EVBUM2614

RSL10-SENSE-GEVK (and RSL10-SENSE-DB-GEVK) User Guide

ON Semiconductor®

www.onsemi.com

EVAL BOARD USER'S MANUAL

Introduction

The RSL10−SENSE−GEVK (and RSL10−SENSE−DB−GEVK) is a comprehensive, compact, node−to−cloud IoT sensor platform that allows development of various Bluetooth Low Energy based use cases. Along with the hardware and software, the RSL10−SENSE−GEVK includes a mobile app to interact with sensors and actuators. The board features RSL10, Industry's lowest power Bluetooth® 5 SoC and several sensors from ON Semiconductor and Bosch. By combining motion, environmental, ambient light sensing with the ultra−low power of the Bluetooth 5 Certified RSL10 and will enable customers to realize a new class of battery powered static, mobile and wearable smart sensors targeting consumer and industrial applications in the IoT.

The overall deep sleep consumption of 20 μA results in a battery life of over 1 year. For further increase in battery life, software configuration wizard allows flexible timing setup as discussed in the following sections.

Variants

There are two SKUs of the RSL10 Sensor kit. Both variants are pre−loaded with an ultra−low power firmware and include a 3 V CR2032 coin cell and a flexible NFC antenna.

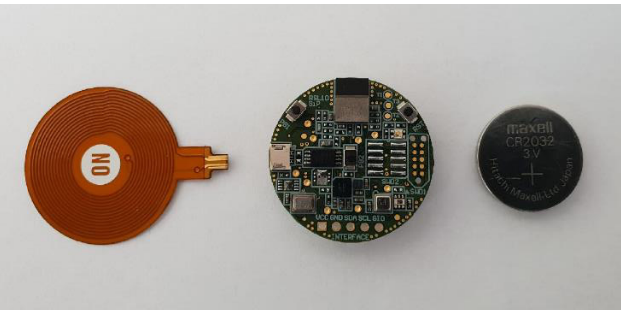


Figure 1.

RSL10−SENSE−GEVK: Firmware can be flashed via 10−pin needle adapter (e.g. TC2050 from Tag−Connect) (not included). The 10−pin header for debugger is not populated on the board.

RSL10−SENSE−DB−GEVK: The "debug" (−DB) version of the board also includes a low cost Segger debugger J−Link LITE CortexM and a USB cable. Users can directly debug/communicate/flash the firmware over the populated−pin header.

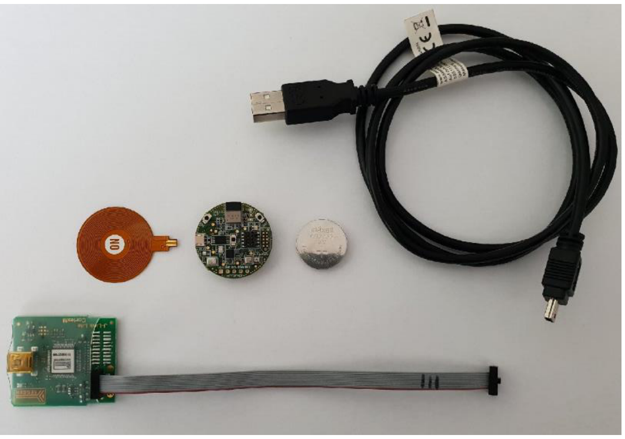


Figure 2.

Scope

The board starts functioning as soon as the coin cell is in place. This document covers the setup, software architecture, 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.

Default Configuration

In addition to the RSL10 SiP (System−in−Package), the following sensors are present on the board.

- NOA1305, ambient Light sensor
- N24RF64, NFC EEPROM
- BME680, environmental sensor (temperature, humidity, pressure, air quality)
- BHI160 + BMM150, 3−axis accelerometer, gyroscope, magnetometer. Together returnabsolute orientation supported in software
- INMP522 −> ultra−low power microphone for audio applications
- User can insert NFC flexible antenna into dedicated connector and bend underneath the battery holder for custom packaging / cases.

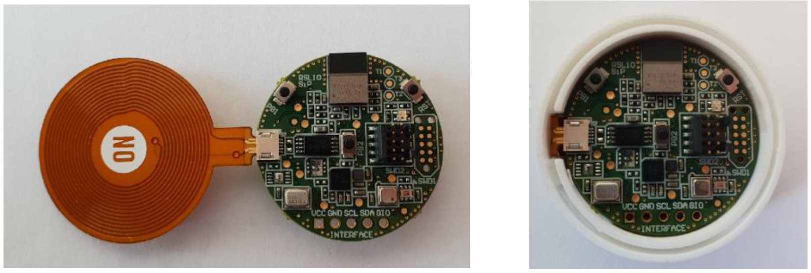


Figure 3.

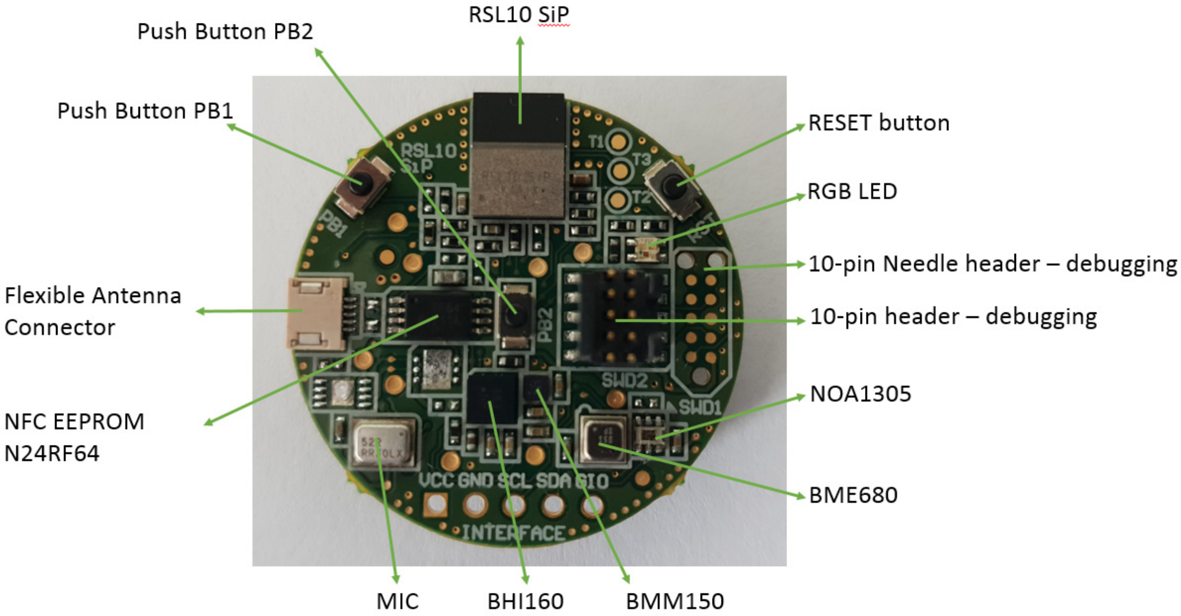


Figure 4.

Both kits are shipped with the ultra−low power firmware pre−loaded into the boards.

Powering the Board

To power RSL10−SENSE−GEVK, one has to insert CR2032 (3 V) battery into battery holder located on the bottom side of the board.

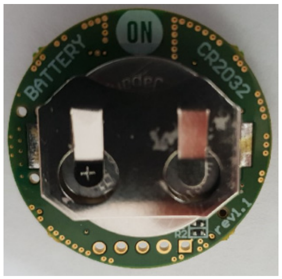


Figure 5.

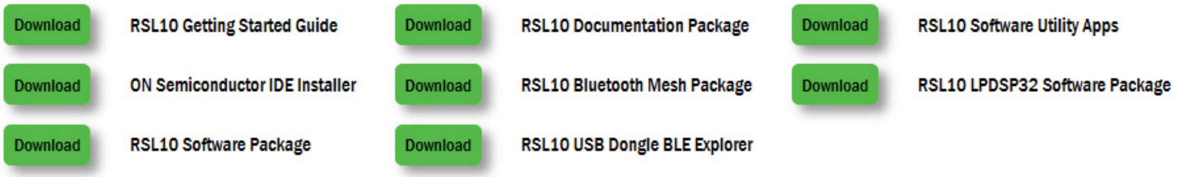
SOFTWARE

The RSL10−SENSE−GEVK boards are, by default, configured with the ultra−low power firmware. For users that want to download different firmware versions, this section details the involved steps.

Prerequisities

1. Install 64−bit version of Java from https://www.java.com/en/download/
2. Install J−Link Version 6.32i 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		

Figure 6.

4. Download the B−ID 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

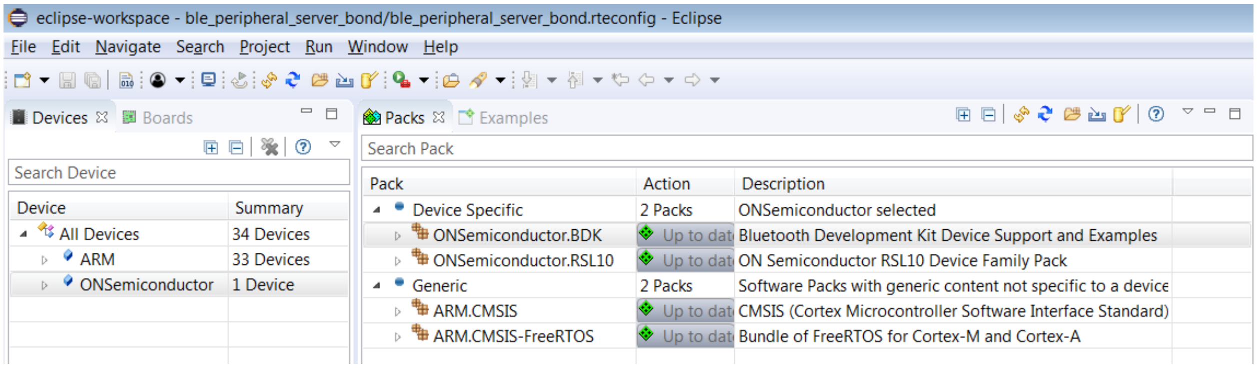
The next section provides details on importing the downloaded CMSIS packs into the SDK.

Importing CMSIS Packages

1. Launch the RSL10 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 CMSIS pack (step 3.a) in the Prerequisites section)

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 (Figure 7)



eclipse-workspace - ble_peripheral_server_bond/ble_peripheral_server_bond.rteconfig - Eclipse

Search Device

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

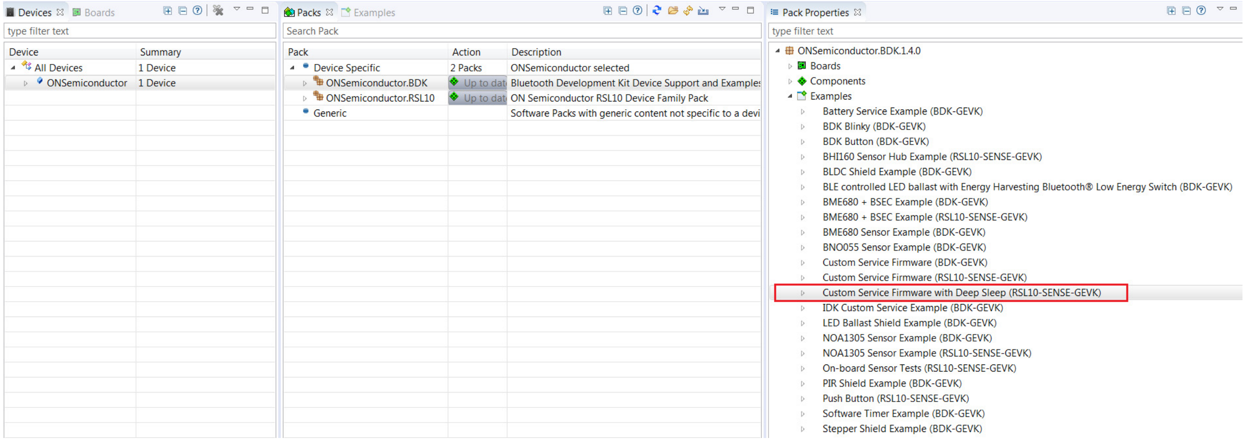
Search Pack

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

Figure 7.

Compiling and Flashing of Ultra Low Power Firmware

4. Examples related to RSL10−SENSE−GEVK are highlighted in brackets. Choose the example *Custom Service Firmware with Deep Sleep (RSL10−SENSE−GEVK)*



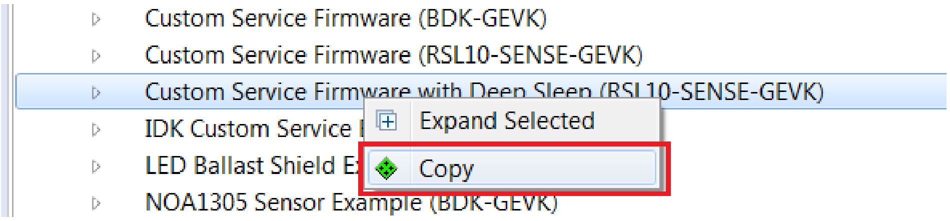
Device	Summary
All Devices	1 Device
ONSemiconductor	1 Device

Pack	Action	Description
Device Specific	2 Packs	ONSemiconductor selected
ONSemiconductor.BDK	Up to dat	Bluetooth Development Kit Device Support and Example
ONSemiconductor.RSL10	Up to dat	ON Semiconductor RSL10 Device Family Pack
Generic		Software Packs with generic content not specific to a devi

```
ONSemiconductor.BDK.1.4.0
  Boards
  Components
  Examples
    Battery Service Example (BDK-GEVK)
    BDK Blinky (BDK-GEVK)
    BDK Button (BDK-GEVK)
    BHI160 Sensor Hub Example (RSL10-SENSE-GEVK)
    BLDC Shield Example (BDK-GEVK)
    BLE controlled LED ballast with Energy Harvesting Bluetooth® Low Energy Switch (BDK-GEVK)
    BME680 + BSEC Example (BDK-GEVK)
    BME680 + BSEC Example (RSL10-SENSE-GEVK)
    BME680 Sensor Example (BDK-GEVK)
    BNO055 Sensor Example (BDK-GEVK)
    Custom Service Firmware (BDK-GEVK)
    Custom Service Firmware (RSL10-SENSE-GEVK)
    Custom Service Firmware with Deep Sleep (RSL10-SENSE-GEVK)
    IDK Custom Service Example (BDK-GEVK)
    LED Ballast Shield Example (BDK-GEVK)
    NOA1305 Sensor Example (BDK-GEVK)
    NOA1305 Sensor Example (RSL10-SENSE-GEVK)
    On-board Sensor Tests (RSL10-SENSE-GEVK)
    PIR Shield Example (BDK-GEVK)
    Push Button (RSL10-SENSE-GEVK)
    Software Timer Example (BDK-GEVK)
    Stepper Shield Example (BDK-GEVK)
```

Figure 8.

