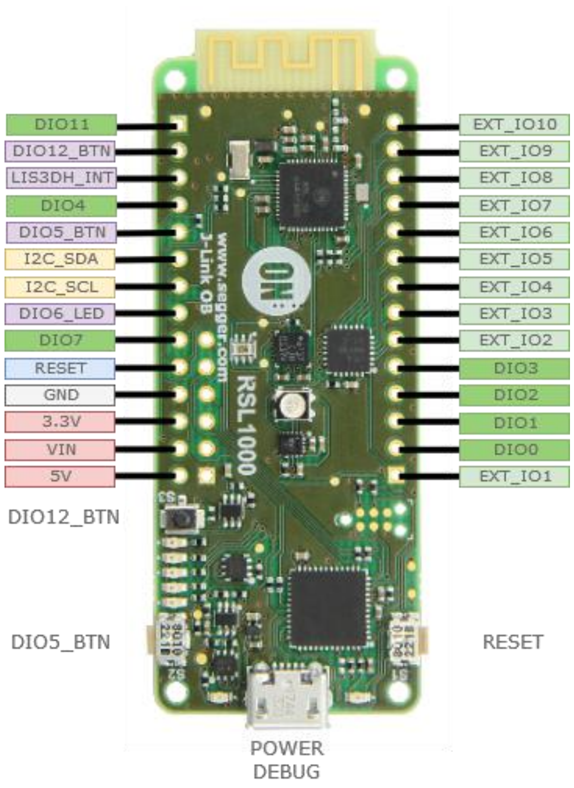
Klaus Kohl-Schoepe – Arrow Central Europe GmbH

Using RSL1000 (V1.0)

RSL1000 is an Arduino MKR size evaluation board based on the OnSemi RSL10 Bluetooth® 5 Radio System-on-Chip (SoC) or RSL10 SiP (at RSL1000S) developed from Arrow and Trenz Electronics. The same form factor is also used in some other FPGA boards like MAX1000 (Intel MAX10), CYC1000 (Intel Cyclone 10), or SMF2000 (Microchip Smart-Fusion2).

Next to the RSL10 there are a lot of peripheral available on the RSL1000:

- NCP5623 - Triple Output I2C Controlled RGB LED Driver
- NOA1305 - Ambient Light Sensor with I2C Interface and Dark Current Compensation
- PCA9655 - Remote 16-bit I/O Expander for I2C Bus with Interrupt
- LIS3DH - MEMS digital output motion sensor



Because all development software from OnSemi are free of charge and supports the integrated J-Link OB JTAG debugger on the board you need only the board, USB cable, an PC (I will use Windows) 10 and an internet connection to download all software and drivers.

There is also a getting started guide for the RSL1000(S) but be careful because until version 1.2 of this guide the pinout of the board is wrong. On the left side you see the correct version.

I write this documentation because the getting started guide covers only the preinstalled demo using an actual iPhone. Here I will cover the required software installation and show some examples using the actual available software.

I also realize a RSL10-Collection with ~1.5GByte who includes all documentation and the examples showed here. To get access please send an email to klaus.kohl-schoepe@arrow.com.

1.Install Software

Most of OnSemi RSL10 documentation and software are at https://www.onsemi.com/products/wireless-connectivity/wireless-rf-transceivers/rsl10

Following programs are used for this development:

- OnSemi IDE (actual version 3.5 – still listed as 3.4 on the web)
- RSL10 Software Package (version 3.5 – includes RSL10 CMSIS pack)
- ARM.CMSIS Pack (from https://www.keil.com/pack/ARM.CMSIS.5.6.0.pack)

Helpful documentation are:

- RSL10 Getting Started (Rev. P8) (also included in the documentation package)
- RSL10 Documentation Package (Rev. 3.5)

During installation of OnSemi IDE maybe you will be ask to install the newest J-Link (V6.96) driver. During start of the IDE you have to select a workspace folder. Please be careful don't use a space in the workspace path because this can be a problem for Eclipse IDE.

