

Gowin GW1NRF Getting Started Guide

Testing GW1NRF4 Development Board Functionality

Step 1: Download and Install

Computer

- GW1NRF SocSdkSetup_1.0.exe
- GOWIN EDA for Windows
 - [Download - https://www.gowinsemi.com/en/support/download_eda/](https://www.gowinsemi.com/en/support/download_eda/)
 - [License - https://www.gowinsemi.com/en/support/license/](https://www.gowinsemi.com/en/support/license/)

Smartphone/Tablet

- Alpwise i-BLE App in iOS App Store or Google Play Store

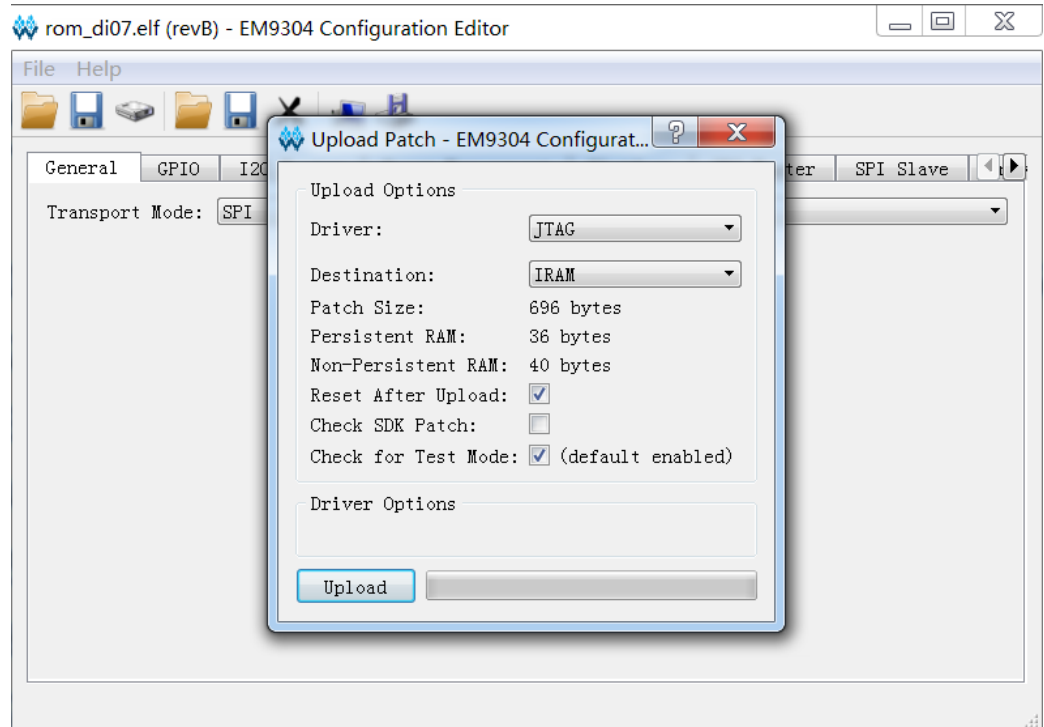
Step 2: Program and Test FPGA

1. Plug in DK-BLE-CEIT-ASSEM Development Board
2. Ensure drivers install and jumpers are set properly
3. Open GOWIN Programmer (Start → Gowin → Gowin Programmer)
4. Scan for devices
 - a). Click 'Scan Device' icon
 - b). Select 'GW1NRF-4B' and click 'OK'
5. Set FPGA bitstream to program
 - a). Double click the blank cell under 'FS File' for the GW1NRF device
 - b). Access Mode: Embedded Flash Mode
 - c). File name: (Navigate to GW1NRF4_FPGA_BLINK_1_21_2020.fs)
 - d). Click Save
6. Click 'Program/Configure' icon
7. Observe blue LED blinking from successful FPGA programming

