EVBUM2589

Bluetooth® Low Energy IoT Development Kit (B-IDK) Getting Started Guide

ON Semiconductor®

www.onsemi.com

EVAL BOARD USER’S MANUAL

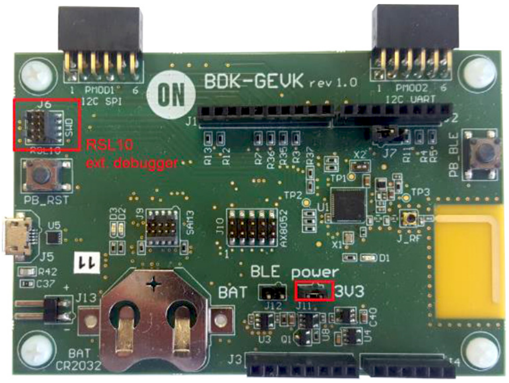


Figure 1. Board Photo

INTRODUCTION

This document helps you get started with the Bluetooth Low Energy IoT Development Kit (B−IDK). The B−IDK is a comprehensive node−to−cloud and a modular IoT platform that allows development of various BLE based use cases. Along with the hardware and software, the B−IDK includes a mobile app to interact with sensors and actuators.

The B−IDK features RSL10, Industry’s lowest power Bluetooth 5 SoC and comprises of a baseboard (BDK−GEVK) and several sensor and actuator daughter cards. For a complete listing of available daughter cards, please visit https://www.onsemi.com/B−IDK. The daughter cards connect to the baseboard, via the two PMOD connectors and/or the Arduino connector to enable various use cases.

Scope

This document covers the hardware setup, software architecture, B−IDK documentation and provides instructions on downloading firmware to the board. The details regarding the mobile app and cloud connectivity are not covered in this document.

HARDWARE

- BDK−GEVK − B−IDK Baseboard
- Daughter Cards – Optional
- BDK−DCDC−GEVB – Power Shield For Use With Higher Power Daughter Cards – Optional

Default Configuration

The BDK−GEVK is shipped with the following jumper configuration. As the board supports OBD, there is no need for an external debugger. In case an external debugger is used, connect it to SWD header, J6.

Powering the Board

Multiple options are available to power the BDK−GEVK.

- USB
- Coin Cell (CR2032)
- External AC/DC Adapter plus power shield (BDK−DCDC−GEVB)
- External Supply

When higher power daughter cards (listed below) are attached to the baseboard, external supply either using the power shield or direct is required.

Higher Power Daughter Cards

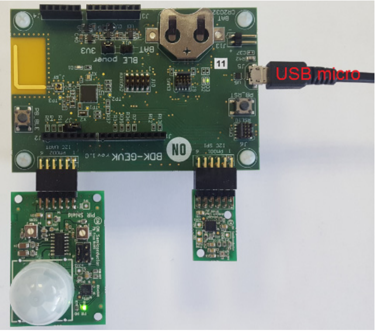
- D−LED−B−GEVK Dual LED Ballast
- D−STPR−GEVK Dual Stepper Motor Driver
- BLDC−GEVK BLDC Motor Driver

© Semiconductor Components Industries, LLC, 2015

September, 2019 − Rev. 4

USB

The B−IDK can be powered via the USB port when the use case doesn’t need any higher power daughter cards. An example configuration with the baseboard and a couple of sensor boards is shown below.



Coin Cell

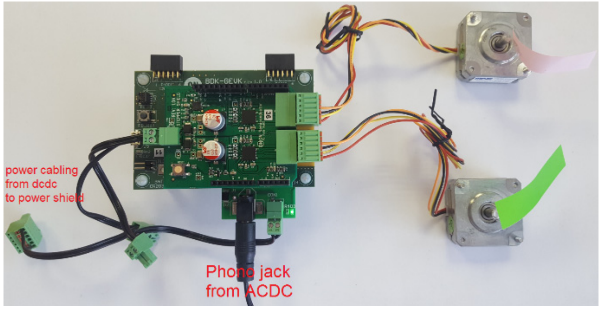
Once the firmware is flashed onto the baseboard, a coin cell (CR2032) may be used to power the system. Similar to USB based power supply, this method of powering is for use cases that don’t utilize the higher power daughter cards. The jumper configuration must match the below table to allow for various power modes.

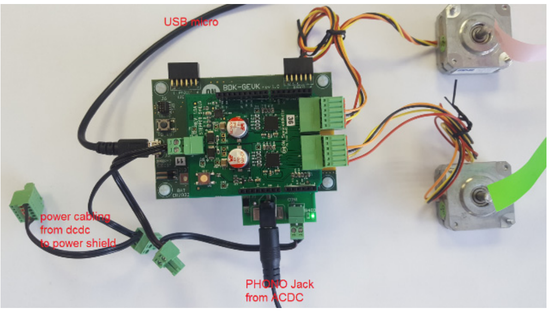
Table 1. JUMPERS

J11	J12	Usage
IN	X	Programming and Power over USB
X	IN	After programming. Only RSL10 is powered.
IN	IN	After programming. Both RSL 10 and OBD Microcontroller are powered

External AC/DC Adapter Plus Power Shield (BDK−DCDC−GEVB)

For use cases that utilize higher power daughter cards, an external AC/DC power supply (Ex: SMI24−12−V−P6) plus the power shield (BDK−DCDC−GEVB) are needed to power the system. While the 3.3 V supply to the baseboard is provided by the power shield via the Arduino connector, power cables (Green connector) are required between BDK−DCDC−GEVB and the higher power daughter card. For firmware flashing and debugging, the USB cable may be plugged in simultaneously with this mode as shown below.





External Supply

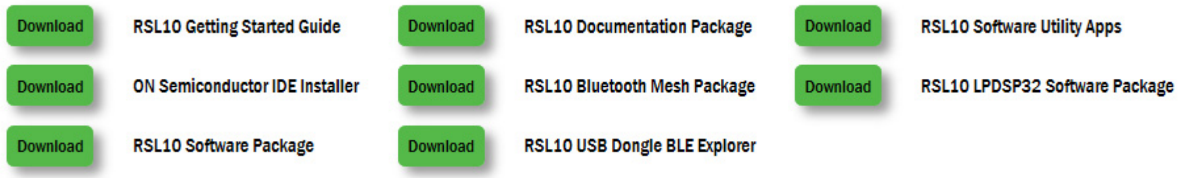
The B−IDK can be powered by an external supply via J13. In this mode, the battery cannot be installed. Jumpers J11 and J12 must be installed.

SOFTWARE

The B−IDK software allows for rapid development of various use cases. This section details the prerequisites and detailed steps in downloading firmware onto the baseboard.

Prerequisites

1. Install 64−bit version of Java from https://www.java.com/en/download/
2. Install J−Link Version 6.32f or later from https://www.segger.com/downloads/jlink (select J−Link software and documentation pack)
3. Download and install“On Semiconductor IDE Installer” from https://www.onsemi.com/PowerSolutions/product.do?id=RSL10
   a. Download the RSL10 SDK Getting Started Guide and RSL10 CMSIS pack under “RSL10 Software Package” from the above site. All of these are highlighted in the picture below. Save the CMSIS pack in a folder, for example, C:\cmsis_packs



Download	RSL10 Getting Started Guide	Download	RSL10 Documentation Package	Download	RSL10 Software Utility Apps
Download	ON Semiconductor IDE Installer	Download	RSL10 Bluetooth Mesh Package	Download	RSL10 LPDSP32 Software Package
Download	RSL10 Software Package	Download	RSL10 USB Dongle BLE Explorer		

4. Download the B−IDK CMSIS pack from https://www.onsemi.com/B−IDK and save it in the same folder as the RSL10 CMSIS pack (see 3.a above)
5. CMSIS pack at item 4. is dependent on ARM CMSIS pack as well. Please install ARM CMSIS pack 5.5.1 or higher after download from: https://github.com/ARM−software/CMSIS_5/releases
6. CMSIS pack at item 4. is also dependent on ARM CMSIS – FreeRTOS version 10.2.0 or higher for users exposed to design the code under FreeRTOS with RSL10: https://github.com/ARM−software/CMSIS−FreeRTOS/releases

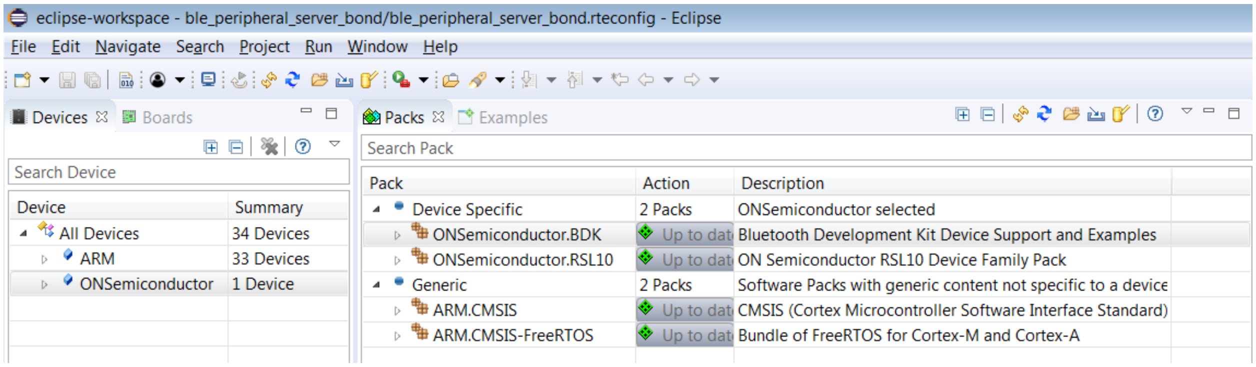
The next section provides details on importing the downloaded CMSIS packs into the SDK.

Importing CMSIS Packages

1. Launch the RSL10 SDK ON Semiconductor IDE

NOTE: Please import RSL10 CMSIS pack first as the B−IDK CMSIS pack (step 4 in the Prerequisites section) depends on the RSL10.

2. Refer to Chapter 3 of RSL10 SDK Getting Started Guide (step 3.a) for step−by−step instructions on importing the CMSIS packs.
3. Once all packs are successfully imported, they can be viewed in the CMSIS pack manager perspective as shown below.



eclipse-workspace - ble_peripheral_server_bond/ble_peripheral_server_bond.rteconfig - Eclipse
File Edit Navigate Search Project Run Window Help
Devices
Boards
Packs
Examples

Search Device

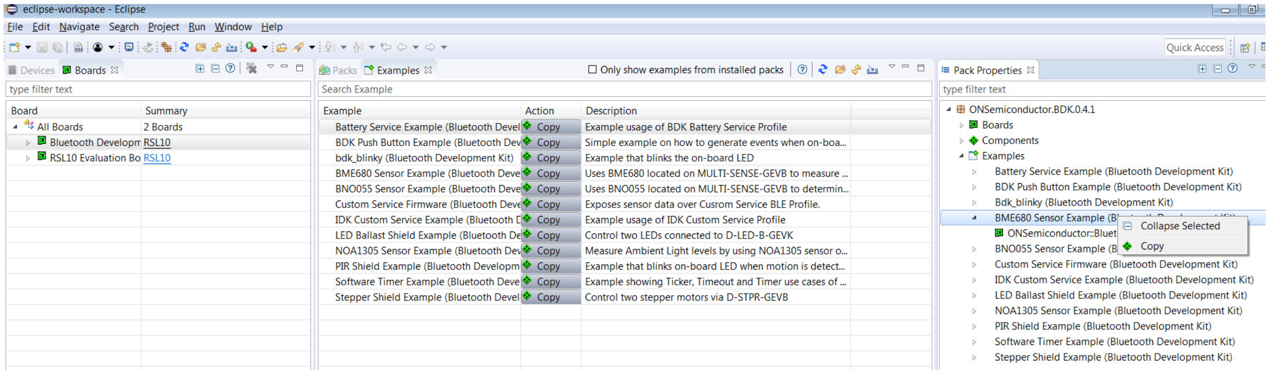
Device	Summary
All Devices	34 Devices
ARM	33 Devices
ONSemiconductor	1 Device

Search Pack

Pack	Action	Description
Device Specific	2 Packs	ONSemiconductor selected
ONSemiconductor.BDK	Up to dat	Bluetooth Development Kit Device Support and Examples
ONSemiconductor.RSL10	Up to dat	ON Semiconductor RSL10 Device Family Pack
Generic	2 Packs	Software Packs with generic content not specific to a device
ARM.CMSIS	Up to dat	CMSIS (Cortex Microcontroller Software Interface Standard)
ARM.CMSIS-FreeRTOS	Up to dat	Bundle of FreeRTOS for Cortex-M and Cortex-A

Compiling and Flashing

1. Choose an example (for example, pr_shield_example) to flash by copying it to the workspace.

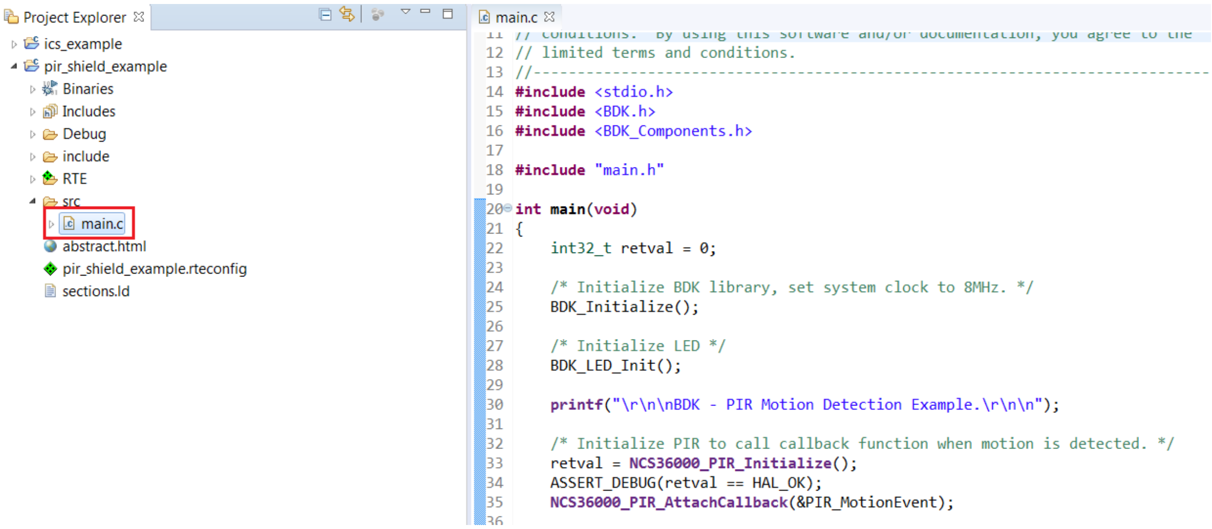

eclipse-workspace - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Devices
Boards
Packs
Examples
☐ Only show examples from installed packs
Pack Properties
type filter text
Search Example
Collapse Selected
Copy

Board	Summary
All Boards	2 Boards
Bluetooth Developm	RSL10
RSL10 Evaluation Bo	RSL10

Example	Action	Description
Battery Service Example (Bluetooth Devel	Copy	Example usage of BDK Battery Service Profile
BDK Push Button Example (Bluetooth Dev	Copy	Simple example on how to generate events when on-boa...
bdk_blinky (Bluetooth Development Kit)	Copy	Example that blinks the on-board LED
BME680 Sensor Example (Bluetooth Deve	Copy	Uses BME680 located on MULTI-SENSE-GEVB to measure ...
BNO055 Sensor Example (Bluetooth Deve	Copy	Uses BNO055 located on MULTI-SENSE-GEVB to determin...
Custom Service Firmware (Bluetooth Deve	Copy	Exposes sensor data over Cusrom Service BLE Profile.
IDK Custom Service Example (Bluetooth D	Copy	Example usage of IDK Custom Service Profile
LED Ballast Shield Example (Bluetooth De	Copy	Control two LEDs connected to D-LED-B-GEVK
NOA1305 Sensor Example (Bluetooth Dev	Copy	Measure Ambient Light levels by using NOA1305 sensor o...
PIR Shield Example (Bluetooth Developm	Copy	Example that blinks on-board LED when motion is detect...
Software Timer Example (Bluetooth Deve	Copy	Example showing Ticker, Timeout and Timer use cases of ...
Stepper Shield Example (Bluetooth Devel	Copy	Control two stepper motors via D-STPR-GEVB

```
ONSemiconductor.BDK.0.4.1
  Boards
  Components
  Examples
    Battery Service Example (Bluetooth Development Kit)
    BDK Push Button Example (Bluetooth Development Kit)
    Bdk_blinky (Bluetooth Development Kit)
    BME680 Sensor Example (B
      ONSemiconductor::Bluet
    BNO055 Sensor Example (B
    Custom Service Firmware (Bluetooth Development Kit)
    IDK Custom Service Example (Bluetooth Development Kit)
    LED Ballast Shield Example (Bluetooth Development Kit)
    NOA1305 Sensor Example (Bluetooth Development Kit)
    PIR Shield Example (Bluetooth Development Kit)
    Software Timer Example (Bluetooth Development Kit)
    Stepper Shield Example (Bluetooth Development Kit)
```

