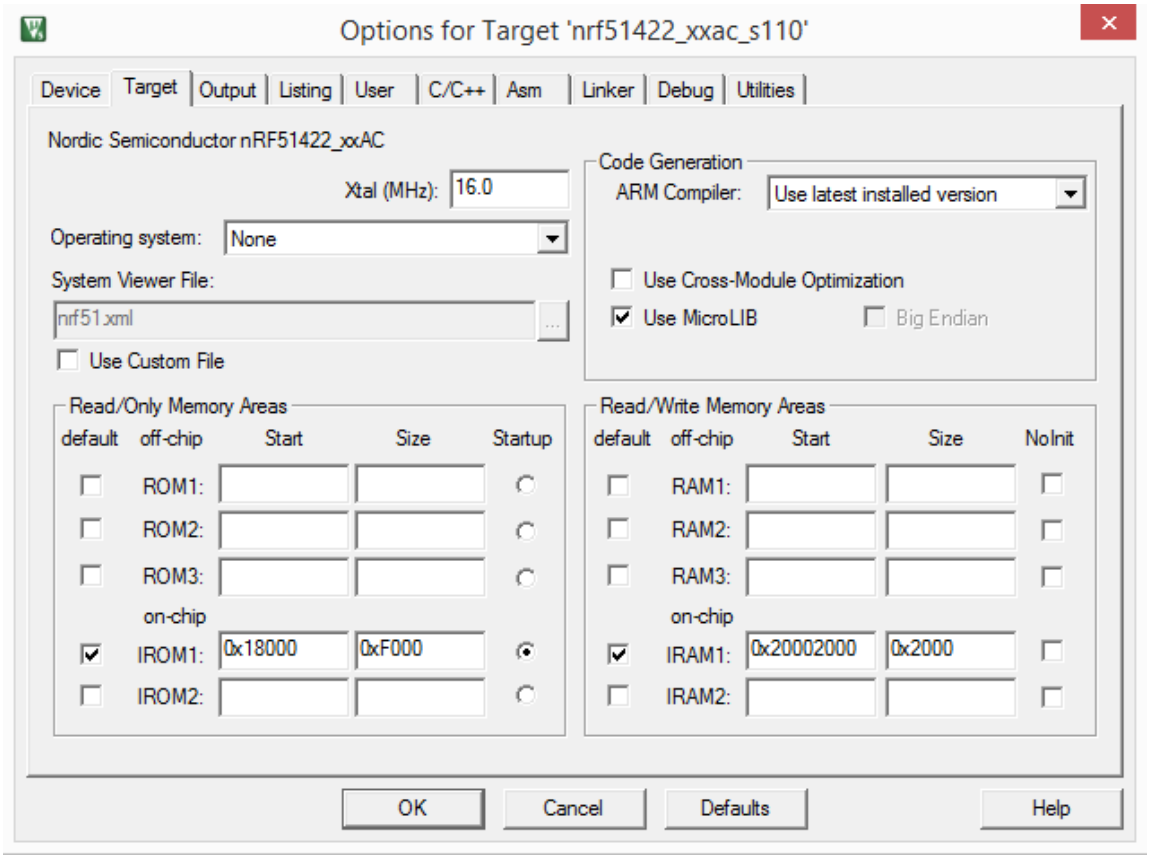

Figure 6- Keil Flash Tools Target window

Note: The memory locations in Figure 6 are based on the size of the S110 SoftDevice. If a different SoftDevice is used, these values will need to be adjusted accordingly.