Step 3: Program BLE and MCU

1. Start → Programs → Gowin Semiconductor → GW1NRF Config Editor
2. Select Driver: Jtag
3. Select 'Read from Device'
4. File → Upload Patch
5. Navigate to 'findme_MCU_GPIO7.emp'
6. Select Driver: Jtag
7. Destination: IRAM

8. Check boxes 'Reset After Upload' & 'Check for Test Mode'
9. Click Upload; Red LED should turn on after upload



Step 4: Test BLE and MCU

1. Open the Alpwis i-BLE App on your smartphone or tablet
2. Click 'Play Bluetooth low energy'
3. Click 'GW1NRF FindMe' from the list of Bluetooth devices
4. Click the 'Find Me' Profile
5. Push 'No Alert' → Red LED on the board should turn off
6. Push 'High Alert' → Red LED on the board should turn on



Creating an GW1NRF MCU and FPGA Project

Step 1: Setup Synopsys Metaware Lite

1. [Download and get license for Metaware Lite](https://www.synopsys.com/cgi-bin/arcmwtk_lite/reg1.cgi)
https://www.synopsys.com/cgi-bin/arcmwtk_lite/reg1.cgi
2. Install Metaware Lite
3. Start → Programs → Gowin Semiconductor → 'Setup Project Files'

Step 2: Setup Metaware Lite for GW1NRF4

1. Start → Programs → Synopsys Inc → MetaWare Lite IDE P-2019.09-1
2. Create an Eclipse workspace path and click 'OK'
3. Click 'Create a New Workspace'
4. Specify the following path: 'C:\ProgramData\GW1NRFsdk'
5. Click 'OK'; Metaware Lite will reopen
6. Run the scrip of the C:\ProgramData\GW1NRFsdk\sw\ide_project.bat'
7. File → Import → General, Existing Projects
8. Select Root Directory: 'C:\ProgramData\GW1NRFsdk\sw'
9. Click 'Finish'

Note!

Metaware compiler version 11.8 in Metaware 2020.03 introduced new optimizations impacting functionality of GW1NRF SDK. On temporary basis, the workaround proposed is to modify include file "platform.h" in directory <libs/em_hw_api/includes>. The external reference to **gPlatform_Config** declaration should be changed:

From:

```
extern const volatile Platform_Configuration_t gPlatform_Config;
```

To:

```
extern volatile Platform_Configuration_t gPlatform_Config;
```

Step 3: Compile MCU C Code

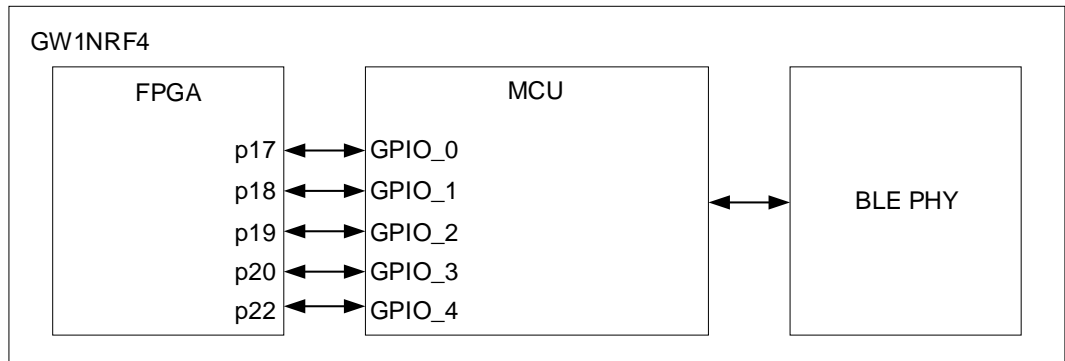
1. Navigate in Project Explorer to sw → projects → findme → findme.c
2. On line 153 change the alert LED to GPIO 0
`#define GPIO_FINDME_ALERT_LED (0)`
3. Project → Build All (if there is any error run 'clean')

Note!

There is some path issue with the BLESW_EM9304FOTA path in the GW1NRF SoC SDK Install.