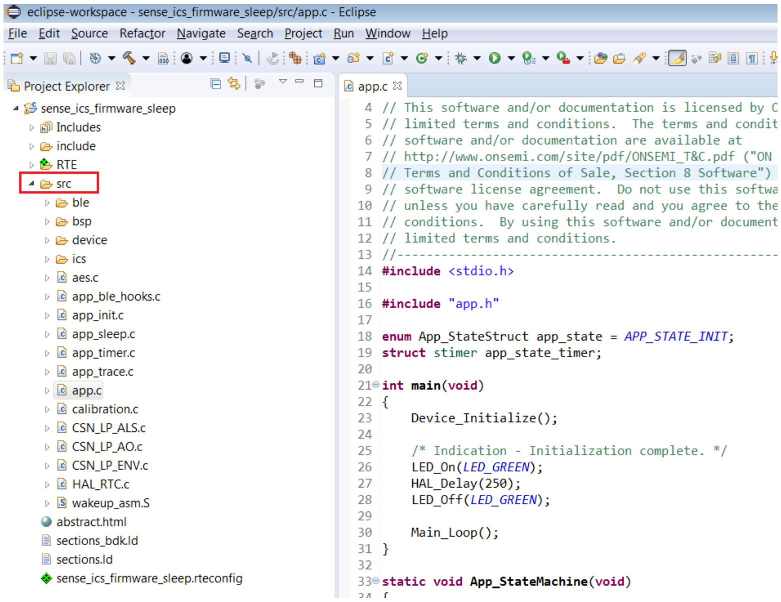
5. Right click and copy the project into workspace



Custom Service Firmware (BDK-GEVK)
Custom Service Firmware (RSL10-SENSE-GEVK)
Custom Service Firmware with Deep Sleep (RSL10-SENSE-GEVK)
IDK Custom Service E
LED Ballast Shield E
NOA1305 Sensor Example (BDK-GEVK)
Expand Selected
Copy

Figure 9.

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



eclipse-workspace - sense_ics_firmware_sleep/src/app.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app.c

```
sense_ics_firmware_sleep
  Includes
  include
  RTE
  src
    ble
    bsp
    device
    ics
    aes.c
    app_ble_hooks.c
    app_init.c
    app_sleep.c
    app_timer.c
    app_trace.c
    app.c
    calibration.c
    CSN_LP_ALS.c
    CSN_LP_AO.c
    CSN_LP_ENV.c
    HAL_RTC.c
    wakeup_asm.S
  abstract.html
  sections_bdk.ld
  sections.ld
  sense_ics_firmware_sleep.rteconfig
```

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

#include "app.h"

enum App_StateStruct app_state = APP_STATE_INIT;
struct stimer app_state_timer;

int main(void)
{
    Device_Initialize();

    /* Indication - Initialization complete. */
    LED_On(LED_GREEN);
    HAL_Delay(250);
    LED_Off(LED_GREEN);

    Main_Loop();
}

static void App_StateMachine(void)
{
```

Figure 10.

6. Now user has to build the project as this creates binaries to be flashed to RSL10−SENSE−GEVK. For the sensor board, there are two options:
   a.) 1 Debug
   b.) 2 Release – go to hammer icon inside IDE and click Release. Project is automatically build

Debug mode enables user to debug application over serial terminal connected to GPIO pin on expansion connector. It’s the option how to fine tune the sleep mode code. RSL10−SENSE−GEVK natively doesn’t support serial communication, only RTT over JTAG.

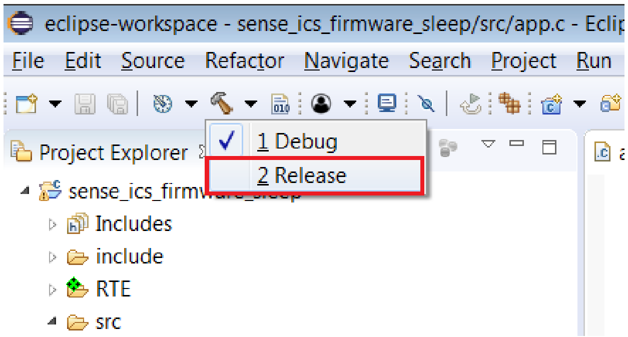
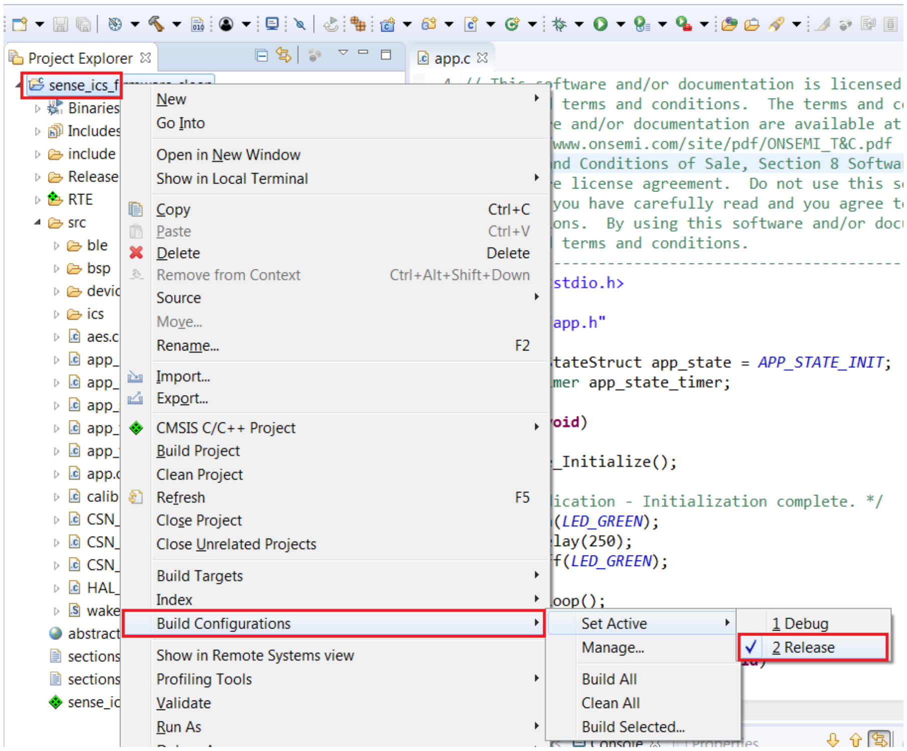

eclipse-workspace - sense_ics_firmware_sleep/src/app.c - Eclip
File Edit Source Refactor Navigate Search Project Run
Project Explorer
1 Debug
2 Release
sense_ics_firmw
Includes
include
RTE
src

Figure 11.

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

Alternatively you can build the project: right click on project under Project Explorer −> Build Configurations −> Set Active −> 2 Release



```
// This software and/or documentation is licensed
 terms and conditions.  The terms and c
e and/or documentation are available at
www.onsemi.com/site/pdf/ONSEMI_T&C.pdf
nd Conditions of Sale, Section 8 Softwa
e license agreement.  Do not use this s
you have carefully read and you agree t
ons.  By using this software and/or doc
 terms and conditions.
------------------------------------------
stdio.h>

app.h"

tateStruct app_state = APP_STATE_INIT;
mer app_state_timer;

oid)

_Initialize();

ication - Initialization complete. */
(LED_GREEN);
lay(250);
f(LED_GREEN);

oop();
```

Figure 12.

7. Once the build is done, the code is ready to be flashed to the RSL10−SENSE−GEVK.
   a.) Insert the battery into the board. Mandatory step as it creates the voltage reference for SWD logic signals.
   b.) Connect the low cost Debugger (RSL10−SENSE−DB−GEVK version) / 10−pin needle adapter with J−LINK (RSL10−SENSE−GEVK version)

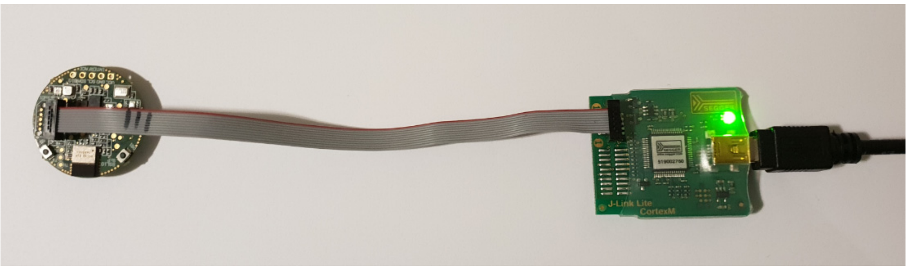
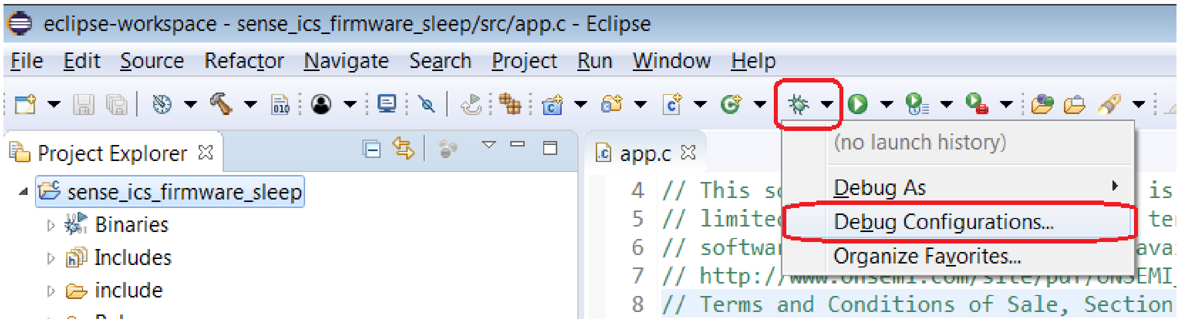


Figure 13.

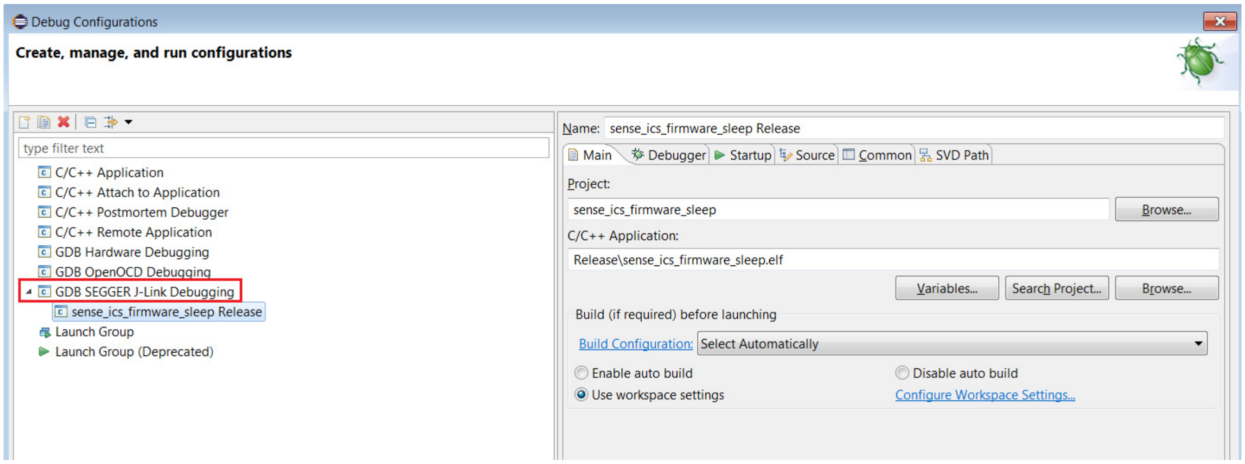
8. Select the project (sense_ics_firmware_sleep), and go to debug configurations as shown below.