In the getting started guide on page 8 you see how to install the CMSIS pack for RSL10 on the OnSemi IDE. The actual version is "ONSemiconductor.RSL10.3.5.285.pack". For some examples you need also the ARM.CMSIS pack (version 5.6.0 for this RSL10 pack) if you see error "API is missing".

Because the RSL1000 use the same buttons and leds the first examples like Blinky should work without changes. All other interfaces like UART or I2C using other DIO's so you have to adapt the RTE_Device.h file (in Device/RSL10 folder) using the CMSIS Configuration Wizard. Also if no changes possible please check the project folder that all files are writable.

2.Examples

All examples are also available on Github: www.github.com/klakosch/RSL1000

2.1. blinky

The first example in the guide is the blinky example – so please load as documented in the getting started guide. Because the button and LEDs it works out of the box. The first 4 LEDs are controlled by the I2C port expander there are (nearly) off.

All outputs from this program are disabled. But you can activate one of two interfaces to show the debug output. The first one is the J-Link RTT interface using SWD/JTAG debugger to communication with the PC. For this simple add "OUTPUT_INTERFACE=1" to the preprocessor symbols in the assembler/C compiler tool settings and recompile the project.

If you want to use the UART interface over the debug interface you have to use "OUTPUT_INTERFACE=0" as a preprocessor symbol. But there are some steps more to do to get an output to PC:

- Disable the JTAG on pins DIO[13] … DIO[15] (see chapter RSL1000 I/O's)
- Change UART_TX (to 14) and UART_RX (to 13) in printf.c
- Change RX (to 13) and TX (to 14) data gpio pad in RTE_Device.h (not really required)

But be careful because printf.c is not copied to the project and used also for future exports. You should clean and recompile the project because of changed preprocessor symbols.

In my installation of IDE the preprocessor symbols are not saved and reopening the project requires again to set this symbols.

2.2. hardware-selftest

This program is used to test the RSL1000. All peripherals of the RSL1000 are configured and drivers for all four I2C devices are included. But only the additional LEDs and the RGB LED are used. With the button on top of the PCB you can speed up the changes.