MCU and FPGA Connectivity



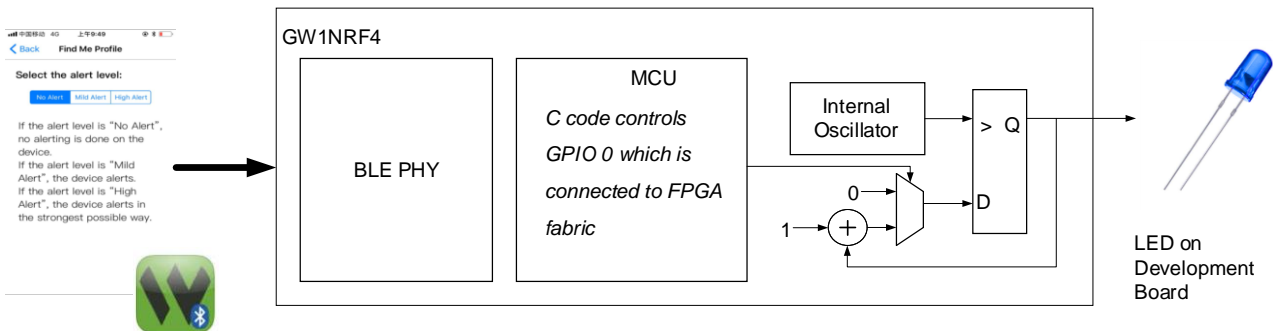
FPGA Constraint File IO Name	MCU IO Name
p17	GPIO 0
p18	GPIO 1
p19	GPIO 2
p20	GPIO 3
p22	GPIO 4

Step 4: Synthesize and Load FPGA Design

1. Open GOWIN EDA
2. Open the 'fpga_led_blink' project
3. Observe led.v
led.v creates a counter that controls an out 'led'. The counter runs when input 'enable' is active high.
4. Observe led.cst
 - a). 'enable' is set to connect to p17, which is connected to MCU GPIO 0 based on the table in the previous slide
 - b). 'led' is connected to package pin 8; this is connected to the blue led on the development board
5. Click the 'Process' tab in GOWIN EDA; right click on 'Place & Route' and select 'Rerun All'



Resulting Design



Now, we use Alpwis i-BLE App to control enable of a counter design in the FPGA.
The enable turns the LED off or allows it to blink with the counter inside the FPGA.

Load and test MCU and FPGA designs

1. Load FPGA file:
 - a). *\\GW1NRF_Getting_Started_v1\\fpga_led_blink\\imp\\pnr\\fpga_project.fs
 - b). Same procedure as “Testing GW1NRF4 Development Board Functionality → Step 2: Program and Test FPGA”
2. Load MCU elf file:
 - a). C:\\ProgramData\\GW1NRFsdk\\sw\\projects\\findme\\findme.elf
 - b). Same procedure as “Testing GW1NRF4 Development Board Functionality → Step 4: Test BLE and MCU Step 2: Program and Test FPGA”
3. Test control of the FPGA led via BLE
 - a). Run Alpwis i-BLE App
 - b). Same procedure as “Testing GW1NRF4 Development Board Functionality → Step 4: Test BLE and MCU Step 2: Program and Test FPGA”
 - c). led should be off or blinking based on the alert state in the Alpwis App

Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

Revision History

Date	Version	Description
04/03/2020	1.0E	Initial version published.
11/10/2020	1.1E	“Step 2: Setup Metaware Lite for GW1NRF4” in the section of “Creating an GW1NRF MCU and FPGA Project” updated.
03/19/2021	1.2E	The note information in the section of “Creating an GW1NRF MCU and FPGA Project” updated.

Copyright©2021 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

No part of this document may be reproduced or transmitted in any form or by any denotes, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI[®], LittleBee[®], Arora, and the GOWINSEMI logos are trademarks of GOWINSEMI and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders, as described at www.gowinsemi.com. GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.