```
// This sc                                         is
// limite                                          te
// softwa                                          ava
// http://www.onsemi.com/site/pdf/ONSEMI_
// Terms and Conditions of Sale, Section
```

Figure 14.

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



Name: sense_ics_firmware_sleep Release

Project: sense_ics_firmware_sleep

C/C++ Application: Release\sense_ics_firmware_sleep.elf

Build (if required) before launching

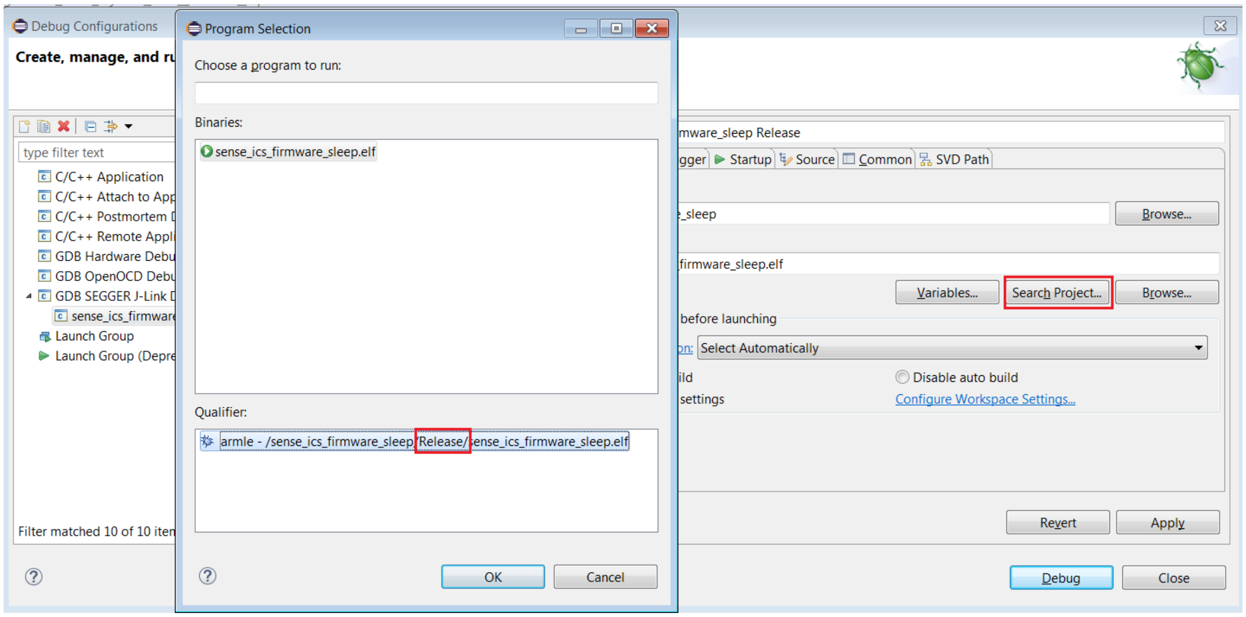
Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

Figure 15.

NOTE: The debug configuration for the selected example is automatically saved and there's no need to re−create it. Make sure you have the Release version of binary (.elf). Click on Search Project and Qualifier returns *Release* in the path. For debugging purposes you can build and switch Debug version as discussed in step 12 a/b.



Program Selection

Choose a program to run:

Binaries:

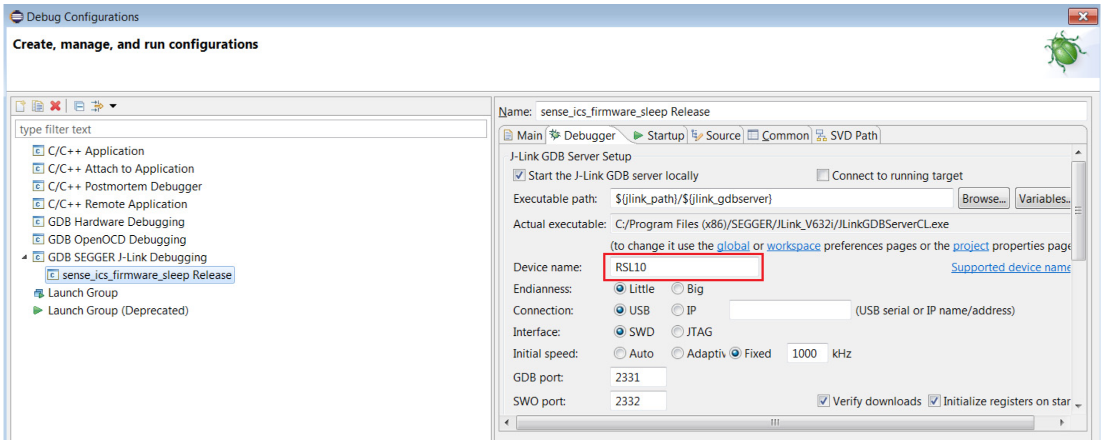
sense_ics_firmware_sleep.elf

Qualifier:

armle - /sense_ics_firmware_sleep/Release/sense_ics_firmware_sleep.elf

Figure 16.

b.) On the Debugger tab, set RSL10 as the device name.
Click Debug.



Name: sense_ics_firmware_sleep Release

J-Link GDB Server Setup

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

Executable path: ${jlink_path}/${jlink_gdbserver}

Actual executable: C:/Program Files (x86)/SEGGER/JLink_V632i/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

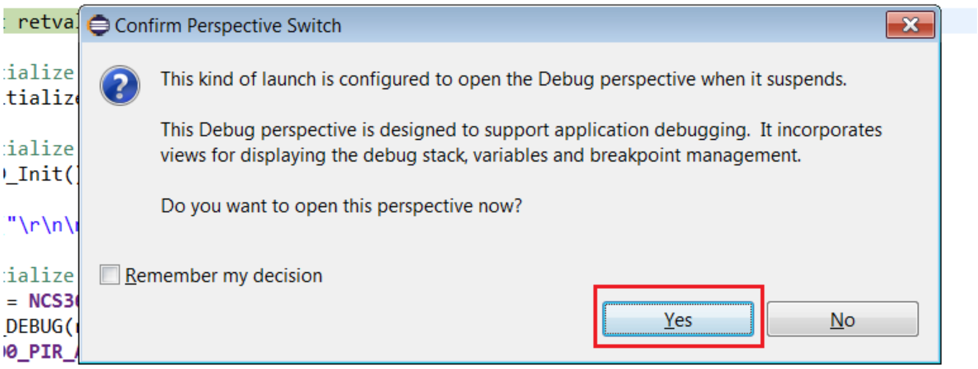
GDB port: 2331

SWO port: 2332

☑ Verify downloads ☑ Initialize registers on star

Figure 17.

9. 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.

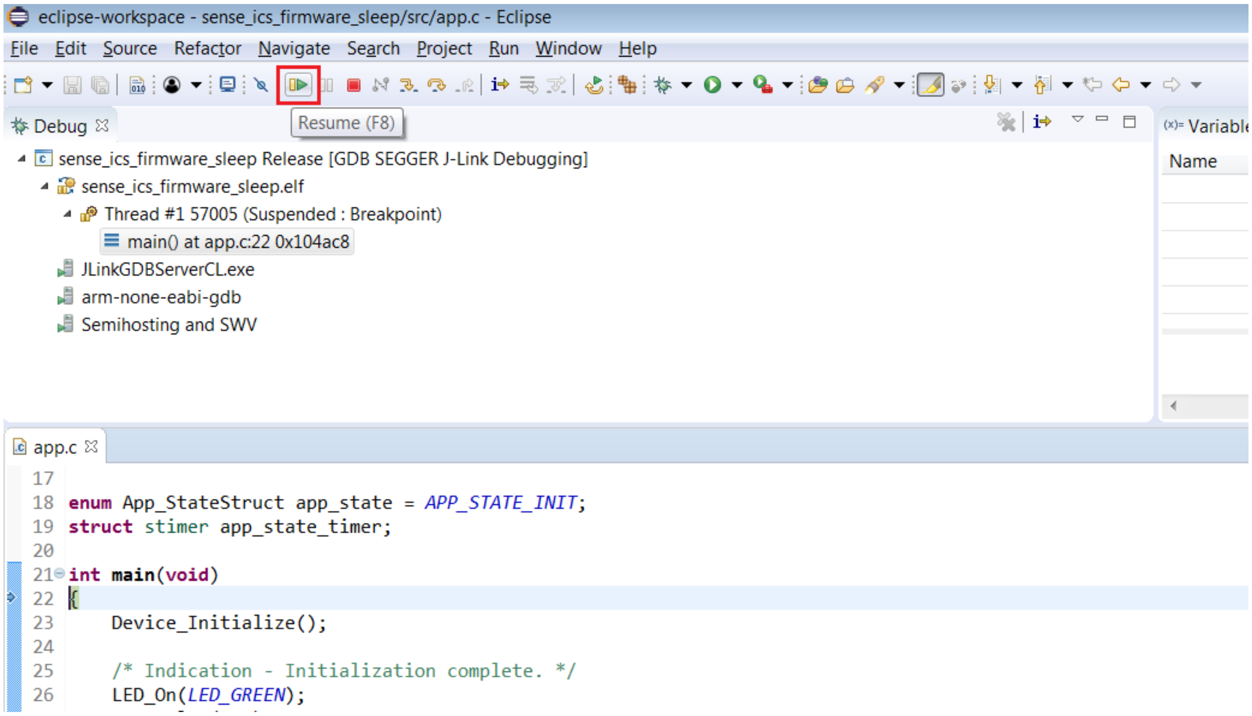
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

☐ Remember my decision

Figure 18.

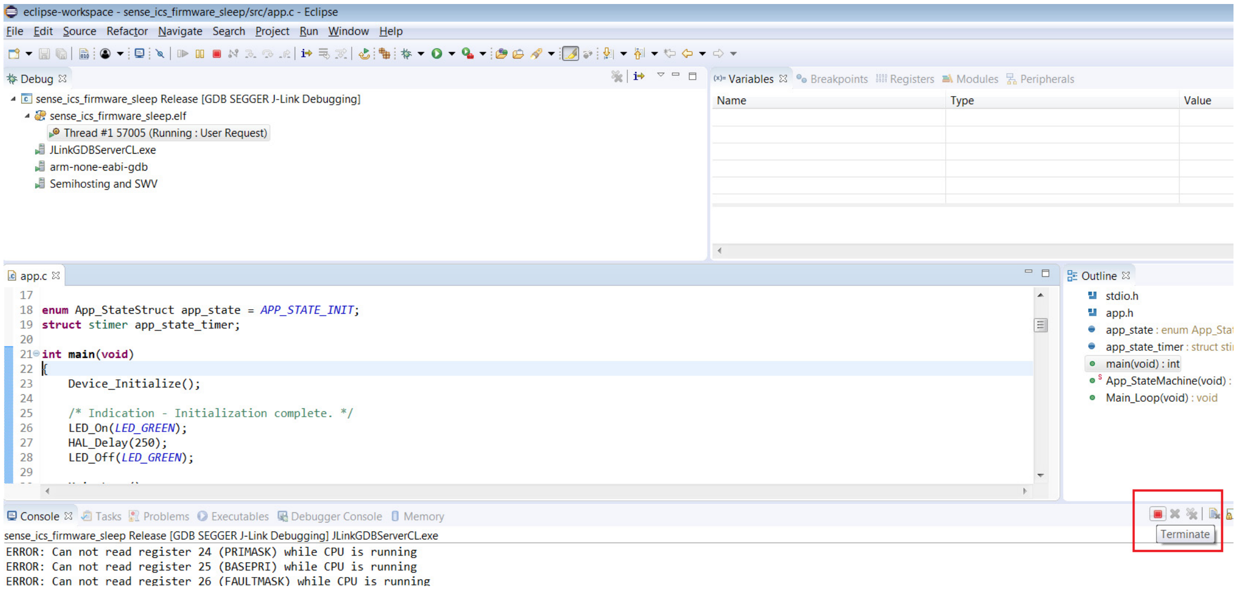
10. The debug session is now launched. Click Resume (F8) to start the target CPU. Green LED briefly flashes. By default, in Release version is no Logging option and terminal doesn't return useful data. By terminating the session, user closes connection with DBG server.


eclipse-workspace - sense_ics_firmware_sleep/src/app.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Resume (F8)
Debug
Variables
Name
app.c

sense_ics_firmware_sleep Release [GDB SEGGER J-Link Debugging]
sense_ics_firmware_sleep.elf
Thread #1 57005 (Suspended : Breakpoint)
main() at app.c:22 0x104ac8
JLinkGDBServerCL.exe
arm-none-eabi-gdb
Semihosting and SWV

```

enum App_StateStruct app_state = APP_STATE_INIT;
struct stimer app_state_timer;

int main(void)
{
    Device_Initialize();

    /* Indication - Initialization complete. */
    LED_On(LED_GREEN);
```

Figure 19.



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

```

enum App_StateStruct app_state = APP_STATE_INIT;
struct stimer app_state_timer;

int main(void)
{
    Device_Initialize();

    /* Indication - Initialization complete. */
    LED_On(LED_GREEN);
    HAL_Delay(250);
    LED_Off(LED_GREEN);

```