4. Go to the 'C/C++' tab, and in the field marked 'Define' change "…BOARD_PCA10028" to "…BOARD_CUSTOM" as shown in Figure 7. This ensures the board support file downloaded in section 5.3.2 is used in place of the default development board)

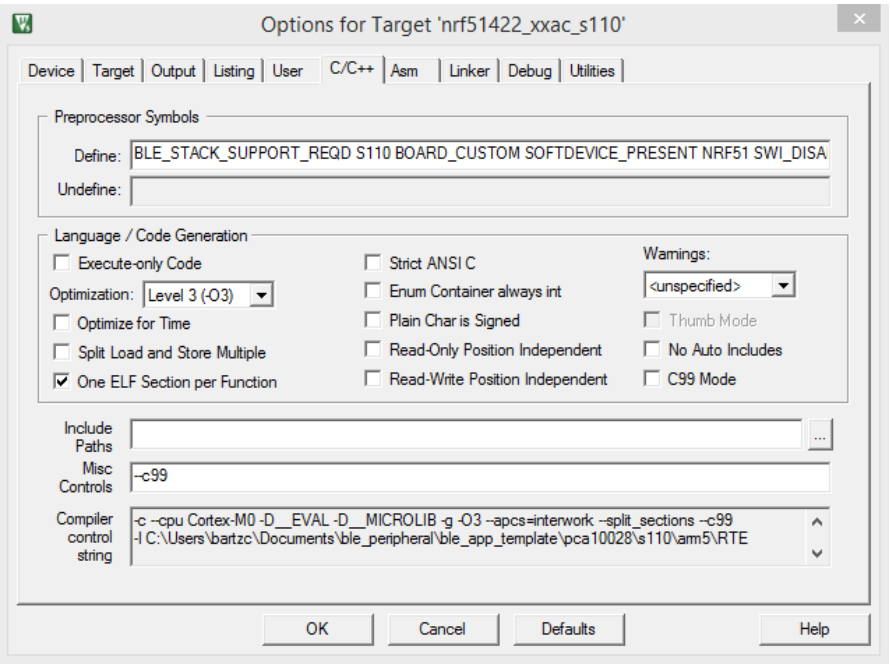

Figure 7- Keil Flash Tools Target window

5. In the 'Debug' tab of the same 'Configure Flash Tools' window, select 'J-LINK/J-TRACE Cortex' from the upper right hand drop down menu then click 'Settings.' The same serial number from the nRFgo program should be in the 'SN' field to acknowledge Keil recognizes the correct J-Link programmer

