

## MANUAL GUIDE

# Firmware debugging with Eclipse

Version 1.0



#### **Revision history**

Version	Date	Note	Contributor(s)	Approver
1.0	12 May 2021	Initial version	Nguyen Hoang Hoan	Nguyen Hoang Hoan



Copyright © 2019 I-SYST, all rights reserved.

3514, 1re Rue, Saint-Hubert, QC., Canada J3Y 8Y5

This document may not be reproduced in any form without, express written consent from I-SYST.



### Contents

1. Introduction	5
2. Debugging and Flashing Firmware with Eclipse IDE	5
2.1 Debugging Firmware with OpenODC	6
2.2 Flashing Firmware	9
2.3 Debugging and Flashing BleAdvertiser Firmware	10



#### 1. Introduction

This document shows step-by-step how to debug and flash firmware with Eclipse IDE and Iosonata which was installed in the Installation guide "Eclipse IDE in firmware development with IOsonata".

#### 2. Debugging and Flashing Firmware with Eclipse IDE



Connect IDK-BLYST-NANO to your computer.



#### 2.1 Debugging Firmware with OpenODC

We will start with Blinky Project as an example.

Select Blinky Project, Right click select Debug Configuration





Debug Configurations	-		×
Create, manage, and run configurations		Ŕ	ñ
<ul> <li>C/C++ Application</li> <li>C/C++ Application</li> <li>C/C++ Container Launcher</li> <li>C/C++ Container Launcher</li> <li>C/C++ Remote Application</li> <li>C/C++ Vint</li> <li>GDB Hardware Debugging</li> <li>GDB OpenOCD Debugging</li> <li>GDB OpenOCD Debugging</li> <li>GDB SPOCD Debugging</li> <li>GDB SEGGER J-Link Debugging</li> <li>GDB SEGGER J-Link Debugging</li> <li>GDB SEGGER J-Link Debugging</li> <li>GDB SEGGER J-Link Debugging</li> <li>BluePyro Debug</li> <li>Launch Group</li> </ul>	Name: Blinky Debug Main Startup Source Common Project: Blinky C/C++ Application: Debug/Blinky.elf Build (if required) before launching Build Configuration: Select Automatically C Enable auto build O Disable auto build O Disable auto build O Use workspace settings	rowse	
Filter matched 16 of 19 items	Revert	Apply	
?	Debug	Close	

Double click on GDB OpenOCD Debugging

In Main tab, at C/C++ Application click Search Project

Program Selection			×		
Choose a program to run:					
Binaries:					
Blinky.elf					
Qualifier:					
参 armle - /Blinky/Debug/Blinky.elf 参 armle - /Blinky/Release/Blinky.elf					
? ок		Cance	I		

Select Blinky.elf



Debug Configurations				- 🗆	×
Create, manage, and run configurations				Ŕ	ñ.
Image: Second	Name: Blinky Debu Main P Debu OpenOCD Setup Start OpenOC Executable: GDB port: Telnet port: Config options: Allocate cons GDB Client Setup Executable: Other options: Commands: Remote Target	Ig gger Startup Source Common D locally D:\i_syst\Development\openocd-0.10.0\bin\openocd.exe 3333 4444 -1 "interface/cmsis-dap.cfg" -1 "target/nf52\cfg" ole for OpenOCD Allocate console for the telnet connection D:\i_syst\Development\GNU Arm Embedded Toolchain\10 2020-q4-major\bin\arm-none-eabi-gdb.exe set mem inaccessible-by-default off	Browse	Variables	
Filter matched 17 of 20 items			Revert	Apply	
?			Debug	Close	:

In Debugger tab, set Config options

-f "interface/cmsis-dap.cfg"

-f "target/nrf52.cfg"

Browse OpenOCD executable file and ARM GDB executable file.