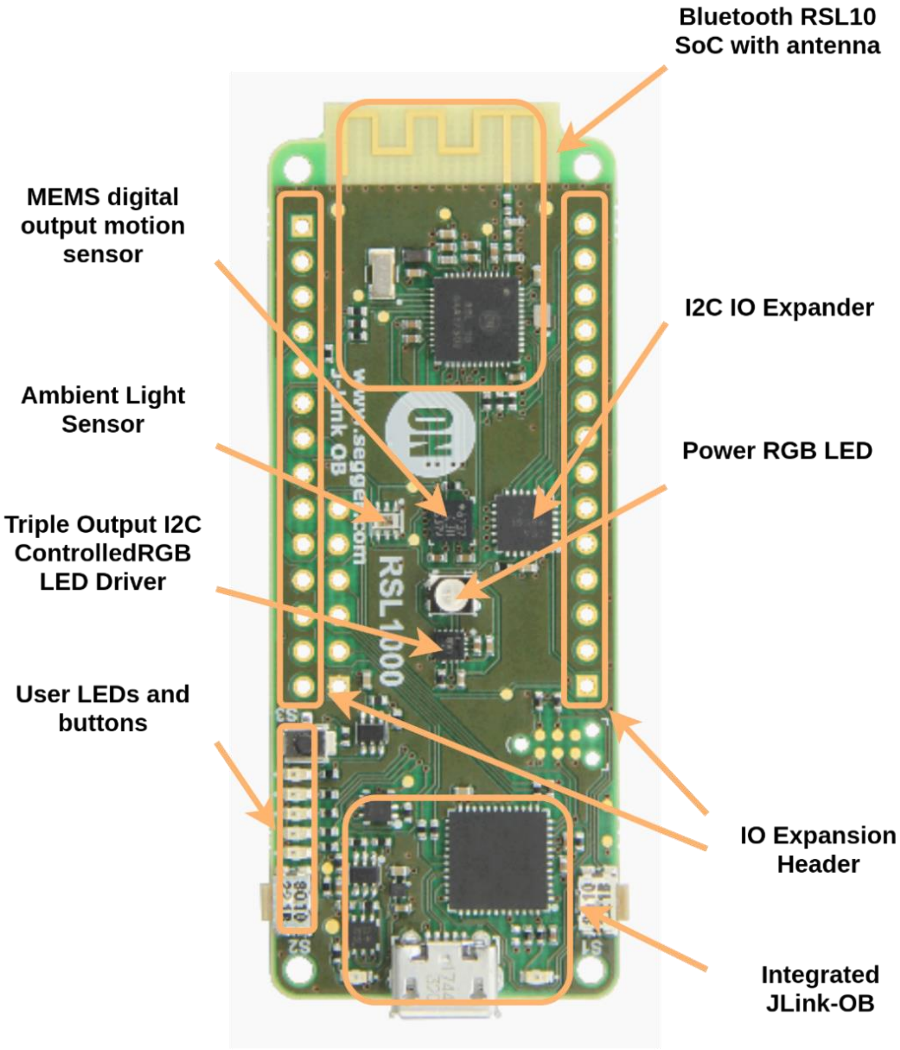
2.3. RSL1000

A complete demonstration of RSL1000 interfaces shows:
- changing LEDs (using GPIO-Expander)
- RGB LEDs (using RGB driver)
- Light sensor (showing values on terminal and RGB LEDs
- Acceleration sensor (also on terminal and RGB LEDs)
You can use the button at the side next to the LEDs to change between these four modes.

I used the same driver as in hardware-selftest so it is easier to modify this demo.

3. Background Information’s

There are a lot of peripherals available on the board:



3.1. RSL1000 Power Supply

There are three options to supply this board:

- 3.3V direct on pin J2-12 (3.3V)
- 4-5.5V on pin J2-13 (VIN)
- 5V from USB

USB or VIN – depending on what is higher - are also accessible at J2-14 (5V).

3.2. RSL10 Bluetooth 5 certified SoC

https://www.onsemi.com/products/wireless-connectivity/wireless-rf-transceivers/rsl10

RSL10 is a Bluetooth 5, multi-protocol radio System on Chip (SoC) based on Arm® Cortex®−M3 Processor there and in addition a LPDSP32 DSP core usable for audio algorithm. This chip allows ultra-low-power Bluetooth Low Energy wireless applications. Offering the industry’s lowest power consumption, RSL10 enables advanced wireless features while optimizing system size and battery life. Bluetooth Low Energy mesh is supported with the RSL10 Mesh Platform.

There are three version of RSL10 available:

- RSL10 SoC (System on Chip)
- RSL10 SiP (System in Package with integrated antenna)
- NCV-RSL10 SoC certified for automotive

3.3. RSL1000 I/O’s

The pinout for the I/O’s on the RSL1000 is visible on the first page. Here are more information’s about how the DIO of RSL10 are used on this board.

DIO	Usage	J1	J2	J4	Port Name
DIO[0]	I/O	2			DIO0
DIO[1]	I/O	3			DIO1
DIO[2]	I/O	4			DIO2
DIO[3]	I/O	5			DIO3
DIO[4]	I/O		4		DIO4
DIO[5]	Button (open=high)		5		DIO5_BTN
DIO[6]	Led 5 (high=on)		8		DIO6_LED
DIO[7]	I/O		9		DIO7
DIO[8]	I2C data		6		I2C_SDA
DIO[9]	I2C clock		7		I2C_SCL
DIO[10]	Accelerator Interrupt		3		LIS3DH_INT
DIO[11]	I/O		1		DIO11
DIO[12]	I/O and Reflash Button (open=high)		2		DIO12_BTN
DIO[13]	UART RX (TX from Debugger)			2	OB_TX
DIO[14]	UART TX (RX to Debugger)			5	OB_RX
DIO[15]	I/O or Debug SWO				OB_SWO

Connector J1 adds some Pins from the PCA9655 I2C Port Expander:

DIO	Usage	J1	Port Name
	PCA9655 IO0_0	1	EXT_IO1
DIO[0]	I/O	2	DIO0
DIO[1]	I/O	3	DIO1
DIO[2]	I/O	4	DIO2
DIO[3]	I/O	5	DIO3
	PCA9655 IO0_1	6	EXT_IO2
	PCA9655 IO0_2	7	EXT_IO3
	PCA9655 IO0_3	8	EXT_IO4
	PCA9655 IO0_4	9	EXT_IO5

	PCA9655 IO0_5	10	EXT_IO6
	PCA9655 IO0_6	11	EXT_IO7
	PCA9655 IO0_7	12	EXT_IO8
	PCA9655 IO1_0	13	EXT_IO9
	PCA9655 IO1_1	14	EXT_IO10

Connector J2 includes power and other I/O’s from the RSL10

DIO	Usage	J2	Port Name
DIO[11]	I/O	1	DIO11
DIO[12]	I/O and Reflash Button	2	DIO12_BTN
DIO[10]	Accelerator Interrupt	3	LIS3DH_INT
DIO[4]	I/O	4	DIO4
DIO[5]	Button	5	DIO5_BTN
DIO[8]	I2C data	6	I2C_SDA
DIO[9]	I2C clock	7	I2C_SCL
DIO[6]	Led 5	8	DIO6_LED
DIO[7]	I/O	9	DIO7
	Reset Input	10	RESET
	Ground	11	GND
	3.3V processor supply after LDO	12	3.3V
	Additional power supply 4-5.5V	13	VIN
	USB or VIN voltage before LDO	14	5V

Connector J4 is the short one next to J2 and includes power and other I/O’s from the RSL10

DIO	Usage	J4	Port Name
	Ground	1	GND
DIO[13]	UART RX (TX from Debugger)	2	OB_TX
DIO[16]	JTAG/SWD clock	3	OB_SWCLK
DIO[15]	JTAG test data out	4	OB_SWO
DIO[14]	UART TX (RX to Debugger)	5	OB_RX
DIO[17]	Debug SWDIO/JTAG TMS	6	OB_SWDIO

If you want to use UART RX/TX on this board you have to deactivate the CM3_JTAG_DATA_EN and CM3_JTAG_TRST_EN bit in DIO_JTAG_SW_PAD_CFG. One way is to disable JTAG data enable and TRST enable in the CMSIS Configuration Manager. The other way is to change register 0x4000079c in the SystemInit routine (system_rsl10.c):

```
*((uint32_t*) (0x4000079c)) = 0x1d;
```

3.4. PCA9655 Port Expander

https://www.onsemi.com/pdf/datasheet/pca9655e-d.pdf

I2C address: 0b**0100000**x (0x20 in program)

With this chip additional 10 I/O and 4 LEDs on RSL1000 are controlled by I2C.

The PCA9655E is a 16 lines general purpose input output (GPIO) expansion controlled by I2C. The PCA9655E can change the direction and value of each port and can activate a polarity inversion for reading the port. At power on, all I/Os default to inputs with an pull-up. That’s the reason why the LEDs are slightly dimming if this chip is not initialized.

Expander	Usage	J1	Port Name
IO0_0	I/O	1	EXT_IO1
IO0_1	I/O	6	EXT_IO2
IO0_2	I/O	7	EXT_IO3
IO0_3	I/O	8	EXT_IO4
IO0_4	I/O	9	EXT_IO5
IO0_5	I/O	10	EXT_IO6
IO0_6	I/O	11	EXT_IO7
IO0_7	I/O	12	EXT_IO8
IO1_0	I/O	13	EXT_IO9
IO1_1	I/O	14	EXT_IO10
IO1_2	(not used)		
IO1_3	(not used)		
IO1_4	LED 1		LED1
IO1_5	LED 2		LED2
IO1_6	LED 3		LED3
IO1_7	LED 4		LED4

Following commands are available.

Address	Command	Description
$20	$00	Input Port 0 (IO0_0 … IO0_7)
$20	$01	Input Port 1 (IO1_0 … IO1_7)
$20	$02	Output Port 0
$20	$03	Output Port 1
$20	$04	Polarity Inversion Port 0 (1: Input inverted)
$20	$05	Polarity Inversion Port 1 (1: Input inverted)
$20	$06	Configuration Port 0 (1=Input with Pull-Up; 0=Output)
$20	$07	Configuration Port 1 (1=Input with Pull-Up; 0=Output)

Options like interrupt output or changing of I2C address are not supported on the RSL1000.

3.5. NCP5623 Triple Output RGB LED Driver

https://www.onsemi.com/pdf/datasheet/ncp5623-d.pdf

Address: 0b**0111000**0 (0x38 in program - write only device)

The NCP5623 mixed analog circuit is a triple output LED driver dedicated to the RGB illumination or backlight LCD display and controls the three-color LED on the RSL.

Address	Command	Description
$38	%000xxxxx	System Shut Down (xxxxx ignored)
$38	%001xxxxx	Set maximum output LED current (xxxxx = value)
$38	%010xxxxx	Set Red PWM (xxxxx = value)
$38	%011xxxxx	Set Green PWM (xxxxx = value)
$38	%100xxxxx	Set Blue PWM (xxxxx = value)
$38	%101xxxxx	Set Up the IEND Upward (xxxxx = value)
$38	%110xxxxx	Set Up the IEND Downward (xxxxx = value)
$38	%111xxxxx	Set Up the Gradual Dimming Time and run (xxxxx = n*8ms)

3.6. NOA1305 Ambient Light Sensor

https://www.onsemi.com/pdf/datasheet/noa1305-d.pdf

I2C address: 0b**0111001**x (0x39 in program)

The NOA1305 ambient light sensor (ALS) is designed for handheld applications and integrates a 16-bit ADC with I2C interface, internal clock oscillator and a power down mode. The built in dynamic dark current compensation and precision calibration capability coupled with excellent IR and 50/60 Hz flicker rejection enables highly accurate measurements from very low light levels to full sunlight. The device can support simple count equals lux readings in interrupt−driven or polling modes. The NOA1305 employs proprietary CMOS image sensing technology from ON Semiconductor to provide large signal to noise ratio (SNR) and wide dynamic range (DR) over the entire operating temperature range. The optical filter used with this chip provides a light response similar to that of the human eye.

Address	Register	Value	Description
$39	$00	$00 $08	Power Down Power On
$39	$01	$10	Reset ALS data to 0000
$39	$02	x	Integration Time 6.25ms * 2^(7-x) => 6.25ms … 800ms
$39	$03	$01 $02 $03	Interrupt Low->High Interrupt High->Low Interrupt inactive (always High)
$39	$04	x	Interrupt threshold, LSB
$39	$05	x	Interrupt threshold, MSB
$39	$06	x	ASL data, LSB
$39	$07	x	ALS data, MSB
$39	$08	$19	Device ID, LSB
$39	$09	$05	Device ID, MSB

3.7. LIS3DH 3-axis Accelerometer

https://www.st.com/resource/en/datasheet/cd00274221.pdf

I2C address: 0b**0011000**x (0x18 in program)

The LIS3DH is a ultra-low-power high-performance three-axis accelerometer. The INT1 pin from the chip is routed to the DIO[10] pin of RSL10. There are a lot of register but for starting measurement and reading the data you need only a few of this:

Address	Register	Value	Description
$18	$0F	$33	WHO_AM_I - this register read only value is $33
$18	$01	x	CTRL_REG1 configs: - Bit 7:4: 0=Down, 1..7=1/10/25/50/100/200/400Hz - Bit 3: Low-Power-Mode enable (default 0) - Bit 2: Z-axis enable (default 1) - Bit 1: Y-axis enable (default 1) - Bit 0: X-axis enable (default 1)
$18	$28-$2D	3*16bit	X-/Y-/Z-axis data (LSB first)