```
sense_ics_firmware_sleep Release [GDB SEGGER J-Link Debugging] JLinkGDBServerCL.exe
ERROR: Can not read register 24 (PRIMASK) while CPU is running
ERROR: Can not read register 25 (BASEPRI) while CPU is running
ERROR: Can not read register 26 (FAULTMASK) while CPU is running
```

Figure 20.

11. Disconnect debugger, download and open the mobile app available under store (Android and iOS). App name is *RSL10 Sense and Control* (www.onsemi.com/b−idk)

IMPORTANT NOTE:

When the board is flashed, Green LED shortly blinks. Board starts BLE advertising only and is visible on the mobile app. When connection with mobile app is not made for next 60s (by default), blue LED blinks and RSL10−SENSE−GEVK goes into deep sleep mode. You can resume operation by holding button PB1 for >1s. Green LED blinks and process repeats. See below the state diagram.

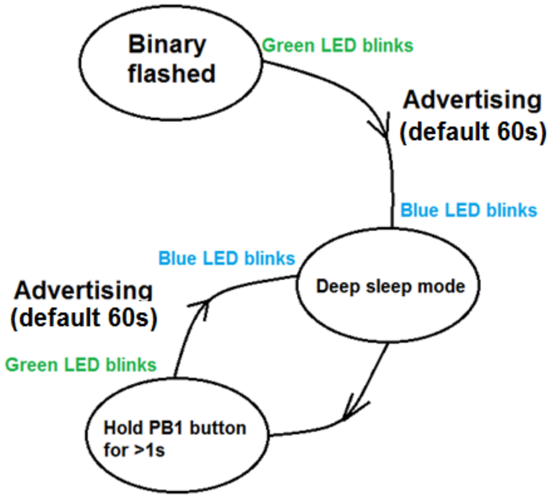


Figure 21.

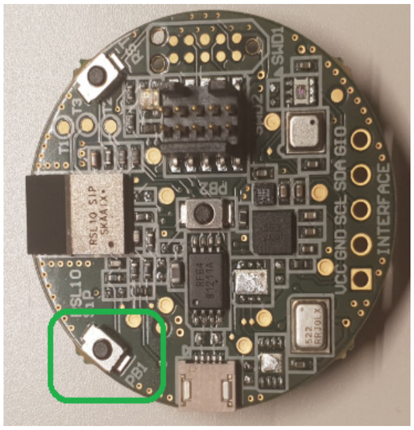
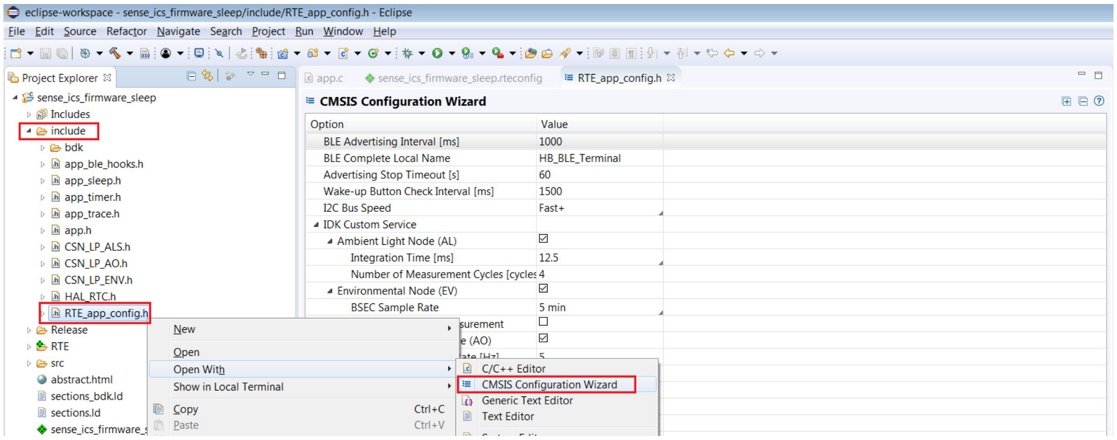


Figure 22.

12. User is exposed to set various parameters that have impact on battery longevity. Three main parameters can be configured in CMSIS:
    a.) BLE Advertising Interval (Default 1000 ms)
    b.) Advertising stop Timeout (Default 60 s)
    c.) Wake−up Button Check Interval (Default 1500 ms)

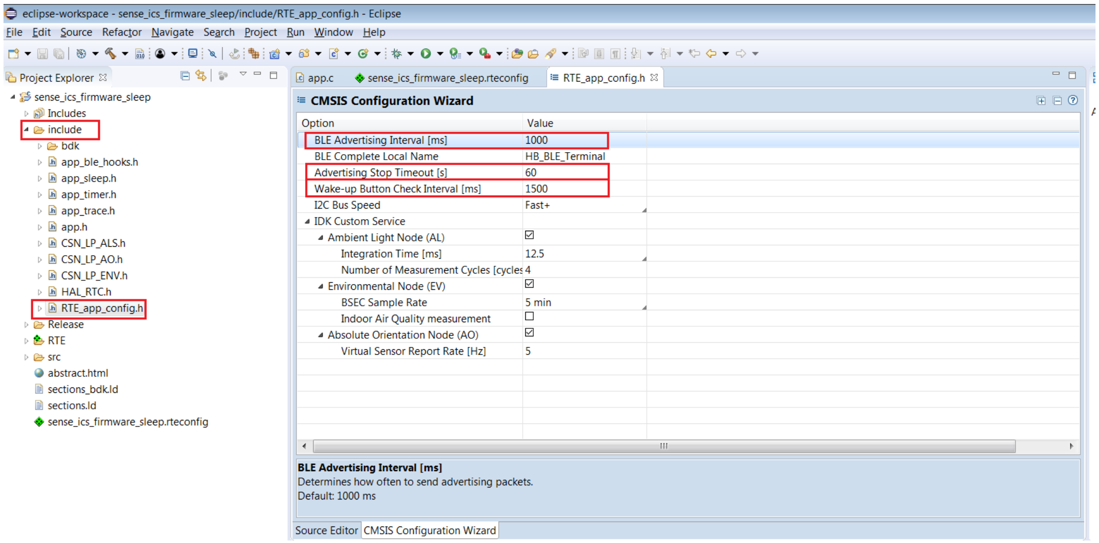
To get into the CMSIS Configuration Wizard, right click on RTE_app_config.h and open CMSIS Configuration Wizard. Change parameters, save the project and build it starting from step 12.



CMSIS Configuration Wizard

Option	Value
BLE Advertising Interval [ms]	1000
BLE Complete Local Name	HB_BLE_Terminal
Advertising Stop Timeout [s]	60
Wake-up Button Check Interval [ms]	1500
I2C Bus Speed	Fast+
IDK Custom Service	
Ambient Light Node (AL)	☑
Integration Time [ms]	12.5
Number of Measurement Cycles [cycles	4
Environmental Node (EV)	☑
BSEC Sample Rate	5 min
surement	☐
e (AO)	☑
ate [Hz]	5

Figure 23.


eclipse-workspace - sense_ics_firmware_sleep/include/RTE_app_config.h - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
sense_ics_firmware_sleep
Includes
include
bdk
app_ble_hooks.h
app_sleep.h
app_timer.h
app_trace.h
app.h
CSN_LP_ALS.h
CSN_LP_AO.h
CSN_LP_ENV.h
HAL_RTC.h
RTE_app_config.h
Release
RTE
src
abstract.html
sections_bdk.ld
sections.ld
sense_ics_firmware_sleep.rteconfig
app.c
sense_ics_firmware_sleep.rteconfig
RTE_app_config.h
CMSIS Configuration Wizard
Source Editor
CMSIS Configuration Wizard

Option	Value
BLE Advertising Interval [ms]	1000
BLE Complete Local Name	HB_BLE_Terminal
Advertising Stop Timeout [s]	60
Wake-up Button Check Interval [ms]	1500
I2C Bus Speed	Fast+
IDK Custom Service	
Ambient Light Node (AL)	☑
Integration Time [ms]	12.5
Number of Measurement Cycles [cycles	4
Environmental Node (EV)	☑
BSEC Sample Rate	5 min
Indoor Air Quality measurement	☐
Absolute Orientation Node (AO)	☑
Virtual Sensor Report Rate [Hz]	5

BLE Advertising Interval [ms]
Determines how often to send advertising packets.
Default: 1000 ms

Figure 24.

Mobile App Usage

13. Within the *Advertising Stop Timeout* interval, board is visible on the app screen. RSL10−SENSE−GEVK is advertising only over BLE. When multiple boards (sensor nodes) are present, each has unique MAC address and user selects the desired one –> HB_BLE_Terminal.

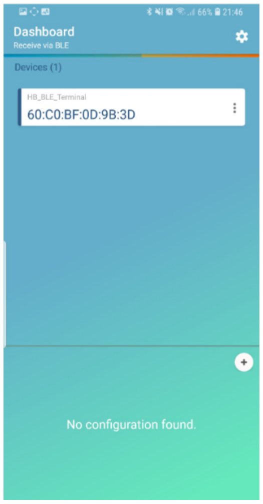
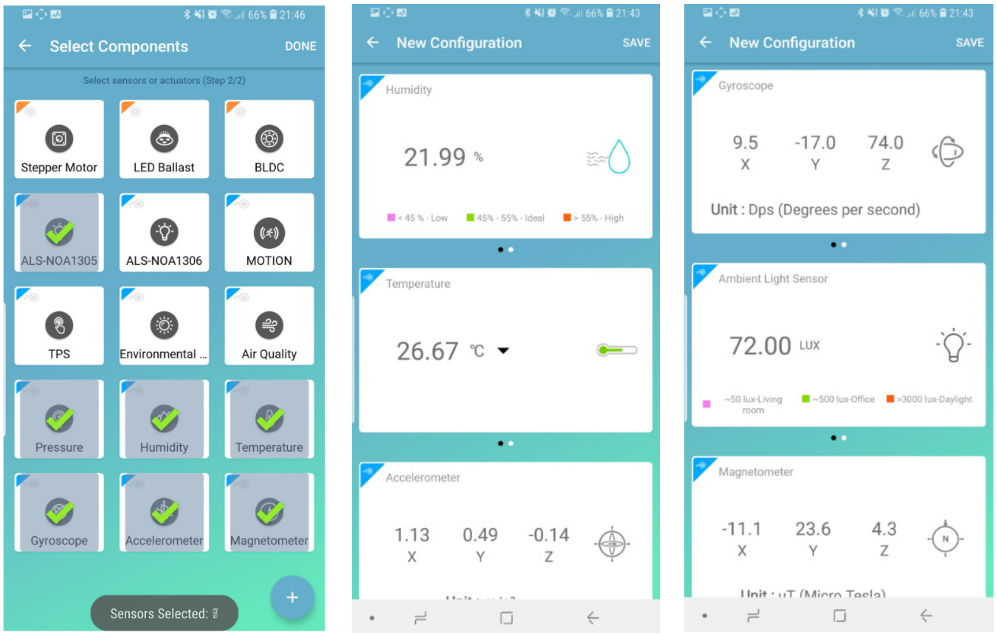


Figure 25.

14. When the appropriate board is selected, one can choose what sensor data to observe. Below are depicted all supported sensors and quantities taken. More simultaneous sensors in place equals more power required.

    RSL10−SENSE−GEVK supports also cloud connectivity via the same mobile app that functions as a gateway.

NOTE: Air quality is not supported in this low power mode example due to heating element and consequent need for higher power consumption. However it’s available under *BME680 + BSEC example* or *Custom Service Firmware* in CMSIS.



Select Components DONE

Select sensors or actuators (Step 2/2)

Stepper Motor · LED Ballast · BLDC · ALS-NOA1305 · ALS-NOA1306 · MOTION · TPS · Environmental ... · Air Quality · Pressure · Humidity · Temperature · Gyroscope · Accelerometer · Magnetometer

Sensors Selected: 7

New Configuration SAVE

Humidity: 21.99 %
🟪 < 45 % - Low 🟩 45% - 55% - Ideal 🟧 > 55% - High

Temperature: 26.67 °C

Accelerometer: 1.13 X, 0.49 Y, -0.14 Z

New Configuration SAVE

Gyroscope: 9.5 X, -17.0 Y, 74.0 Z
Unit : Dps (Degrees per second)

Ambient Light Sensor: 72.00 LUX
🟪 ~50 lux-Living room 🟩 ~500 lux-Office 🟧 >3000 lux-Daylight

Magnetometer: -11.1 X, 23.6 Y, 4.3 Z
Unit : µT (Micro Tesla)

Figure 26.

Ultra−low Power Firmware Modes

15. The following are the low power features of above described firmware:

- BDK libraries adapted for use with deep sleep mode of RSL10.
- HAL library for RTC and RTC based low power timer.
- Low power IDK Custom Service nodes for:
  a.) ALS (NOA1305)
  b.) Environmental sensing (BME680 + BSEC software)