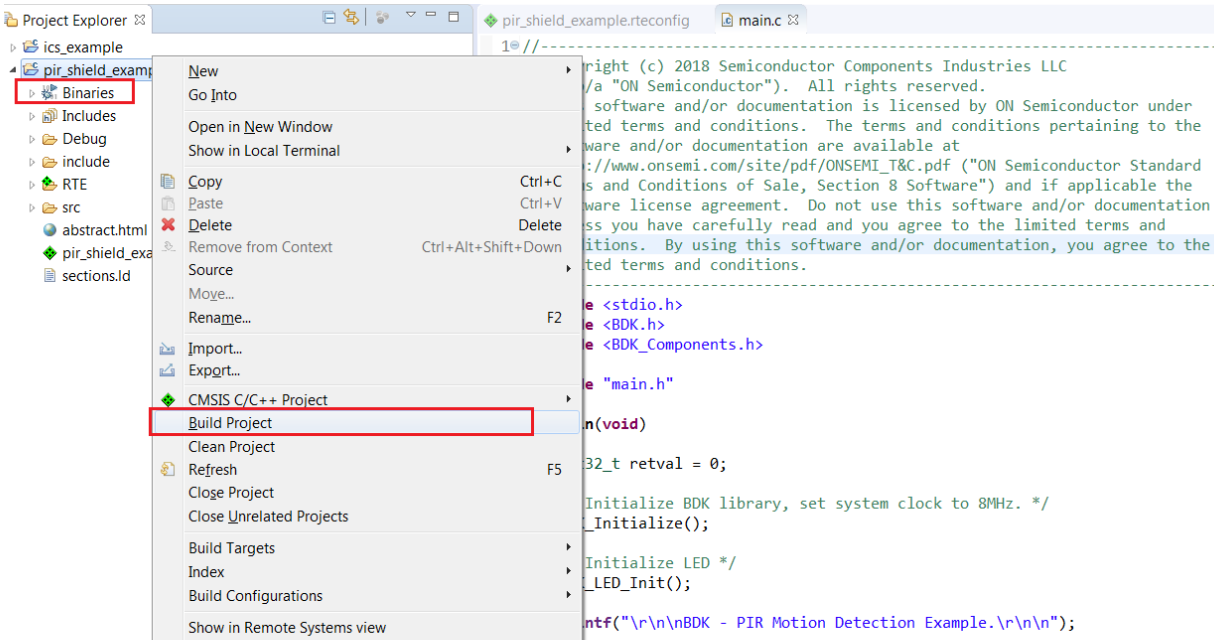
NOTE: Once the example is copied, it can be viewed under Project Explorer. All source files including main are located in the src folder.



```
ics_example
pir_shield_example
  Binaries
  Includes
  Debug
  include
  RTE
  src
    main.c
  abstract.html
  pir_shield_example.rteconfig
  sections.ld
```

```
// conditions.  By using this software and/or documentation, you agree to the
// limited terms and conditions.
//-----------------------------------------------------------------------------
#include <stdio.h>
#include <BDK.h>
#include <BDK_Components.h>

#include "main.h"

int main(void)
{
    int32_t retval = 0;

    /* Initialize BDK library, set system clock to 8MHz. */
    BDK_Initialize();

    /* Initialize LED */
    BDK_LED_Init();

    printf("\r\n\nBDK - PIR Motion Detection Example.\r\n\n");

    /* Initialize PIR to call callback function when motion is detected. */
    retval = NCS36000_PIR_Initialize();
    ASSERT_DEBUG(retval == HAL_OK);
    NCS36000_PIR_AttachCallback(&PIR_MotionEvent);
```

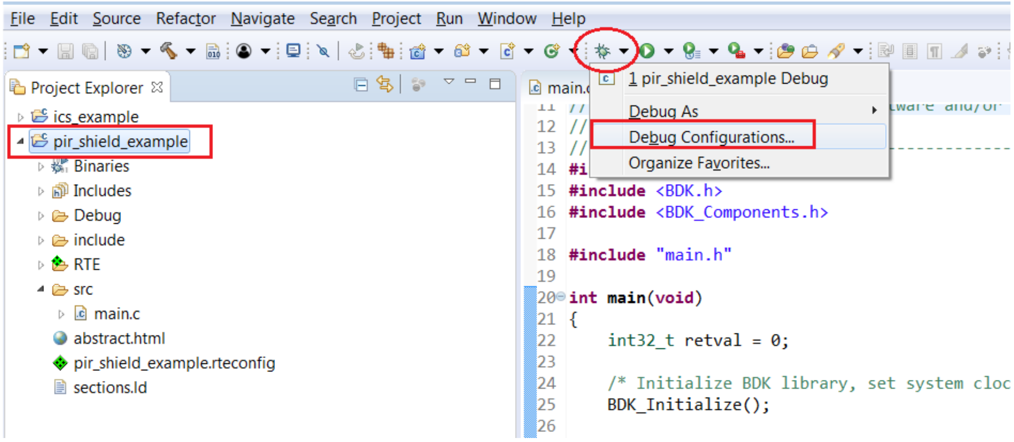
2. Right click and build the project. This creates binaries to be flashed to BDK−GEVK.



```
//-----------------------------------------------------------------------------
right (c) 2018 Semiconductor Components Industries LLC
/a "ON Semiconductor").  All rights reserved.
 software and/or documentation is licensed by ON Semiconductor under
ted terms and conditions.  The terms and conditions pertaining to the
ware and/or documentation are available at
://www.onsemi.com/site/pdf/ONSEMI_T&C.pdf ("ON Semiconductor Standard
s and Conditions of Sale, Section 8 Software") and if applicable the
ware license agreement.  Do not use this software and/or documentation
ss you have carefully read and you agree to the limited terms and
itions.  By using this software and/or documentation, you agree to the
ted terms and conditions.
-----------------------------------------------------------------------------
e <stdio.h>
e <BDK.h>
e <BDK_Components.h>

e "main.h"

n(void)

32_t retval = 0;

Initialize BDK library, set system clock to 8MHz. */
_Initialize();

Initialize LED */
_LED_Init();

ntf("\r\n\nBDK - PIR Motion Detection Example.\r\n\n");
```

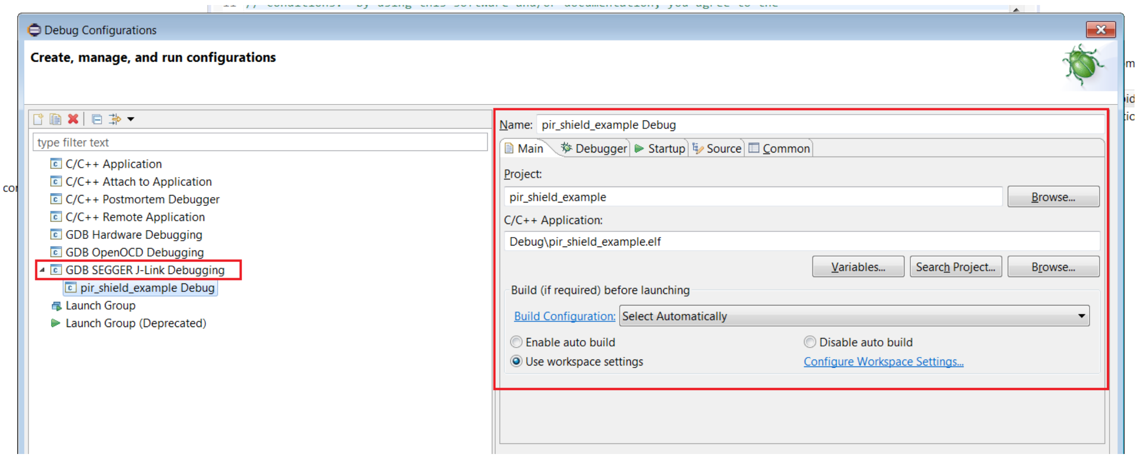
NOTE: If the binaries are not seen, press F5 (refresh).

3. Once the build is done, the code is ready to be flashed to the BDK−GEVK. Select the project (pir_shield_example), and go to the debug configurations as shown below.



```
//                                      ware and/or
//
//                                      -------------
#i
#include <BDK.h>
#include <BDK_Components.h>

#include "main.h"

int main(void)
{
    int32_t retval = 0;

    /* Initialize BDK library, set system cloc
    BDK_Initialize();

```

4. Double click **GDB Segger J−Link Debugging** to create the debug configuration for the selected example.



Debug Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- GDB Hardware Debugging
- GDB OpenOCD Debugging
- GDB SEGGER J-Link Debugging
  - pir_shield_example Debug
- Launch Group
- Launch Group (Deprecated)

Name: pir_shield_example Debug

Main | Debugger | Startup | Source | Common

Project: pir_shield_example [Browse...]

C/C++ Application: Debug\pir_shield_example.elf [Variables...] [Search Project...] [Browse...]

Build (if required) before launching

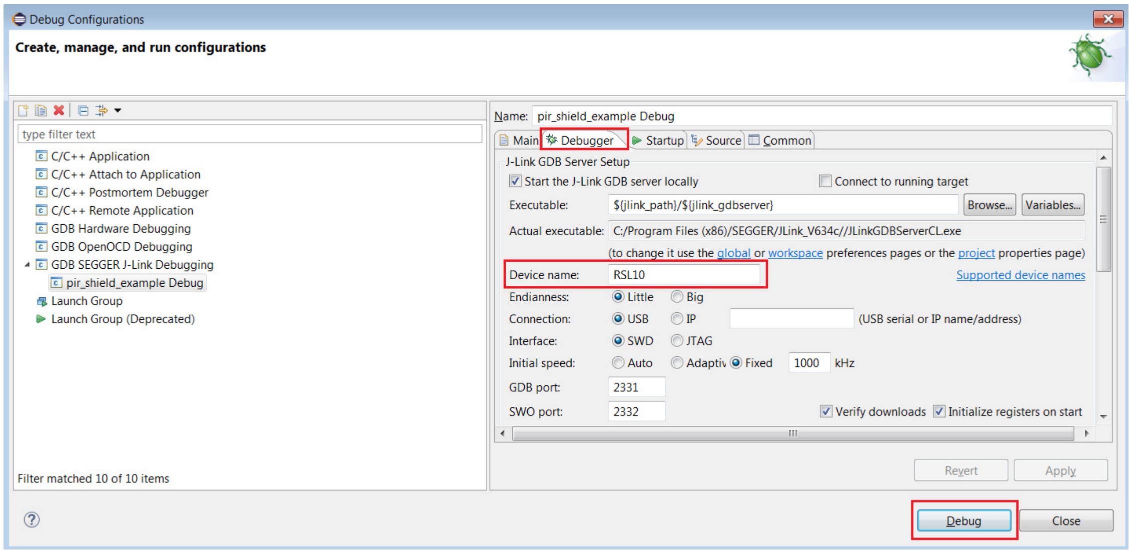
Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

NOTE: The debug configuration for the selected example is automatically saved and there’s no need to re−create it.

5. On the **Debugger** tab, set RSL10 as the device name. Click **Debug** to launch the code.



Debug Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- GDB Hardware Debugging
- GDB OpenOCD Debugging
- GDB SEGGER J-Link Debugging
  - pir_shield_example Debug
- Launch Group
- Launch Group (Deprecated)

Filter matched 10 of 10 items

Name: pir_shield_example Debug

Main | Debugger | Startup | Source | Common

J-Link GDB Server Setup

☑ Start the J-Link GDB server locally ☐ Connect to running target

Executable: ${jlink_path}/${jlink_gdbserver} [Browse...] [Variables...]

Actual executable: C:/Program Files (x86)/SEGGER/JLink_V634c//JLinkGDBServerCL.exe

(to change it use the global or workspace preferences pages or the project properties page)

Device name: RSL10 Supported device names

Endianness: ● Little ○ Big

Connection: ● USB ○ IP (USB serial or IP name/address)

Interface: ● SWD ○ JTAG

Initial speed: ○ Auto ○ Adaptiv ● Fixed 1000 kHz

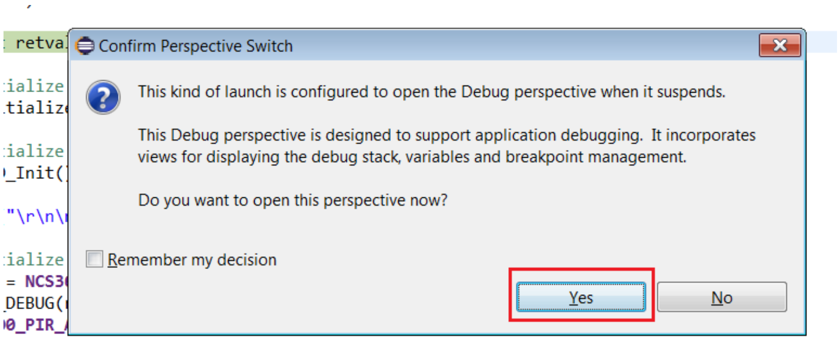
GDB port: 2331

SWO port: 2332 ☑ Verify downloads ☑ Initialize registers on start

[Revert] [Apply]

[Debug] [Close]

6. For application debugging, confirm perspective switch by clicking Yes.



Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

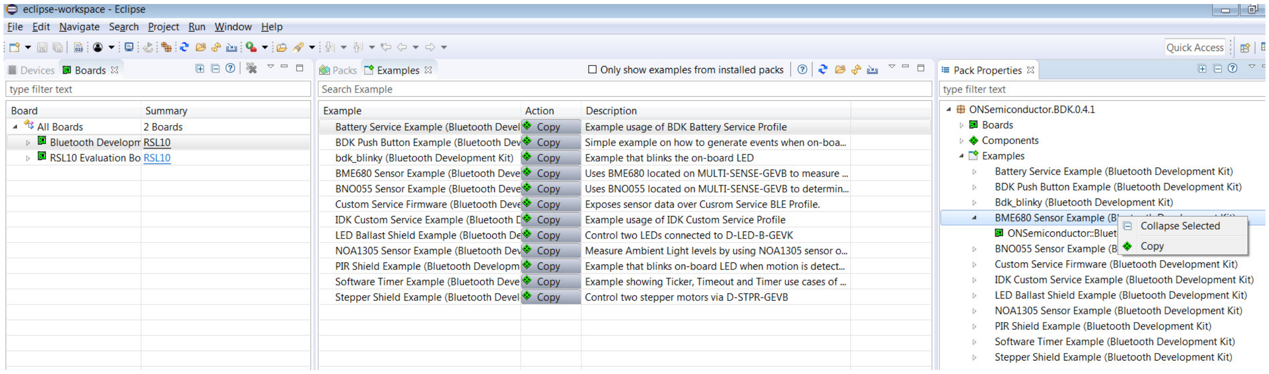
☐ Remember my decision

[Yes] [No]

7. The debug session is now launched. Click Resume (F8) to start the target CPU.

Compiling and Flashing

1. Choose an example (for example, pr_shield_example) to flash by copying it to the workspace.

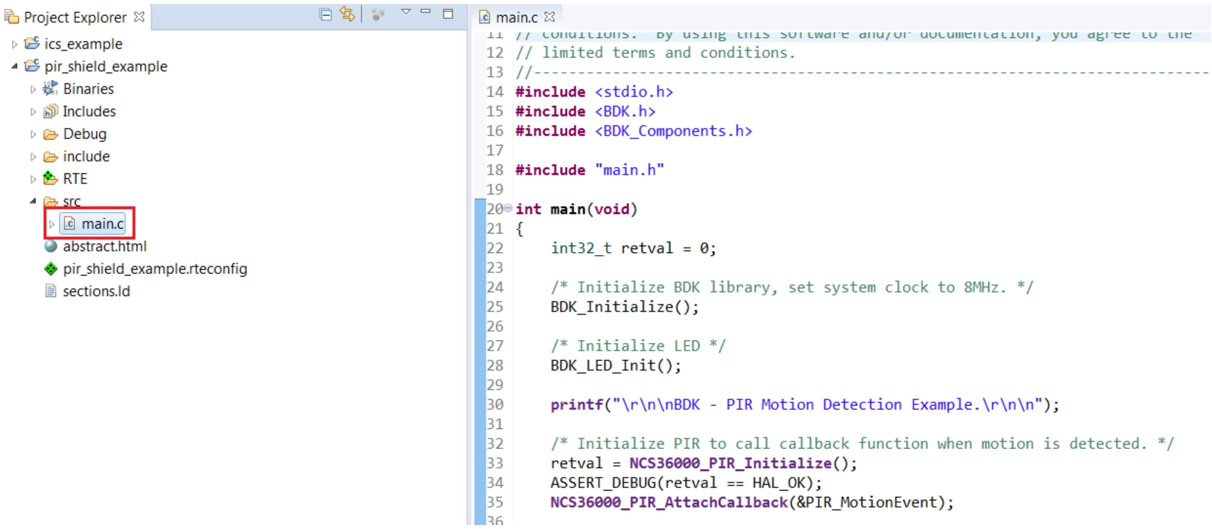


Board	Summary
All Boards	2 Boards
Bluetooth Developm	RSL10
RSL10 Evaluation Bo	RSL10

Example	Action	Description
Battery Service Example (Bluetooth Devel	Copy	Example usage of BDK Battery Service Profile
BDK Push Button Example (Bluetooth Dev	Copy	Simple example on how to generate events when on-boa...
bdk_blinky (Bluetooth Development Kit)	Copy	Example that blinks the on-board LED
BME680 Sensor Example (Bluetooth Deve	Copy	Uses BME680 located on MULTI-SENSE-GEVB to measure ...
BNO055 Sensor Example (Bluetooth Deve	Copy	Uses BNO055 located on MULTI-SENSE-GEVB to determin...
Custom Service Firmware (Bluetooth Deve	Copy	Exposes sensor data over Cusrom Service BLE Profile.
IDK Custom Service Example (Bluetooth D	Copy	Example usage of IDK Custom Service Profile
LED Ballast Shield Example (Bluetooth De	Copy	Control two LEDs connected to D-LED-B-GEVK
NOA1305 Sensor Example (Bluetooth Dev	Copy	Measure Ambient Light levels by using NOA1305 sensor o...
PIR Shield Example (Bluetooth Developm	Copy	Example that blinks on-board LED when motion is detect...
Software Timer Example (Bluetooth Deve	Copy	Example showing Ticker, Timeout and Timer use cases of ...
Stepper Shield Example (Bluetooth Devel	Copy	Control two stepper motors via D-STPR-GEVB

ONSemiconductor.BDK.0.4.1
- Boards
- Components
- Examples
  - Battery Service Example (Bluetooth Development Kit)
  - BDK Push Button Example (Bluetooth Development Kit)
  - Bdk_blinky (Bluetooth Development Kit)
  - BME680 Sensor Example (B
    - ONSemiconductor::Bluet
  - BNO055 Sensor Example (B
  - Custom Service Firmware (Bluetooth Development Kit)
  - IDK Custom Service Example (Bluetooth Development Kit)
  - LED Ballast Shield Example (Bluetooth Development Kit)
  - NOA1305 Sensor Example (Bluetooth Development Kit)
  - PIR Shield Example (Bluetooth Development Kit)
  - Software Timer Example (Bluetooth Development Kit)
  - Stepper Shield Example (Bluetooth Development Kit)

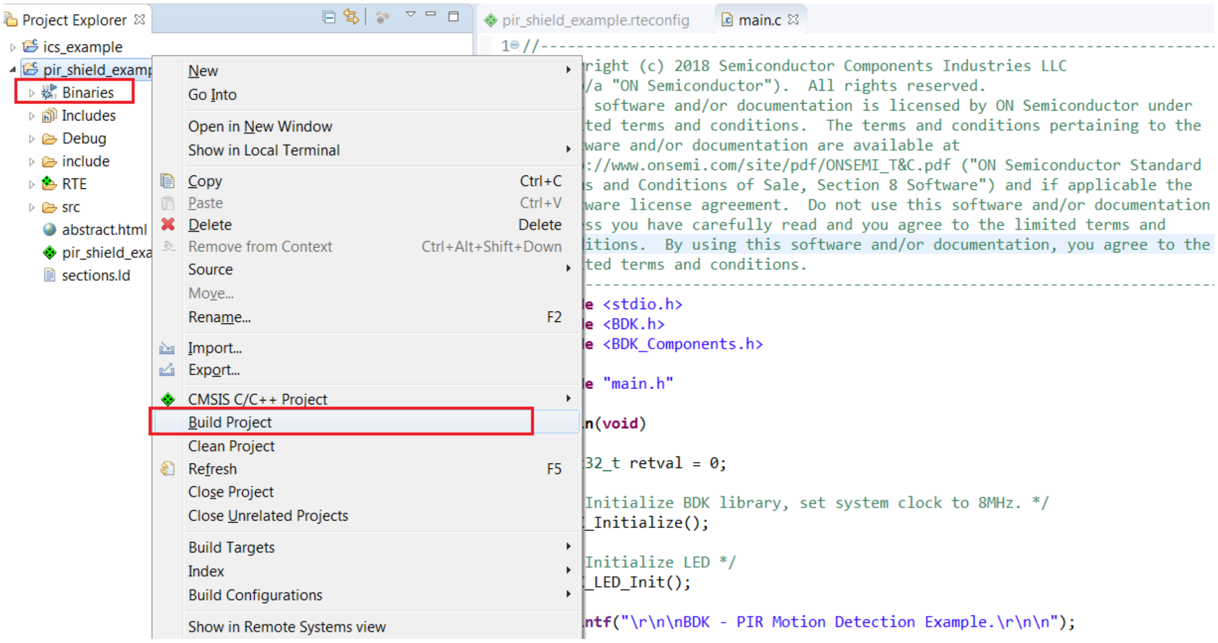
NOTE: Once the example is copied, it can be viewed under Project Explorer. All source files including main are located in the src folder.



