Using the BL653 or BL653µ and Nordic SDK v17.0.2

BL653/ BL653µ

Application Note *v2.0*

1 Introduction

This application note is intended to help developers who want to use BL653 or BL653µ module to develop applications using freely available tools and the Nordic SDK. The core instructions for building, loading, and debugging firmware are available at the Nordic DevZone. This document provides additional instructions for using these tools on the Laird Connectivity BL653 series development kit and modules.

2 Requirements

The following are required to complete this step-by-step guide:

- Windows 7 (or newer) machine
- For BL653 development kits:
  - BL653 development kit (product code 453-00039-K1 or 453-00041-K1) or
  - USB A to micro-USB cable (provided with BL653 development kit)
- For BL653/BL653µ modules:
  - BL653 (product code 453-00039 or 453-00041) or BL653µ (product code 453-00059 or 453-00060) module with appropriate SWD connections.

Note: Tutorials provided by Nordic Semiconductor provide the most recent supported versions of software and tools used for development. The tutorials can be found at: https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/development-with-gcc-and-eclipse

© Copyright 2025 Ezurio
All Rights Reserved

3 Development Environment Setup

3.1 Nordic SDK

The Nordic Software development kit offers a rich development environment and examples for the BL653 module. It can easily be used to develop applications for the BL653 when using freely available software and when C language is preferred. The SDK offers a wide selection of drivers, libraries, and examples for the module and its peripherals.

To use Nordic's SDK for BL653 development, complete the following steps:

1. Download the Nordic SDK v17.0.2 zip file from https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF5-SDK. The file is located in the **DOWNLOADS** tab.
2. Once downloaded, extract and place the SDK in a suitable location on your machine, e.g.: *D:\Work\nRF5_SDK* (Figure 1).

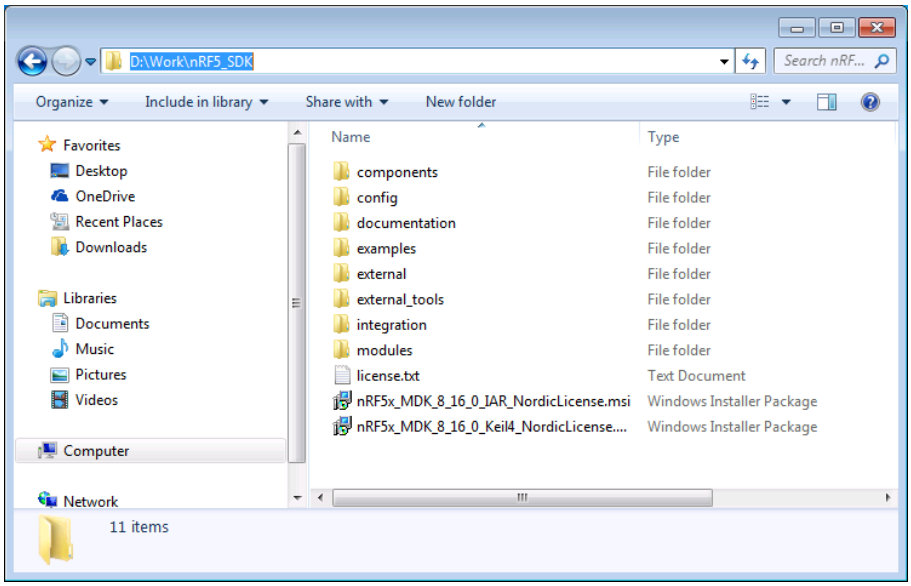

Figure 1: nRF5 SDK folder structure

Note: By default, the BL653 uses the nRF52833's on-chip RC oscillator as its clock source, as opposed to the nRF52833 development kits which use an external crystal. The difference in oscillator source should therefore be reflected in the SDK.