  c.) Absolute Orientation (BHI160 + BMM150)
- Automatic on demand sensor activation.
- Automatically stops BLE advertising if no connection is made.
- BLE advertising can be restarted by holding push button (PB1).
- Configurable using RTE configuration header.

16. Environmental Sensing Node (BME680 + BSEC software):

- Provides two sample rates (every 3 seconds or every 5 minutes). By default, due to reduced power consumption, environmental sensors updates each 5 minutes.
- Option to disable IAQ measurement to save power.
- Long term average power consumption:
  a.) Outputs: Indoor Air Quality, Compensated Temperature, Compensated Humidity, Pressure
    i. Low Power mode – 900 µA (3 s sample rate),
    ii. Ultra−low power mode – 90 µA (5 min sample rate)
  b.) Outputs: Compensated Temperature, Compensated Humidity, Pressure
    i. < 5.2 µA (1 s sample rate)
- Gas sensor uses too much power and is not suitable for CR2032 battery powered systems. By default, this feature is disabled in ultra−low power firmware

17. Ambient Light Node (NOA1305):

- Sensor is activated only when ambient light value is requested by peer device.
- Power consumption depends on number of requests received from peer device.
  a.) ~80 µA current draw when sensor is active
- Sensor remains active for 4 measurement cycles (integration times) to stabilize sensor output.
- Integration time and number of cycles are configurable from RTE header.

18. BLE Connection Interval – Possible Power savings:

- BLE allows devices to negotiate connection parameter, most notably **Slave Connection Interval**
  a.) BLE communication always occurs at every connection interval even if the devices do not have anything to exchange (just send empty packets).
  b.) Configurable from 1.25 ms up to 4000 ms.
  c.) Bigger Slave Connection Interval −> Less energy consumed by HB.
  d.) Master (phone) has complete control over used connection interval.
- Android allows only 3 specific interval configurations:
  i. High (11.25 – 15 ms), Balanced (30 – 50 ms), Low Power (100 – 125 ms)
- All Android apps do not allow to set connection interval and always force balanced mode.

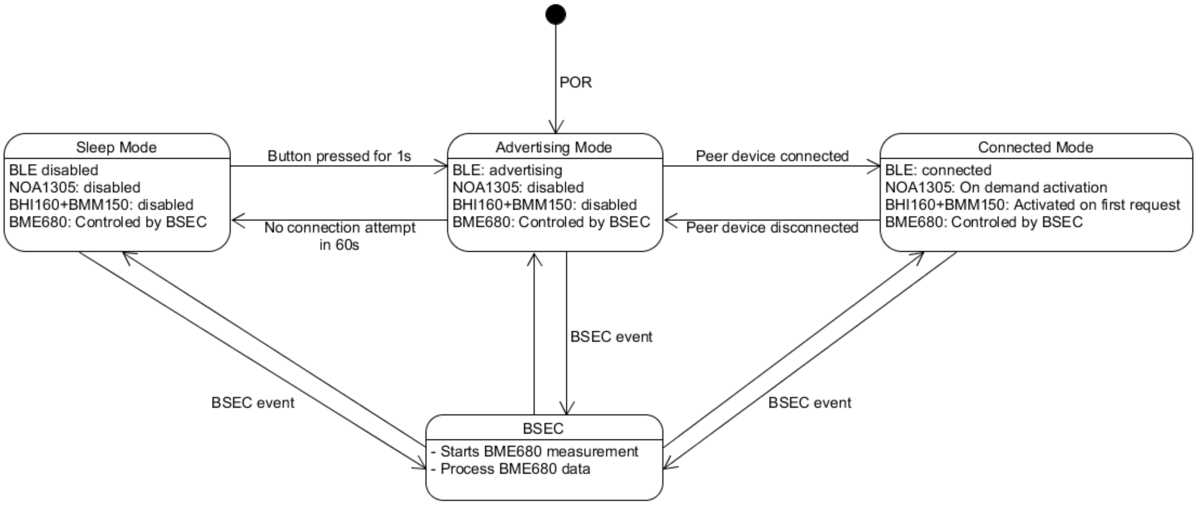


Figure 27.

19. Short term power consumption (100s interval):

- Deep Sleep mode:
  a.) 1.5 s periodic button check
  b.) Consumption: 18.7 uA @ 3 V
- Advertising mode:
  c.) 1 Hz BLE advertising interval
  d.) Consumption: 24.5 uA @ 3 V
- Connected mode:
  e.) Full operation connected to RSL10 Sense & Control:
    i. ALS: 1 s measurement & report rate
    ii. Absolute Orientation: 12.5 Hz measure rate, 1 s report rate
    iii. Environmental data (IAQ disabled): 5 min. measure rate, 3 s report rate
  f.) Consumption: ~3000 uA @ 3 V

20. Low power firmware – block diagram. The diagram in Figure 27 depicts detailed high level operation of ultra–low power firmware.

Compiling and Flashing of the Rest Examples Attached to RSL10−SENSE

In this section user is guided on how to flash software for all remaining examples in the CMSIS pack. The procedure is similar as for Ultra−Low power FW. Let's pick up *On−board Sensor Tests* that enables microphone functionality and returns sensors data into console.

21. Right click and copy the project into workspace

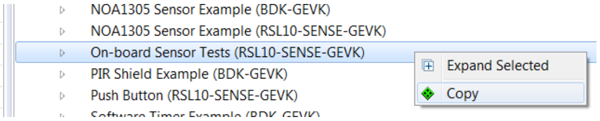

NOA1305 Sensor Example (BDK-GEVK)
NOA1305 Sensor Example (RSL10-SENSE-GEVK)
On-board Sensor Tests (RSL10-SENSE-GEVK)
PIR Shield Example (BDK-GEVK)
Push Button (RSL10-SENSE-GEVK)
Software Timer Example (BDK-GEVK)
Expand Selected
Copy

Figure 28.

22. Right click on the project and build it

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

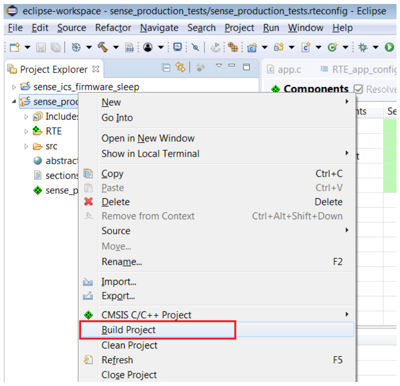


Figure 29.

23. Once the project is build, go to Debug configurations, double click on GDB SEGGER J–Link Debugging that automatically creates Session and import binaries ready to be flashed. Click on Debug button.

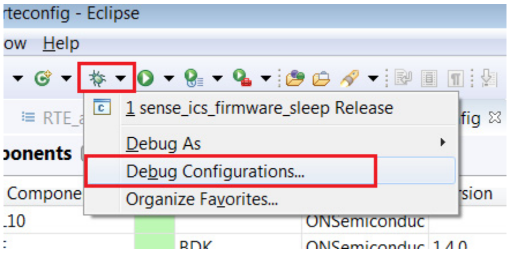
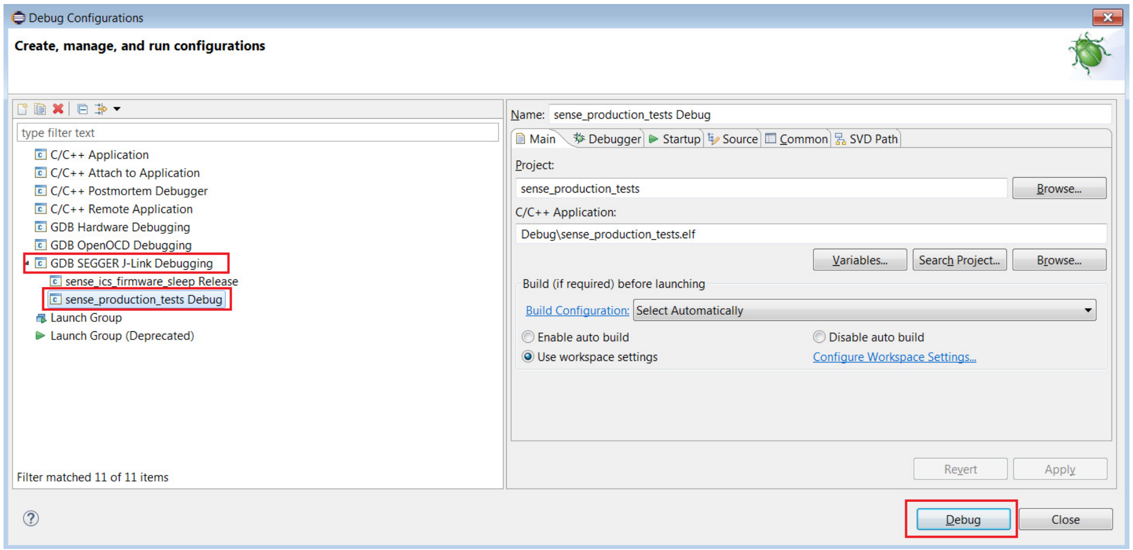


Figure 30.



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
  - sense_ics_firmware_sleep Release
  - sense_production_tests Debug
- Launch Group
- Launch Group (Deprecated)

Filter matched 11 of 11 items

Name: sense_production_tests Debug

Project: sense_production_tests

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

Build (if required) before launching

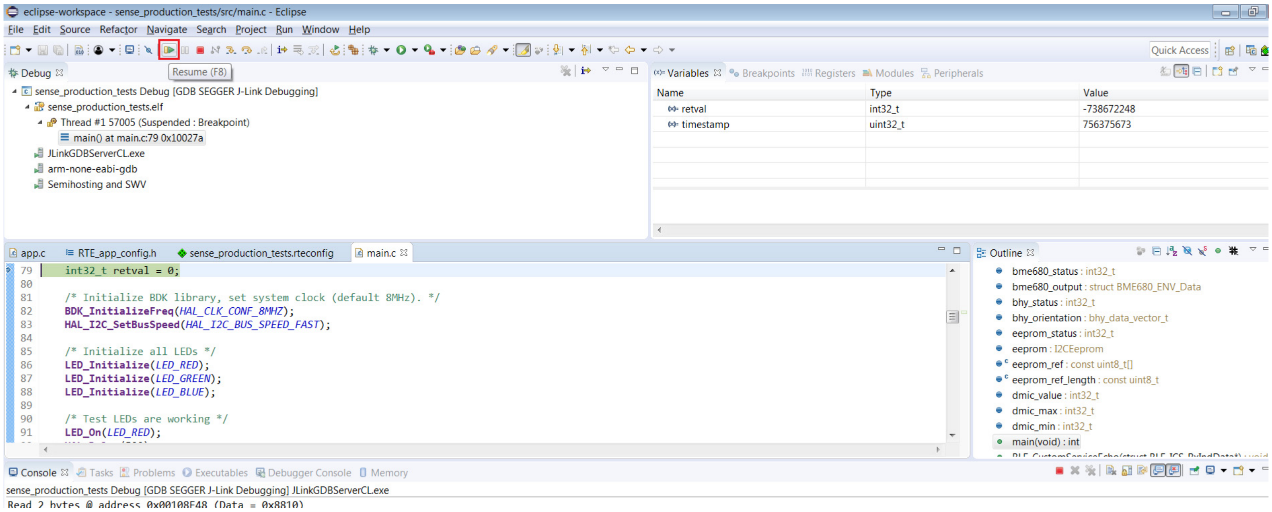
Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings

Figure 31.

24. When the debug session is launched, click on Resume button (F8).



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

- sense_production_tests Debug [GDB SEGGER J-Link Debugging]
  - sense_production_tests.elf
    - Thread #1 57005 (Suspended : Breakpoint)
      - main() at main.c:79 0x10027a
  - JLinkGDBServerCL.exe
  - arm-none-eabi-gdb
  - Semihosting and SWV

Name	Type	Value
retval	int32_t	-738672248
timestamp	uint32_t	756375673

```
    int32_t retval = 0;

    /* Initialize BDK library, set system clock (default 8MHz). */
    BDK_InitializeFreq(HAL_CLK_CONF_8MHZ);
    HAL_I2C_SetBusSpeed(HAL_I2C_BUS_SPEED_FAST);

    /* Initialize all LEDs */
    LED_Initialize(LED_RED);
    LED_Initialize(LED_GREEN);
    LED_Initialize(LED_BLUE);

    /* Test LEDs are working */
    LED_On(LED_RED);
```

- bme680_status : int32_t
- bme680_output : struct BME680_ENV_Data
- bhy_status : int32_t
- bhy_orientation : bhy_data_vector_t
- eeprom_status : int32_t
- eeprom : I2CEeprom
- eeprom_ref : const uint8_t[]
- eeprom_ref_length : const uint8_t
- dmic_value : int32_t
- dmic_max : int32_t
- dmic_min : int32_t
- main(void) : int

```
sense_production_tests Debug [GDB SEGGER J-Link Debugging] JLinkGDBServerCL.exe
Read 2 bytes @ address 0x00108F48 (Data = 0x8810)
```

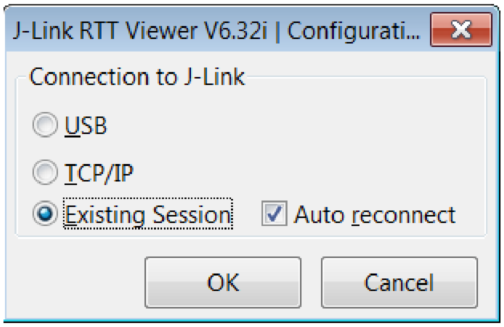
Figure 32.

Logging/Debugging

This is the next step after launching the session. For logging/debugging the downloaded Firmware, either J−Link RTT or Eclipse Console may be used. This section provides instructions for both.

Using J−Link RTT

25. After step 30 is done, open J−Link RTT viewer 6.32i (should be installed when J−Link software package was installed per Step 2)
26. Select USB / Existing session and click OK. As the debugger is in operation, easiest way is to utilize Existing Session.



J-Link RTT Viewer V6.32i | Configurati...

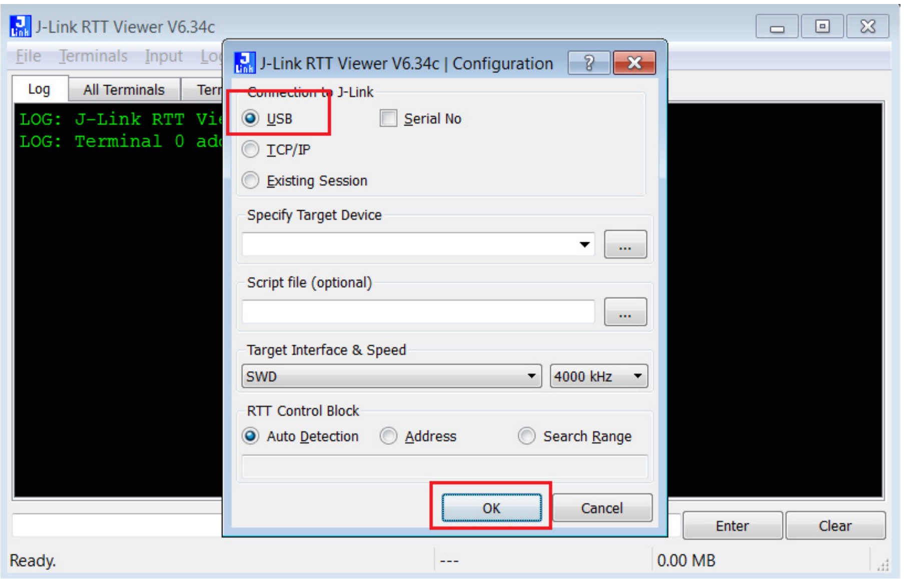
Connection to J-Link

○ USB

○ TCP/IP

● Existing Session ☑ Auto reconnect

Figure 33.



J-Link RTT Viewer V6.34c

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

J-Link RTT Viewer V6.34c | Configuration

Connection to J-Link

● USB □ Serial No

○ TCP/IP

○ Existing Session

Specify Target Device

Script file (optional)

Target Interface & Speed

SWD 4000 kHz

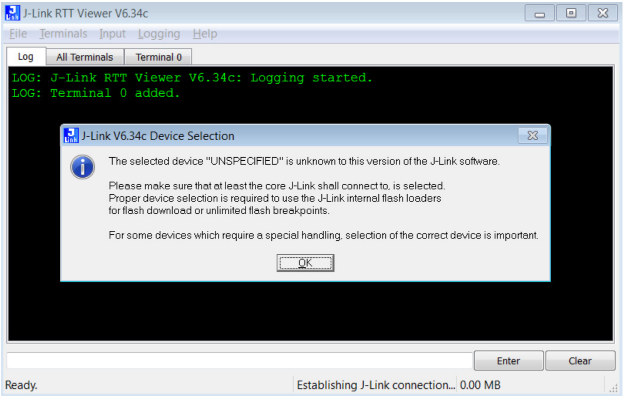
RTT Control Block

● Auto Detection ○ Address ○ Search Range

Ready. --- 0.00 MB

Figure 34.

27. 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

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

J-Link V6.34c Device Selection

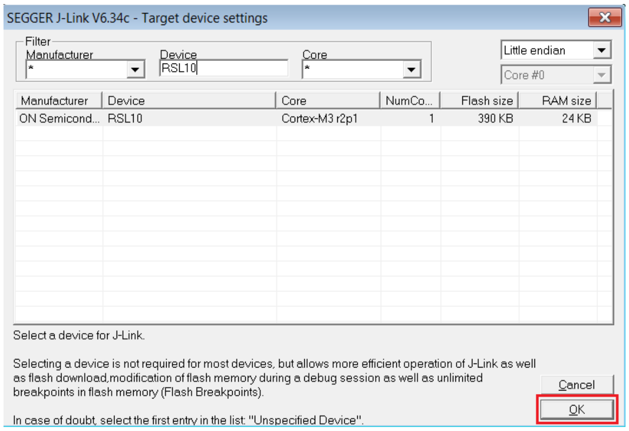
The selected device "UNSPECIFIED" is unknown to this version of the J-Link software.

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

Figure 35.



SEGGER J-Link V6.34c - Target device settings

Filter

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

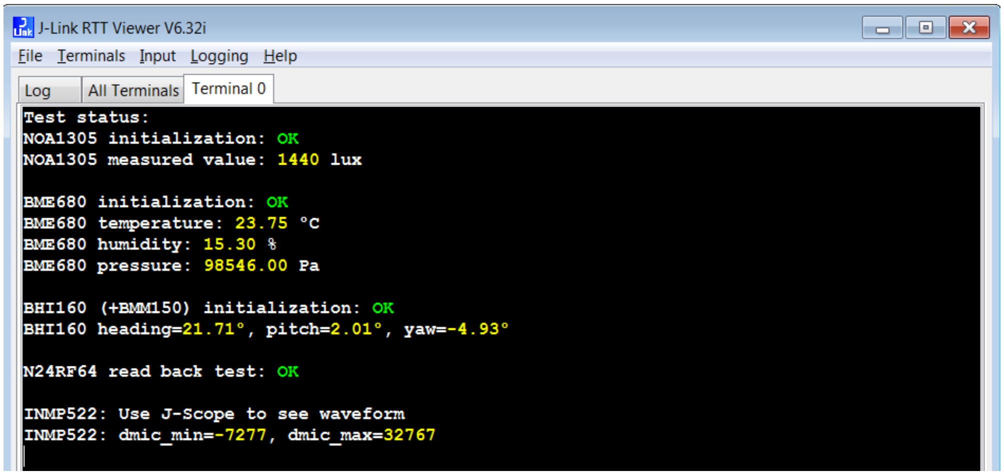
Select a device for J-Link.

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".

Figure 36.

28. Console returns the actual values from all sensors assembled on the board

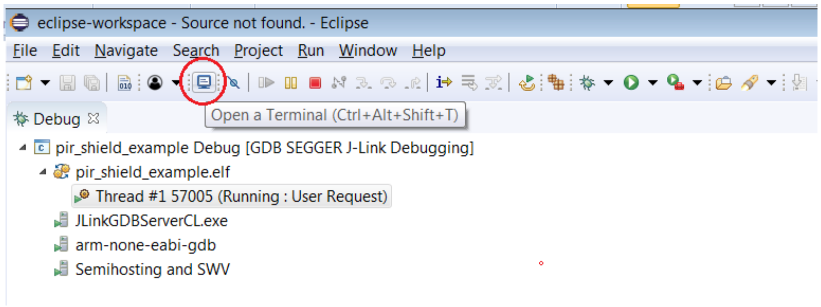


```
Test status:
NOA1305 initialization: OK
NOA1305 measured value: 1440 lux

BME680 initialization: OK
BME680 temperature: 23.75 °C
BME680 humidity: 15.30 %
BME680 pressure: 98546.00 Pa

BHI160 (+BMM150) initialization: OK
BHI160 heading=21.71°, pitch=2.01°, yaw=-4.93°

N24RF64 read back test: OK

INMP522: Use J-Scope to see waveform
INMP522: dmic_min=-7277, dmic_max=32767
```

Figure 37.

Using Eclipse RTT Console

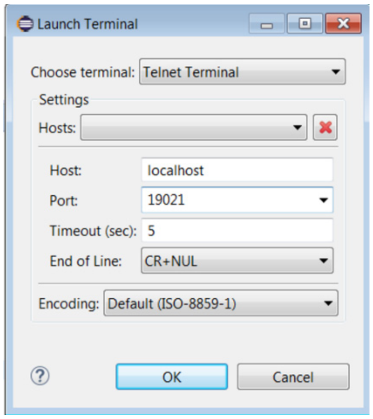
29. Click the Open a Terminal Icon


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

Figure 38.

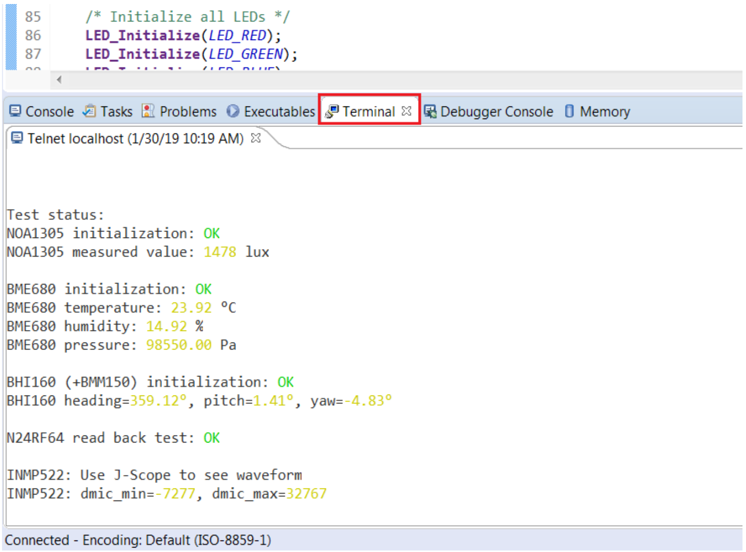
30. Enter the values shown below and launch the session. The incoming events are printed on the terminal window.



Choose terminal: Telnet Terminal
Hosts:
Host: localhost
Port: 19021
Timeout (sec): 5
End of Line: CR+NUL
Encoding: Default (ISO-8859-1)

Figure 39.

31. Terminal window returns the same data format as using the J–LINK RTT viewer.



```
    /* Initialize all LEDs */
    LED_Initialize(LED_RED);
    LED_Initialize(LED_GREEN);
```

```
Test status:
NOA1305 initialization: OK
NOA1305 measured value: 1478 lux

BME680 initialization: OK
BME680 temperature: 23.92 °C
BME680 humidity: 14.92 %
BME680 pressure: 98550.00 Pa

BHI160 (+BMM150) initialization: OK
BHI160 heading=359.12°, pitch=1.41°, yaw=-4.83°

N24RF64 read back test: OK

INMP522: Use J-Scope to see waveform
INMP522: dmic_min=-7277, dmic_max=32767
```

Figure 40.

Using J–scope for MIC data visualization

32. Launch Segger J–Scope and click on New project

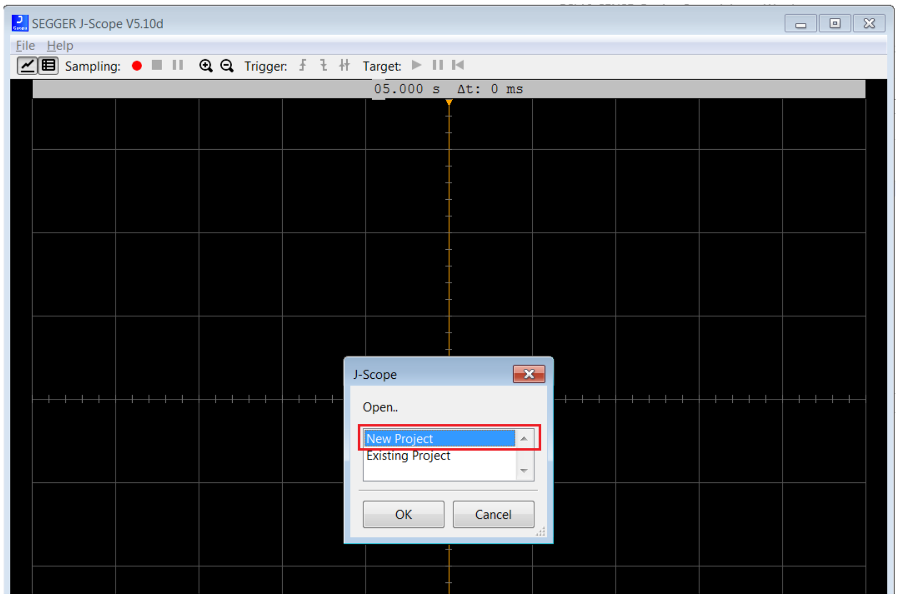
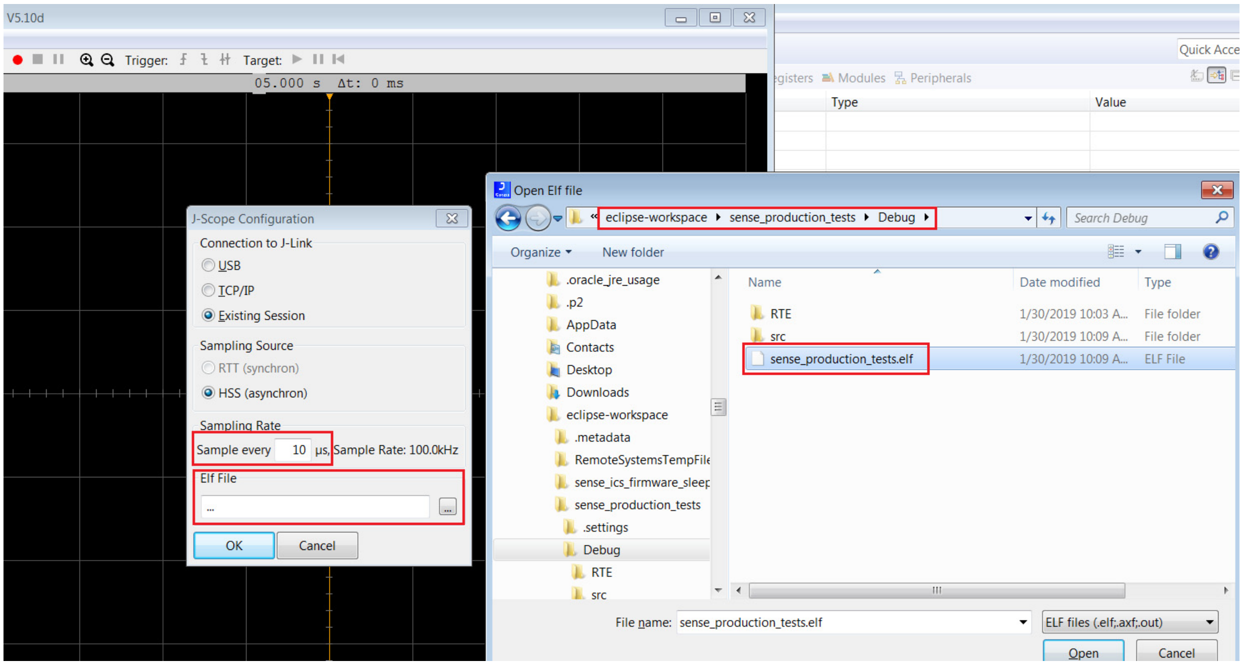


Figure 41.

33. When your Eclipse debug session is launched, use existing Session (alternatively you use USB), set Sample rate every 10us and load elf. file (binary) that is located under Eclipse–workspace and Debug folder.



J-Scope Configuration

Connection to J-Link

○ USB
○ TCP/IP
● Existing Session

Sampling Source

○ RTT (synchron)
● HSS (asynchron)

Sampling Rate

Sample every 10 µs, Sample Rate: 100.0kHz

Elf File

...

OK Cancel

Open Elf file

« eclipse-workspace ▸ sense_production_tests ▸ Debug ▸

.oracle_jre_usage
.p2
AppData
Contacts
Desktop
Downloads
eclipse-workspace
.metadata
RemoteSystemsTempFile
sense_ics_firmware_sleep
sense_production_tests
.settings
Debug
RTE
src

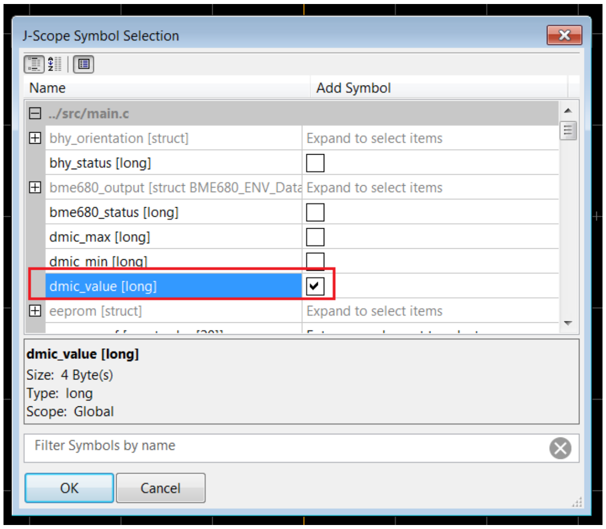
Name	Date modified	Type
RTE	1/30/2019 10:03 A...	File folder
src	1/30/2019 10:09 A...	File folder
sense_production_tests.elf	1/30/2019 10:09 A...	ELF File

File name: sense_production_tests.elf

ELF files (.elf;.axf;.out)

Figure 42.

34. J−Scope symbol section opens. Check dmic_value box and hit OK button.



J-Scope Symbol Selection

Name	Add Symbol
⊟ ../src/main.c	
⊞ bhy_orientation [struct]	Expand to select items
bhy_status [long]	□
⊞ bme680_output [struct BME680_ENV_Data	Expand to select items
bme680_status [long]	□
dmic_max [long]	□
dmic_min [long]	□
dmic_value [long]	☑
⊞ eeprom [struct]	Expand to select items

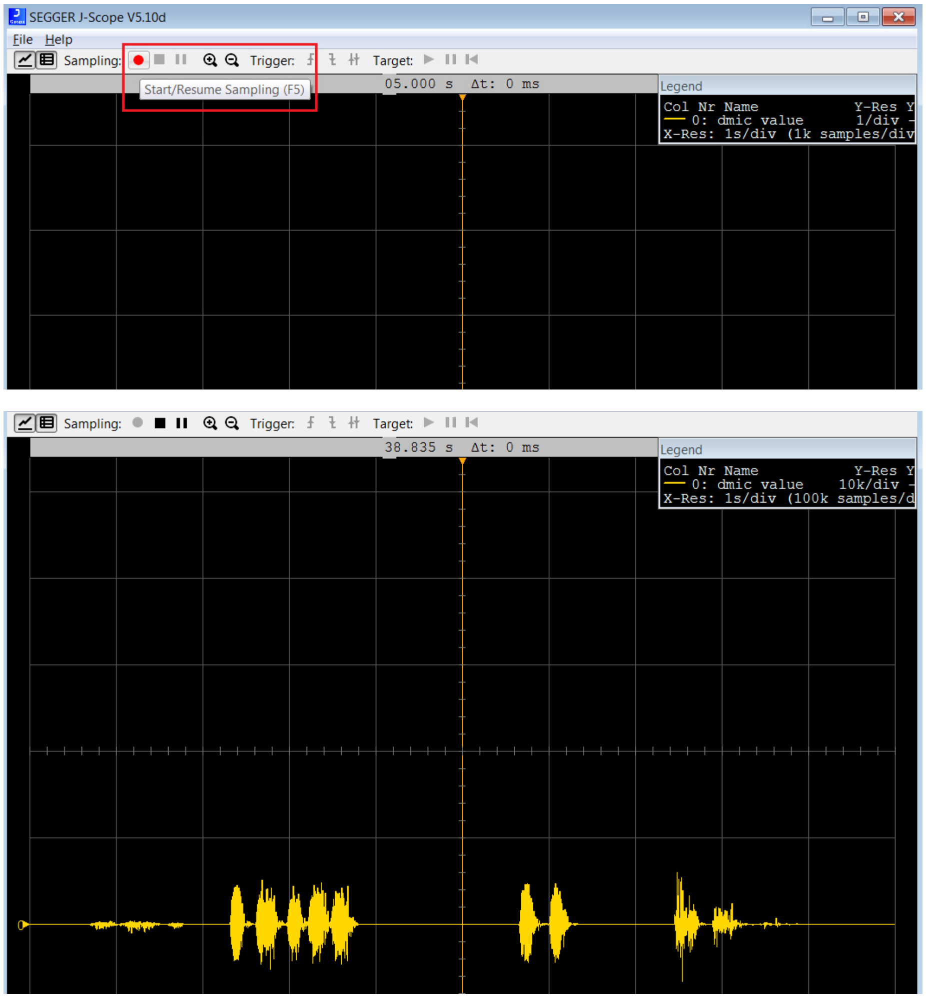
dmic_value [long]
Size: 4 Byte(s)
Type: long
Scope: Global

Filter Symbols by name

OK Cancel

Figure 43.

35. Visualization of the audio is started when Red Sampling button is pushed (or F5.)



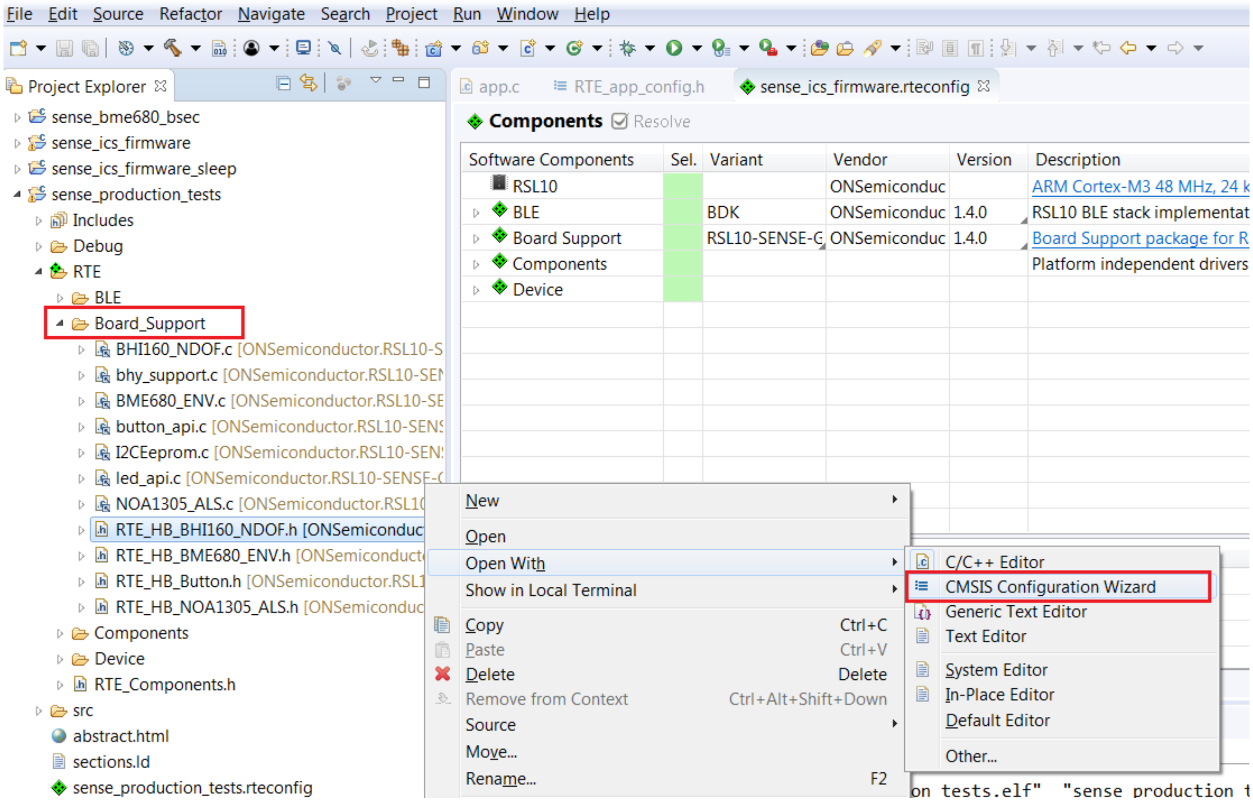
```
05.000 s   Δt: 0 ms
Col Nr Name           Y-Res Y
 ── 0: dmic value      1/div -
X-Res: 1s/div (1k samples/div
```

```
38.835 s   Δt: 0 ms
Col Nr Name           Y-Res Y
 ── 0: dmic value    10k/div -
X-Res: 1s/div (100k samples/d
```

Figure 44.

Configuration Setup

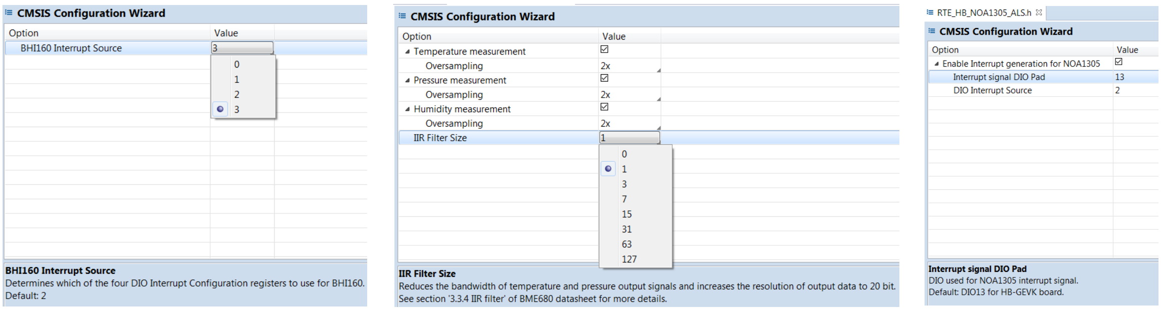
System settings can be configured directly from within the CMSIS pack. Each example is equipped with basic system configuration that covers three main categories. These are accessible in the RTE/BDK folder within the project. Each system configuration starts with “RTE_”. As shown below, opening the RTE_... 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
sense_bme680_bsec
sense_ics_firmware
sense_ics_firmware_sleep
sense_production_tests
Includes
Debug
RTE
BLE
Board_Support
BHI160_NDOF.c [ONSemiconductor.RSL10-S
bhy_support.c [ONSemiconductor.RSL10-SEN
BME680_ENV.c [ONSemiconductor.RSL10-SE
button_api.c [ONSemiconductor.RSL10-SENS
I2CEeprom.c [ONSemiconductor.RSL10-SEN
led_api.c [ONSemiconductor.RSL10-SENSE-C
NOA1305_ALS.c [ONSemiconductor.RSL10
RTE_HB_BHI160_NDOF.h [ONSemiconduc
RTE_HB_BME680_ENV.h [ONSemiconduct
RTE_HB_Button.h [ONSemiconductor.RSL1
RTE_HB_NOA1305_ALS.h [ONSemiconduc
Components
Device
RTE_Components.h
src
abstract.html
sections.ld
sense_production_tests.rteconfig
app.c
RTE_app_config.h
sense_ics_firmware.rteconfig
Components
Resolve
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
C/C++ Editor
CMSIS Configuration Wizard
Generic Text Editor
Text Editor
System Editor
In-Place Editor
Default Editor
Other...
on tests.elf" "sense production t

Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 24 k
BLE		BDK	ONSemiconduc	1.4.0	RSL10 BLE stack implementat
Board Support		RSL10-SENSE-G	ONSemiconduc	1.4.0	Board Support package for R
Components					Platform independent drivers
Device					

Figure 45.

A brief description on the header files is given in the wizard for various sensors.


CMSIS Configuration Wizard
0
1
2
3
CMSIS Configuration Wizard
0
1
3
7
15
31
63
127
RTE_HB_NOA1305_ALS.h
CMSIS Configuration Wizard

Option	Value
BHI160 Interrupt Source	3

BHI160 Interrupt Source
Determines which of the four DIO Interrupt Configuration registers to use for BHI160.
Default: 2

Option	Value
Temperature measurement	☑
Oversampling	2x
Pressure measurement	☑
Oversampling	2x
Humidity measurement	☑
Oversampling	2x
IIR Filter Size	1

IIR Filter Size
Reduces the bandwidth of temperature and pressure output signals and increases the resolution of output data to 20 bit.
See section '3.3.4 IIR filter' of BME680 datasheet for more details.

Option	Value
Enable Interrupt generation for NOA1305	☑
Interrupt signal DIO Pad	13
DIO Interrupt Source	2

Interrupt signal DIO Pad
DIO used for NOA1305 interrupt signal.
Default: DIO13 for HB-GEVK board.

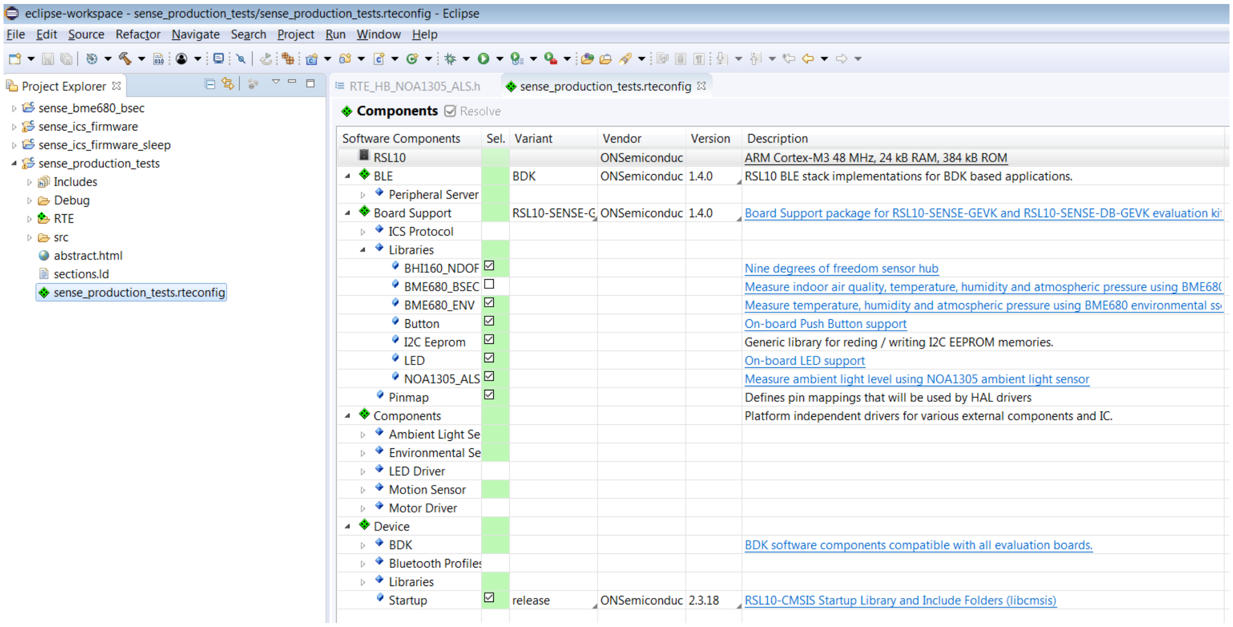
Figure 46.

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.

*.rteconfig

The *.rteconfig file lists the software components within the CMSIS pack. To access the components, double click *.rteconfig file. Extensive help is provided under the description tab.

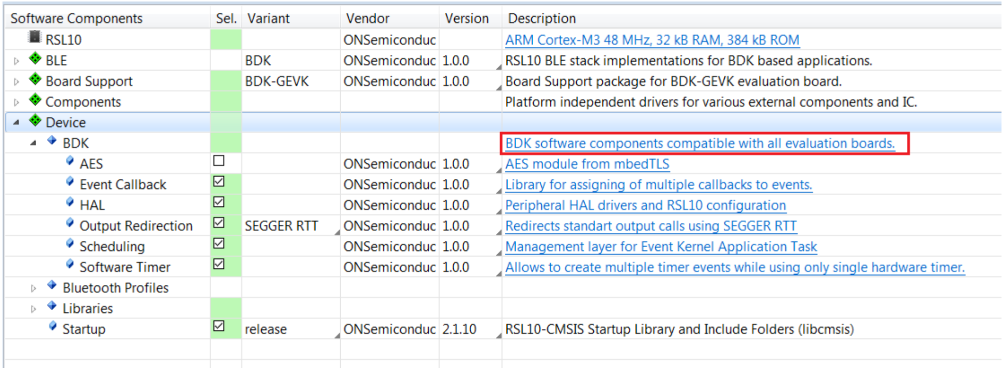

eclipse-workspace - sense_production_tests/sense_production_tests.rteconfig - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
sense_bme680_bsec
sense_ics_firmware
sense_ics_firmware_sleep
sense_production_tests
Includes
Debug
RTE
src
abstract.html
sections.ld
sense_production_tests.rteconfig
RTE_HB_NOA1305_ALS.h
sense_production_tests.rteconfig
Components
Resolve

Software Components	Sel.	Variant	Vendor	Version	Description
RSL10			ONSemiconduc		ARM Cortex-M3 48 MHz, 24 kB RAM, 384 kB ROM
BLE		BDK	ONSemiconduc	1.4.0	RSL10 BLE stack implementations for BDK based applications.
Peripheral Server					
Board Support		RSL10-SENSE-G	ONSemiconduc	1.4.0	Board Support package for RSL10-SENSE-GEVK and RSL10-SENSE-DB-GEVK evaluation ki
ICS Protocol					
Libraries					
BHI160_NDOF	☑				Nine degrees of freedom sensor hub
BME680_BSEC	☐				Measure indoor air quality, temperature, humidity and atmospheric pressure using BME68(
BME680_ENV	☑				Measure temperature, humidity and atmospheric pressure using BME680 environmental ss
Button	☑				On-board Push Button support
I2C Eeprom	☑				Generic library for reding / writing I2C EEPROM memories.
LED	☑				On-board LED support
NOA1305_ALS	☑				Measure ambient light level using NOA1305 ambient light sensor
Pinmap	☑				Defines pin mappings that will be used by HAL drivers
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 Profile:					
Libraries					
Startup	☑	release	ONSemiconduc	2.3.18	RSL10-CMSIS Startup Library and Include Folders (libcmsis)

Figure 47.

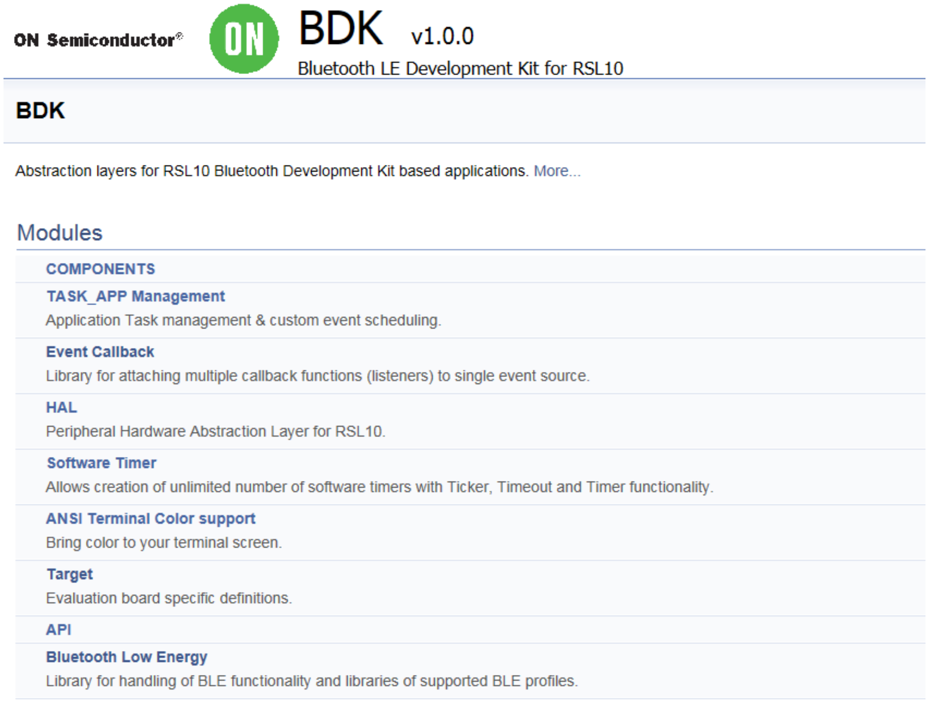
Main Help Page

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)

Figure 48.



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.

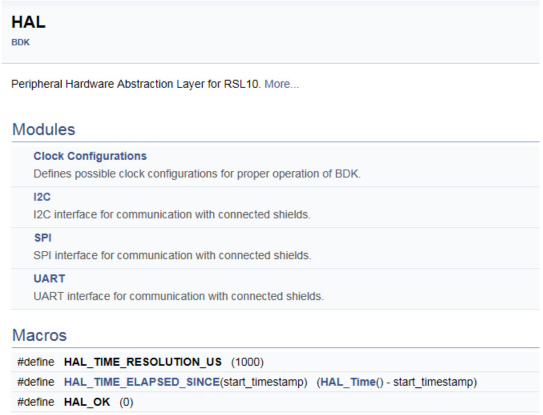
Target
Evaluation board specific definitions.

API

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

Figure 49.

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.

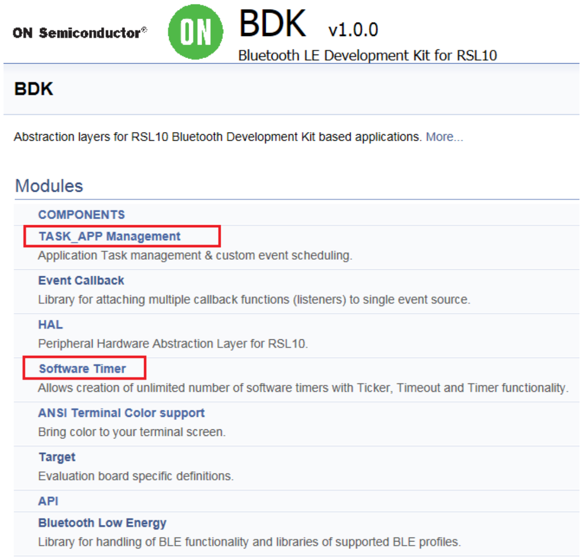
UART
UART interface for communication with connected shields.

Macros

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

Figure 50.

CMSIS 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.

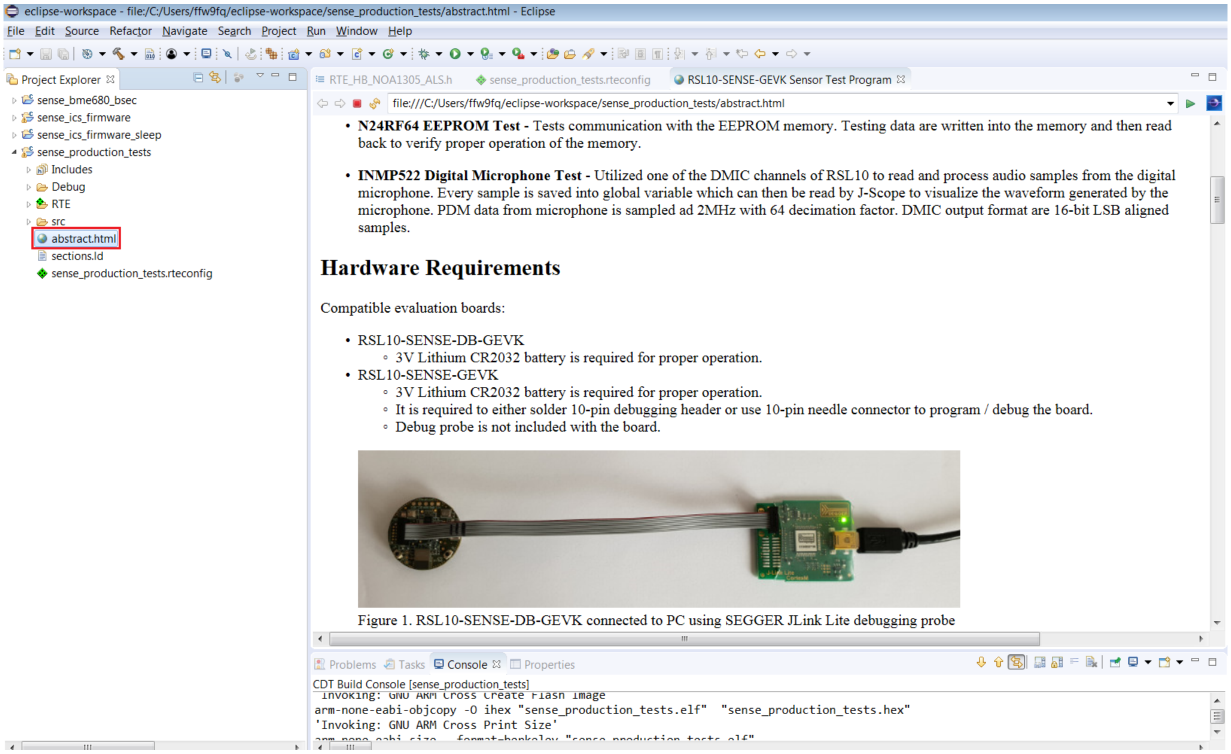
Target
Evaluation board specific definitions.

API

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

Figure 51.

Every example attached to the RSL10−SENSE−GEVK is equipped with addl. help under *abstract.html*


eclipse-workspace - file:/C:/Users/ffw9fq/eclipse-workspace/sense_production_tests/abstract.html - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
sense_bme680_bsec
sense_ics_firmware
sense_ics_firmware_sleep
sense_production_tests
Includes
Debug
RTE
src
abstract.html
sections.ld
sense_production_tests.rteconfig
RTE_HB_NOA1305_ALS.h
sense_production_tests.rteconfig
RSL10-SENSE-GEVK Sensor Test Program
file:///C:/Users/ffw9fq/eclipse-workspace/sense_production_tests/abstract.html
Problems
Tasks
Console
Properties

- **N24RF64 EEPROM Test -** Tests communication with the EEPROM memory. Testing data are written into the memory and then read back to verify proper operation of the memory.

- **INMP522 Digital Microphone Test -** Utilized one of the DMIC channels of RSL10 to read and process audio samples from the digital microphone. Every sample is saved into global variable which can then be read by J-Scope to visualize the waveform generated by the microphone. PDM data from microphone is sampled ad 2MHz with 64 decimation factor. DMIC output format are 16-bit LSB aligned samples.

Hardware Requirements

Compatible evaluation boards:

- RSL10-SENSE-DB-GEVK
  - 3V Lithium CR2032 battery is required for proper operation.
- RSL10-SENSE-GEVK
  - 3V Lithium CR2032 battery is required for proper operation.
  - It is required to either solder 10-pin debugging header or use 10-pin needle connector to program / debug the board.
  - Debug probe is not included with the board.

Figure 1. RSL10-SENSE-DB-GEVK connected to PC using SEGGER JLink Lite debugging probe

CDT Build Console [sense_production_tests]

```
'Invoking: GNU ARM Cross Create Flash Image'
arm-none-eabi-objcopy -O ihex "sense_production_tests.elf"  "sense_production_tests.hex"
'Invoking: GNU ARM Cross Print Size'
arm-none-eabi-size --format=berkeley "sense_production_tests.elf"
```

Figure 52.

Bluetooth is a registered trademark of Bluetooth SIG.

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