- ics_example
- pir_shield_example
  - Binaries
  - Includes
  - Debug
  - include
  - RTE
  - src
    - main.c
  - abstract.html
  - pir_shield_example.rteconfig
  - sections.ld

```
// conditions.  By using this software and/or documentation, you agree to the
// limited terms and conditions.
//-----------------------------------------------------------------------------
#include <stdio.h>
#include <BDK.h>
#include <BDK_Components.h>

#include "main.h"

int main(void)
{
    int32_t retval = 0;

    /* Initialize BDK library, set system clock to 8MHz. */
    BDK_Initialize();

    /* Initialize LED */
    BDK_LED_Init();

    printf("\r\n\nBDK - PIR Motion Detection Example.\r\n\n");

    /* Initialize PIR to call callback function when motion is detected. */
    retval = NCS36000_PIR_Initialize();
    ASSERT_DEBUG(retval == HAL_OK);
    NCS36000_PIR_AttachCallback(&PIR_MotionEvent);
```

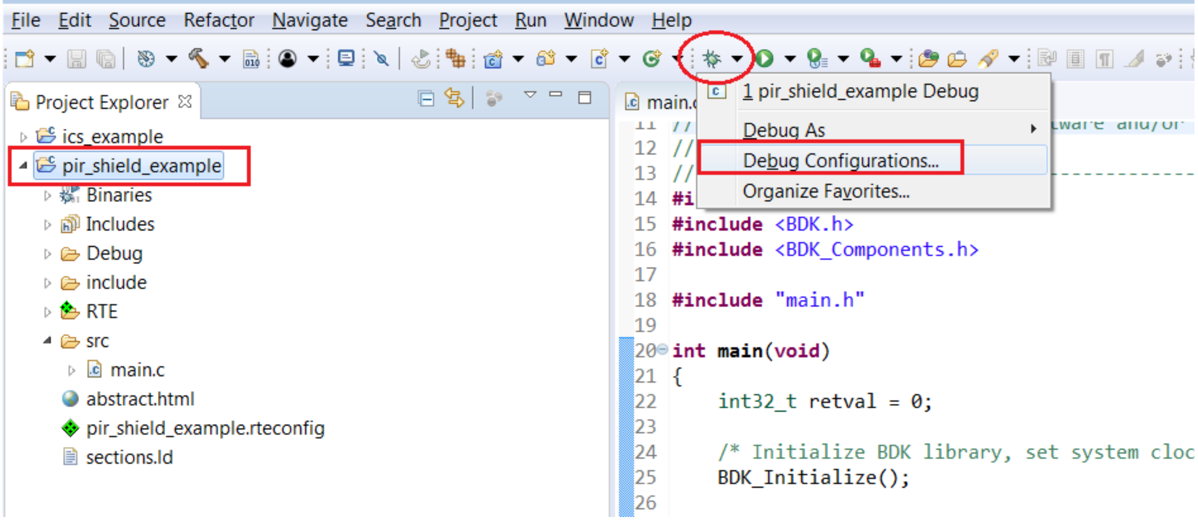
2. Right click and build the project. This creates binaries to be flashed to BDK−GEVK.



```
//-----------------------------------------------------------------------
right (c) 2018 Semiconductor Components Industries LLC
/a "ON Semiconductor").  All rights reserved.
 software and/or documentation is licensed by ON Semiconductor under
ted terms and conditions.  The terms and conditions pertaining to the
ware and/or documentation are available at
://www.onsemi.com/site/pdf/ONSEMI_T&C.pdf ("ON Semiconductor Standard
s and Conditions of Sale, Section 8 Software") and if applicable the
ware license agreement.  Do not use this software and/or documentation
ss you have carefully read and you agree to the limited terms and
itions.  By using this software and/or documentation, you agree to the
ted terms and conditions.
-----------------------------------------------------------------------
e <stdio.h>
e <BDK.h>
e <BDK_Components.h>

e "main.h"

n(void)

32_t retval = 0;

Initialize BDK library, set system clock to 8MHz. */
_Initialize();

Initialize LED */
_LED_Init();

ntf("\r\n\nBDK - PIR Motion Detection Example.\r\n\n");
```

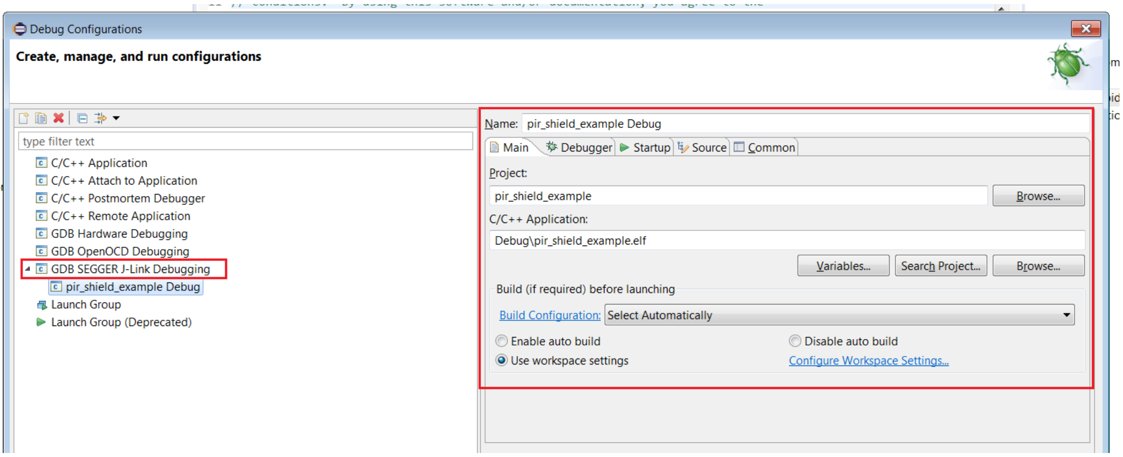
NOTE: If the binaries are not seen, press F5 (refresh).

3. Once the build is done, the code is ready to be flashed to the BDK−GEVK. Select the project (pir_shield_example), and go to debug configurations as shown below.



```
//                                                      ware and/or
//
//   ------------------------------------------------------------
#i
#include <BDK.h>
#include <BDK_Components.h>

#include "main.h"

int main(void)
{
    int32_t retval = 0;

    /* Initialize BDK library, set system cloc
    BDK_Initialize();

```

4. Double click GDB Segger J−Link Debugging to create the debug configuration for the selected example.


Debug Configurations

Create, manage, and run configurations

Name: pir_shield_example Debug

Project: pir_shield_example

C/C++ Application: Debug\pir_shield_example.elf

Build (if required) before launching

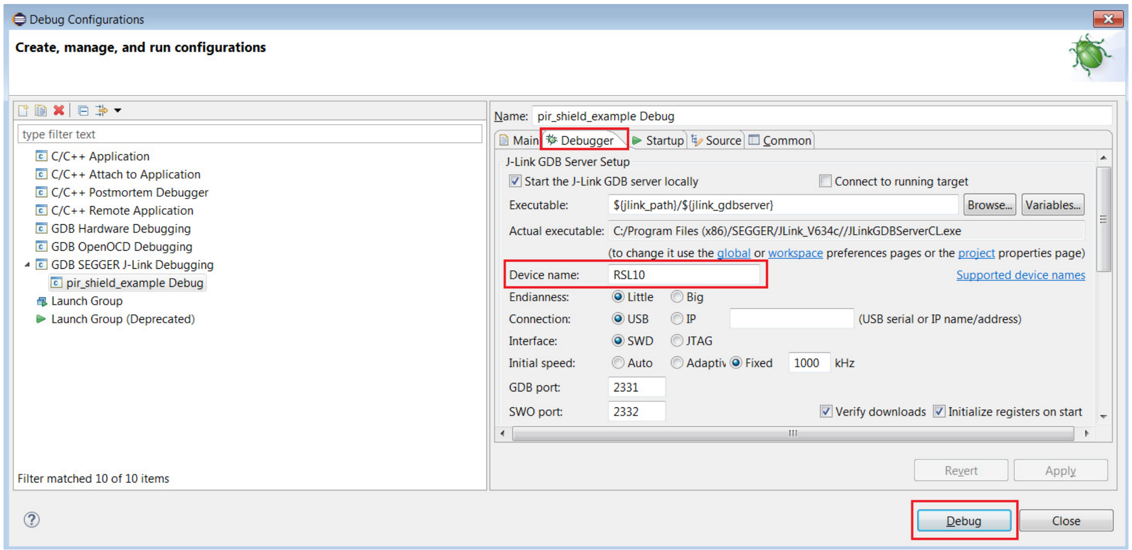
Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

NOTE: The debug configuration for the selected example is automatically saved and there’s no need to re−create it.

5. On the Debugger tab, set RSL10 as the device name. Click Debug to launch the code.


Debug Configurations

Create, manage, and run configurations

Name: pir_shield_example Debug

J-Link GDB Server Setup

☑ Start the J-Link GDB server locally ☐ Connect to running target

Executable: ${jlink_path}/${jlink_gdbserver}

Actual executable: C:/Program Files (x86)/SEGGER/JLink_V634c//JLinkGDBServerCL.exe

(to change it use the global or workspace preferences pages or the project properties page)

Device name: RSL10

Endianness: ● Little ○ Big

Connection: ● USB ○ IP (USB serial or IP name/address)

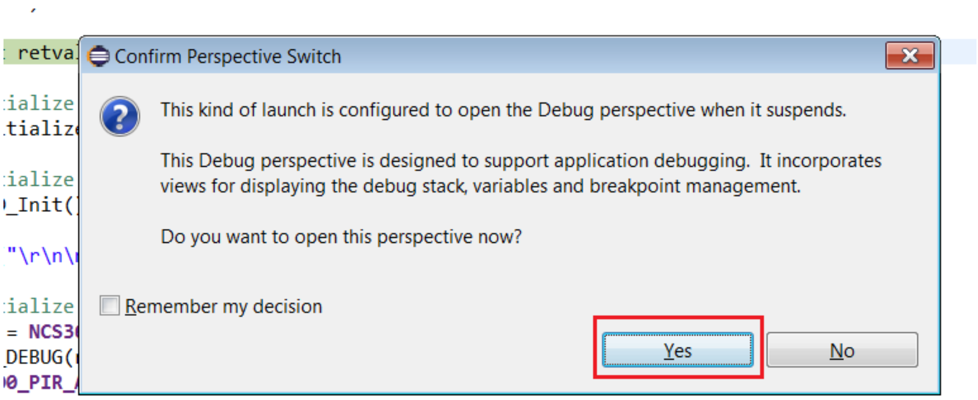
Interface: ● SWD ○ JTAG

Initial speed: ○ Auto ○ Adaptiv ● Fixed 1000 kHz

GDB port: 2331

SWO port: 2332 ☑ Verify downloads ☑ Initialize registers on start

6. For application debugging, confirm perspective switch by clicking Yes.


Confirm Perspective Switch

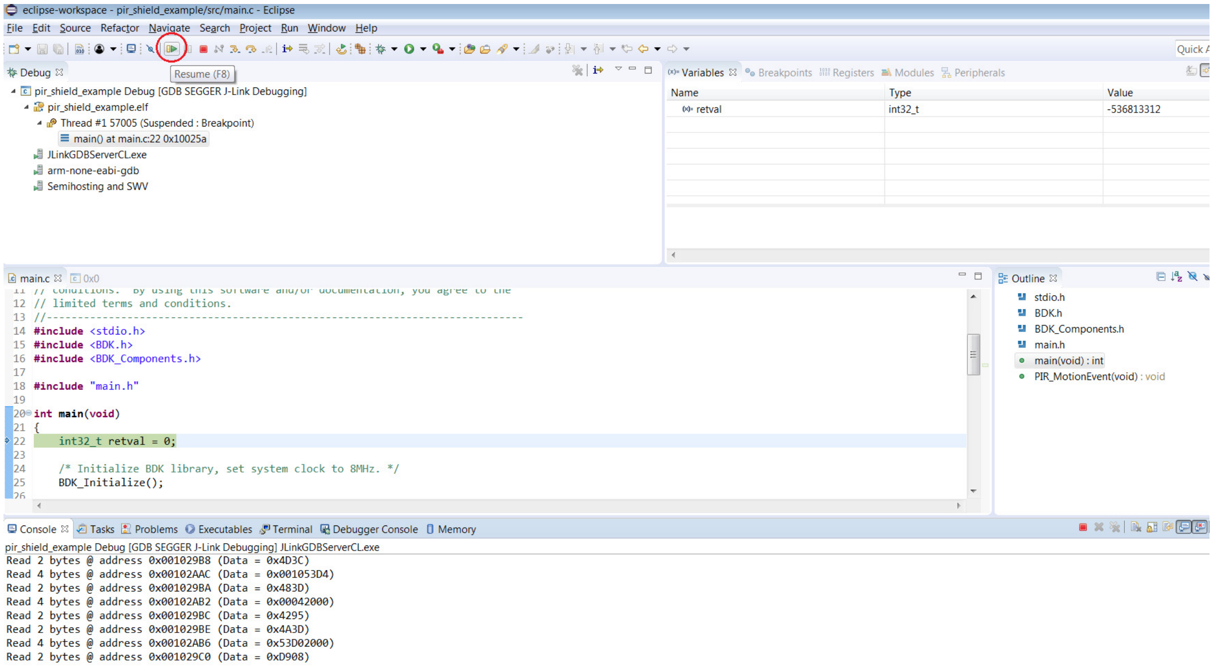
This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

☐ Remember my decision

7. The debug session is now launched. Click Resume (F8) to start the target CPU.



pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Suspended : Breakpoint)
main() at main.c:22 0x10025a
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

Name	Type	Value
retval	int32_t	-536813312

Outline: stdio.h, BDK.h, BDK_Components.h, main.h, main(void) : int, PIR_MotionEvent(void) : void

```
// conditions.  By using this software and/or documentation, you agree to the
// limited terms and conditions.
//-------------------------------------------------------------------------
#include <stdio.h>
#include <BDK.h>
#include <BDK_Components.h>

#include "main.h"

int main(void)
{
    int32_t retval = 0;

    /* Initialize BDK library, set system clock to 8MHz. */
    BDK_Initialize();
```

```
pir_shield_example Debug [GDB SEGGER J-Link Debugging] JLinkGDBServerCL.exe
Read 2 bytes @ address 0x001029B8 (Data = 0x4D3C)
Read 4 bytes @ address 0x00102AAC (Data = 0x001053D4)
Read 2 bytes @ address 0x001029BA (Data = 0x483D)
Read 4 bytes @ address 0x00102AB2 (Data = 0x00042000)
Read 2 bytes @ address 0x001029BC (Data = 0x4295)
Read 2 bytes @ address 0x001029BE (Data = 0x4A3D)
Read 4 bytes @ address 0x00102AB6 (Data = 0x53D02000)
Read 2 bytes @ address 0x001029C0 (Data = 0xD908)
```

Logging/Debugging

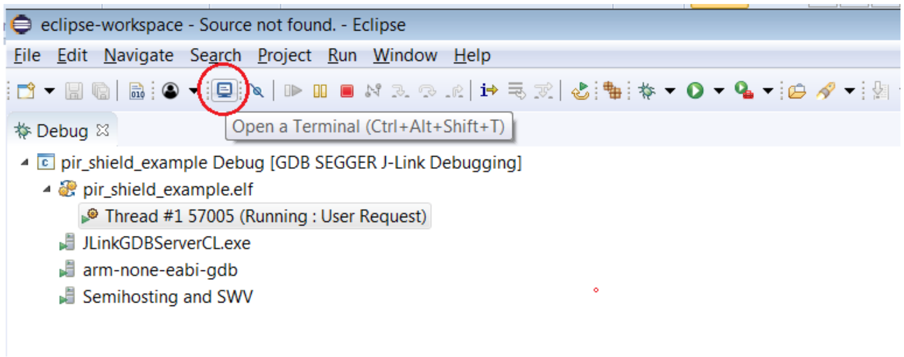
The following options are available to log/debug the downloaded firmware:

- Eclipse RTT Console
- J−Link RTT
- AX8052F100 UART−SPI bridge

This section provides instructions for each of the above options.