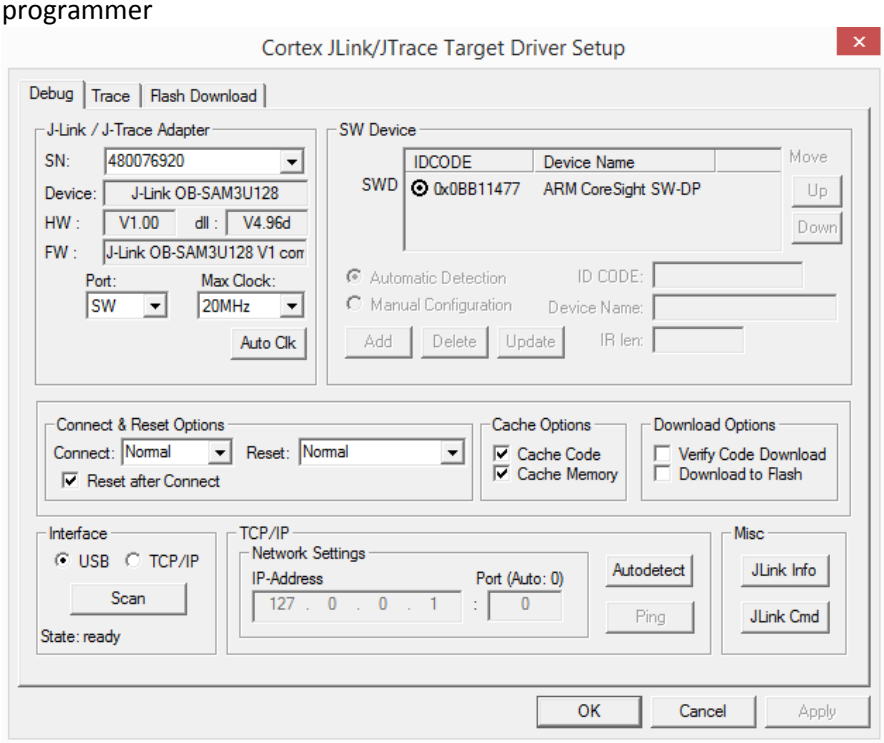

Figure 8- Keil Flash Tools Target window

6. Build example program (, 'F7') and download to the BMD-200 (, 'Flash'-> 'Download')

6. BMD-200 Pinout

Top View

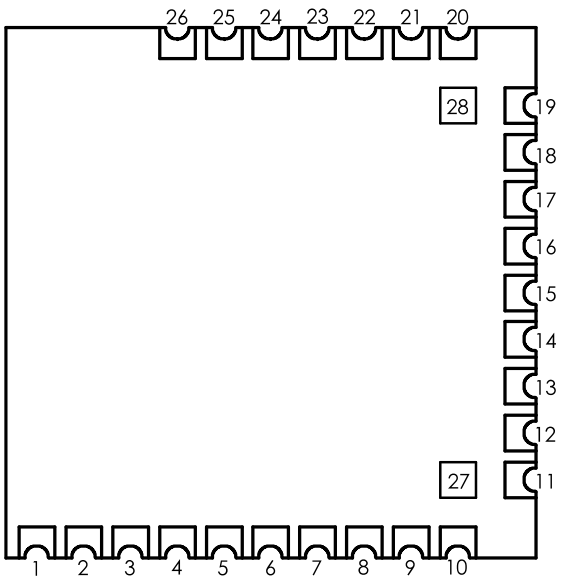

Figure 9- BMD-200 Pin-out

Pin description			
Name	*Pin*	*Direction*	*Description*
P0.24	5	In/Out	GPIO
P0.25	6	In/Out	GPIO
P0.26	8	In/Out	GPIO/AIN1/XTAL2 (32.768kHz)
P0.27	9	In/Out	GPIO/AIN0/XTAL1(32.768kHz)
P0.00	11	In/Out	GPIO/AREF0
P0.01	12	In/Out	GPIO/AIN2
P0.02	13	In/Out	GPIO/AIN3
P0.03	14	In/Out	GPIO/AIN4
P0.04	15	In/Out	GPIO/AIN5
P0.05	16	In/Out	GPIO/AIN6
P0.06	17	In/Out	GPIO/AIN7/AREF1
P0.08	20	In/Out	GPIO
P0.09	21	In/Out	GPIO
P0.10	22	In/Out	GPIO
P0.11	23	In/Out	GPIO
SWDIO	24	In/Out	SWD IO/ $\overline{\text{RESET}}$
SWCLK	25	In	SW Clock [1]
VCC	18	Power	+2.1 to +3.6VDC input [2]
GND	1, 2, 3, 4, 7, 10, 19, 26, (27, 28 opt.)	Power	Electrical Ground

Note 1: SWDCLK has an internal 12 kΩ pull-down resistor.
Note 2: An external capacitor for V_{CC} is not strictly required, however using a 1µF - 4.7µF ceramic capacitor is recommended.

7. Related Documents

Rigado Documents:

- BMD-200-DS: Module Datasheet
- BMD-200-AN-1: MAC Address Provisioning

Nordic Semiconductor Documents:

- nRF51822-PS: nRF51822 Product Specification
- nRF51 RM: nRF51 Series Reference Manual
- S110-SDS: nRF51822 S110 SoftDevice Specification
- S120-SDS: nRF51822 S120 SoftDevice Specification
- S130-SDS: nRF51822 S130 SoftDevice Specification

8. Document History

Revision	Date	Changes / Notes
0.1	1/30/15	Initial document draft
1.0	2/5/15	Initial release
1.1	8/5/15	Updated for Nordic SDK 8