---

3. Open and edit the *nRF5_SDK\components\softdevice\common\nrf_sdh.c file* which is originally as shown in Figure 2.

```
    // Notify observers about SoftDevice enable request.
    if (sdh_request_observer_notify(NRF_SDH_EVT_ENABLE_REQUEST) == NRF_ERROR_BUSY)
    {
        // Enable process was stopped.
        return NRF_SUCCESS;
    }

    // Notify observers about starting SoftDevice enable process.
    sdh_state_observer_notify(NRF_SDH_EVT_STATE_ENABLE_PREPARE);

    nrf_clock_lf_cfg_t const clock_lf_cfg =
    {
        .source       = NRF_SDH_CLOCK_LF_SRC,
        .rc_ctiv      = NRF_SDH_CLOCK_LF_RC_CTIV,
        .rc_temp_ctiv = NRF_SDH_CLOCK_LF_RC_TEMP_CTIV,
        .accuracy     = NRF_SDH_CLOCK_LF_ACCURACY
    };
```

Figure 2: Original, unchanged nrf_sdh.c

© Copyright 2025 Ezurio
All Rights Reserved

4. Change the oscillator so that it uses the BL653 configuration as shown in Figure 3.

```
// Notify observers about SoftDevice enable request.
if (sdh_request_observer_notify(NRF_SDH_EVT_ENABLE_REQUEST) == NRF_ERROR_BUSY)
{
    // Enable process was stopped.
    return NRF_SUCCESS;
}

// Notify observers about starting SoftDevice enable process.
sdh_state_observer_notify(NRF_SDH_EVT_STATE_ENABLE_PREPARE);

nrf_clock_lf_cfg_t const clock_lf_cfg =
{
    .source       = NRF_CLOCK_LF_SRC_RC,
    .rc_ctiv      = 16,
    .rc_temp_ctiv = 2,
    .accuracy     = NRF_CLOCK_LF_ACCURACY_500_PPM
};
```

Figure 3: Changes to oscillator values in nrf_sdh.c

3.2 Eclipse

Follow Nordic's guide on installing the most up to date supported version of Eclipse.

3.3 Using GPIO BUTTON4 and SPI EEPROM

The GPIO for BUTTON4 is connected to P0.25 on the Nordic nRF52833 devboard (PCA10100), whilst the BL653 module has BUTTON4 connection to P0.22.

The pins connected to the SPI EEPROM on BL653-DVK are:

Pin Description	Pin Number
SPI SCK	41 (P1.09)
SPI MOSI	40 (P1.08)
SPI MISO	4 (P0.04)
SPI CS	23 (P0.23)

The rest of the pin map retains 1:1 mapping between PCA10100 and BL653-DVK.

3.4 BL653µ GPIO differences

The BL653µ has fewer GPIOs than BL653 due to being on a smaller package. The following GPIOs will not be useable when developing with BL653µ:

- P0.19
- P0.20
- P0.21
- P0.25
- P1.02
- P1.03
- P1.04
- P1.05
- P1.06
- P1.07

© Copyright 2025 Ezurio
All Rights Reserved

4 Reference

Further information relating to different utilities used in this app note can be accessed from the following links:

- BL653 Product Page - https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl653-series-bluetooth-51-802154-nfc-module
- Make - https://www.gnu.org/software/make/
- ARM Toolchain - https://launchpad.net/gcc-arm-embedded
- Eclipse - https://eclipse.org/org/
- Nordic Tutorial on using Eclipse and GCC - https://devzone.nordicsemi.com/tutorials/b/getting-started/posts/development-with-gcc-and-eclipse
- Nordic Blog on using Eclipse and GCC - https://devzone.nordicsemi.com/b/blog/posts/development-with-eclipse-and-gcc

5 Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	01 June 2020	Initial version.	Kieran Mackey	Jonathan Kaye
1.1	29 Jan 2020	Updated guide to latest SDK 17.0.2 and detailed GPIO changes for BL653µ. Added pin map for SPI EEPROM	Kieran Mackey	Jonathan Kaye
2.0	13 Mar 2025	Ezurio rebranding	Sue White	Dave Drogowski

Ezurio's products are subject to standard Terms & Conditions.

© Copyright 2025 Ezurio
All Rights Reserved