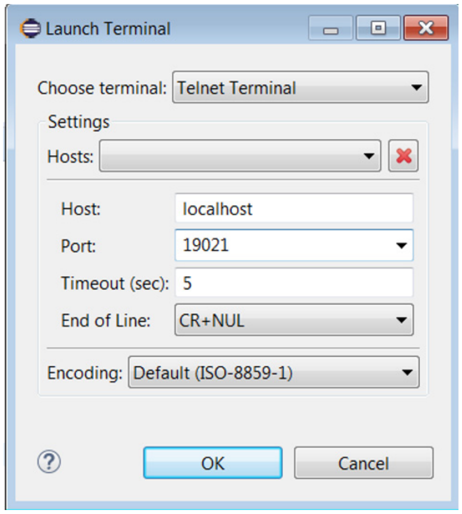
Using Eclipse Console

1. Click the Open a Terminal Icon


eclipse-workspace - Source not found. - Eclipse
File Edit Navigate Search Project Run Window Help
Open a Terminal (Ctrl+Alt+Shift+T)
Debug

pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

2. Enter the values shown below and launch the session. The incoming events are printed on the terminal window.



Launch Terminal

Choose terminal: Telnet Terminal

Settings

Hosts:

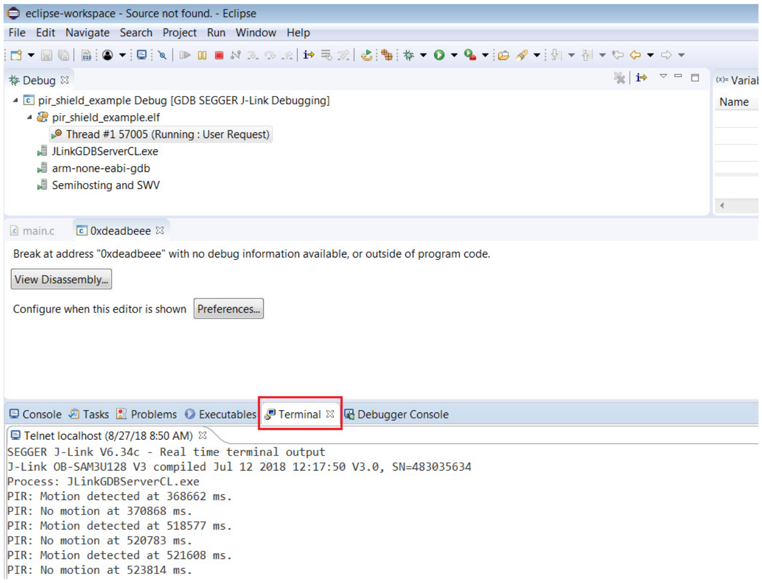
Host: localhost

Port: 19021

Timeout (sec): 5

End of Line: CR+NUL

Encoding: Default (ISO-8859-1)



eclipse-workspace - Source not found. - Eclipse

pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

Break at address "0xdeadbeee" with no debug information available, or outside of program code.

Configure when this editor is shown

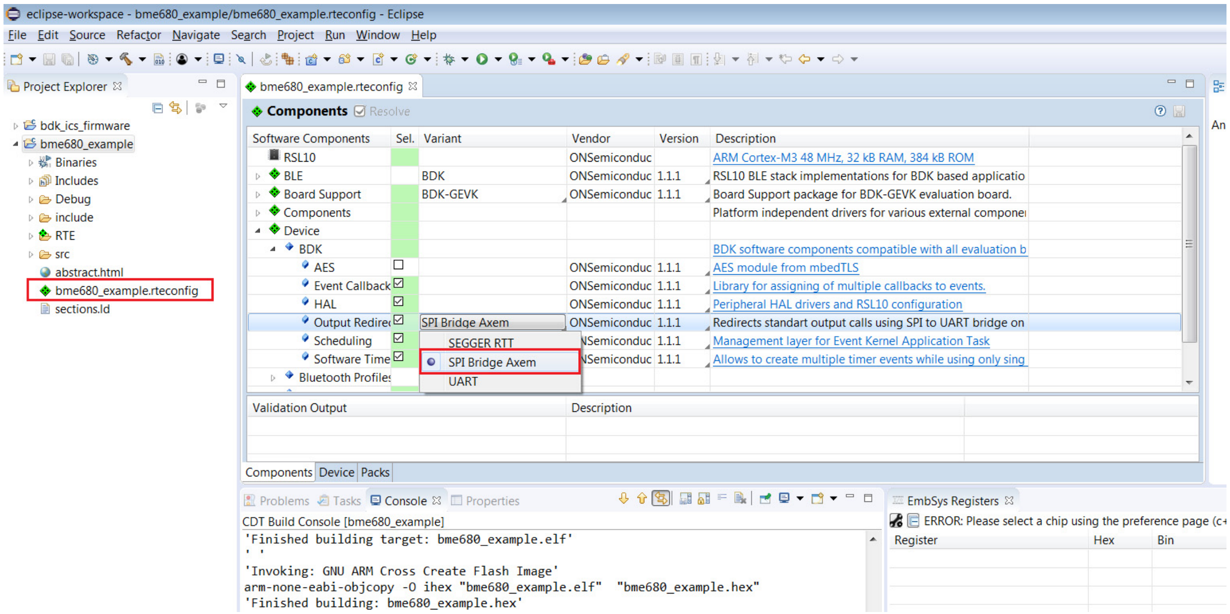
Telnet localhost (8/27/18 8:50 AM)

```
SEGGER J-Link V6.34c - Real time terminal output
J-Link OB-SAM3U128 V3 compiled Jul 12 2018 12:17:50 V3.0, SN=483035634
Process: JLinkGDBServerCL.exe
PIR: Motion detected at 368662 ms.
PIR: No motion at 370868 ms.
PIR: Motion detected at 518577 ms.
PIR: No motion at 520783 ms.
PIR: Motion detected at 521608 ms.
PIR: No motion at 523814 ms.
```

Using Eclipse Serial Console via UART–SPI Bridge.

When you do not want to use the Segger RTT viewer as serial console, the BDK–GEVK board is equipped with UART–SPI uC AX8052F100 flashed with special firmware, taking care of the entire serial communication with values returned on Terminal.

3. Click on the example’s rteconfig file and choose under *Device/BDK/Output redirection*, SPI Bridge AXEM. Save, compile and flash the whole project.



Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 32 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.1.1	RSL10 BLE stack implementations for BDK based applicatio
Board Support		BDK-GEVK	ONSemiconduc	1.1.1	Board Support package for BDK-GEVK evaluation board.
Components					Platform independent drivers for various external compone
Device					
BDK					BDK software components compatible with all evaluation b
AES	☐		ONSemiconduc	1.1.1	AES module from mbedTLS
Event Callback	☑		ONSemiconduc	1.1.1	Library for assigning of multiple callbacks to events.
HAL	☑		ONSemiconduc	1.1.1	Peripheral HAL drivers and RSL10 configuration
Output Redire	☑	SPI Bridge Axem	ONSemiconduc	1.1.1	Redirects standart output calls using SPI to UART bridge on
Scheduling	☑	SEGGER RTT	NSemiconduc	1.1.1	Management layer for Event Kernel Application Task
Software Time	☑	● SPI Bridge Axem	NSemiconduc	1.1.1	Allows to create multiple timer events while using only sing
Bluetooth Profiles		UART			

Validation Output	Description

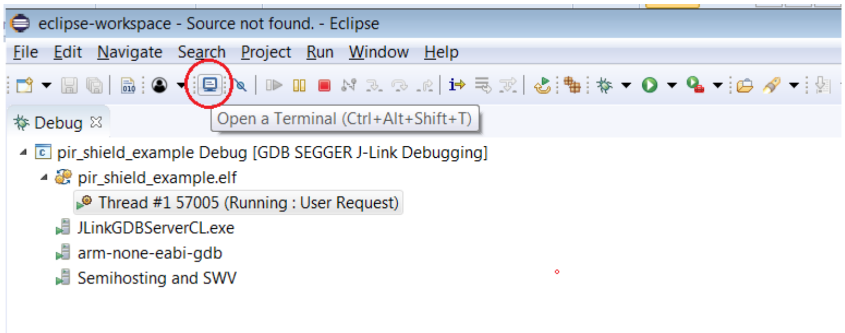
CDT Build Console [bme680_example]

```
'Finished building target: bme680_example.elf'
' '
'Invoking: GNU ARM Cross Create Flash Image'
arm-none-eabi-objcopy -O ihex "bme680_example.elf"  "bme680_example.hex"
'Finished building: bme680_example.hex'
```

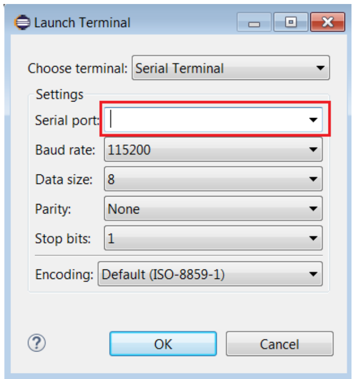
ERROR: Please select a chip using the preference page (c

Register	Hex	Bin

4. When the project runs, Click the **Open a Terminal** Icon.


eclipse-workspace - Source not found. - Eclipse
File Edit Navigate Search Project Run Window Help
Open a Terminal (Ctrl+Alt+Shift+T)
Debug
pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

5. Enter the appropriate COM port as shown below, and launch the session. The incoming events are entered on the terminal window.



Choose terminal: Serial Terminal

Serial port:

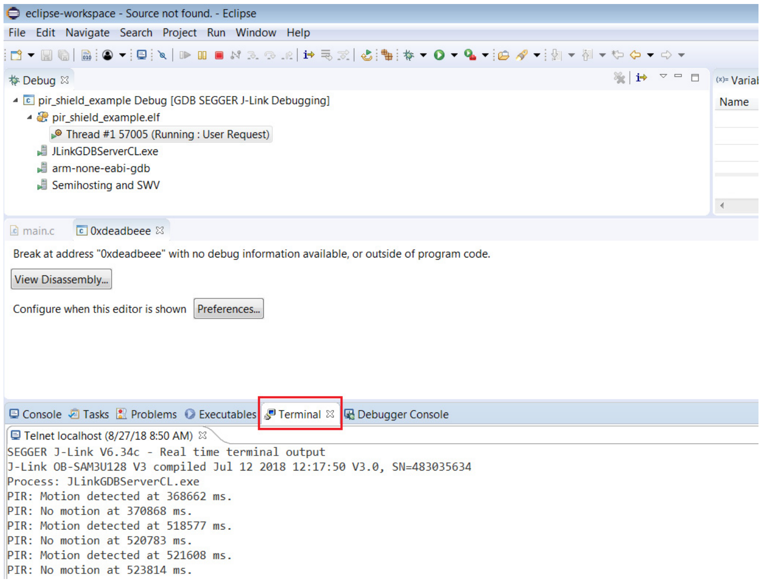
Baud rate: 115200

Data size: 8

Parity: None

Stop bits: 1

Encoding: Default (ISO-8859-1)



pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

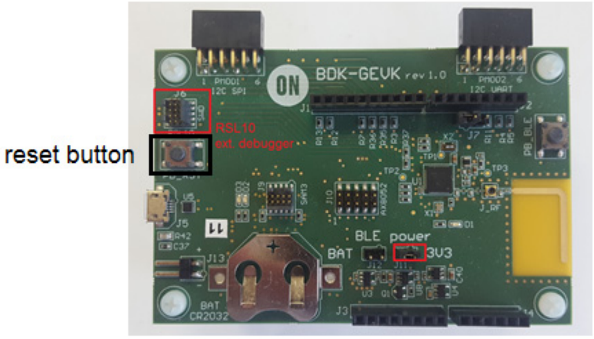
Break at address "0xdeadbeee" with no debug information available, or outside of program code.

Configure when this editor is shown

Telnet localhost (8/27/18 8:50 AM)

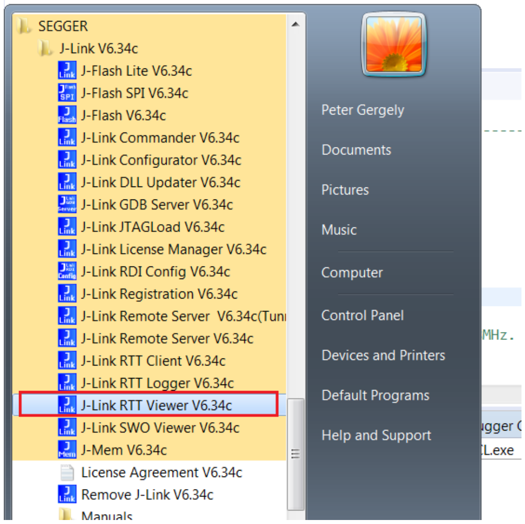
```
SEGGER J-Link V6.34c - Real time terminal output
J-Link OB-SAM3U128 V3 compiled Jul 12 2018 12:17:50 V3.0, SN=483035634
Process: JLinkGDBServerCL.exe
PIR: Motion detected at 368662 ms.
PIR: No motion at 370868 ms.
PIR: Motion detected at 518577 ms.
PIR: No motion at 520783 ms.
PIR: Motion detected at 521608 ms.
PIR: No motion at 523814 ms.
```

NOTE: You may reset (PB_RST) the BDK−GEVK (shown below) to launch the RTT terminal without needing to launch Eclipse.

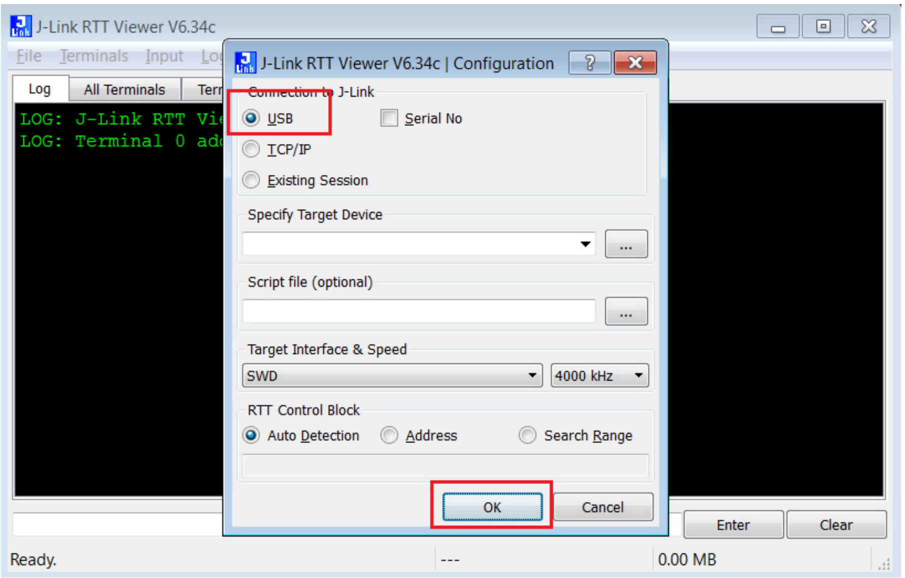


Using J−Link RTT

6. After step 14 is done, open J−Link RTT viewer (should be installed when J−Link software package was installed per Step 2).



7. Select USB and click OK.



```
LOG: J-Link RTT Vi
LOG: Terminal 0 ad
```

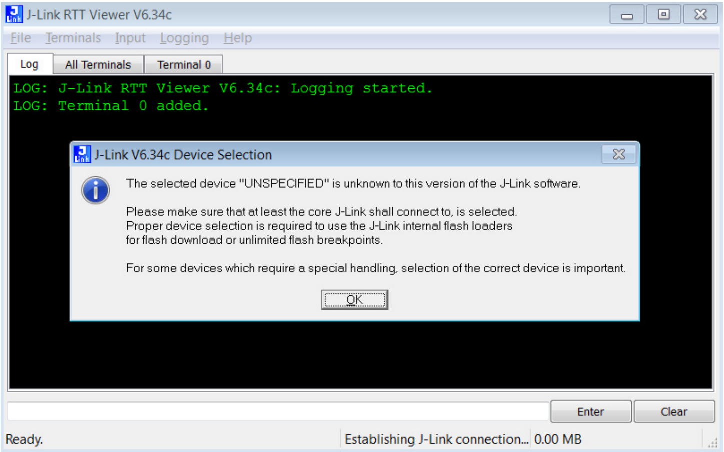
Connection to J-Link: ● USB ☐ Serial No ○ TCP/IP ○ Existing Session

Target Interface & Speed: SWD 4000 kHz

RTT Control Block: ● Auto Detection ○ Address ○ Search Range

Ready. --- 0.00 MB

8. RTT prompts you to select the appropriate microcontroller. Select RSL10 and click OK. The serial terminal is ready to use and the events from RSL10 can be observed by clicking the All Terminals Window.


J-Link RTT Viewer V6.34c
File Terminals Input Logging Help
Log All Terminals Terminal 0
J-Link V6.34c Device Selection
OK
Enter Clear

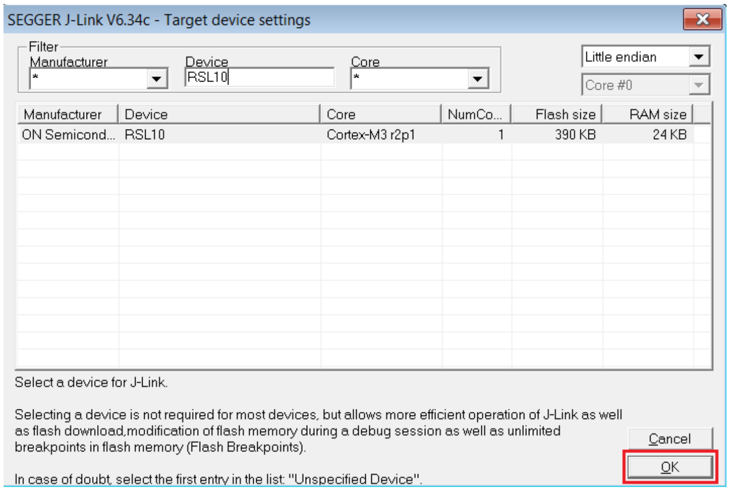
```
LOG: J-Link RTT Viewer V6.34c: Logging started.
LOG: Terminal 0 added.
```

The selected device "UNSPECIFIED" is unknown to this version of the J-Link software.

Please make sure that at least the core J-Link shall connect to, is selected.
Proper device selection is required to use the J-Link internal flash loaders
for flash download or unlimited flash breakpoints.

For some devices which require a special handling, selection of the correct device is important.

Ready. Establishing J-Link connection... 0.00 MB



Manufacturer: * Device: RSL10 Core: *

Little endian

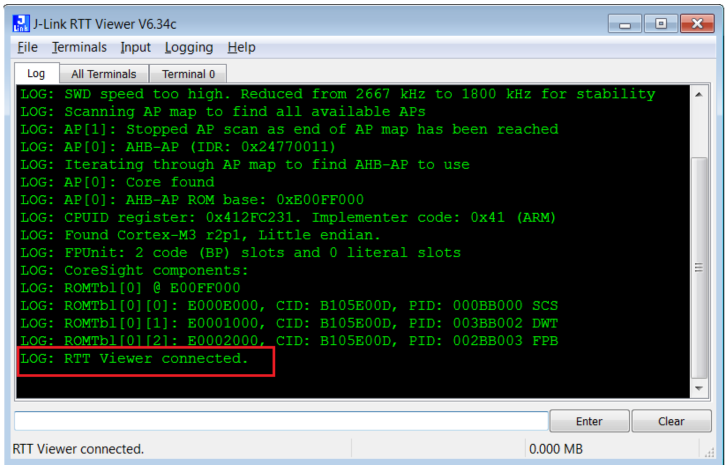
Core #0

Manufacturer	Device	Core	NumCo...	Flash size	RAM size
ON Semicond...	RSL10	Cortex-M3 r2p1	1	390 KB	24 KB

Select a device for J-Link.