Click Debug



🔦 🎄 🔳 🎄 Debug	🗸 💽 Blinky Debug 💦 🗸 🔅 🔹 🐨 🐨 🖏 🐨 💊 🕶 🐘 🔛 🐂 🔯 🗣 📾 🖛 📾 🕬 🔊 🐼 🕫 🐼 🗸	🖸 • G • 🕸 • 🗘 • 🤮 • 🍓 • 🍅 📁 🛷 • 🍠 🗟 🗉 🗉	
월 - 월 - 번 라 슈 - 아 - 1 클		Q 😰 🖬 🐂	*
Project Explorer 🐹 📄 😫 🍸 🖇 📟	🗈 💦 blinky.c 🛛 📸 pulse_train.c 🕞 board.h 📧 0x0	😤 Outli 🛞 Buil 🛷 Sear 🎬 Disa 🏇 Debug 💥 🦉	
<ul> <li>Signification (In Edipox)</li> <li>Signification (In Edipox)</li> <li>Signification (Intersection)</li> <li>Signin (Intersection)</li> <li>Signification (Intersection)&lt;</li></ul>	<pre>100 // 101 102=int min() 103 103 104 105 105 105 105 105 105 105 105 105 105</pre>	<ul> <li>■ % i+ 8</li> <li>■ % i+ 8</li> <li>■ % i+ 8</li> <li>■ % i+ 100000000000000000000000000000000000</li></ul>	
	124 msDelay(1000); 125 IOPinSet(0, 28);	v	
	4	>	
	🖹 Problems 🧓 Tasks 📮 Console 🛛 🔲 Properties 🧬 Terminal 😭 Include Browser	📕 💥 🔆 📴 🚱 💭 💭 ଟ 💭 🕶 🗂 🕶 🗆	
	Binky Debug (DGB DeredOD Debugging)           (3) df (/4): exb00000000000000           (3) df (/4): exb0000000000000           (3) df (/4): exb0000000000000           (3) df (/4): exb0000000000000           (4) df (/4): exb00000000000000           (5) df (/6): exb0000000000000           (3) df (/6): exb0000000000000           (4) df (/6): exb0000000000000           (3) df (/6): exb0000000000000           (4) df (/6): exb00000000000000           (4) df (/6): exb000000000000000           (4) df (/6): exb00000000000000           (4) df (/6): exb000000000000000           (4) df (/6): exb000000000000000000000000000000000000		
<	<b>x</b>	· · · · · · · · · · · · · · · · · · ·	
	Writable Smart Inser	t 112 : 79 : 3693	-

After you start the debugger, it will stop at main(). Now you can debug the firmware by clicking the step button (F5, F6) to trace your source code line by line.

#### 2.2 Flashing Firmware

Click Run button to run the firmware on your device







#### 2.3 Debugging and Flashing BleAdvertiser Firmware

BleAdvertiser requires NRF SDK softdevice components so we have to flash softdevice first. Use IDAPnRFProg to flash NRF softdevice using IDAP-Link. Download here: <u>IDAP-Link/M - Browse</u>/<u>Windows at SourceForge.net</u>

Run IDAPnRFProg by following command line:

```
$.\IDAPnRFProg.exe D:\i_syst\external\nRF5_SDK\components\softdevice\s132\hex\s132_nrf52_7.2.0_softdevice.hex
```

```
IDAPnRFProg Ver. 1.8.201221
Copyright 2014-2020, I-SYST inc. All rights reserved
Found IDAP-Link/M - S/N : 4030318000216, Firmware : 1.6.201011
Target device found : 1
IDAP-Link/M-4030318000216 : nRF52832-CIAAB0, Rev.-1, HWID = 0x52832, DEVID =
0x5A6E40192D427DB2
IDAP-Link/M-4030318000216 : Flash size = 524288, Ram size = 65536
IDAP-Link/M-4030318000216 : Device address = 0x237DE024308F
IDAP-Link/M-4030318000216 : Erase all
```



IDAP-Link/M-4030318000216 : Blank checking... IDAP-Link/M-4030318000216 : Chip erased IDAP-Link/M-4030318000216 : Programming 153964 bytes... IDAP-Link/M-4030318000216 : Programmed 150 KB in 8.696 sec at rate 17.290 KB/s IDAP-Link/M-4030318000216 : Verifying... IDAP-Link/M-4030318000216 : Programming succeeded. Total programmed 1 nRF5x devices in 11.177 sec, 300 KB R/W transfered, rate = 26.840 KB/s

After flashing Bluetooth Low Energy Module in IDK-BLYST-NANO with IDAPnRFProg, we can now debug and flash firmware BleAdvertiser on IDK-BLYST-NANO

Cecipse-workspace - BleAdvertiser/srz/BleAdverti	×
	h **
By Detection 1:         UN (Note: Compute State And	
invalid arguments' Writable Smart Insert 105: 14: 3405	5