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

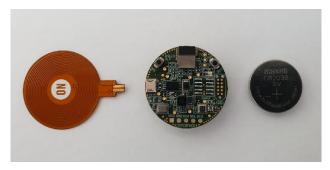
#### Introduction

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  $\mu$ A 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.



# **ON Semiconductor®**

www.onsemi.com

# **EVAL BOARD USER'S MANUAL**

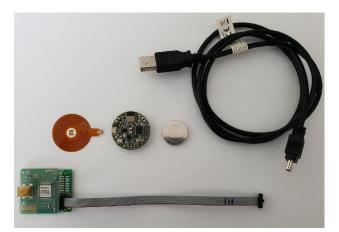


Figure 2.

#### 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.

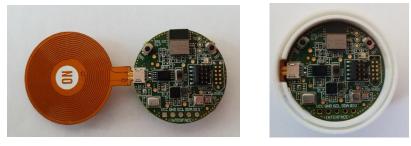


Figure 3.

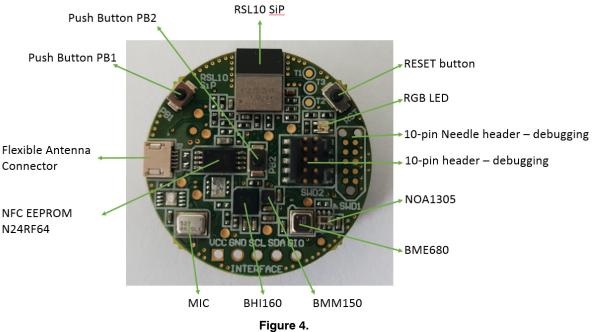


Figure 4

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 <u>https://www.segger.com/downloads/jlink</u> (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.

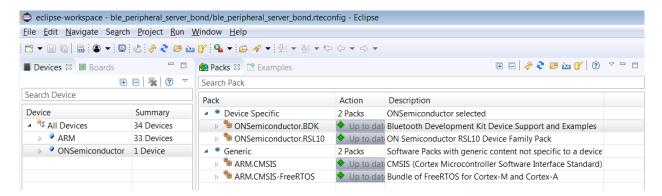
- 4. Download the B–ID CMSIS pack from https://www.onsemi.com/B–IDK and save it in the same folder as the RSL10 CMSIS pack (see 3.a above)
- 5. CMSIS pack at item 4. is dependent on ARM CMSIS pack as well. Please install ARM CMSIS pack 5.5.1 or higher after download from: <u>https://github.com/ARM-software/CMSIS\_5/relea</u> <u>ses</u>
- 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-FreeR TOS/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)





### 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)

E Devices 🖾 🐻 Boards	🕀 🖻 🕐 💥 🐃 🗖	😫 Packs 🛛 📑 Examples	🕀 🖻 😨 🍣 🐸 🔻 🗖 🗖	💷 Pack Properties 😂	
type filter text		Search Pack		type filter text	
y entre dation Device	Summary 1 Device 1 Device	Pack     Pevice Specific     ONSemiconductor/RSL10     Genetic	Description ONSemiconductor selected Bluetooth Device Support and Example: ON Semiconductor RSLID Device Family Pack Software Packs with generic content not specific to a devi		SENSE-GEVK) y Harvesting Bluetooth® Low Energy Switch (BDK-GEVK) ) SE-GEVK) ) SE-GEVK SE-GEVK SE-GEVK K0 K0 SE-GEVK SE

Figure 8.

5. Right click and copy the project into workspace

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



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/ap	nn c - Fclinse
Eile Edit Source Refactor Navigate Search Proje	
	₫ ▼ 않 ▼ ♂ ▼ ☆ ▼ 0 ▼ 9 ▼ 4 ▼ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2 ∞ 2
Project Explorer 🛛 📄 🐄 🛡 🖻	□ 🖻 app.c 🛛
<ul> <li>Sense.ics.firmware_sleep</li> <li>Sense.ics.firmware_sleep</li> <li>Sirncludes</li> <li>Sense.ics.firmware_sleep</li> <li>Sentions.ics.jodkld</li> <li>Sections.idd</li> </ul>	<pre>4 // This software and/or documentation is licensed by 5 // limited terms and conditions. The terms and condit 6 // software and/or documentation are available at 7 // http://www.onsemi.com/site/pdf/ONSEMI_T&amp;C.pdf ("ON 8 // Terms and Conditions of Sale, Section 8 Software") 9 // software license agreement. Do not use this softw 10 // unless you have carefully read and you agree to th 11 // conditions. By using this software and/or documen 12 // limited terms and conditions. 13 //</pre>
sense_ics_firmware_sleep.rteconfig	330 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.

🖨 e	clipse	-workspa	ice - sense	_ics_firmwa	re_sleep/	src/app.c	- Eclip
<u>F</u> ile	<u>E</u> dit	<u>S</u> ource	Refactor	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> roject	<u>R</u> un
<b>1</b>	. 8	r 🛞 🗣	- 🐔 🗕 📷	🛨	× &	•	
		Explorer	110	ebug	5		
		e_ics_firm	2 R	elease			c
_		cludes	Winne_bree	٢			
	🕞 in						
	🚖 R						
	🗁 sr	c					
			Fia	ure 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

<u>-</u>	≪ <b>- - - - - - - -</b>	₫ • @ • ₢ • ₢ •	☆ ▼ () ▼ 😪 ▼ 💁 ▼ 🎒 🗀 🖋 ▼ 🌙 😜 📴 🗐	
Project Explorer	🛪 🕒 🖻 🖘 🖓	🗖 🔂 app.c 🛙		
<ul> <li>Project Explorer</li> <li>Sense_ics_f</li> <li>Binaries</li> <li>Binaries</li> <li>Includes</li> <li>Include</li> <li>Release</li> <li>Release</li> <li>RTE</li> <li>Src</li> <li>Srbe</li> <li>Strain</li> <li>Sense</li> <li>Sense</li></ul>	New A // This A		<pre>e and/or documentation are available a www.onsemi.com/site/pdf/ONSEMI_T&amp;C.pdf nd Conditions of Sale, Section 8 Softw e license agreement. Do not use this you have carefully read and you agree ons. By using this software and/or do terms and conditions. stdio.h&gt; app.h"</pre>	
v 🔤 abb	Exp <u>o</u> rt		<pre>tateStruct app_state = APP_STATE_INIT; mer app_state_timer;</pre>	
▷ ⓒ app_ � ▷ ⓒ app_ ▷ ⓒ app.c	CMSIS C/C++ Project <u>B</u> uild Project Clean Project	•	oid) _Initialize();	
<ul> <li>▷</li></ul>	Refresh Cloge Project Close <u>Unrelated Projects</u> Build Targets	F5	<pre>ication - Initialization complete. */ (LED_GREEN); lay(250); f(LED_GREEN);</pre>	
▷ 🖻 HAL_ ▷ 🗟 waker	Index	+	oop();	
abstract	Build Configurations	•	Set Active • <u>1</u> Debug	
<ul> <li>sections</li> <li>sections</li> <li>sense_ic</li> </ul>	Show in Remote Systems view Profiling Tools <u>V</u> alidate	•	Manage ✓ 2 Release Build All Clean All	
	Run As	•	Build Selected	

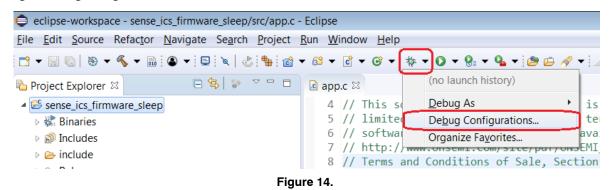
Figure 12.

- 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.



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

Create, manage, and run configurations		Ŕ
3 🗎 🗙 🖪 🌣 ▾	Name: sense_ics_firmware_sleep Release	
type filter text C //C++ Application C //C++ Attach to Application C //C++ Postmortem Debugger C //C++ Remote Application C GDB Hardware Debugging C GDB SEGGER J-Link Debugging C GDB SEGGER J-Link Debugging C sense_ics_firmware_sleep Release Launch Group ► Launch Group (Deprecated)	Main  Debugger  Startup  Source  Common  Story  SVD Path Project:	
	sense_ics_firmware_sleep  C/C++ Application:  Release\sense_ics_firmware_sleep.elf	growse Browse
	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.

Configurations	Program Selection	- • ×	8
Create, manage, and ru	Choose a program to run:		- A
Image: Second Secon		re sleep.elf	mware_sleep_Release gger   Startup   Source   Common   So SVD Path s_sleep   Browse firmware_sleep.elf before launching pr: Select Automatically ild   Disable auto build settings   Configure Workspace Settings
Filter matched 10 of 10 iter	Э ОК	Cancel	Reyert Apply Debug Close

Figure 16.

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

Debug Configurations	
Create, manage, and run configurations	
C @ ¥ = > ▼	Name: sense_jcs_firmware_sleep Release
type filter text	🕒 Main ≉ Debugger 🕒 Startup 🦻 Source 🗖 Common 🛃 SVD Path
C/C++ Application	J-Link GDB Server Setup Start the J-Link GDB server locally Connect to running target
<ul> <li>C/C++ Postmortem Debugger</li> <li>C/C++ Remote Application</li> </ul>	Executable path: \$(jlink_path)/\$(jlink_gdbserver) Browse Variable
GDB Hardware Debugging     GDB OpenOCD Debugging	Actual executable: C/Program Files (x86)/SEGGER/ILink_V632i/ILinkGDBServerCLexe (to change it use the global or workspace preferences pages or the project properties page
GDB SEGGER J-Link Debugging     E sense ics_firmware_sleep Release	Device name: RSL10 Supported device name
Caunch Group	Endianness: 💿 Little 🔘 Big
Launch Group (Deprecated)	Connection:  OUSB  IP (USB serial or IP name/address)
	Interface:      SWD      JTAG
	Initial speed: O Auto Adaptiv  Fixed 1000 kHz
	GDB port: 2331
	SWO port: 2332 Verify downloads 🗹 Initialize registers on sta

Figure 17.

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

retva	Confirm Perspective Switch	۲
ialize tialize	This kind of idunct is configured to open the Debug perspective when it suspends.	
ialize	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.	
"\r\n\ı	Do you want to open this perspective now?	
ialize = NCS30 DEBUG(1 0_PIR_0		

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	
<u>Eile E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
! ◘ ▾ ◙ ◎ ▮ ● ▾ ! ▣ ! ኊ ■ ₩ ૠ ☜ ☜ !! !> ≂ ☆   ₺ = ₩ ₩ ▼ ● ▼ ● ≁ ! ⊅ = ∅ ↓ ▼ ! ⊅ ♥ ↓ ▼ !!	- ⇒ -
☆ Debug 🛛         Resume (F8)         ¾         I ➡         □         □	(x)= Variable
sense_ics_firmware_sleep Release [GDB SEGGER J-Link Debugging]	Name
<ul> <li>Sense_ics_firmware_sleep.elf</li> <li>Thread #1 57005 (Suspended : Breakpoint)</li> <li>main() at app.c:22 0x104ac8</li> <li>JLinkGDBServerCL.exe</li> <li>arm-none-eabi-gdb</li> <li>Semihosting and SWV</li> </ul>	
li app.c ☆	



#### Figure 19.

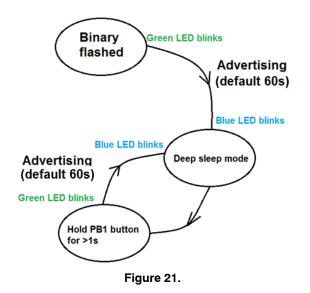
eclipse-workspace - sense_ics_firmware_sleep/src/app.c - Eclipse					
<u>File Edit Source Refactor Navigate Search Project Run Window H</u> elp					
C → □ □ □ □ □ ↓ □ ↓ □ □ ■ ⋈ ≥ ○ .c ↓ → = ∞ ↓ ≥ □ ↓ = 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0	i • †i • †≎ ¢ •	⇔ -			
₩ Debug 😂	🍇 i i 🔹 🗖 🗖	* Variables 13 • Breakpoints III Registers	🛋 Modules 🛛 🔒 Periph	ierals	
<ul> <li>C sense_ics_firmware_sieep Release (GDB SEGGER J-Link Debugging)</li> <li>C sense_ics_firmware_sieep.elf</li> <li>InikGDBServerCLexe</li> <li>TunkGDBServerCLexe</li> <li>arm-none-eabi-gdb</li> <li>Semihosting and SWV</li> </ul>		Name	Туре		Value
la app.c ∞			- 8	BE Outline 🖾	
<pre>17 18 enum App_StateStruct app_state = APP_STATE_INIT; 19 struct stimer app_state_timer; 20 21=int main(void)</pre>				<ul> <li>stdio.h</li> <li>app.h</li> <li>app_state:ei</li> <li>app_state_tin</li> </ul>	ner : struct stil
<pre>22 [ 22 Device_Initialize(); 24 25 /* Indication - Initialization complete. */ 26 LED_On(LED_GREEN); 27 HAL_Delay(250); 28 LED_Off(LED_GREEN); 29</pre>				<ul> <li>main(void) : i</li> <li><sup>5</sup> App_StateMa</li> <li>Main_Loop(v)</li> </ul>	chine(void) :
· · · · · · · · · · · · · · · · · · ·			Þ		
Console ☆ ② Tasks இ Problems					🗙 🔆 🗎 🚡 erminate

#### Figure 20.

 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.



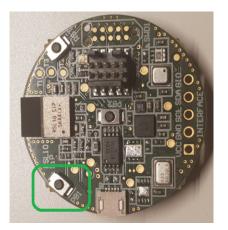


Figure 22.

- 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.

eclipse-workspace - sense_ics	_firmware_sleep/include/RT	E_app_config.h - Eclipse			
Eile Edit Source Refactor Na	avigate Se <u>a</u> rch <u>P</u> roject [	<u>un W</u> indow <u>H</u> elp			
📑 🕶 🔛 💿   🕸 🕶 🗞 🕶 🔝 🖉		◎ • ◎ • ◎ • ♦ • 0 • 9 • 9	• 🙋 🖨 🛷 • 🗟 🗉 🖬 🖗	○ ◆ ○ ◆ ○ ◆	
မြဲ Project Explorer 🛛	🗏 😫 🛸 🗠 🗖	B app.c	rteconfig 🛛 🛤 RTE_app_config.	.h 23	
<ul> <li>Sense_ics_firmware_sleep</li> </ul>		CMSIS Configuration Wizard			⊞ ⊟ ⊘
Includes		Option	Value		
bdk		BLE Advertising Interval [ms]	1000		
but		BLE Complete Local Name	HB_BLE_Terminal		
app_sleep.h		Advertising Stop Timeout [s]	60		
app_timer.h		Wake-up Button Check Interval [ms]	1500		
app_trace.h		I2C Bus Speed	Fast+	4	
▶		<ul> <li>IDK Custom Service</li> </ul>			
CSN_LP_ALS.h		<ul> <li>Ambient Light Node (AL)</li> </ul>			
CSN_LP_AO.h		Integration Time [ms]	12.5		
CSN_LP_ENV.h		Number of Measurement Cycle			
B HAL_RTC.h		<ul> <li>Environmental Node (EV)</li> </ul>			
RTE_app_config.h		BSEC Sample Rate	5 min	4	
Release	New	, suremen	t 🗆		
🖻 🍉 RTE	Open	e (AO)	5		
> 🗁 SFC	Open With	> C/C			
abstract.html	Show in Local Terminal		SIS Configuration Wizard		
sections_bdk.ld		Gen Gen	eric Text Editor		
in occurring the	Сору	Ctrl+C E Text	Editor		
sense_ics_firmware_s	Paste	Ctrl+V			

Figure 23.

eclipse-workspace - sense_ics_firmware_sleep/include/RT	E_app_config.h - Eclipse			
Eile Edit Source Refactor Navigate Search Project I	Run <u>W</u> indow <u>H</u> elp			
≝ - 🗉 🖏 - 🍕 - 🐘 🎱 - 🖾 😓 🍪 -	• 🚳 • 🖻 • @ •   🕸 • O • 9: • 9. • 1	۵۵ 🖉 🖻 🖉 🗸 🖉	▼ ⅔ ▼ ♥ ♥ ▼ ♥ ▼	
Project Explorer 🛛 📄 😫 😨 🔻 🗖 🗖	c app.c	nfig 🛛 🗏 RTE_app_config.h	1 22	
<ul> <li>✓ Sense_ics_firmware_sleep</li> <li>▷ S Includes</li> </ul>	CMSIS Configuration Wizard			• • •
<ul> <li>include</li> <li>is bdk</li> <li>is app_ble_hooksh</li> <li>is app_limerh</li> <li>is app_limerh</li> <li>is app_h</li> <li>is app_h</li> <li>is app_h</li> <li>is CSN_LP_ACSh</li> <li>is CSN_LP_ACSh</li></ul>	Option BLE Advertising Interval [ms] BLE Complete Local Name Advertising Stop Timeout [6] Wake-up Button Check Interval [ms] IZC Bus Speed IDK Custom Service A Ambient Light Node (AL) Integration Time [ms] Number of Measurement Cycles (cyc BSEC Sample Rate Indoor Air Quality measurement Absolute Orientation Node (AO) Virtual Sensor Report Rate [Hz]	Value 1000 HB_BLE_Terminal 60 1500 Fast+ 2 12.5		
	BLE Advertising Interval [ms] Determines how often to send advertising pac Default: 1000 ms Source Editor [CMSIS Configuration Wizard]	ikets.	19	,



### 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.

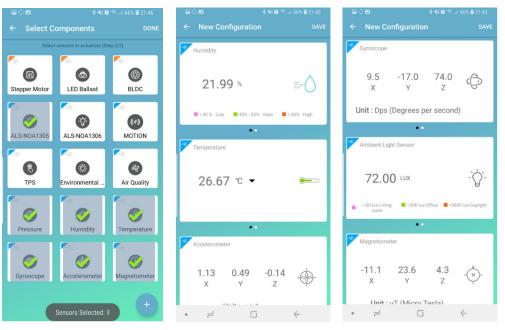


Figure 25.

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.





### **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  $-900 \,\mu\text{A}$  (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  $\mu$ A (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.

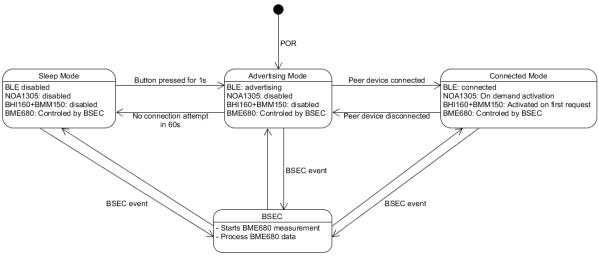


Figure 27.

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.

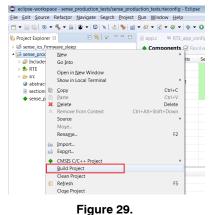
21. Right click and copy the project into workspace



Figure 28.

22. Right click on the project and build it

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



- -igule 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.

rteconfig -	Eclip	se					
ow <u>H</u> elp							
- © -	脊	• • •	<b>9</b> - 9	- 2	) 🕞 🔗 🤜		
≔ RTE_a	1 sense ics firmware sleen Release						fig ដ
onents		<u>D</u> ebug	, As				•
Jonenta		Debug Configurations					
Compone		Organ	ize Fa <u>v</u> or	ites			sion
.10	_				ONSemic	onduc	
:			RUK		ONSemic	onduc	1/10
			Figure	e 30.			

Name: sense_production_tests Debug	purce 🔲 Common 🛱 SVD Path			
Project:         Browse           sense_production_tests         Browse           C/C++ Application:         C/C++				
Variables         Search Project           Build Configuration         Select Automatically				
Use workspace settings	Configure Workspace Settings			
	Reyert	Apply		
	C/C++ Application: Debug\sense_production_tests.elf Build (if required) before launching Build Configuration; Select Automatica © Enable auto build	C/C++ Application: Debug\sense_production_testself UariablesSearch Project Build (if required) before launching Build Configuration: Select Automatically © Enable auto build © Disable auto build © Use workspace settings Configure Workspace Settings.		

Figure 31.

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

	• ↔ ↔ •  { •  { •  { •  { •    •    •			Quick Access 🔡 🛱
Debug 12 Resume (15) E sense_production, tests Debug (GDB SEGGER J-Link Debugging)	74   JF	% 14 ° ° ° □ 0• Variables 21 % Breakpoints III Ro Name 6+ retval 6+ timestamp		Value -738672248 756375673
app.c ■ RTE_app_config.h ◆ sense_production_tests.rteconfig @ main.c № 70 int32_t retval = 0;		4		Outline 22 문 년 전 文 • 북 · · · bme680_status : imi32_t
<pre>80 /* Initialize BDK library, set system clock (default 80Hz). */ 82 BDK_InitializeFreq(HAL_CLK_COMP_80HZ); 83 HAL_IZC_SetBusSpeed(HAL_IZC_BUS_SPEED_FAST); 84 85 /* Initialize all LEDs */ 85 LED_Initialize(LED_BEED); 87 LED_Initialize(LED_BEED); 81 LED_Initialize(LED_BEED); 89 97 /* Test LEDs are working */ 89</pre>		Ξ.	<ul> <li>bme680_output: struct BME680_ENV_Data</li> <li>bhy_status : int32_t</li> <li>bhy_ointation: bhy_data_vector_t</li> <li>eeprom_status: int32_t</li> <li>eeprom_iZCEeprom</li> <li>eeprom_ef: const uint3_t]</li> <li>eeprom_ref_length: const uint3_t</li> <li>dmic_value: int32_t</li> </ul>	
91 LED_On(LED_RED);			-	<ul> <li>dmic_min : int32_t</li> <li>main(void) : int</li> </ul>



## 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
© <u>U</u> SB
<u> Т</u> СР/IР
Existing Session Auto reconnect
OK Cancel

Figure 33.

J-Link RTT Viewer V6.34c		
<u>File</u> <u>Terminals</u> Input Log	🔝 J-Link RTT Viewer V6.34c   Configuration 🛛 🔹	]
Log All Terminals Terr LOG: J-Link RTT Vie LOG: Terminal 0 add	Connection to J-Link USB ICP/IP Existing Session Specify Target Device 	
	Script file (optional)	
	Target Interface & Speed	
	SWD	
	RTT Control Block  Auto Detection Address Search Range  OK Cancel	
-		Enter Clear
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 View	ver V6.34c		• ×
<u>Eile</u> <u>T</u> erminals <u>I</u>	Input Logging <u>H</u> elp		
Log All Termina	als Terminal 0		
LOG: J-Link LOG: Termina	RTT Viewer V6.34c: Logging started. al 0 added.		
🔜 J-Lin	nk V6.34c Device Selection	X	
0	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 downlaad or unlimited flash breakpoints. For some devices which require a special handling, selection of the correct device is		
		Enter	Clear
Ready.	Establishing J-Link connection 0.00 M	В	

Figure 35.

GGER J-Link V	6.34c - Ta	arget device set	tings				<b>—</b> ×
Filter Manufacturer *	•	Device RSL10	Core	,	•	Little	endian 💌
Manufacturer	Device		Core		NumCo	Flash size	RAM size
ON Semicond	RSL10		Cortex-N	13 r2p1	1	390 KB	24 KB
s flash downloa	e is not re d, modifice	quired for most d ation of flash mem y (Flash Breakpo	ory during a deb				<u>C</u> ancel
case of doubt :	select the	first entry in the lis	t "Unspecified E	levice".			<u>0</u> K

Figure 36.

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

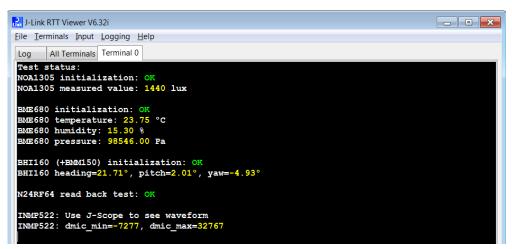
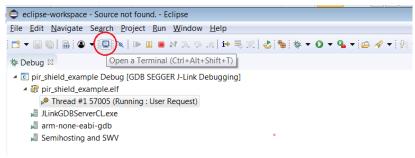


Figure 37.

# Using Eclipse RTT Console

29. Click the Open a Terminal Icon





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

Caunch Terminal	
Choose terminal:	Telnet Terminal
Settings	
Hosts:	- *
Host:	localhost
Port:	19021 🔹
Timeout (sec):	5
End of Line:	CR+NUL -
Encoding: Defa	ult (ISO-8859-1) 👻
?	OK Cancel

Figure 39.

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

<pre>85  /* Initialize all LEDs */ 86  LED_Initialize(LED_RED); 87  LED_Initialize(LED_GREEN); 90  (100 01100); 91  (100 01100); 92  (100 01100); 93  (100 01100); 94  (100 01100); 95  (100 0100); 95  (100 01</pre>
🕒 Console 🧔 Tasks 🖹 Problems 🕡 Executables 🐙 Terminal 🕺 🔀 Debugger Console 🕕 Memory
■ Telnet localhost (1/30/19 10:19 AM) 🕸
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= <mark>359.12°, pitch=1.41°, yaw=-4.83°</mark>
N24RF64 read back test: OK
INMP522: Use J-Scope to see waveform INMP522: dmic_min=-7277, dmic_max=32767
Connected - Encoding: Default (ISO-8859-1)

Figure 40.

#### Using J-scope for MIC data visualization 32. Launch Segger J-Scope and click on New project

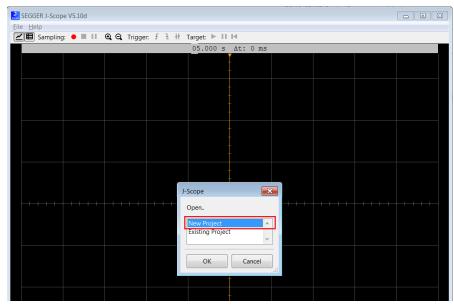


Figure 41.

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.

V5.10d		
● ■ II @ Q Trigger. f t # Target: ▶ II M		Quick Acce
05.000 s Δt: 0 ms	gisters 🛋 Modules 🔀 Peripherals	1 <b>1</b>
	Туре	Value
J-Scope Configuration	Open Elf file	Search Debug
Connection to J-Link	Organize • New folder	≣ • □ 0
CEP/IP  Existing Session  Sampling Source  RTT (synchron)  HSS (asynchron)  Sampling Rate	Loracle_jre_usage     J.p2     AppData     Contacts     Desktop     Downloads     Lecipse-workspace	Date modified         Type           1/30/2019 10:03 A         File folder           1/30/2019 10:09 A         File folder           1/30/2019 10:09 A         ELF File
Sample every 10 µs Sample Rate: 100.0kHz Elf File	metadata     RemoteSystemsTempFile     sense_jroduction_tests     settings     Debug     RTE	
	File game: sense_production_tests.elf	▼ ELF files (.elf;axf;out) ▼ Qpen Cancel

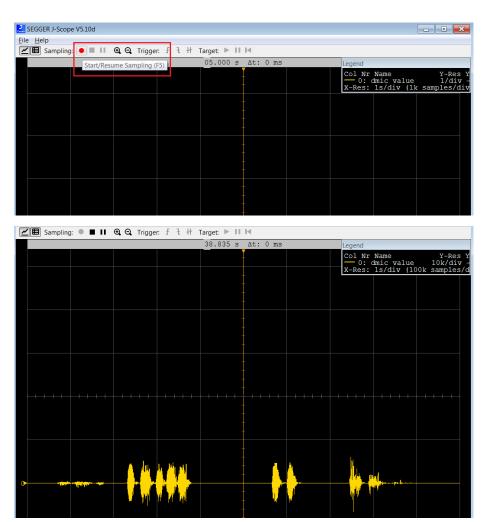
Figure 42.

34. J–Scope symbol section opens. Check dmic\_value box and hit OK button.

J	-Sc	ope Symbol Selection		×			
Ō	ī.	\$II   III					
	Na	me	Add Symbol				
		/src/main.c		-			
	Ŧ	bhy_orientation [struct]	Expand to select items	H			
		bhy_status [long]					
	Ŧ	bme680_output [struct BME680_ENV_Data	Expand to select items				
		bme680_status [long]					
		dmic_max [long]					
	_	dmic min [long]					
		dmic_value [long]					
	Ŧ	eeprom [struct]	Expand to select items	-			
			e				
	dmic_value [long]           Size: 4 Byte(s)           Type: long           Scope: Global						
	Fil	Iter Symbols by name		$\bigotimes$			
		OK Cancel		đ	-		

Figure 43.

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





#### **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.

<u>File Edit Source Refactor Navigate Search Project</u>	<u>R</u> un <u>W</u> indow <u>H</u> elp						
🖆 🕶 🔚 🏀   🛞 🕶 🔦 🕶 📾   🕥 🕶   😂   🔌   🕹   🍓 :	• 🚳 • 🖻 • 🚱 • 🔯 •	•	8 • <b>%</b> • 🙋	🗁 🔗 🔻	R I		▼ 🖓 ▼ 🏷 🗘 ▼ 🖒 ▼
ြာ Project Explorer 🛛 🕞 🧐 😨 🖓 🗖	ⓓ app.c ≔ RTE_app_cc	onfig.h	sense_ics	_firmware.	rtecor	nfig 🛙	
<ul> <li>Sense_bme680_bsec</li> <li>Sense ics firmware</li> </ul>	💠 Components 🗹 Res	olve					
Sense_ics_firmware_sleep	Software Components RSL10		Variant Vendor ONSemico		anduc	Version	Description ARM Cortex-M3 48 MHz, 24 k
<ul> <li>▲ Sense_production_tests</li> <li>▶ Some Debug</li> <li>▲ Debug</li> <li>▲ RTE</li> <li>▶ BLE</li> <li>▲ Board_Support</li> <li>▶ BHI160_NDOF.c [ONSemiconductor.RSL10-SE</li> <li>▶ Buby_support.c [ONSemiconductor.RSL10-SE</li> </ul>	<ul> <li>▶ ◆ BLE</li> <li>▶ ◆ Board Support</li> <li>▶ ◆ Components</li> <li>▶ ◆ Device</li> </ul>			ONSemico	onduc	1.4.0	RSL10 BLE stack implementat Board Support package for R Platform independent drivers
<ul> <li>BME680_ENV.c [ONSemiconductor.RSL10-SE</li> <li>Botton_api.c [ONSemiconductor.RSL10-SENS</li> <li>C2CEeprom.c [ONSemiconductor.RSL10-SENS</li> <li>C2CEeprom.c [ONSemiconductor.RSL10-SENSE-C</li> <li>NOA1305_ALS.c [ONSemiconductor.RSL10</li> <li>NOA1305_ALS.c [ONSemiconductor.RSL10</li> <li>M RTE_HB_BHI160_NDOF.h [ONSemiconductor.RSL10</li> </ul>	New Open			•			
<ul> <li>RTE_HB_BME680_ENV.h [ONSemiconduct</li> <li>RTE_HB_Button.h [ONSemiconductor.RSL1</li> <li>RTE_HB_NOA1305_ALS.h [ONSemiconduct</li> <li>Components</li> <li>Device</li> <li>RTE_Components.h</li> <li>Strc</li> <li>abstract.html</li> <li>sections.ld</li> <li>sense_production_tests.rteconfig</li> </ul>	Open With Show in Local Terminal Copy Paste Delete Remove from Context Source Moye Rename	Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down			Generic Te Text Editor System Ed In-Place Ec Default Ed Other	nfiguration Wizard	

Figure 45.

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

CMSIS Configuration Wizard		CMSIS Configuration Wizard	CMSIS Configuration Wizard					
Option	Value	Option	Value	CMSIS Configuration Wizard				
BHI160 Interrupt Source	3	<ul> <li>Temperature measurement</li> </ul>		Option	Value			
	0	Oversampling	2x	<ul> <li>Enable Interrupt generation for NOA1305</li> </ul>				
	1	<ul> <li>Pressure measurement</li> </ul>		Interrupt signal DIO Pad	13			
	2	Oversampling	2x	DIO Interrupt Source	2			
	3     3	<ul> <li>Humidity measurement</li> </ul>						
		Oversampling	2x					
		IIR Filter Size	1					
			0					
			• 1					
			3					
			7					
			15					
			31					
			63					
			127					
BHI160 Interrupt Source Determines which of the four DIO Interrup Default: 2	ot Configuration registers to use for BHI	1160. Reduces the bandwidth of temperature a See section '3.3.4 IIR filter' of BME680 dat	nd pressure output signals and increases the resolution of output data to 20	Interrupt signal DIO Pad DIO used for NOA1305 interrupt signal. Dit. Default: DIO13 for HB-GEVK board.				



# 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.

<u>Eile E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>Pro</u>									
🖆 🕶 🔲 🐚 🖌 🗣 🦠 🕶 📾 🔍 🔍 😓 🐂	@ ▼ @ ▼ @ ▼ @ ▼ *	0 -	9: • 9 <b>.</b> • 🙆	• 🗁 🛷 👻 📴		▼ 型 ▼ <sup>(1)</sup> ← ▼ <sup>(2)</sup> ▼			
🖕 Project Explorer 🛛 📃 🖘 🔍 🖻	° □ 🔚 RTE_HB_NOA1305_ALS	.h	sense_produc	tion_tests.rteconfi	g 🛙				
> 🐸 sense_bme680_bsec		♦ Components 🖉 Resolve							
Sense_ics_firmware	Software Components	امک	Variant	Vendor	Version	Description			
Sense_ics_firmware_sleep	RSI 10	501	variant	ONSemiconduc		ARM Cortex-M3 48 MHz. 24 kB RAM. 384 kB ROM			
sense_production_tests	A SELO		BDK	ONSemiconduc		RSL10 BLE stack implementations for BDK based applications.			
Includes	Peripheral Serve		BUK	ONSemiconduc	1.4.0	A RELIGIBLE stack implementations for BDK based applications.			
Debug	<ul> <li>Peripheral Server</li> <li>Board Support</li> </ul>	-	RSI 10-SENSE-	G ONSemiconduc	140	Board Support package for RSL10-SENSE-GEVK and RSL10-SENSE-DB-GEVK evaluation			
🖻 🏝 RTE	ICS Protocol		HOLIO DENDE	e on on on on one	1.1.0	bound support package for North Service Gerre and North Service Sob Gerre evaluation			
SIC	▲ ◆ Libraries								
abstract.html sections.ld	P BHI160 NDC	F 🛛				Nine degrees of freedom sensor hub			
	BME680 BSE					Measure indoor air quality, temperature, humidity and atmospheric pressure using BME			
sense_production_tests.rteconfig	BME680 ENV					Measure temperature, humidity and atmospheric pressure using BME680 environmenta			
	Button					On-board Push Button support			
	I2C Eeprom					Generic library for reding / writing I2C EEPROM memories.			
	LED					On-board LED support			
	NOA1305_AI	s⊠				Measure ambient light level using NOA1305 ambient light sensor			
	Pinmap					Defines pin mappings that will be used by HAL drivers			
	<ul> <li>Components</li> </ul>					Platform independent drivers for various external components and IC.			
	Ambient Light S	ie							
	Environmental 5	Se							
	LED Driver								
	Motion Sensor								
	Motor Driver								
	A								
	BDK					BDK software components compatible with all evaluation boards.			
	Bluetooth Profil	e							
	Libraries								
	Startup		release	ONSemiconduc	2.3.18	RSL10-CMSIS Startup Library and Include Folders (libcmsis)			



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 <sup>®</sup> 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)

pheral Hardware Abstraction Layer for RSL10. More  bdules  Clock Configurations Defines possible clock configurations for proper operation of BDK.  Izc IzC Interface for communication with connected shields.  SPI SPI Interface for communication with connected shields. UART Interface for communication with connected shields.  CLART Interface for communication with connected shields.  CLAR
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 UART Interface for communication with connected shields.  CIOS  efine HAL_TIME_RESOLUTION_US (1000)
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.  CONS  efine HAL_TIME_RESOLUTION_US (1000)
2C interface for communication with connected shields.  SPI SPI interface for communication with connected shields. UART UART Interface for communication with connected shields.  acros  efine HAL_TIME_RESOLUTION_US (1000)
SPI Interface for communication with connected shields. UART UART Interface for communication with connected shields. acros efine HAL_TIME_RESOLUTION_US (1000)
UART interface for communication with connected shields.  acros  efine HAL_TIME_RESOLUTION_US (1000)
efine HAL_TIME_RESOLUTION_US (1000)
efine HAI TIME ELAPSED SINCE(start timestamp) (HAI Time() - start timestamp)
(integration (can game (can game a more
efine HAL_OK (0)

Figure 50.

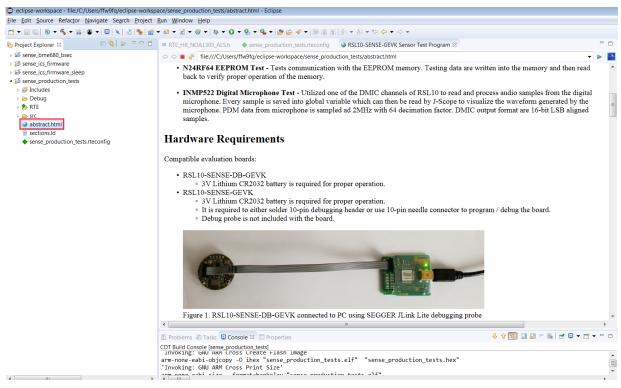
CMSIS also provides software timers and applications task manager abstraction layers to enable management of specific tasks and timing within the event kernel.



IVI	oddies
	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.
L	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* 





onsemi, ONSEMI, and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "onsemi" or its affiliates and/or subsidiaries in the United States and/or other countries. onsemi owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of onsemi's product/patent coverage may be accessed at <a href="https://www.onsemi.com/site/pdf/Patent-Marking.pdf">www.onsemi.com/site/pdf/Patent-Marking.pdf</a>, onsemi 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 not available for sale to consumers. The board is only intended for research, development, demonstration and evaluation purposes and will 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 purpose is strictly prohibited.

THE BOARD IS PROVIDED BY ONSEMI TO YOU "AS IS" AND WITHOUT ANY REPRESENTATIONS OR WARRANTIES WHATSOEVER. WITHOUT LIMITING THE FOREGOING, ONSEMI (AND ITS LICENSORS/SUPPLIERS) HEREBY DISCLAIMS ANY AND ALL REPRESENTATIONS AND WARRANTIES IN RELATION TO THE BOARD, ANY MODIFICATIONS, OR THIS AGREEMENT, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY AND ALL REPRESENTATIONS AND WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, AND THOSE ARISING FROM A COURSE OF DEALING, TRADE USAGE, TRADE CUSTOM OR TRADE PRACTICE.

onsemi 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 **onsemi** shall not constitute any representation or warranty by **onsemi**, and no additional obligations or liabilities shall arise from **onsemi** having provided such information or services.

onsemi products including 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. You agree to indemnify, defend and hold harmless onsemi, its directors, officers, employees, representatives, agents, subsidiaries, affiliates, distributors, and assigns, against any and all liabilities, losses, costs, damages, judgments, and expenses, arising out of any claim, demand, investigation, lawsuit, regulatory action or cause of action arising out of or associated with any unauthorized use, even if such claim alleges that onsemi was negligent regarding the design or manufacture of any products and/or 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 **onsemi** 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.

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

LIMITATIONS OF LIABILITY: **onsemi** 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 **onsemi** is advised of the possibility of such damages. In no event shall **onsemi**'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.

The board is provided to you subject to the license and other terms per **onsemi**'s standard terms and conditions of sale. For more information and documentation, please visit www.onsemi.com.

#### PUBLICATION ORDERING INFORMATION

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

onsemi 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

# **X-ON Electronics**

Largest Supplier of Electrical and Electronic Components

Click to view similar products for RF Development Tools category:

Click to view products by ON Semiconductor manufacturer:

Other Similar products are found below :

MAAM-011117 MAAP-015036-DIEEV2 EV1HMC1113LP5 EV1HMC6146BLC5A EV1HMC637ALP5 EVAL-ADG919EBZ ADL5363-EVALZ LMV228SDEVAL SKYA21001-EVB SMP1331-085-EVB EV1HMC618ALP3 EVAL01-HMC1041LC4 MAAL-011111-000SMB MAAM-009633-001SMB MASW-000936-001SMB 107712-HMC369LP3 107780-HMC322ALP4 SP000416870 EV1HMC470ALP3 EV1HMC520ALC4 EV1HMC244AG16 EV1HMC539ALP3 EV1HMC6789BLC5A MAX2614EVKIT# 124694-HMC742ALP5 SC20ASATEA-8GB-STD MAX2837EVKIT+ MAX2612EVKIT# MAX2692EVKIT# EV1HMC629ALP4E SKY12343-364LF-EVB 108703-HMC452QS16G EV1HMC863ALC4 EV1HMC427ALP3E 119197-HMC658LP2 EV1HMC647ALP6 ADL5725-EVALZ MAX2371EVKIT# 106815-HMC441LM1 EV1HMC1018ALP4 UXN14M9PE MAX2016EVKIT EV1HMC939ALP4 MAX2410EVKIT MAX2204EVKIT+ EV1HMC8073LP3D SIMSA868-DKL SIMSA868C-DKL SKY65806-636EK1 SKY68020-11EK1