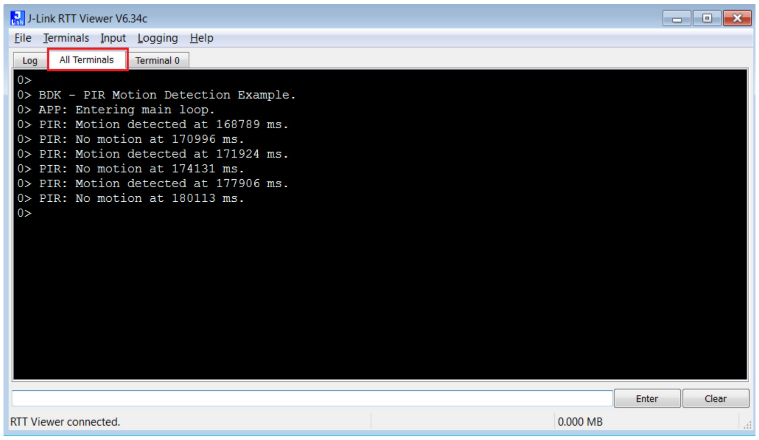
Selecting a device is not required for most devices, but allows more efficient operation of J-Link as well as flash download,modification of flash memory during a debug session as well as unlimited breakpoints in flash memory (Flash Breakpoints).

In case of doubt, select the first entry in the list: "Unspecified Device".


J-Link RTT Viewer V6.34c
File Terminals Input Logging Help
Log All Terminals Terminal 0
Enter Clear

```
LOG: SWD speed too high. Reduced from 2667 kHz to 1800 kHz for stability
LOG: Scanning AP map to find all available APs
LOG: AP[1]: Stopped AP scan as end of AP map has been reached
LOG: AP[0]: AHB-AP (IDR: 0x24770011)
LOG: Iterating through AP map to find AHB-AP to use
LOG: AP[0]: Core found
LOG: AP[0]: AHB-AP ROM base: 0xE00FF000
LOG: CPUID register: 0x412FC231. Implementer code: 0x41 (ARM)
LOG: Found Cortex-M3 r2p1, Little endian.
LOG: FPUnit: 2 code (BP) slots and 0 literal slots
LOG: CoreSight components:
LOG: ROMTbl[0] @ E00FF000
LOG: ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB000 SCS
LOG: ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
LOG: ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
LOG: RTT Viewer connected.
```

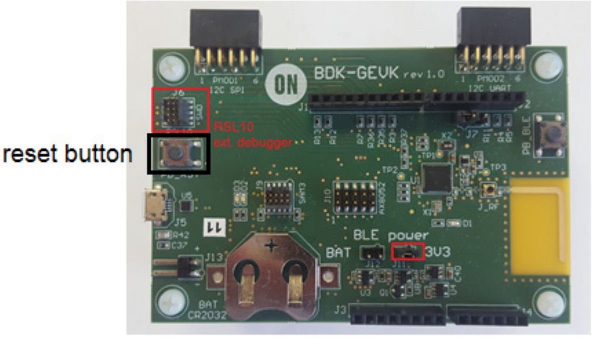
RTT Viewer connected. 0.000 MB



```
0>
0> BDK - PIR Motion Detection Example.
0> APP: Entering main loop.
0> PIR: Motion detected at 168789 ms.
0> PIR: No motion at 170996 ms.
0> PIR: Motion detected at 171924 ms.
0> PIR: No motion at 174131 ms.
0> PIR: Motion detected at 177906 ms.
0> PIR: No motion at 180113 ms.
0>
```

RTT Viewer connected. 0.000 MB

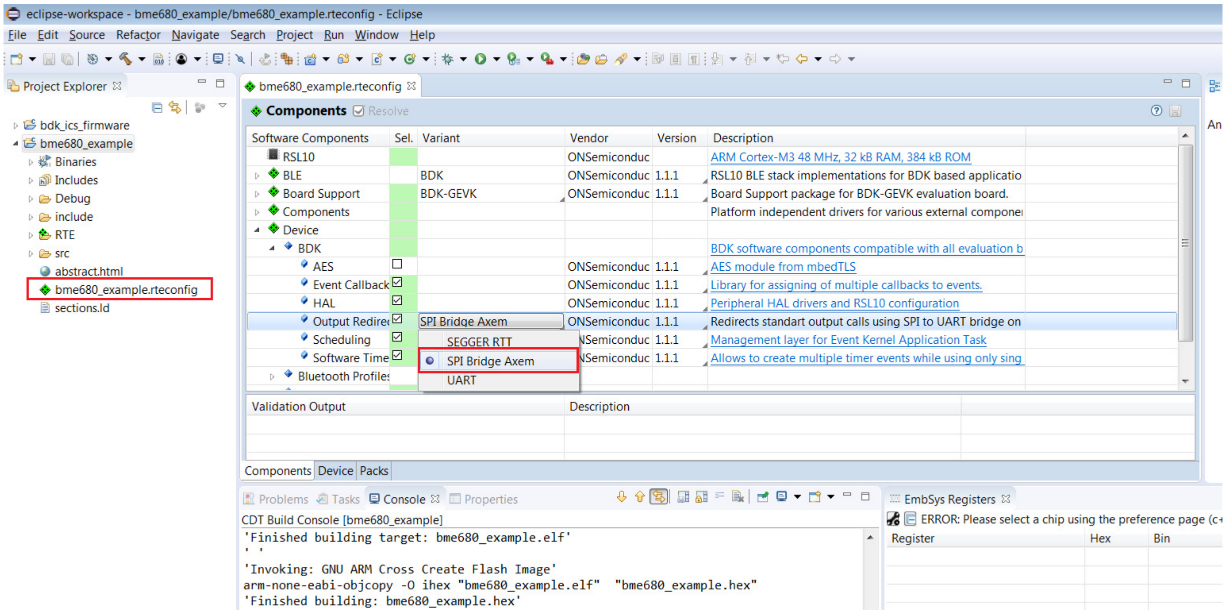
NOTE: You may reset (PB_RST) the BDK−GEVK (shown below) to launch RTT terminal without needing to launch Eclipse.



Using Eclipse Serial Console via UART−SPI Bridge

The BDK−GEVK board is equipped with UART−SPI microcontroller AX8052F100 flashed with special firmware, to enable serial communication with values returned to Terminal.

9. Click on example’s rteconfig file and choose “SPI Bridge AXEM” under *Device/BDK/Output redirection*. Save, compile and flash the whole project.



Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 32 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.1.1	RSL10 BLE stack implementations for BDK based applicatio
Board Support		BDK-GEVK	ONSemiconduc	1.1.1	Board Support package for BDK-GEVK evaluation board.
Components					Platform independent drivers for various external componei
Device					
BDK					BDK software components compatible with all evaluation b
AES	☐		ONSemiconduc	1.1.1	AES module from mbedTLS
Event Callback	☑		ONSemiconduc	1.1.1	Library for assigning of multiple callbacks to events.
HAL	☑		ONSemiconduc	1.1.1	Peripheral HAL drivers and RSL10 configuration
Output Redire	☑	SPI Bridge Axem	ONSemiconduc	1.1.1	Redirects standart output calls using SPI to UART bridge on
Scheduling	☑		NSemiconduc	1.1.1	Management layer for Event Kernel Application Task
Software Time	☑		NSemiconduc	1.1.1	Allows to create multiple timer events while using only sing
Bluetooth Profiles					

Validation Output	Description

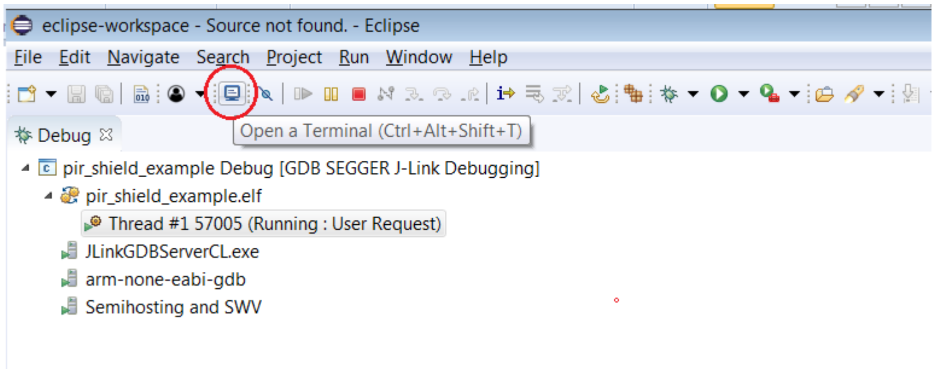
CDT Build Console [bme680_example]

```
'Finished building target: bme680_example.elf'
' '
'Invoking: GNU ARM Cross Create Flash Image'
arm-none-eabi-objcopy -O ihex "bme680_example.elf"  "bme680_example.hex"
'Finished building: bme680_example.hex'
```

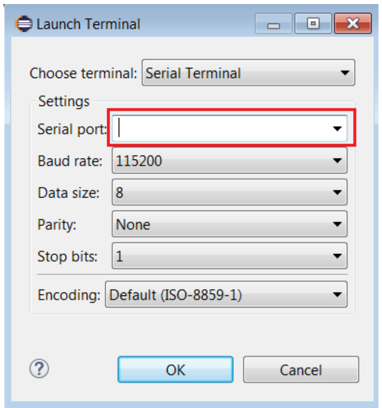
ERROR: Please select a chip using the preference page (c+

Register	Hex	Bin

10. When the project runs, Click the Open a Terminal Icon.


eclipse-workspace - Source not found. - Eclipse
File Edit Navigate Search Project Run Window Help
Open a Terminal (Ctrl+Alt+Shift+T)
Debug
pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

11. Enter the appropriate COM port as shown below and launch the session. The incoming events are printed on the terminal window.



Choose terminal: Serial Terminal

Serial port:

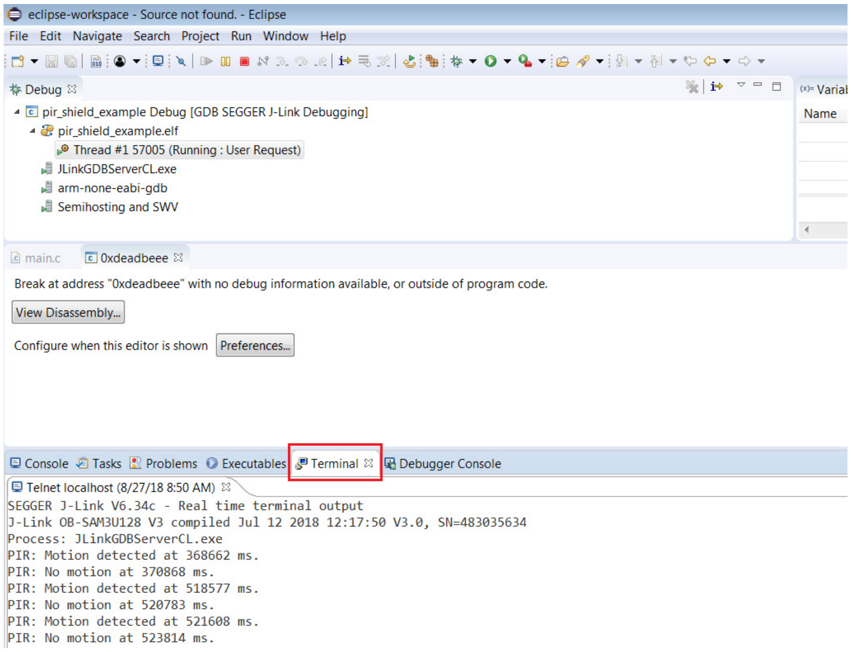
Baud rate: 115200

Data size: 8

Parity: None

Stop bits: 1

Encoding: Default (ISO-8859-1)


eclipse-workspace - Source not found. - Eclipse
File Edit Navigate Search Project Run Window Help
Debug
Variał
Name
pir_shield_example Debug [GDB SEGGER J-Link Debugging]
pir_shield_example.elf
Thread #1 57005 (Running : User Request)
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV
main.c
0xdeadbeee
View Disassembly...
Preferences...
Console
Tasks
Problems
Executables
Terminal
Debugger Console
Telnet localhost (8/27/18 8:50 AM)

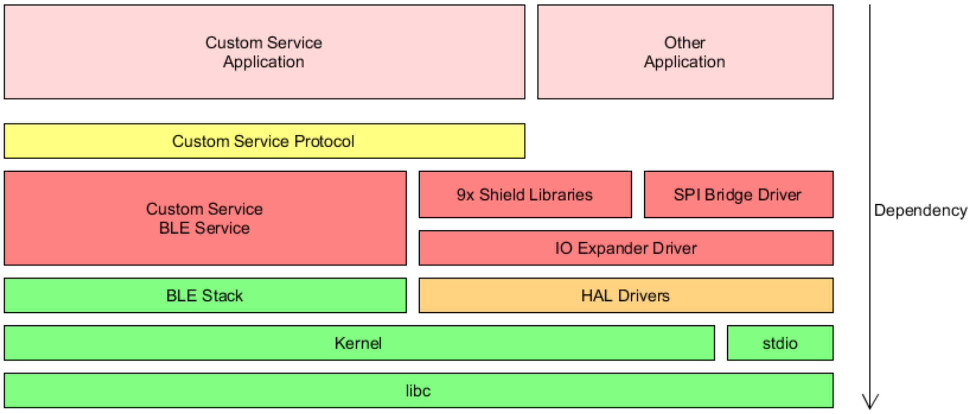
Break at address "0xdeadbeee" with no debug information available, or outside of program code.

Configure when this editor is shown

```
SEGGER J-Link V6.34c - Real time terminal output
J-Link OB-SAM3U128 V3 compiled Jul 12 2018 12:17:50 V3.0, SN=483035634
Process: JLinkGDBServerCL.exe
PIR: Motion detected at 368662 ms.
PIR: No motion at 370868 ms.
PIR: Motion detected at 518577 ms.
PIR: No motion at 520783 ms.
PIR: Motion detected at 521608 ms.
PIR: No motion at 523814 ms.
```

SOFTWARE ORGANIZATION

For users modifying the example code and building new projects, the following sections detail the B−IDK software organization. The stack overview is shown below.

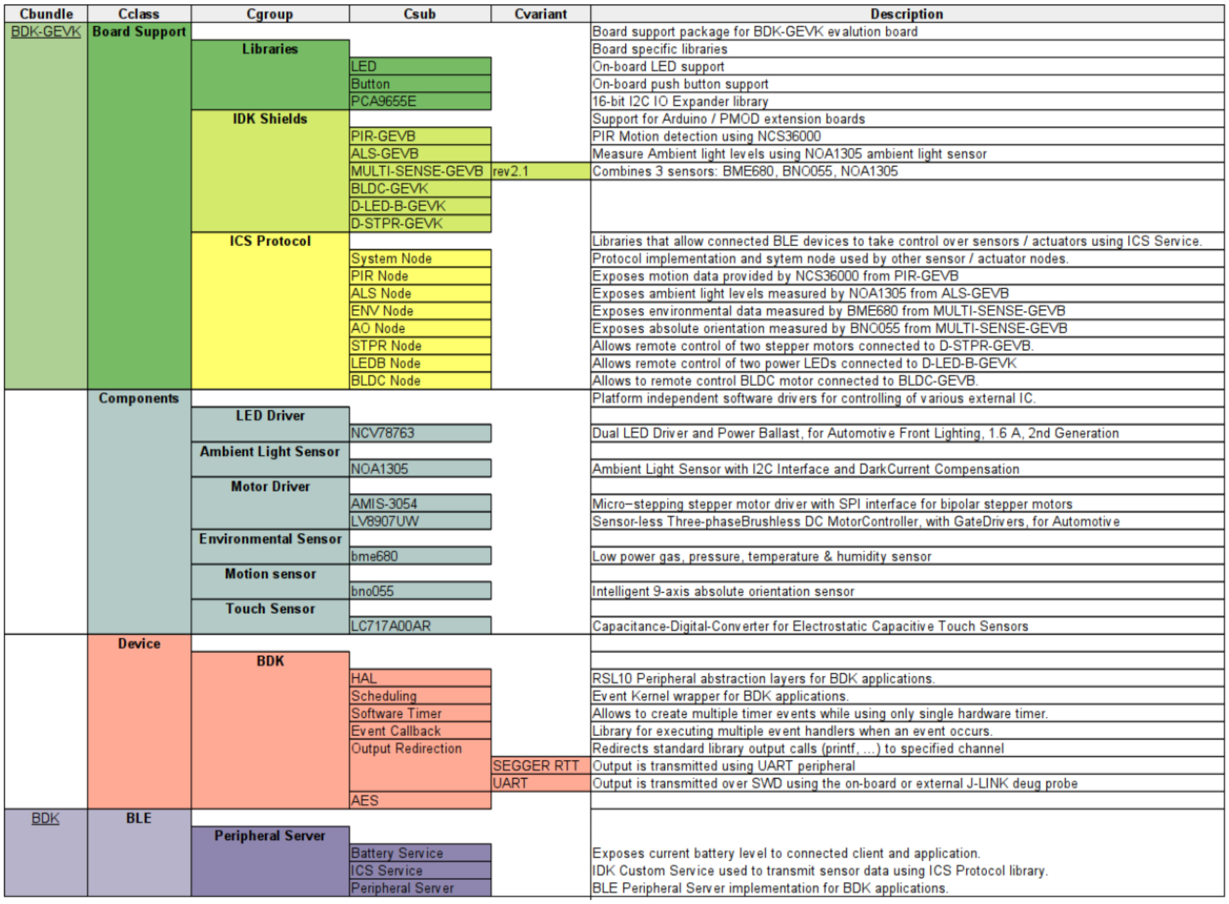


B−IDK CMSIS Software Organization

CMSIS pack and the associated software components handle multiple evaluation boards as different bundles of the standardized Board Support Cclass.

- This bundle shows only components supported by ON Semiconductor for a given board
- No confusing component variants

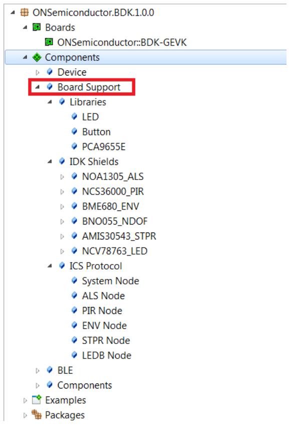
Common libraries and HAL are in a separate group within the Device class



Cbundle	Cclass	Cgroup	Csub	Cvariant	Description
BDK-GEVK	Board Support				Board support package for BDK-GEVK evalution board
		Libraries			Board specific libraries
			LED		On-board LED support
			Button		On-board push button support
			PCA9655E		16-bit I2C IO Expander library
		IDK Shields			Support for Arduino / PMOD extension boards
			PIR-GEVB		PIR Motion detection using NCS36000
			ALS-GEVB		Measure Ambient light levels using NOA1305 ambient light sensor
			MULTI-SENSE-GEVB	rev2.1	Combines 3 sensors: BME680, BNO055, NOA1305
			BLDC-GEVK		
			D-LED-B-GEVK		
			D-STPR-GEVK		
		ICS Protocol			Libraries that allow connected BLE devices to take control over sensors / actuators using ICS Service.
			System Node		Protocol implementation and sytem node used by other sensor / actuator nodes.
			PIR Node		Exposes motion data provided by NCS36000 from PIR-GEVB
			ALS Node		Exposes ambient light levels measured by NOA1305 from ALS-GEVB
			ENV Node		Exposes environmental data measured by BME680 from MULTI-SENSE-GEVB
			AO Node		Exposes absolute orientation measured by BNO055 from MULTI-SENSE-GEVB
			STPR Node		Allows remote control of two stepper motors connected to D-STPR-GEVB.
			LEDB Node		Allows remote control of two power LEDs connected to D-LED-B-GEVK
			BLDC Node		Allows to remote control BLDC motor connected to BLDC-GEVB.
	Components				Platform independent software drivers for controlling of various external IC.
		LED Driver			
			NCV78763		Dual LED Driver and Power Ballast, for Automotive Front Lighting, 1.6 A, 2nd Generation
		Ambient Light Sensor			
			NOA1305		Ambient Light Sensor with I2C Interface and DarkCurrent Compensation
		Motor Driver			
			AMIS-3054		Micro−stepping stepper motor driver with SPI interface for bipolar stepper motors
			LV8907UW		Sensor-less Three-phaseBrushless DC MotorController, with GateDrivers, for Automotive
		Environmental Sensor			
			bme680		Low power gas, pressure, temperature & humidity sensor
		Motion sensor			
			bno055		Intelligent 9-axis absolute orientation sensor
		Touch Sensor			
			LC717A00AR		Capacitance-Digital-Converter for Electrostatic Capacitive Touch Sensors
	Device				
		BDK			
			HAL		RSL10 Peripheral abstraction layers for BDK applications.
			Scheduling		Event Kernel wrapper for BDK applications.
			Software Timer		Allows to create multiple timer events while using only single hardware timer.
			Event Callback		Library for executing multiple event handlers when an event occurs.
			Output Redirection		Redirects standard library output calls (printf, ...) to specified channel
				SEGGER RTT	Output is transmitted using UART peripheral
				UART	Output is transmitted over SWD using the on-board or external J-LINK deug probe
			AES		
BDK	BLE				
		Peripheral Server			
			Battery Service		Exposes current battery level to connected client and application.
			ICS Service		IDK Custom Service used to transmit sensor data using ICS Protocol library.
			Peripheral Server		BLE Peripheral Server implementation for BDK applications.

Board Support

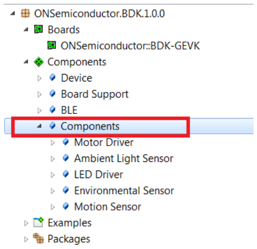
- Libraries to support BDK−GEVK, GPIO Expander, Various daughter cards and custom protocol (required for the mobile app)



- ONSemiconductor.BDK.1.0.0
  - Boards
    - ONSemiconductor::BDK-GEVK
  - Components
    - Device
    - Board Support
      - Libraries
        - LED
        - Button
        - PCA9655E
      - IDK Shields
        - NOA1305_ALS
        - NCS36000_PIR
        - BME680_ENV
        - BNO055_NDOF
        - AMIS30543_STPR
        - NCV78763_LED
      - ICS Protocol
        - System Node
        - ALS Node
        - PIR Node
        - ENV Node
        - STPR Node
        - LEDB Node
    - BLE
    - Components
  - Examples
  - Packages

Components

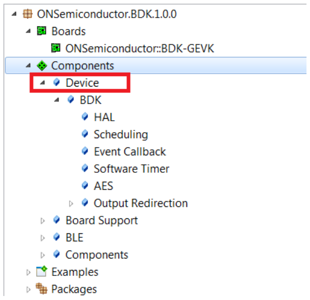
- Libraries attached to board support



- ONSemiconductor.BDK.1.0.0
  - Boards
    - ONSemiconductor::BDK-GEVK
  - Components
    - Device
    - Board Support
    - BLE
    - Components
      - Motor Driver
      - Ambient Light Sensor
      - LED Driver
      - Environmental Sensor
      - Motion Sensor
  - Examples
  - Packages

Device

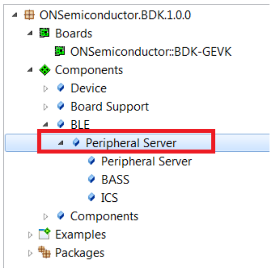
- Abstraction layers for interfaces, timers, AES, serial re−direction, etc.



- ONSemiconductor.BDK.1.0.0
  - Boards
    - ONSemiconductor::BDK-GEVK
  - Components
    - Device
      - BDK
        - HAL
        - Scheduling
        - Event Callback
        - Software Timer
        - AES
        - Output Redirection
    - Board Support
    - BLE
    - Components
  - Examples
  - Packages

BLE

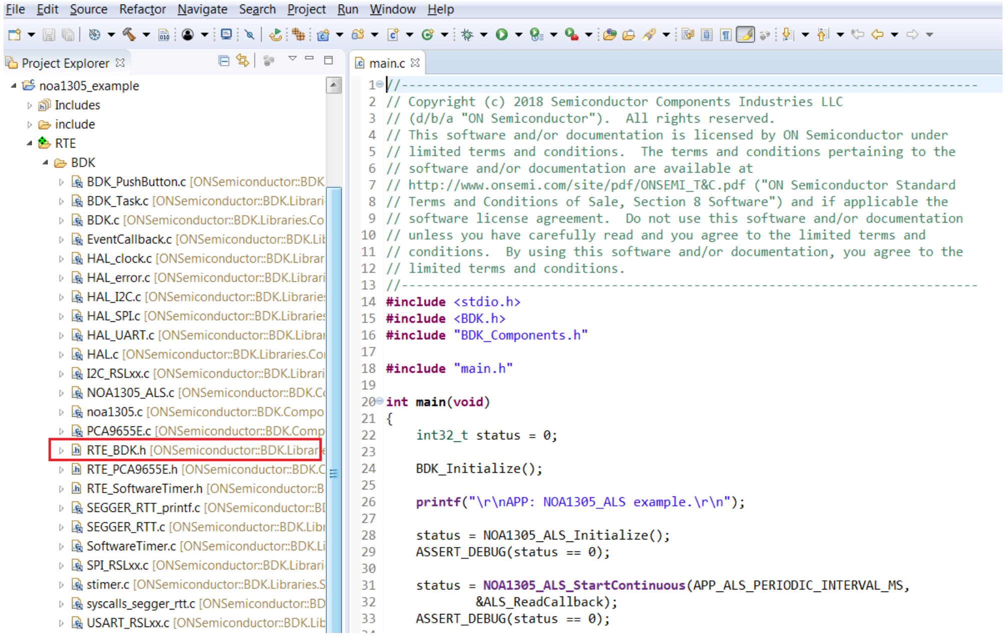
- Peripheral Server Support



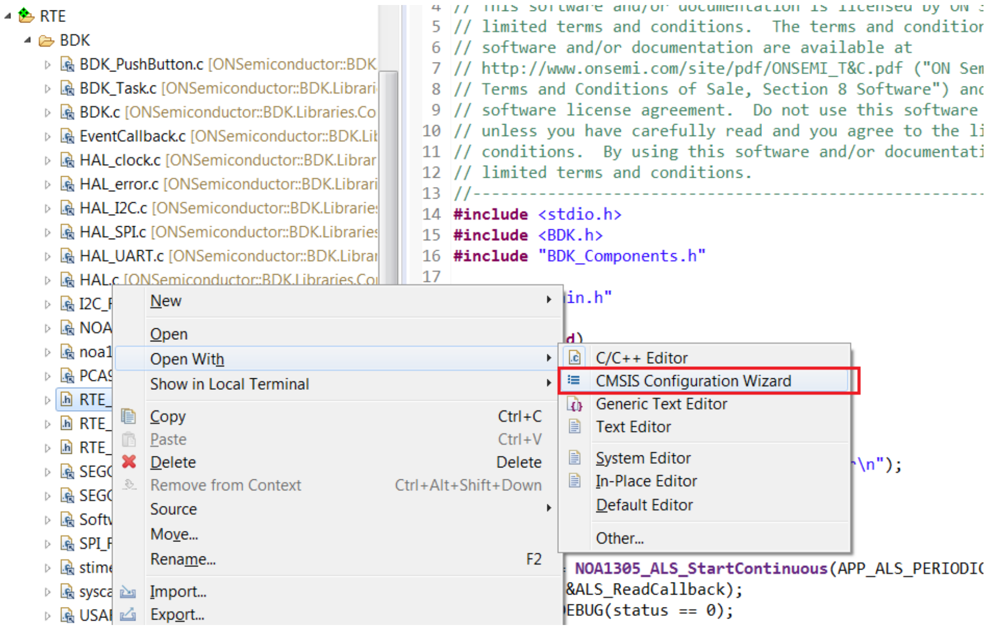
- ONSemiconductor.BDK.1.0.0
  - Boards
    - ONSemiconductor::BDK-GEVK
  - Components
    - Device
    - Board Support
    - BLE
      - Peripheral Server
        - Peripheral Server
        - BASS
        - ICS
    - Components
  - Examples
  - Packages

CONFIGURATION SETUP

System settings can be configured directly from within the CMSIS pack. Each example is equipped with basic system configuration that covers three main categories. These are accessible in the RTE/BDK folder within the project. Each system configuration starts with “RTE_”. As shown below, opening the RTE_... header files using the CMSIS configuration wizard (right click on the header file), displays the configuration table. Various application specific parameters can be set. This allows pre−configuration of RSL10 without the need for explicit programming.


File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
main.c
noa1305_example
Includes
include
RTE
BDK
BDK_PushButton.c [ONSemiconductor::BDK
BDK_Task.c [ONSemiconductor::BDK.Librari
BDK.c [ONSemiconductor::BDK.Libraries.Co
EventCallback.c [ONSemiconductor::BDK.Lit
HAL_clock.c [ONSemiconductor::BDK.Librar
HAL_error.c [ONSemiconductor::BDK.Librari
HAL_I2C.c [ONSemiconductor::BDK.Librarie
HAL_SPI.c [ONSemiconductor::BDK.Libraries
HAL_UART.c [ONSemiconductor::BDK.Librar
HAL.c [ONSemiconductor::BDK.Libraries.Co
I2C_RSLxx.c [ONSemiconductor::BDK.Librari
NOA1305_ALS.c [ONSemiconductor::BDK.C
noa1305.c [ONSemiconductor::BDK.Compo
PCA9655E.c [ONSemiconductor::BDK.Comp
RTE_BDK.h [ONSemiconductor::BDK.Librar
RTE_PCA9655E.h [ONSemiconductor::BDK.C
RTE_SoftwareTimer.h [ONSemiconductor::B
SEGGER_RTT_printf.c [ONSemiconductor::BI
SEGGER_RTT.c [ONSemiconductor::BDK.Libr
SoftwareTimer.c [ONSemiconductor::BDK.Li
SPI_RSLxx.c [ONSemiconductor::BDK.Librari
stimer.c [ONSemiconductor::BDK.Libraries.S
syscalls_segger_rtt.c [ONSemiconductor::BD
USART_RSLxx.c [ONSemiconductor::BDK.Lib

```
//-----------------------------------------------------------------------------
// Copyright (c) 2018 Semiconductor Components Industries LLC
// (d/b/a "ON Semiconductor").  All rights reserved.
// This software and/or documentation is licensed by ON Semiconductor under
// limited terms and conditions.  The terms and conditions pertaining to the
// software and/or documentation are available at
// http://www.onsemi.com/site/pdf/ONSEMI_T&C.pdf ("ON Semiconductor Standard
// Terms and Conditions of Sale, Section 8 Software") and if applicable the
// software license agreement.  Do not use this software and/or documentation
// unless you have carefully read and you agree to the limited terms and
// conditions.  By using this software and/or documentation, you agree to the
// limited terms and conditions.
//-----------------------------------------------------------------------------
#include <stdio.h>
#include <BDK.h>
#include "BDK_Components.h"

#include "main.h"

int main(void)
{
    int32_t status = 0;

    BDK_Initialize();

    printf("\r\nAPP: NOA1305_ALS example.\r\n");

    status = NOA1305_ALS_Initialize();
    ASSERT_DEBUG(status == 0);

    status = NOA1305_ALS_StartContinuous(APP_ALS_PERIODIC_INTERVAL_MS,
            &ALS_ReadCallback);
    ASSERT_DEBUG(status == 0);
```

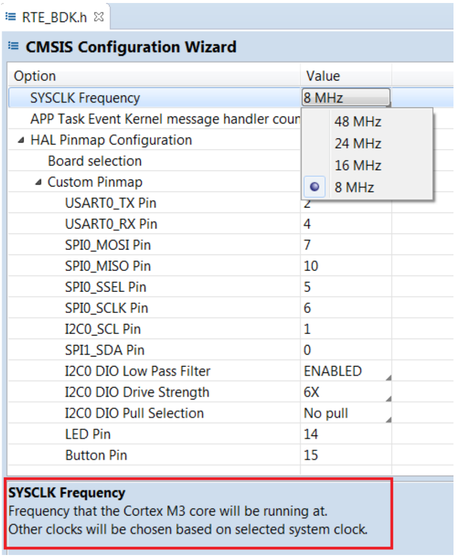

RTE
BDK
BDK_PushButton.c [ONSemiconductor::BDK
BDK_Task.c [ONSemiconductor::BDK.Librari
BDK.c [ONSemiconductor::BDK.Libraries.Co
EventCallback.c [ONSemiconductor::BDK.Lit
HAL_clock.c [ONSemiconductor::BDK.Librar
HAL_error.c [ONSemiconductor::BDK.Librari
HAL_I2C.c [ONSemiconductor::BDK.Librarie
HAL_SPI.c [ONSemiconductor::BDK.Libraries
HAL_UART.c [ONSemiconductor::BDK.Librar
HAL.c [ONSemiconductor::BDK.Libraries.Co
New
Open
Open With
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Source
Move...
Rename... F2
Import...
Export...
C/C++ Editor
CMSIS Configuration Wizard
Generic Text Editor
Text Editor
System Editor
In-Place Editor
Default Editor
Other...

```
// limited terms and conditions.  The terms and conditio
// software and/or documentation are available at
// http://www.onsemi.com/site/pdf/ONSEMI_T&C.pdf ("ON Sem
// Terms and Conditions of Sale, Section 8 Software") and
// software license agreement.  Do not use this software
// unless you have carefully read and you agree to the li
// conditions.  By using this software and/or documentati
// limited terms and conditions.
//-------------------------------------------------------
#include <stdio.h>
#include <BDK.h>
#include "BDK_Components.h"
```

A brief description on the header files is given below.

RTE_BDK.h

Parameters such as system clock frequency and the board that feature RSL10 (default set to BDK−GEVK), etc. can be set. Descriptions of each of these parameters are also provided.

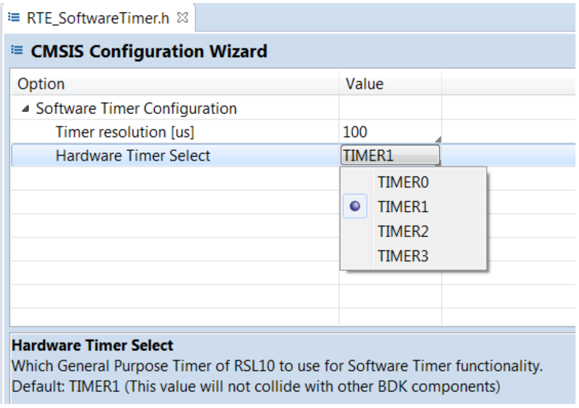


Option	Value
SYSCLK Frequency	8 MHz
APP Task Event Kernel message handler coun	
HAL Pinmap Configuration	
Board selection	
Custom Pinmap	
USART0_TX Pin	2
USART0_RX Pin	4
SPI0_MOSI Pin	7
SPI0_MISO Pin	10
SPI0_SSEL Pin	5
SPI0_SCLK Pin	6
I2C0_SCL Pin	1
SPI1_SDA Pin	0
I2C0 DIO Low Pass Filter	ENABLED
I2C0 DIO Drive Strength	6X
I2C0 DIO Pull Selection	No pull
LED Pin	14
Button Pin	15

SYSCLK Frequency
Frequency that the Cortex M3 core will be running at.
Other clocks will be chosen based on selected system clock.

RTE_Software_Timer.h

Various timers (4) supported by RSL10 can be configured by invoking the CMSIS configuration wizard on this header file. Timer 1 is used for B−IDK components.

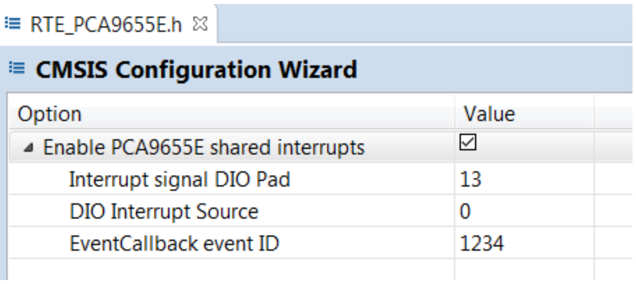


Option	Value
Software Timer Configuration	
Timer resolution [us]	100
Hardware Timer Select	TIMER1

Hardware Timer Select
Which General Purpose Timer of RSL10 to use for Software Timer functionality.
Default: TIMER1 (This value will not collide with other BDK components)

RTE_PCA9655.h

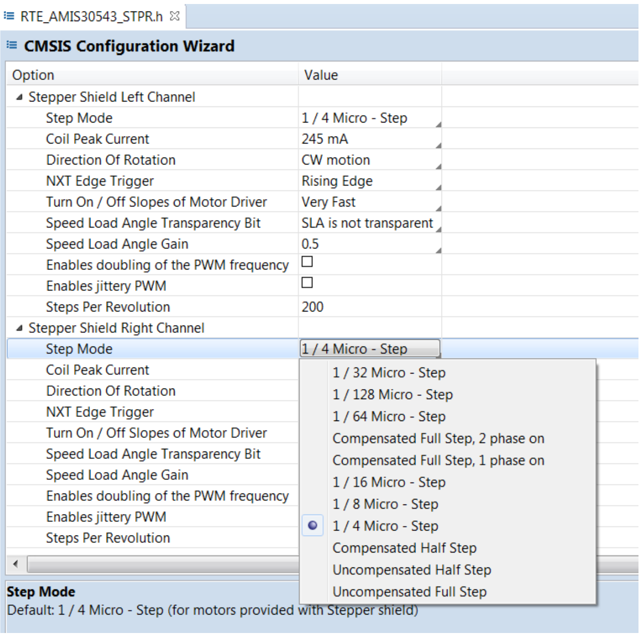
PCA9655 is the GPIO expander chip assembled on most daughter cards to expand interface functionality. Parameters related to this chip can be set here.



Option	Value
Enable PCA9655E shared interrupts	☑
Interrupt signal DIO Pad	13
DIO Interrupt Source	0
EventCallback event ID	1234

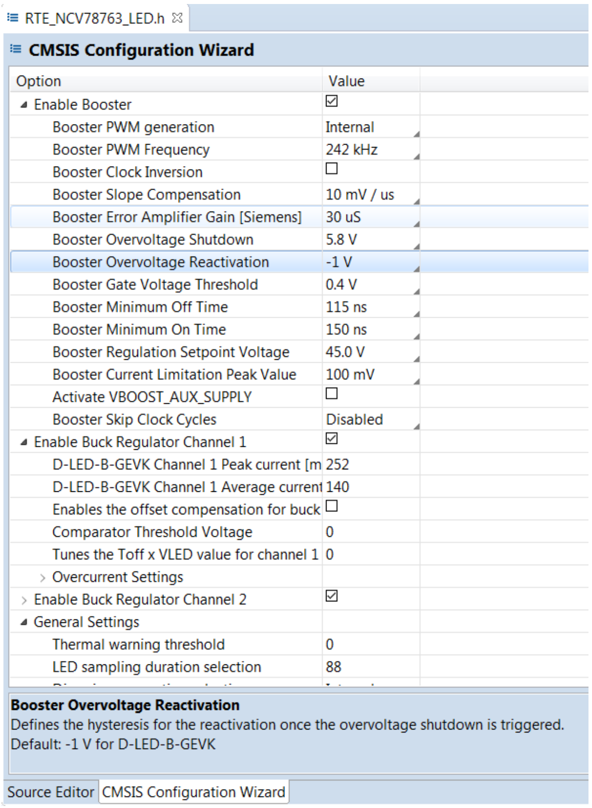
RTE_x.h

In addition to configuring system settings, all the supported daughter cards' parameters can be configured directly using the configuration wizard, without the need for programming. Once the parameters are changed per the application requirements, saving, rebuilding and flashing the project will let the new parameters take effect. Examples for the stepper and LED ballast daughter cards are shown below. Other daughter cards can be configured in a similar fashion.



Option	Value
Stepper Shield Left Channel	
Step Mode	1 / 4 Micro - Step
Coil Peak Current	245 mA
Direction Of Rotation	CW motion
NXT Edge Trigger	Rising Edge
Turn On / Off Slopes of Motor Driver	Very Fast
Speed Load Angle Transparency Bit	SLA is not transparent
Speed Load Angle Gain	0.5
Enables doubling of the PWM frequency	□
Enables jittery PWM	□
Steps Per Revolution	200
Stepper Shield Right Channel	
Step Mode	1 / 4 Micro - Step
Coil Peak Current	
Direction Of Rotation	
NXT Edge Trigger	
Turn On / Off Slopes of Motor Driver	
Speed Load Angle Transparency Bit	
Speed Load Angle Gain	
Enables doubling of the PWM frequency	
Enables jittery PWM	
Steps Per Revolution	

Step Mode
Default: 1 / 4 Micro - Step (for motors provided with Stepper shield)

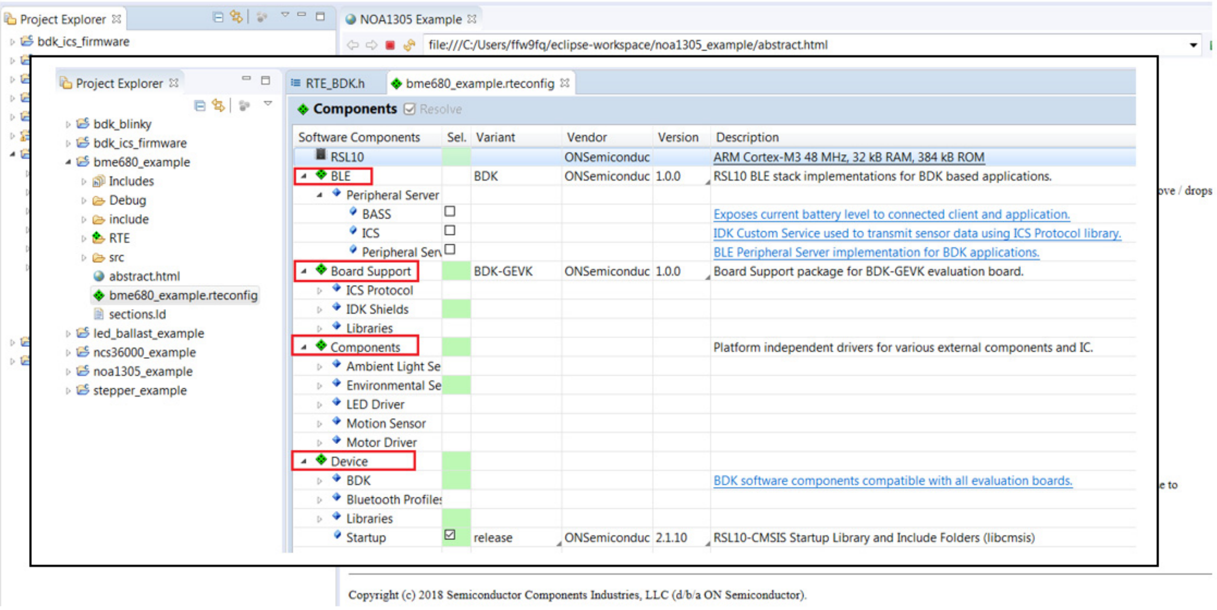


Option	Value
Enable Booster	☑
Booster PWM generation	Internal
Booster PWM Frequency	242 kHz
Booster Clock Inversion	□
Booster Slope Compensation	10 mV / us
Booster Error Amplifier Gain [Siemens]	30 uS
Booster Overvoltage Shutdown	5.8 V
Booster Overvoltage Reactivation	-1 V
Booster Gate Voltage Threshold	0.4 V
Booster Minimum Off Time	115 ns
Booster Minimum On Time	150 ns
Booster Regulation Setpoint Voltage	45.0 V
Booster Current Limitation Peak Value	100 mV
Activate VBOOST_AUX_SUPPLY	□
Booster Skip Clock Cycles	Disabled
Enable Buck Regulator Channel 1	☑
D-LED-B-GEVK Channel 1 Peak current [m	252
D-LED-B-GEVK Channel 1 Average current	140
Enables the offset compensation for buck	□
Comparator Threshold Voltage	0
Tunes the Toff x VLED value for channel 1	0
Overcurrent Settings	
Enable Buck Regulator Channel 2	☑
General Settings	
Thermal warning threshold	0
LED sampling duration selection	88

Booster Overvoltage Reactivation
Defines the hysteresis for the reactivation once the overvoltage shutdown is triggered.
Default: -1 V for D-LED-B-GEVK

DOCUMENTATION

Detailed documentation of all functions, code, APIs, HALs is part of the CMSIS package. Every use case (for a particular daughter card, service, etc.) copied into the workspace has its own manual with key description in the abstract.html page. URL Information and orderable part numbers are also provided as shown below.

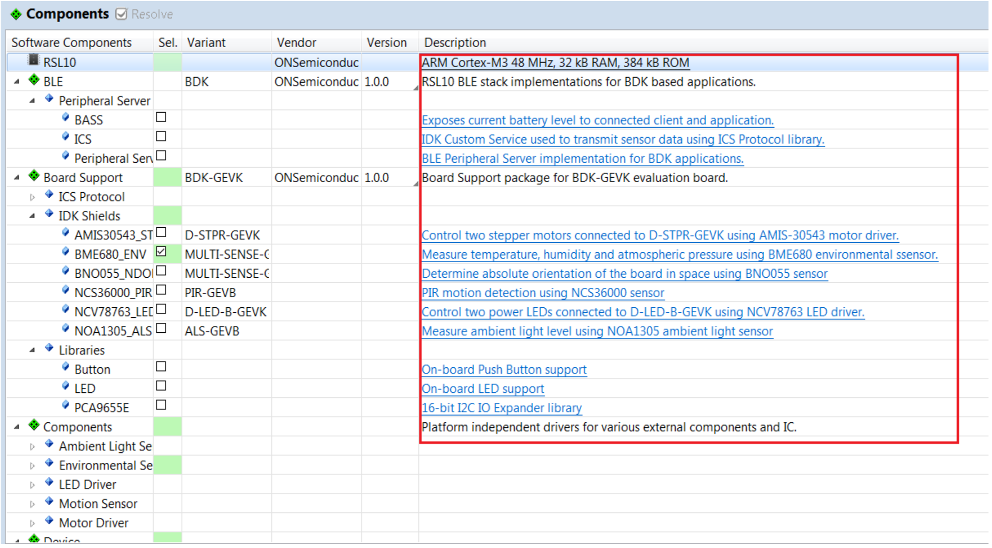


Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 32 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.0.0	RSL10 BLE stack implementations for BDK based applications.
Peripheral Server					
BASS	□				Exposes current battery level to connected client and application.
ICS	□				IDK Custom Service used to transmit sensor data using ICS Protocol library.
Peripheral Serv	□				BLE Peripheral Server implementation for BDK applications.
Board Support		BDK-GEVK	ONSemiconduc	1.0.0	Board Support package for BDK-GEVK evaluation board.
ICS Protocol					
IDK Shields					
Libraries					
Components					Platform independent drivers for various external components and IC.
Ambient Light Se					
Environmental Se					
LED Driver					
Motion Sensor					
Motor Driver					
Device					
BDK					BDK software components compatible with all evaluation boards.
Bluetooth Profiles					
Libraries					
Startup	☑	release	ONSemiconduc	2.1.10	RSL10-CMSIS Startup Library and Include Folders (libcmsis)

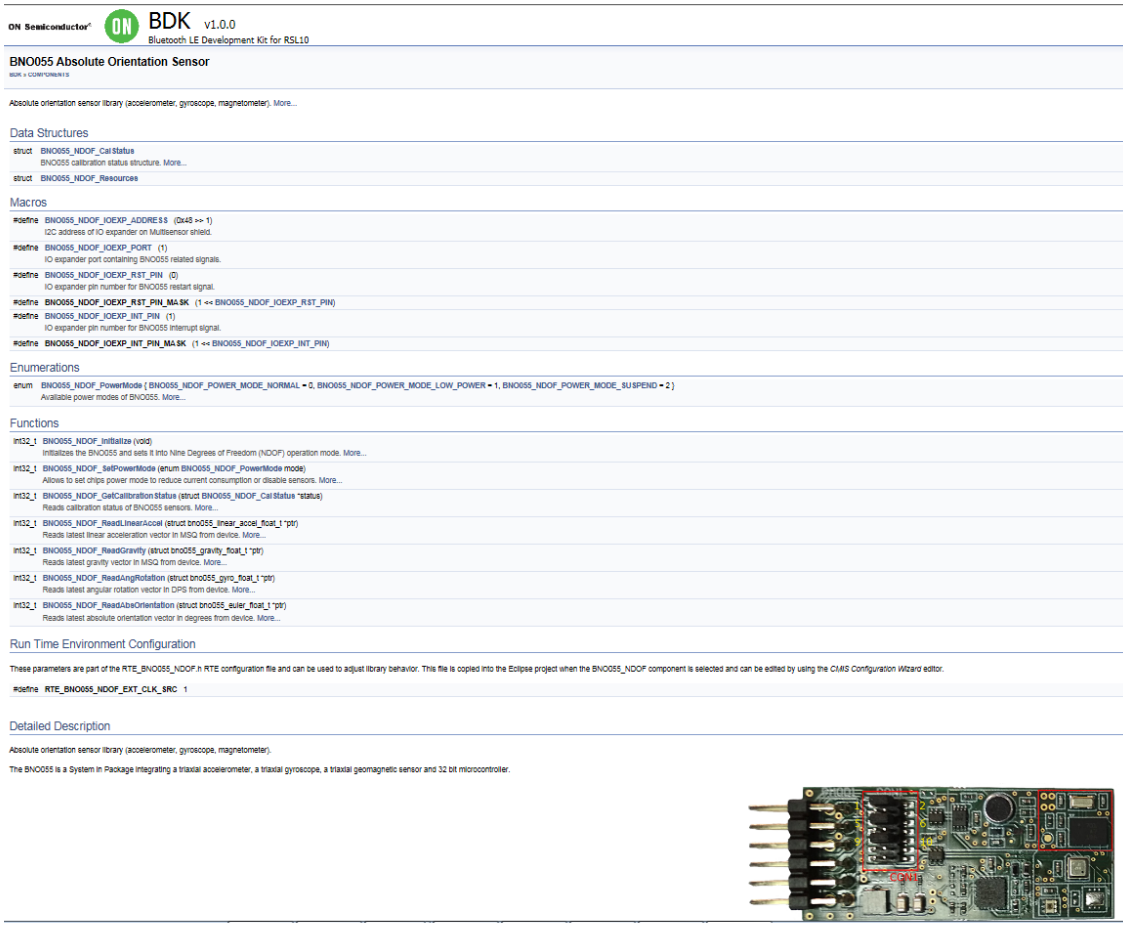
Copyright (c) 2018 Semiconductor Components Industries, LLC (d/b/a ON Semiconductor).

*.rteconfig

The *.rteconfig file lists the software components within the CMSIS pack as described in the B_IDK CMSIS Software Organization section. To access the components, double click *.rteconfig file. Extensive help is provided under the description tab.



Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 32 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.0.0	RSL10 BLE stack implementations for BDK based applications.
Peripheral Server					
BASS	□				Exposes current battery level to connected client and application.
ICS	□				IDK Custom Service used to transmit sensor data using ICS Protocol library.
Peripheral Serv	□				BLE Peripheral Server implementation for BDK applications.
Board Support		BDK-GEVK	ONSemiconduc	1.0.0	Board Support package for BDK-GEVK evaluation board.
ICS Protocol					
IDK Shields					
AMIS30543_ST	□	D-STPR-GEVK			Control two stepper motors connected to D-STPR-GEVK using AMIS-30543 motor driver.
BME680_ENV	☑	MULTI-SENSE-(			Measure temperature, humidity and atmospheric pressure using BME680 environmental ssensor.
BNO055_NDOI	□	MULTI-SENSE-(			Determine absolute orientation of the board in space using BNO055 sensor
NCS36000_PIR	□	PIR-GEVB			PIR motion detection using NCS36000 sensor
NCV78763_LEI	□	D-LED-B-GEVK			Control two power LEDs connected to D-LED-B-GEVK using NCV78763 LED driver.
NOA1305_ALS	□	ALS-GEVB			Measure ambient light level using NOA1305 ambient light sensor
Libraries					
Button	□				On-board Push Button support
LED	□				On-board LED support
PCA9655E	□				16-bit I2C IO Expander library
Components					Platform independent drivers for various external components and IC.
Ambient Light Se					
Environmental Se					
LED Driver					
Motion Sensor					
Motor Driver					
Device					



ON Semiconductor® BDK v1.0.0

Bluetooth LE Development Kit for RSL10

BNO055 Absolute Orientation Sensor

BDK » COMPONENTS

Absolute orientation sensor library (accelerometer, gyroscope, magnetometer). More...

Data Structures

```
struct  BNO055_NDOF_CalStatus
        BNO055 calibration status structure. More...
struct  BNO055_NDOF_Resources
```

Macros

```
#define  BNO055_NDOF_IOEXP_ADDRESS  (0x48 >> 1)
         I2C address of IO expander on Multisensor shield.
#define  BNO055_NDOF_IOEXP_PORT  (1)
         IO expander port containing BNO055 related signals.
#define  BNO055_NDOF_IOEXP_RST_PIN  (0)
         IO expander pin number for BNO055 restart signal.
#define  BNO055_NDOF_IOEXP_RST_PIN_MASK  (1 << BNO055_NDOF_IOEXP_RST_PIN)
#define  BNO055_NDOF_IOEXP_INT_PIN  (1)
         IO expander pin number for BNO055 interrupt signal.
#define  BNO055_NDOF_IOEXP_INT_PIN_MASK  (1 << BNO055_NDOF_IOEXP_INT_PIN)
```

Enumerations

```
enum  BNO055_NDOF_PowerMode { BNO055_NDOF_POWER_MODE_NORMAL = 0, BNO055_NDOF_POWER_MODE_LOW_POWER = 1, BNO055_NDOF_POWER_MODE_SUSPEND = 2 }
      Available power modes of BNO055. More...
```

Functions

```
int32_t  BNO055_NDOF_Initialize (void)
         Initializes the BNO055 and sets it into Nine Degrees of Freedom (NDOF) operation mode. More...
int32_t  BNO055_NDOF_SetPowerMode (enum BNO055_NDOF_PowerMode mode)
         Allows to set chips power mode to reduce current consumption or disable sensors. More...
int32_t  BNO055_NDOF_GetCalibrationStatus (struct BNO055_NDOF_CalStatus *status)
         Reads calibration status of BNO055 sensors. More...
int32_t  BNO055_NDOF_ReadLinearAccel (struct bno055_linear_accel_float_t *ptr)
         Reads latest linear acceleration vector in MSQ from device. More...
int32_t  BNO055_NDOF_ReadGravity (struct bno055_gravity_float_t *ptr)
         Reads latest gravity vector in MSQ from device. More...
int32_t  BNO055_NDOF_ReadAngRotation (struct bno055_gyro_float_t *ptr)
         Reads latest angular rotation vector in DPS from device. More...
int32_t  BNO055_NDOF_ReadAbsOrientation (struct bno055_euler_float_t *ptr)
         Reads latest absolute orientation vector in degrees from device. More...
```

Run Time Environment Configuration

These parameters are part of the RTE_BNO055_NDOF.h RTE configuration file and can be used to adjust library behavior. This file is copied into the Eclipse project when the BNO055_NDOF component is selected and can be edited by using the *CMIS Configuration Wizard* editor.

```
#define  RTE_BNO055_NDOF_EXT_CLK_SRC  1
```

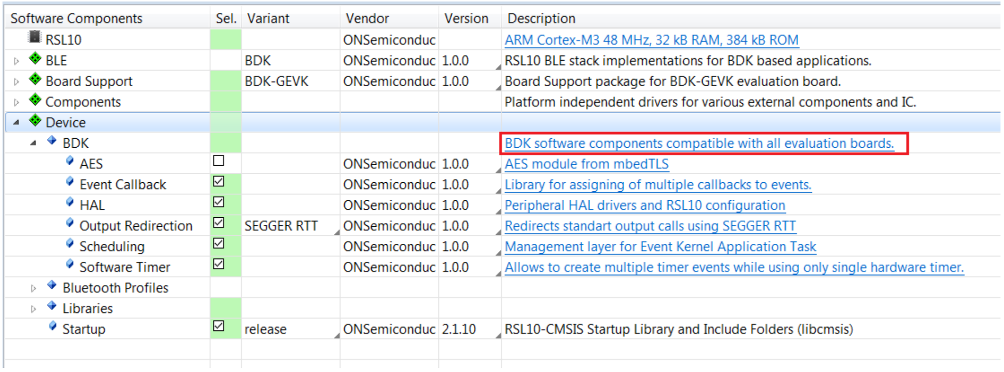
Detailed Description

Absolute orientation sensor library (accelerometer, gyroscope, magnetometer).

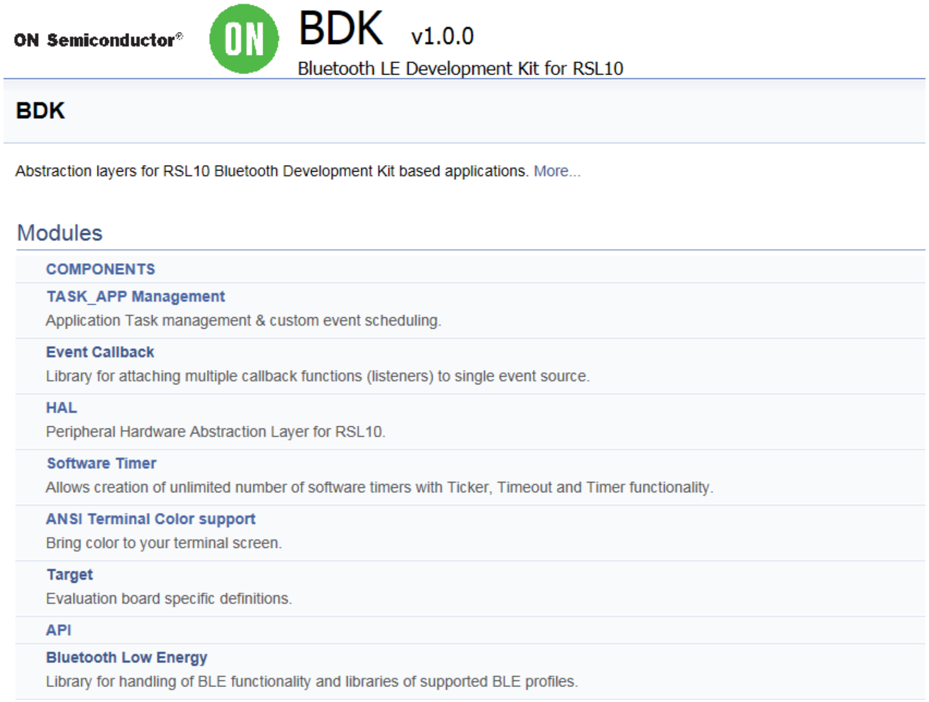
The BNO055 is a System in Package integrating a triaxial accelerometer, a triaxial gyroscope, a triaxial geomagnetic sensor and 32 bit microcontroller.

Main Help Page

The main help page is accessible via Device/BDK, visible for all use cases in *.rteconfig file. It’s further divided into various modules as shown below.



Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 32 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.0.0	RSL10 BLE stack implementations for BDK based applications.
Board Support		BDK-GEVK	ONSemiconduc	1.0.0	Board Support package for BDK-GEVK evaluation board.
Components					Platform independent drivers for various external components and IC.
Device					
BDK					BDK software components compatible with all evaluation boards.
AES	☐		ONSemiconduc	1.0.0	AES module from mbedTLS
Event Callback	☑		ONSemiconduc	1.0.0	Library for assigning of multiple callbacks to events.
HAL	☑		ONSemiconduc	1.0.0	Peripheral HAL drivers and RSL10 configuration
Output Redirection	☑	SEGGER RTT	ONSemiconduc	1.0.0	Redirects standart output calls using SEGGER RTT
Scheduling	☑		ONSemiconduc	1.0.0	Management layer for Event Kernel Application Task
Software Timer	☑		ONSemiconduc	1.0.0	Allows to create multiple timer events while using only single hardware timer.
Bluetooth Profiles					
Libraries					
Startup	☑	release	ONSemiconduc	2.1.10	RSL10-CMSIS Startup Library and Include Folders (libcmsis)



ON Semiconductor®

BDK v1.0.0

Bluetooth LE Development Kit for RSL10

BDK

Abstraction layers for RSL10 Bluetooth Development Kit based applications. More...

Modules

COMPONENTS

TASK_APP Management
Application Task management & custom event scheduling.

Event Callback
Library for attaching multiple callback functions (listeners) to single event source.

HAL
Peripheral Hardware Abstraction Layer for RSL10.

Software Timer
Allows creation of unlimited number of software timers with Ticker, Timeout and Timer functionality.

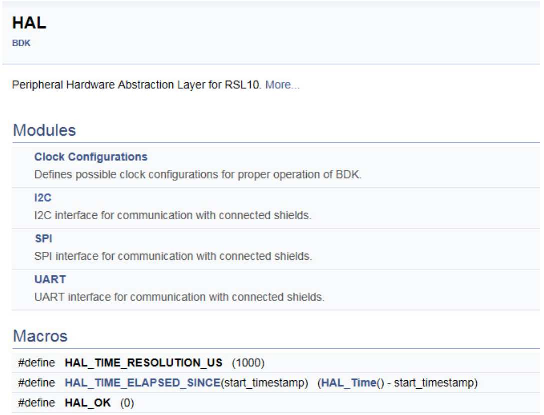
ANSI Terminal Color support
Bring color to your terminal screen.

Target
Evaluation board specific definitions.

API

Bluetooth Low Energy
Library for handling of BLE functionality and libraries of supported BLE profiles.

Sub−sections may be expanded for further information (Ex: HAL interfaces shown below)



HAL
BDK

Peripheral Hardware Abstraction Layer for RSL10. More...

Modules

Clock Configurations
Defines possible clock configurations for proper operation of BDK.

I2C
I2C interface for communication with connected shields.

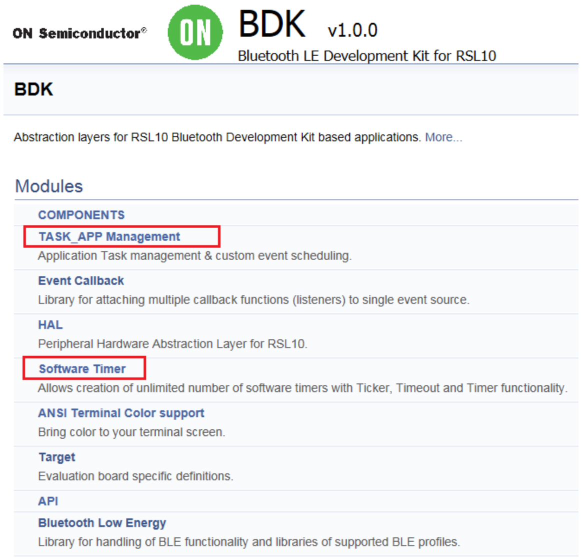
SPI
SPI interface for communication with connected shields.

UART
UART interface for communication with connected shields.

Macros

```
#define  HAL_TIME_RESOLUTION_US  (1000)
#define  HAL_TIME_ELAPSED_SINCE(start_timestamp)  (HAL_Time() - start_timestamp)
#define  HAL_OK  (0)
```

B−IDK also provides software timers and applications task manager abstraction layers to enable management of specific tasks and timing within the event kernel.



ON Semiconductor® BDK v1.0.0

Bluetooth LE Development Kit for RSL10

BDK

Abstraction layers for RSL10 Bluetooth Development Kit based applications. More...

Modules

COMPONENTS

TASK_APP Management
Application Task management & custom event scheduling.

Event Callback
Library for attaching multiple callback functions (listeners) to single event source.

HAL
Peripheral Hardware Abstraction Layer for RSL10.

Software Timer
Allows creation of unlimited number of software timers with Ticker, Timeout and Timer functionality.

ANSI Terminal Color support
Bring color to your terminal screen.

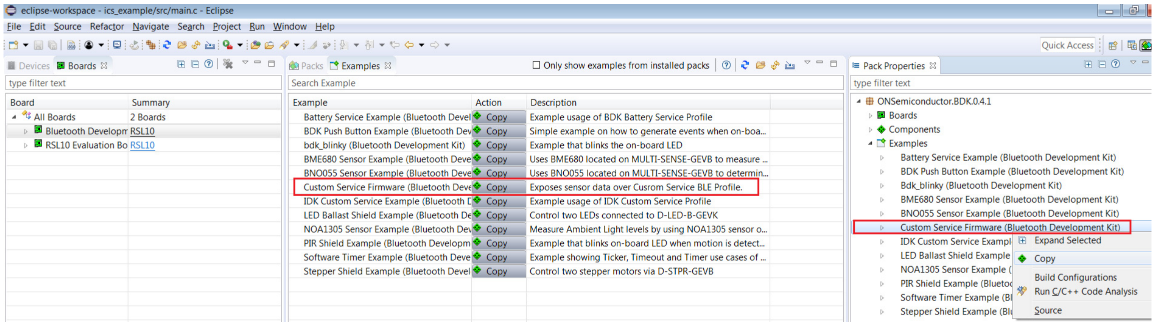
Target
Evaluation board specific definitions.

API

Bluetooth Low Energy
Library for handling of BLE functionality and libraries of supported BLE profiles.

Custom Service Firmware

In order to read sensor data and control actuators connected to the BDK−GEVK from the RSL10 Sense and Control mobile app, the Custom Service Firmware must be downloaded onto the BDK−GEVK. This firmware can be found as Custom Service Firmware under examples in the CMSIS pack.



eclipse-workspace - ics_example/src/main.c - Eclipse

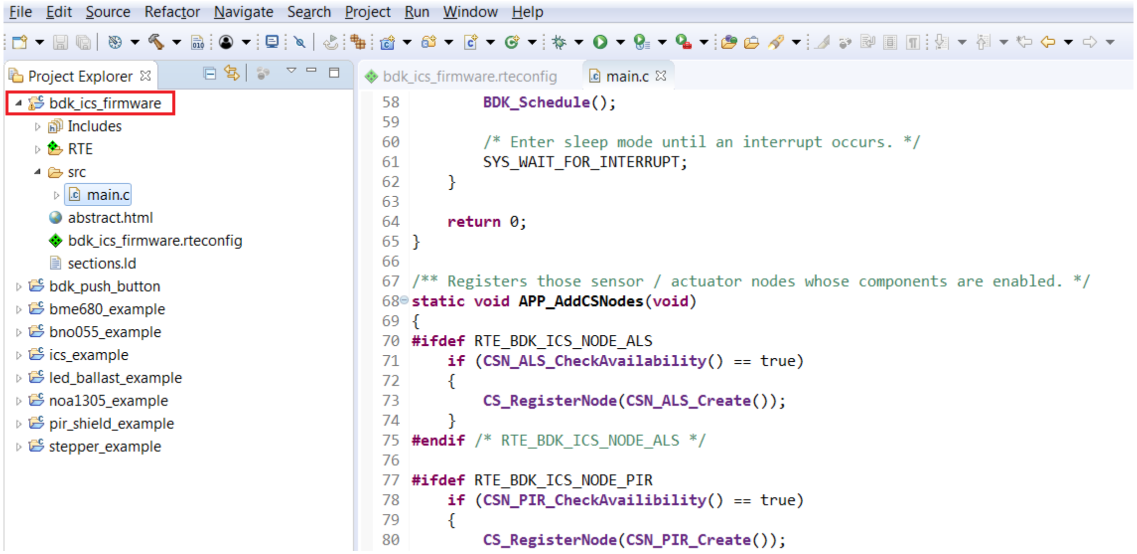
☐ Only show examples from installed packs

Board	Summary
All Boards	2 Boards
Bluetooth Developm	RSL10
RSL10 Evaluation Bo	RSL10

Example	Action	Description
Battery Service Example (Bluetooth Devel	Copy	Example usage of BDK Battery Service Profile
BDK Push Button Example (Bluetooth Dev	Copy	Simple example on how to generate events when on-boa...
bdk_blinky (Bluetooth Development Kit)	Copy	Example that blinks the on-board LED
BME680 Sensor Example (Bluetooth Deve	Copy	Uses BME680 located on MULTI-SENSE-GEVB to measure ...
BNO055 Sensor Example (Bluetooth Deve	Copy	Uses BNO055 located on MULTI-SENSE-GEVB to determin...
Custom Service Firmware (Bluetooth Deve	Copy	Exposes sensor data over Cusrom Service BLE Profile.
IDK Custom Service Example (Bluetooth D	Copy	Example usage of IDK Custom Service Profile
LED Ballast Shield Example (Bluetooth De	Copy	Control two LEDs connected to D-LED-B-GEVK
NOA1305 Sensor Example (Bluetooth Dev	Copy	Measure Ambient Light levels by using NOA1305 sensor o...
PIR Shield Example (Bluetooth Developm	Copy	Example that blinks on-board LED when motion is detect...
Software Timer Example (Bluetooth Deve	Copy	Example showing Ticker, Timeout and Timer use cases of ...
Stepper Shield Example (Bluetooth Devel	Copy	Control two stepper motors via D-STPR-GEVB

ONSemiconductor.BDK.0.4.1
- Boards
- Components
- Examples
  - Battery Service Example (Bluetooth Development Kit)
  - BDK Push Button Example (Bluetooth Development Kit)
  - Bdk_blinky (Bluetooth Development Kit)
  - BME680 Sensor Example (Bluetooth Development Kit)
  - BNO055 Sensor Example (Bluetooth Development Kit)
  - Custom Service Firmware (Bluetooth Development Kit)
  - IDK Custom Service Exampl
  - LED Ballast Shield Example
  - NOA1305 Sensor Example (
  - PIR Shield Example (Blueto
  - Software Timer Example (Bl
  - Stepper Shield Example (Bl

Expand Selected
Copy
Build Configurations
Run C/C++ Code Analysis
Source



File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
bdk_ics_firmware.rteconfig
main.c

- bdk_ics_firmware
  - Includes
  - RTE
  - src
    - main.c
  - abstract.html
  - bdk_ics_firmware.rteconfig
  - sections.ld
- bdk_push_button
- bme680_example
- bno055_example
- ics_example
- led_ballast_example
- noa1305_example
- pir_shield_example
- stepper_example

```
        BDK_Schedule();

        /* Enter sleep mode until an interrupt occurs. */
        SYS_WAIT_FOR_INTERRUPT;
    }

    return 0;
}

/** Registers those sensor / actuator nodes whose components are enabled. */
static void APP_AddCSNodes(void)
{
#ifdef RTE_BDK_ICS_NODE_ALS
    if (CSN_ALS_CheckAvailability() == true)
    {
        CS_RegisterNode(CSN_ALS_Create());
    }
#endif /* RTE_BDK_ICS_NODE_ALS */

#ifdef RTE_BDK_ICS_NODE_PIR
    if (CSN_PIR_CheckAvailibility() == true)
    {
        CS_RegisterNode(CSN_PIR_Create());
```

ON Semiconductor and the ON Semiconductor logo are trademarks of Semiconductor Components Industries, LLC dba ON Semiconductor or its subsidiaries in the United States and/or other countries. ON Semiconductor owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of ON Semiconductor's product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent−Marking.pdf. ON Semiconductor is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

The evaluation board/kit (research and development board/kit) (hereinafter the "board") is not a finished product and is as such not available for sale to consumers. The board is only intended for research, development, demonstration and evaluation purposes and should as such only be used in laboratory/development areas by persons with an engineering/technical training and familiar with the risks associated with handling electrical/mechanical components, systems and subsystems. This person assumes full responsibility/liability for proper and safe handling. Any other use, resale or redistribution for any other purpose is strictly prohibited.

The board is delivered "AS IS" and without warranty of any kind including, but not limited to, that the board is production−worthy, that the functions contained in the board will meet your requirements, or that the operation of the board will be uninterrupted or error free. ON Semiconductor expressly disclaims all warranties, express, implied or otherwise, including without limitation, warranties of fitness for a particular purpose and non−infringement of intellectual property rights.

ON Semiconductor reserves the right to make changes without further notice to any board.

You are responsible for determining whether the board will be suitable for your intended use or application or will achieve your intended results. Prior to using or distributing any systems that have been evaluated, designed or tested using the board, you agree to test and validate your design to confirm the functionality for your application. Any technical, applications or design information or advice, quality characterization, reliability data or other services provided by ON Semiconductor shall not constitute any representation or warranty by ON Semiconductor, and no additional obligations or liabilities shall arise from ON Semiconductor having provided such information or services.

The boards are not designed, intended, or authorized for use in life support systems, or any FDA Class 3 medical devices or medical devices with a similar or equivalent classification in a foreign jurisdiction, or any devices intended for implantation in the human body. Should you purchase or use the board for any such unintended or unauthorized application, you shall indemnify and hold ON Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that ON Semiconductor was negligent regarding the design or manufacture of the board.

This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and may not meet the technical requirements of these or other related directives.

FCC WARNING – This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by ON Semiconductor to be a finished end product fit for general consumer use. It may generate, use, or radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment may cause interference with radio communications, in which case the user shall be responsible, at its expense, to take whatever measures may be required to correct this interference.

ON Semiconductor does not convey any license under its patent rights nor the rights of others.

LIMITATIONS OF LIABILITY: ON Semiconductor shall not be liable for any special, consequential, incidental, indirect or punitive damages, including, but not limited to the costs of requalification, delay, loss of profits or goodwill, arising out of or in connection with the board, even if ON Semiconductor is advised of the possibility of such damages. In no event shall ON Semiconductor's aggregate liability from any obligation arising out of or in connection with the board, under any theory of liability, exceed the purchase price paid for the board, if any.

For more information and documentation, please visit www.onsemi.com.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Email Requests to: orderlit@onsemi.com

ON Semiconductor Website: www.onsemi.com

TECHNICAL SUPPORT
North American Technical Support:
Voice Mail: 1 800−282−9855 Toll Free USA/Canada
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:
Phone: 00421 33 790 2910
For additional information, please contact your local Sales Representative

LifeElectronics
Живое партнерство

ООО "ЛайфЭлектроникс" "LifeElectronics" LLC

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru