RS14100 EVK

Getting Started Guide

Version 1.5

October 2018

Redpine Signals, Inc.
2107 North First Street, #540
San Jose, CA 95131.
Tel: (408) 748-3385
Fax: (408) 705-2019
Email: sales@redpinesignals.com
Website: www.redpinesignals.com

Disclaimer:

The information in this document pertains to information related to Redpine Signals, Inc. products. This information is provided as a service to our customers and may be used for information purposes only.
Redpine assumes no liabilities or responsibilities for errors or omissions in this document. This document may be changed at any time at Redpine's sole discretion without any prior notice to anyone. Redpine is not committed to updating this document in the future.

About this Document

R14100 WiSeMCU™ products include embedded Wi-Fi, TCP/IP, BT 5 and ZigBee Pro stacks along with a Cortex-M4F processor for customer applications. This guide assists users in the initial setup and demonstration of the MCU and Wireless SAPIs using example projects. The guide also explains how to create a new project, compile, download and debug.

Table Of Contents

Table of Figures

Table of Tables

1 Wireless and MCU Simple API (SAPI) Features

- Platform-independent, interrupt-driven drivers written in C.
- SAPIs provide a simpler functional interface and eliminate the need to manage the low-level host interface protocol.
- Supports bare-metal and FreeRTOS out-of-box; other RTOS can be supported through OS abstraction changes.
- Supports Keil uVision – can be ported to other toolchains.

2 Requirements for Running MCU and Wireless SAPIs Project

2.1 Overview

The Redpine MCU and Wireless Simple API (SAPI) is a comprehensive collection of peripheral APIs and driver code to simplify application development on Redpine WiSeMCU. Users can develop application firmware without having to learn the underlying peripheral register interface and other details.
The SAPI is intended to run on Cortex-M4F core available in Redpine's WiSeMCU products. With its uniform API, SAPIs enable easy migration between Redpine products. SAPIs provide CMSIS-Driver API for MCU peripherals that are CMSIS-compliant.

For more details on Wireless and MCU SAPIs, refer to Redpine_Wireless_SAPI_Guide and Redpine_MCU_SAPI_Manual respectively. Refer to EVK User Guide for more details on the EVK board.The Updated EVK Board's Schematics and Pin-out details will be available from the next release.

Power Sequence:

Make sure to power up the USB port of the board before connecting the CMSIS DAP port.

2.2 Hardware Requirements

1. A Micro USB cable (included with the kit)
2. A Wi-Fi access point
3. A desktop PC, Smartphone or Tablet with wired or wireless connectivity
4. Redpine RS14100 EVK with the daughter board
5. CMSIS DAP internal debug adapter board (included with the kit)
6. Segger J-Link debug probe with 10 pin connector cable (external debug adaptor)

2.3 Software Requirement

1. Keil IDE (MDK-ARM). Download the latest version from the link.**https://www.keil.com/download/product/**
2. Keil DFP. Download the latest version from the link.**http://www.redpinesignals.com/downloads/RS14100_DFP/Redpine.RS14100_DFP.1.0.1.pack**
3. Serial terminal application (TeraTerm, Cute, Putty, Docklight etc.)

2.4 Quick Start with Redpine SAPIs

Install Keil uVision version 5.0 or higher along with the J-Link drivers for detecting the JTAG. Segger J-Link is required to download and debug the project.
Install Keil Redpine. RS14100_DFP.1.0.1.pack to get the Redpine device in device library of Keil.

1. Power up the EVK by using USB cable plugged into port J23 (POWER).
2. Connect the DAP Adaptor board to J7 header (JTAG/TRACE); the adaptor card is provided with the kit.
3. Connect the UART port of EVK to PC for getting UART prints.
4. Open the serial terminal utility and set the baud rate to 115200, stop bit 1, parity None to open the COM port that is detected in device manager once the EVK is connected.

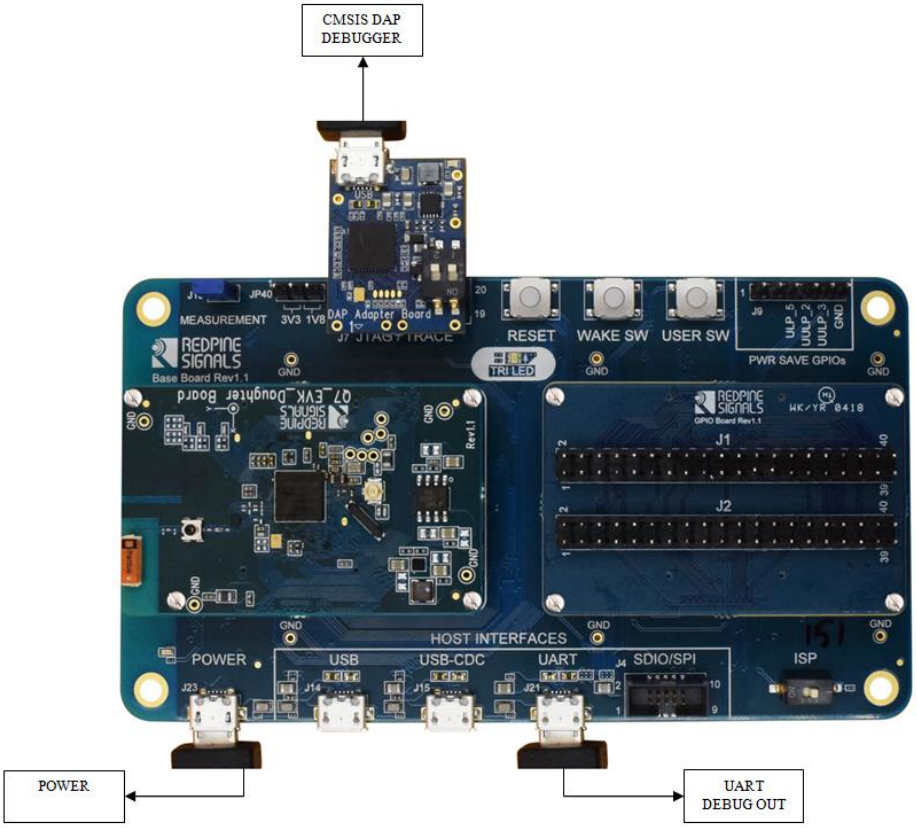


Figure 1: RS14100 with Daughter Board (With Redpine CMSIS DAP Debugger)

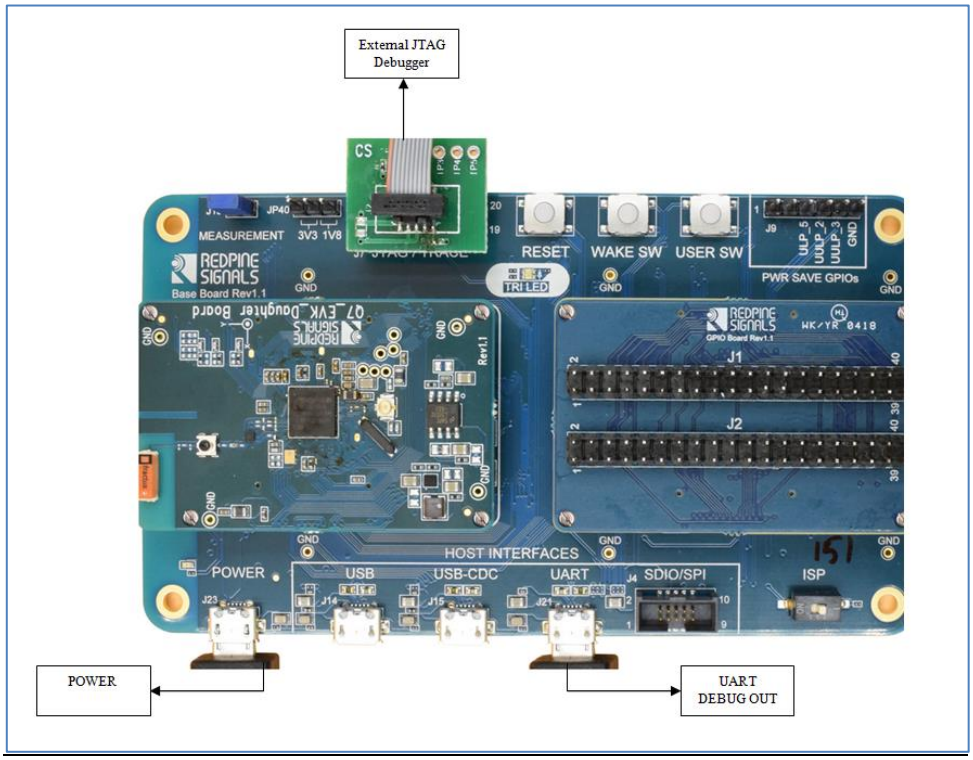


Figure 2: RS14100 with Daughter Board (With External Debugger)

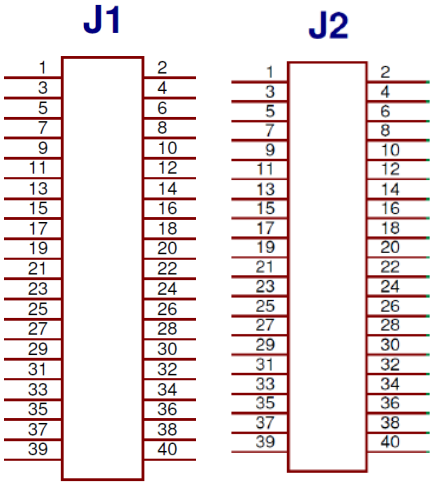


Figure 3: GPIO Header (J1 &J2) Pins

Partial GPIO pins are available for CCI, Ethernet and CAN with the QMS SoC package.

Pin Number	Pin Name	Q7 EVK	Functionality
J1.7	GPIO_6	SUPPORTED	I2C1_SDA,UART2_TX,SSIM_MOSI,QEI_IDX,RMII_TXD1,
J1.9	GPIO_7	SUPPORTED	I2C1_SCL,UART2_RX,QEI_PHB,RMII_TXD0,I2S_2CH_CLK
J1.11	GPIO_8	SUPPORTED	USART0_RX_PIN ,QEI_PHA,RMII_RXCLK
J1.13	GPIO_9	SUPPORTED	USART0_TX_PIN ,SSIM_CS0,QEI_DIR ,SIO_3, RMII_TXEN
J1.15	GPIO_10	SUPPORTED	USART0_CLK_PIN,RMII_ RXD0,
J1.19	GPIO_11	SUPPORTED	I2S_WSCLK,RMII_MDC
J1.21	GPIO_12	SUPPORTED	SSIM_MOSI,I2C1_SCL,CCI_DATA0 ,RMII_MDO,
J1.23	GPIO_13	NA	
J1.25	GPIO_14	NA	
J1.27	GPIO_15	SUPPORTED	UART2_TX,I2C1_SDA,CCI_DATA3 ,SSIM_MISO,RMII_RXD0,
J1.29	GPIO_16	NA	RMII_CRS_DV in Q7 not coming out.
J1.33	GPIO_17	NA	
J1.35	GPIO_18	NA	
J1.37	GPIO_19	NA	
J1.39	GPIO_20	NA	
J1.6	GPIO_21	NA	
J1.8	GPIO_22	NA	
J1.10	GPIO_23	NA	
J1.12	GPIO_24	NA	
J1.14	GPIO_35	NA	
J1.16	GPIO_36	NA	
J1.20	GPIO_37	NA	
J1.22	GPIO_38	NA	
J1.24	GPIO_39	NA	
J1.26	GPIO_40	NA	
J1.28	GPIO_41	NA	
J1.30	GPIO_42	NA	
J1.34	GPIO_43	NA	
J1.36	GPIO_44	NA	
J1.38	GPIO_45	NA	
J1.40	GPIO_46	NA	
J2.5	GPIO_47	NA	
J2.7	GPIO_48	NA	
J2.9	GPIO_49	SUPPORTED	
J2.11	GPIO_50	SUPPORTED	SIO_2,SCT_OUT_1
J2.13	GPIO_51	SUPPORTED	SCT_OUT_0
J2.15	GPIO_52	NA	
J2.19	GPIO_53	NA	
J2.21	UULP_VBAT_GPIO_0	SUPPORTED	Power save application

Pin Number	Pin Name	Q7 EVK	Functionality
J2.23	UULP_VBAT_GPIO_1	NA	
J2.25	UULP_VBAT_GPIO_2	SUPPORTED	Power save application
J2.27	UULP_VBAT_GPIO_3	SUPPORTED	Power save application
J2.29	UULP_VBAT_GPIO_4	NA	
J2.6	ULP_GPIO_0	SUPPORTED	LED_PIN0, Comparator1 positive input pin
J2.8	ULP_GPIO_1	SUPPORTED	ULP_SPI_DOUT, Comparator1 negative input pin
J2.10	ULP_GPIO_2	NA	
J2.12	ULP_GPIO_3	NA	
J2.14	ULP_GPIO_4	SUPPORTED	SSIM_CLK , DAC_OUTPUT,OPAMP_OUT
J2.16	ULP_GPIO_5	SUPPORTED	IR_OUTPUT
J2.20	ULP_GPIO_6	SUPPORTED	ULP_UART_RX,I2S_2CH_DIN_0
J2.22	ULP_GPIO_7	SUPPORTED	GSPI_CLK ,ULP_UART_TX,OPAMP_INP,I2S_2CH_DOUT_0
J2.24	ULP_GPIO_8	SUPPORTED	GSPI_CS0 ,
J2.26	ULP_GPIO_9	SUPPORTED	GSPI_MOSI,
J2.28	ULP_GPIO_10	SUPPORTED	CAN_RX,IR_INPUT
J2.30	ULP_GPIO_11	SUPPORTED	CAN_TX,
J2.34	ULP_GPIO_12	NA	
J2.36	ULP_GPIO_13	NA	
J2.38	ULP_GPIO_14	NA	
J2.40	ULP_GPIO_15	NA	

Table 1: Peripheral/GPIO Board Rev 1.1 and below

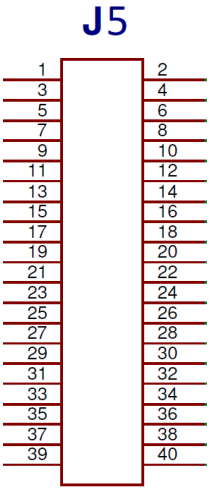


Figure 4: GPIO Header J5 Pins

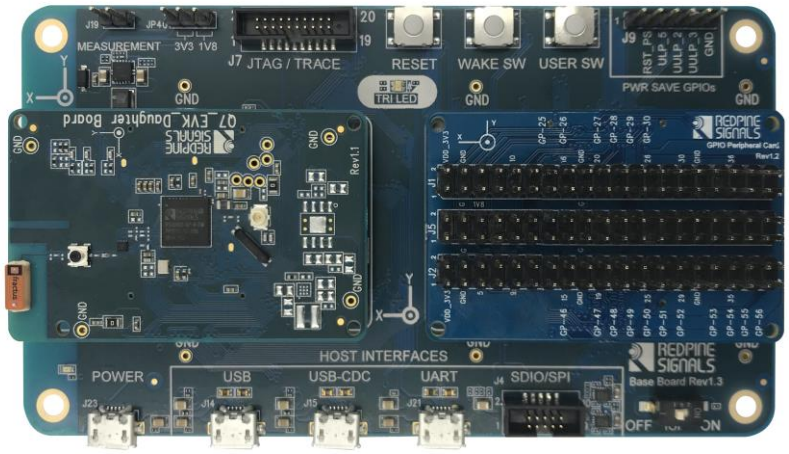


Figure 5 Baseboard Rev 1.3 and GPIO Board 1.2 with J5 Header

Pin Number	Pin Name	Q7 EVK	Functionality
J1.7	GPIO_6	SUPPORTED	I2C1_SDA,UART2_TX,SSIM_MOSI,QEI_IDX,RMII_TXD1,
J1.9	GPIO_7	SUPPORTED	I2C1_SCL,UART2_RX,QEI_PHB,RMII_TXD0,I2S_2CH_CLK
J1.11	GPIO_8	SUPPORTED	USART0_RX_PIN ,QEI_PHA,RMII_RXCLK
J1.13	GPIO_9	SUPPORTED	USART0_TX_PIN ,SSIM_CS0,QEI_DIR ,SIO_3, RMII_TXEN
J1.15	GPIO_10	SUPPORTED	USART0_CLK_PIN,RMII_ RXD0,
J1.19	GPIO_11	SUPPORTED	I2S_WSCLK,RMII_MDC
J1.21	GPIO_12	SUPPORTED	SSIM_MOSI, I2C1_SCL, CCI_DATA0 ,RMII_MDO,
J1.23	GPIO_13	NA	
J1.25	GPIO_14	NA	
J1.27	GPIO_15	SUPPORTED	I2C1_SDA,CCI_DATA3 ,SSIM_MISO,RMII_RXD0,
J1.29	GPIO_16	NA	RMII_CRS_DV in Q7 not coming out.
J1.33	GPIO_17	NA	
J1.35	GPIO_18	NA	
J1.37	GPIO_19	NA	
J1.39	GPIO_20	NA	
J1.6	GPIO_21	NA	
J1.8	GPIO_22	NA	
J1.10	GPIO_23	NA	
J1.12	GPIO_24	NA	
J1.14	GPIO_25	NA	
J1.16	GPIO_26	NA	
J1.20	GPIO_27	NA	
J1.22	GPIO_28	NA	
J1.24	GPIO_29	NA	
J1.26	GPIO_30	NA	

Pin Number	Pin Name	Q7 EVK	Functionality
J1.28	GPIO_35	NA	
J1.30	GPIO_36	NA	
J1.34	GPIO_37	NA	
J1.36	GPIO_38	NA	
J1.38	GPIO_39	NA	
J1.40	GPIO_40	NA	
J2.5	GPIO_41	NA	
J2.7	GPIO_42	NA	
J2.9	GPIO_43	NA	
J2.11	GPIO_44	NA	
J2.13	GPIO_45	NA	
J2.15	GPIO_46	NA	
J2.19	GPIO_47	NA	
J2.21	GPIO_48	NA	
J2.23	GPIO_49	SUPPORTED	
J2.25	GPIO_50	SUPPORTED	SIO_2,SCT_OUT_1
J2.27	GPIO_51	SUPPORTED	SCT_OUT_0
J2.29	GPIO_52	NA	
J2.33	GPIO_53	NA	
J2.35	GPIO_54	NA	
J2.37	GPIO_55	NA	
J2.39	GPIO_56	NA	
J2.6	ULP_GPIO_0	SUPPORTED	LED_PIN0, Comparator1 positive input pin
J2.8	ULP_GPIO_1	SUPPORTED	ULP_SPI_DOUT, Comparator1 negative input pin
J2.10	ULP_GPIO_2	NA	
J2.12	ULP_GPIO_3	NA	
J2.14	ULP_GPIO_4	SUPPORTED	SSIM_CLK , DAC_OUTPUT,OPAMP_OUT
J2.16	ULP_GPIO_5	SUPPORTED	IR_OUTPUT
J2.20	ULP_GPIO_6	SUPPORTED	ULP_UART_RX,I2S_2CH_DIN_0
J2.22	ULP_GPIO_7	SUPPORTED	GSPI_CLK ,ULP_UART_TX,OPAMP_INP,I2S_2CH_DOUT_0
J2.24	ULP_GPIO_8	SUPPORTED	GSPI_CS0 ,
J2.26	ULP_GPIO_9	SUPPORTED	GSPI_MOSI,
J2.28	ULP_GPIO_10	SUPPORTED	CAN_RX,IR_INPUT
J2.30	ULP_GPIO_11	SUPPORTED	CAN_TX,
J2.34	ULP_GPIO_12	NA	
J2.36	ULP_GPIO_13	NA	
J2.38	ULP_GPIO_14	NA	
J2.40	ULP_GPIO_15	NA	

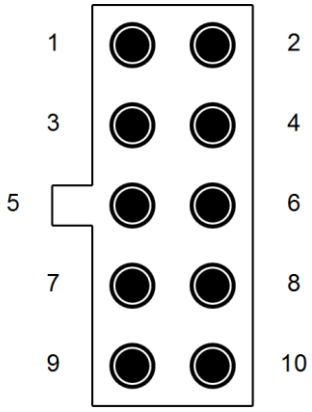
Pin Number	Pin Name	Q7 EVK	Functionality
J5.6	GPIO_57	NA	
J5.20	UULP_VBAT_GPIO_0	SUPPORTED	Power save application
J5.22	UULP_VBAT_GPIO_1	NA	
J5.24	UULP_VBAT_GPIO_2	SUPPORTED	Power save application
J5.26	UULP_VBAT_GPIO_3	SUPPORTED	Power save application
J5.28	UULP_VBAT_GPIO_4	NA	
J5.30	SDB_GPIO_1	NA	
J5.34	SDB_GPIO_2	NA	
J5.36	SDB_GPIO_3	NA	
J5.38	SDB_GPIO_4	NA	
J5.40	SDB_GPIO_5	NA	

Table 2 Peripheral/GPIO Board Rev 1.2 and above

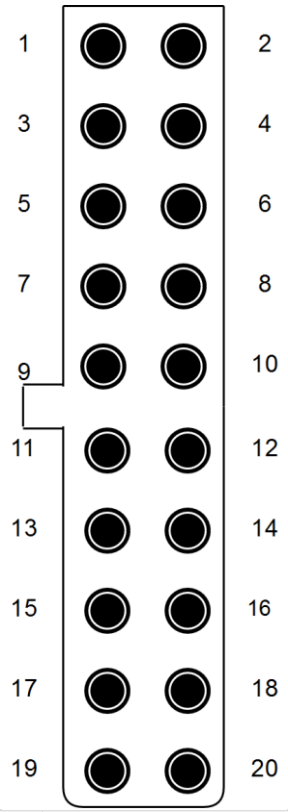
Board Port Descriptions

Port	Description
Measurement Pin (J19)	Can connect an ammeter across these pins to measure current. Otherwise, leave this covered
RF Supply Pin (J40)	Determines Voltage supplied to the RF. Leave this at 3V3 for this guide.
JTAG/TRACE (J7)	Debug adapter can be attached here.
RESET	Soft resets the M4 Processor
WAKE SW	Wake up interrupt source used in sleep mode.
USER SW	Button used by some example applications
POWER (J23)	Power supply to the board
USB (J14)	Not used in WiSeMCU mode
USB-CDC (J15)	Serial host interface. Used for flashing firmware to the RS14100
UART (J21)	Used to view UART messages sent by the M4 Processor. Also supplies power.
SDIO/SPI (J4)	Not used in WiSeMCU mode
ISP	Toggles In-System Programming. If OFF, M4 Processor code will run. If ON, NWP can be interacted with over USB-CDC. By default, should be OFF.

3 Cortex-M JTAG/TRACE 9 pin and 19 pin Header



Pin Number	Pin Name
1	VBATT_GATE
2	TMS
3	GND
4	TCK
5	GND
6	TDO
7	NC
8	TDI
9	NC
10	RESET_N



Pin Number	Pin Name
1	VBATT_GATE
2	TMS
3	GND
4	TCK
5	GND
6	TDO
7	NC
8	TDI
9	NC
10	M4_JTAG_RESET
11	TRACE_5V
12	M4_TRACE_CLK
13	TRACE_5V
14	M4_TRACE_D0
15	GND
16	M4_TRACE_D1
17	GND
18	M4_TRACE_D2
19	GND
20	M4_TRACE_D3

4 Starting a WiSeMCU Multi-Project View

Multiple examples are organized into multi-projects groups for UV5. To select a project group, double-click the appropriate multi-project file in the Keil examples directory.
e.g: *{Release$}*Examples\Reference_Projects\Peripheral_Projects\Keil
Individual examples are also present in the same path with the specific example folder.
e.g: *{Release$}*Examples\Reference_Projects\Peripheral_Projects\Keil\uart
Once a multi-project file is started, all the examples associated with that project will be shown in the Project view. Refer to the image below:

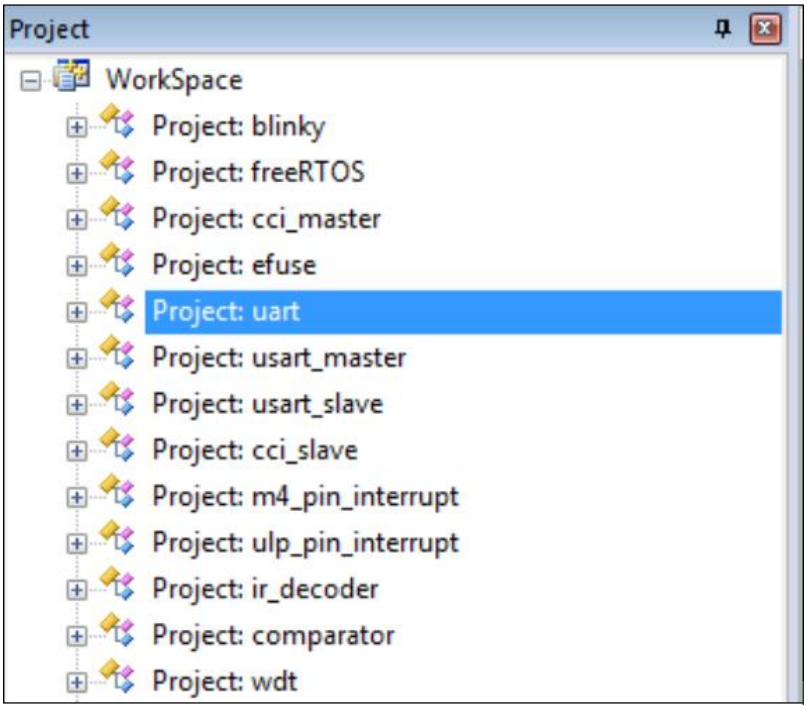


Figure 6: Multi-Project View

4.1 Target Selection

In Project settings, navigate to Device and select Redpine Signals-> RS14100_Series-> RS14100_1MB, then navigate to Debug tab. Select any debugger e.g: **"CMSIS-DAP Debugger"** from the drop-down menu and click the **"Use"** radio button (present beside the dropdown). Once done, hit **OK**. Refer to the image below:

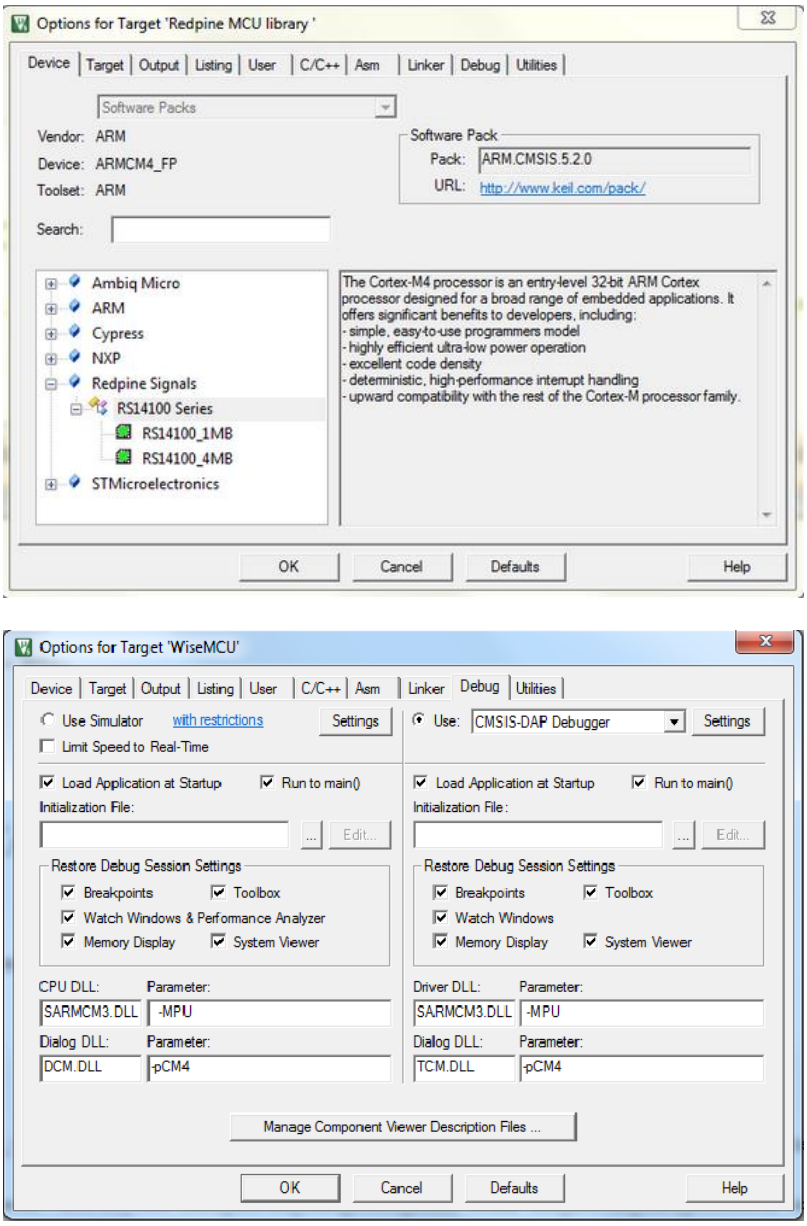


Figure 7: Debug Setting

4.2 Settings for running from RAM

1. In Keil Project, navigate to the **Project** tab and click on **Options for Target**. Select the RAM settings options and hit **OK**. Refer to the image below:

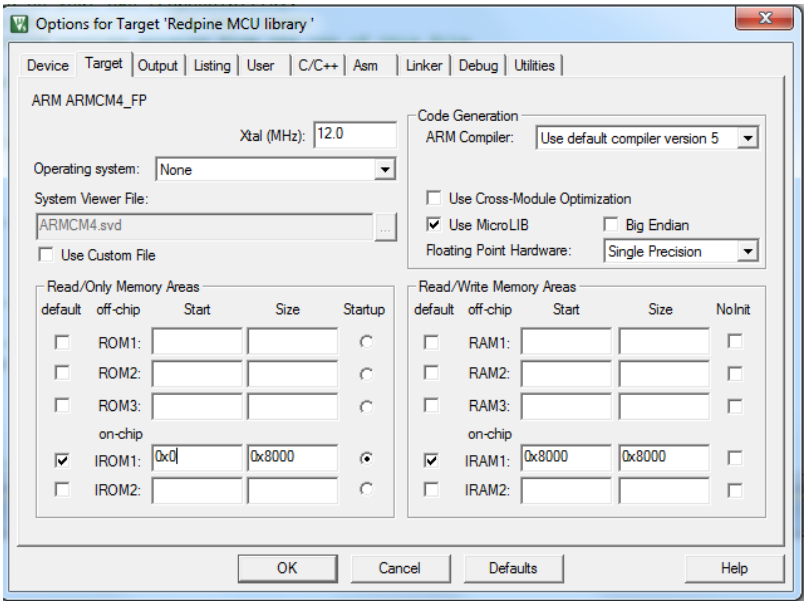


Figure 8: RAM Address Settings

2. When finished, go to Debug tab and select **"CMSIS-DAP Debugger"** as target driver. Browse and select **"Debug_SRAM0_0x00.ini"** as the initialization file and hit **OK**. Refer to the image below:

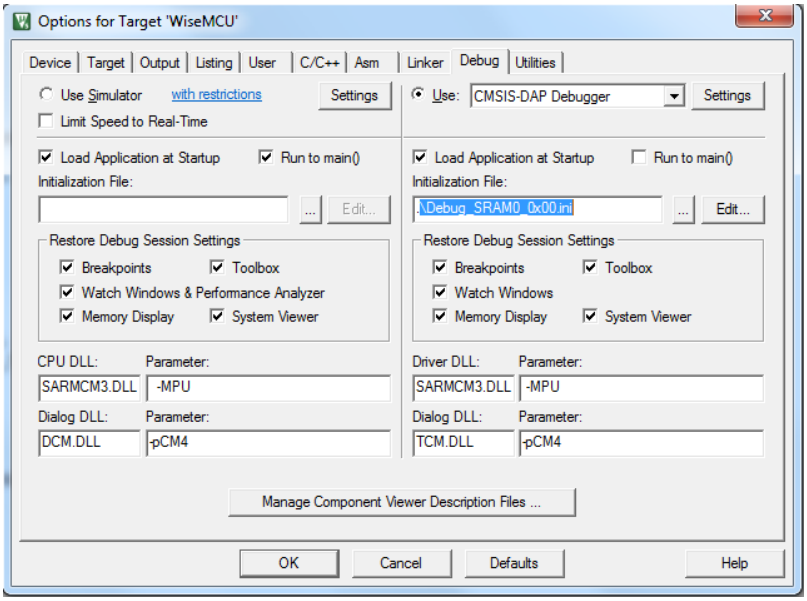


Figure 9: SRAM File Selection

3. In Debug window, click the **Settings** button. A modal window named **Cortex-M Target Driver Setup** will be prompted. There, under **Port** choose **SW** and clock settings as **1MHz**. Once done, hit **OK**. Refer to the image below:

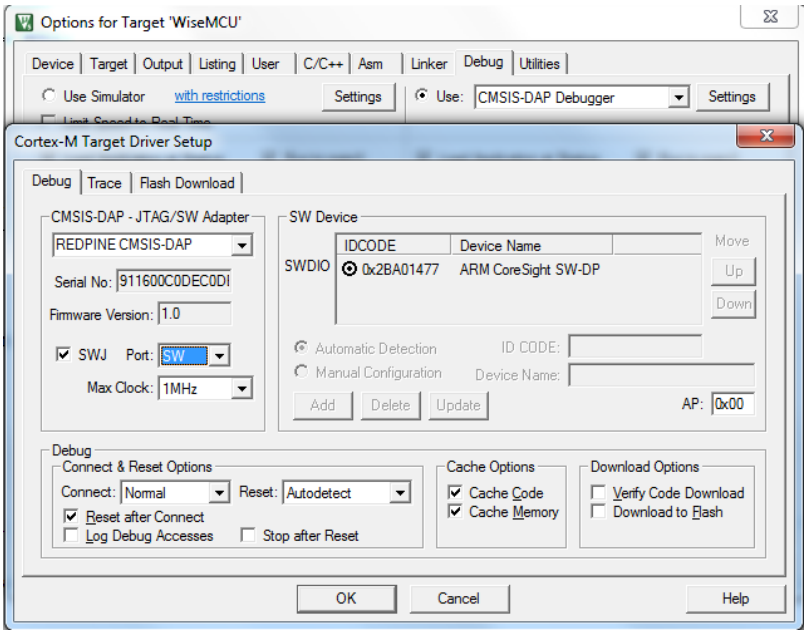


Figure 10: JTAG Clock Setting

5. Next, click on **Flash Download** and select the settings options as shown below. When finished, hit **OK.**

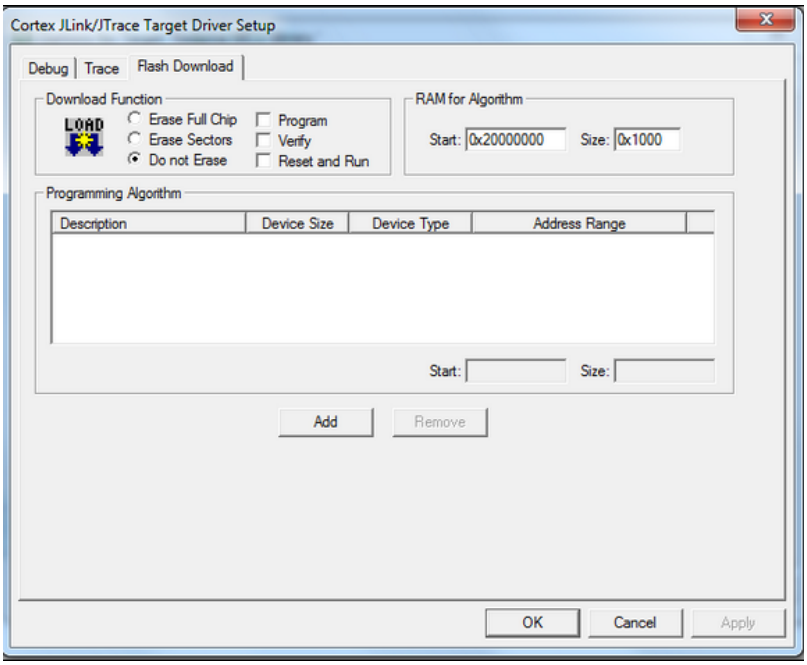
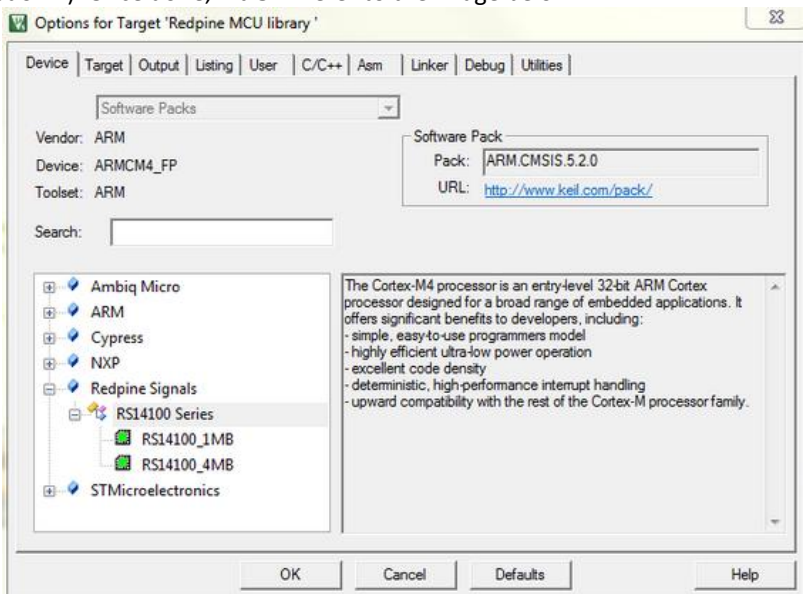
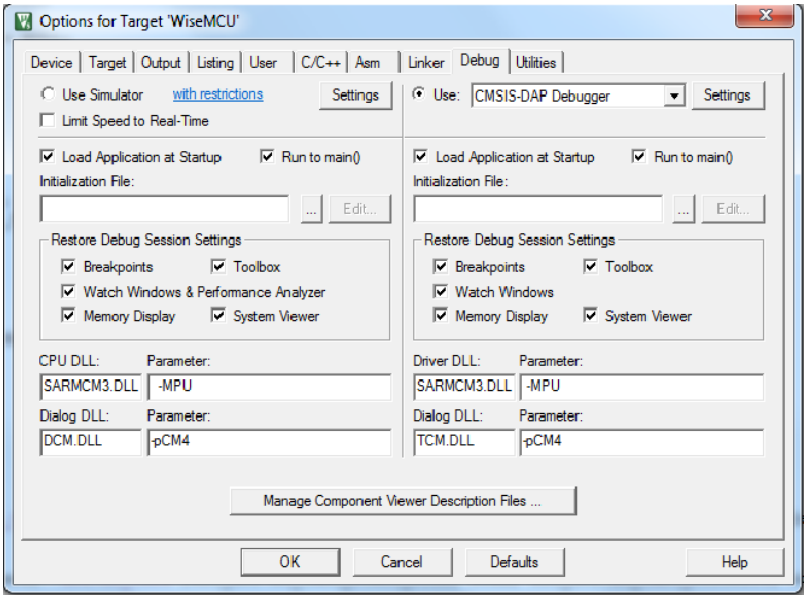


Figure 11: Flash Download Settings

4.3 Settings for running from Flash

In Project settings, navigate to Device and select Redpine Signals-> RS14100_Series-> RS14100_1MB, then navigate to **Debug** tab. Select any debugger e.g: **"CMSIS-DAP Debugger"** from the drop-down menu and click the **"Use"** radio button (present beside the dropdown). Once done, hit **OK**. Refer to the image below:



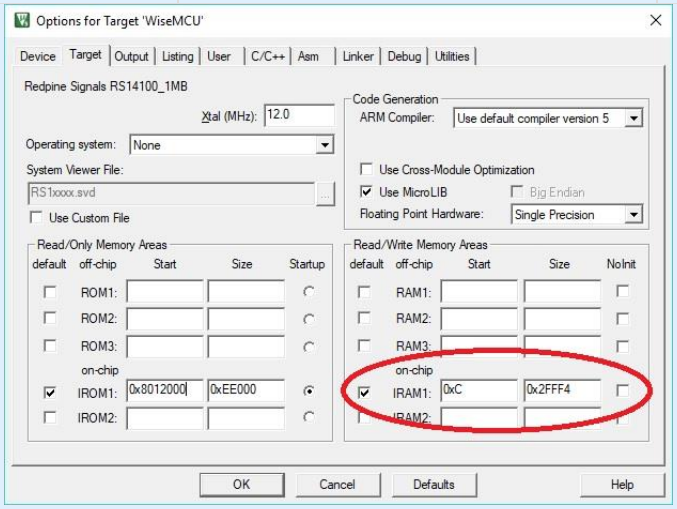


If settings are made for RAM then navigate to Device and select the RS14100_1MB device which will reset the setting for Flash programming.

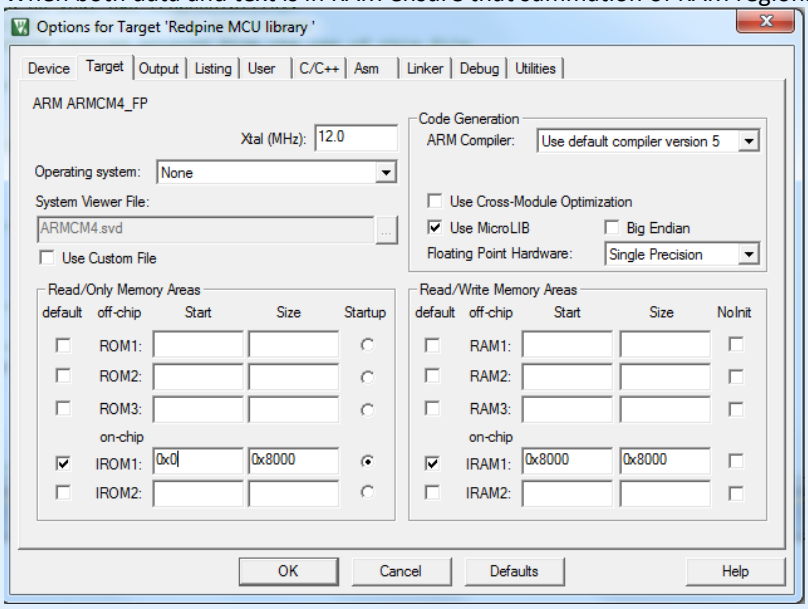
Both M4 and NWP has 192 KB as default memory configuraiton. User can change the memory configuration based on the feature bit map Bit 20 or Bit 21 in the RSI_EXT_CUSTOM_FEATURE_BIT_MAP.

When Text memory in flash user can refer below table

BIT Set	NWP Memory Size	M4 Memory Size	Size
20	320KB	64KB	0xFFF4
21	256KB	128KB	0x1FFF4
Default	192KB	192KB	0x2FFF4



When both data and text is in RAM ensure that summation of RAM regions should be with in the M4 memory size.



4.4 Changing the Default Optimization Level

All library and application projects for Keil are configured to build at full optimization.
Using the level of optimization (such as level 3 (-O3), typically makes the code harder to debug, but gives the best possible code size and performance. You can change the optimization settings by opening the project options by right-clicking the project in the project browser window and selecting Options. After this, select the C/C++ tab and select the optimization level you want for your build images. Using an optimization level of (Level 0 (-O0)) will give a better debug experience, but a larger image.

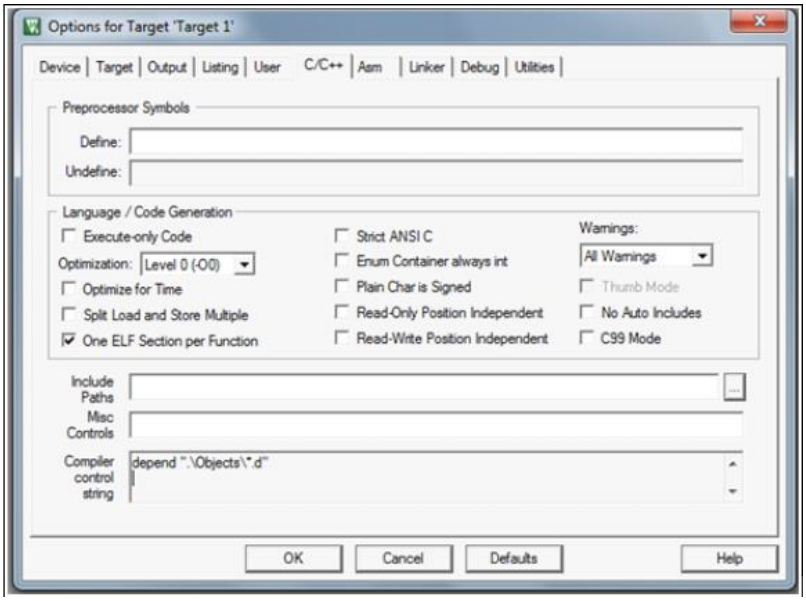


Figure 12: Default Optimization Level

4.5 Select Project

For selecting required project example, right click **"Project:** uart**"** and set as **"Active Project"**.

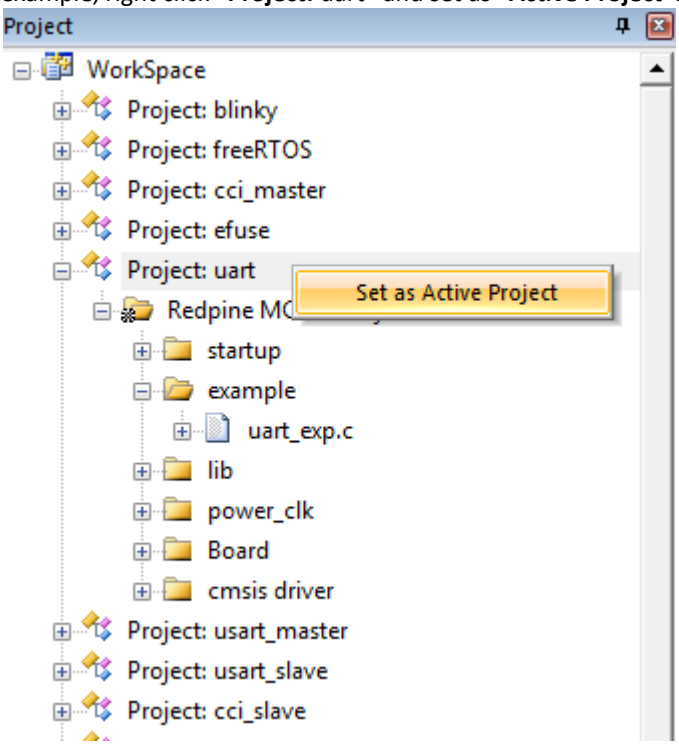


Figure 13: Select Project

Once Flash or RAM settings are finished, you can select the example, debug and can run the project.

Every MCU project has its own Readme file which has the description related to the corresponding MCU example. Check the status of the application by adding the watch variables in the watch window which will be present in **"View"** tab. Also, check the status of the application in the serial window if the application has debug prints enabled.

Once the project has started running and reset is also completed then, EVK and J-Link JTAG needs to be power cycled.

5 Starting a WiSeMCU Wireless Multi-Project View

Multiple examples are organized into multi-project groups for UV5. To select a project group, double-click on the appropriate multi-project file in the Keil examples directory.
There are 4 multi-project examples namely, **Keil_ble**, **Keil_bt**, **Keil_wlan** and **Keil_coex** present in the path below:
***{Release$}*Examples\Reference_Projects\Wireless_Projects**
Once any of the 4 multi-project files is started, all the examples associated with that specific project will be shown in the Project view. Refer to the image below:
*e.g:{Release$}*Examples\Reference_Projects\Wireless_Projects\Keil\wlan

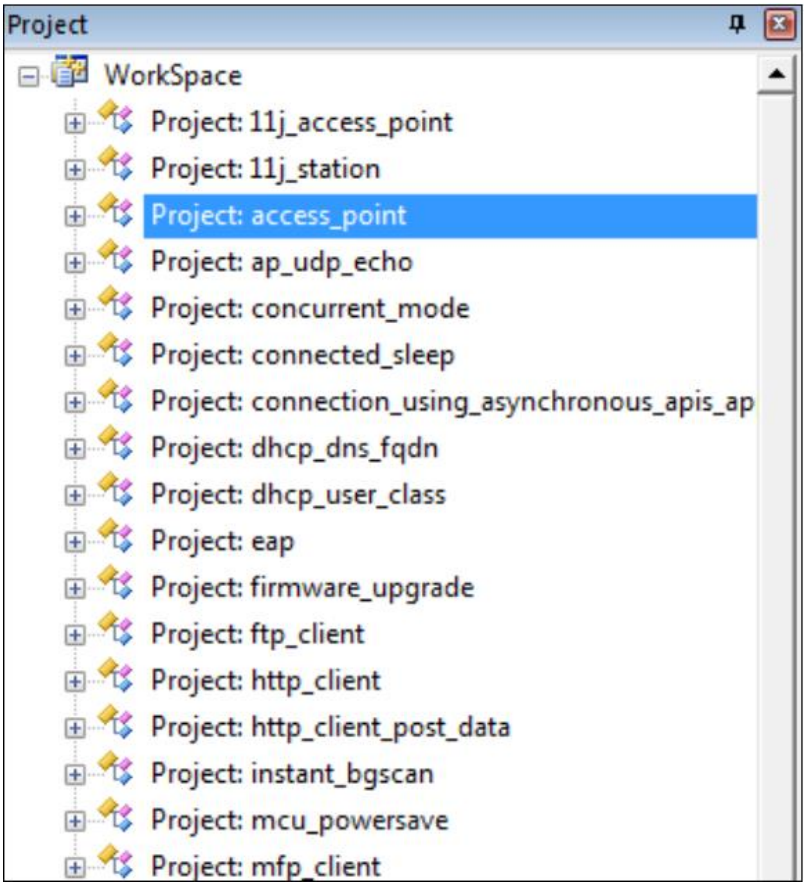


Figure 14: Multi Project View

5.1 Select Project

For selecting the required project example, right click on **"Project: access_point"** and set as Active Project as shown below:

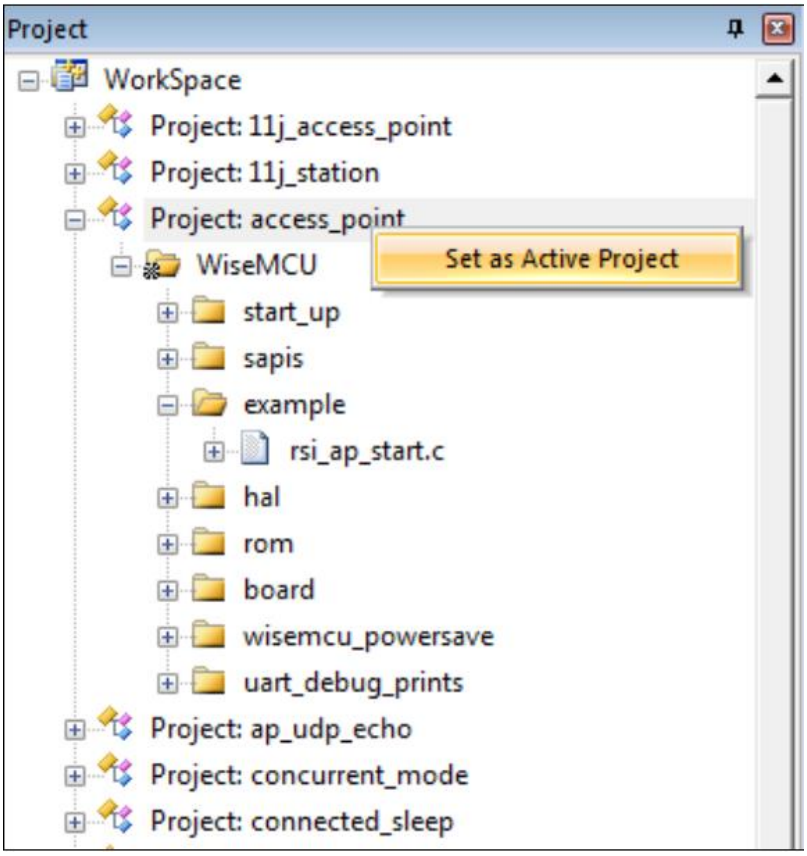


Figure 15: Select Project

When finished, click on Debug and run the application. Also, check the status of the application. Every example has its own user guide explaining the configuration and execution of the example.
e.g:{Release$}Examples\Wireless_Examples\wlan\access_point

6 Create a New Project in Keil IDE with Keil SVD Pack

We have submitted the DFP to Keil for reviewing and integrating into the Keil uVision. Once this is approved, we can use this procedure.
Until then, follow the steps mentioned in the section **"Creating a new project in Keil IDE Without SVD"**.

6.1 Start New Project

Install **"Keil.RedpineMCU_DFP.1.0.1 version"** of Keil SVD pack and create a new project.

6.2 Select Device

Select **"Redpine MCU"** device. Refer to the image below:

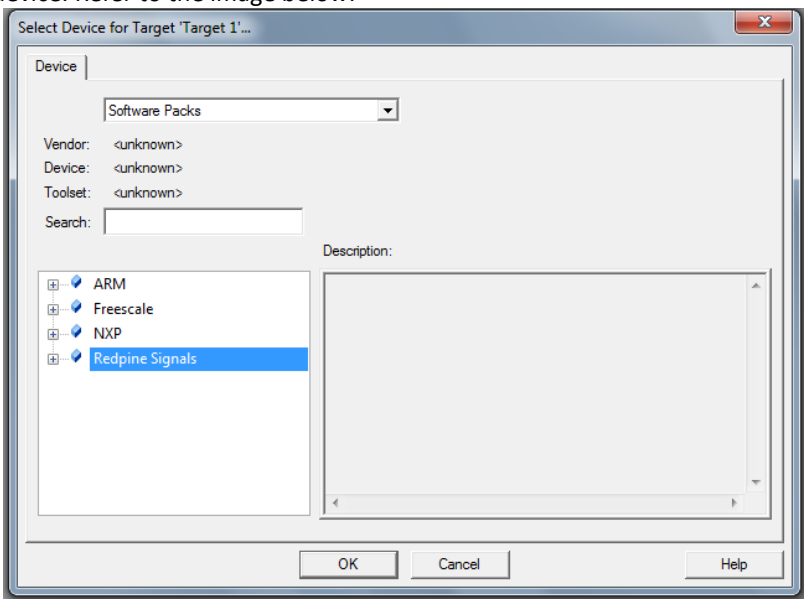


Figure 16: Select Device

6.3 Selecting and Building the Library File for the Examples

All examples depend on a library file. Click the **Manage Run-Time Environment** tab and select the library. Also, select the dependent file as well as the related RSI proprietary and CMSIS driver file. Refer to the image below:

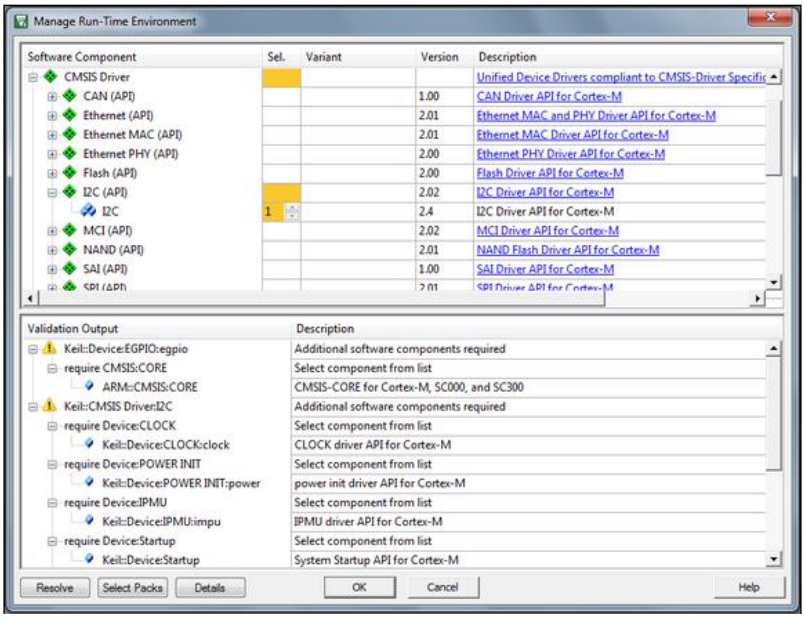


Figure 17: Add Library Files

7 Flashing Firmware using Host Interface

Firmware can be flashed through the NWP using the USB-CDC host interface. It is possible to flash both M4 and NWP firmware in this way. The following guides show how to do both.

7.1 Flashing NWP Firmware

It may be necessary to update the firmware running on the Network Processor.

1. Power the device off and switch ISP to ON.
2. Plug-in a USB cable to USB-CDC and another cable to POWER, in the same order. The device should appear on the Windows PC as a COM port.
3. Open Tera Term and select the COM Port used by the RS14100.
4. Enter the pipe key |. Tera Term should echo back a U. Enter a capital U. This will make the bootloader menu appear. This process is called Auto Baud Rate Detection (ABRD) and is used to set the baud rate of the RS14100.
5. Choose option B and select image 0.
6. Go to File-> Transfer-> Kermit-> Send.
7. Select the image **RS14100.NB0.WM.GENR.X.Y.Z.rps** in <Package>\NWP\Firmware. Tera Term will begin sending this image.
8. RS14100 will send the message **"Upgradation Successful"** once the flashing process is completed.
9. Turn off the device and switch ISP back to OFF. Your device is now running the latest wireless firmware.

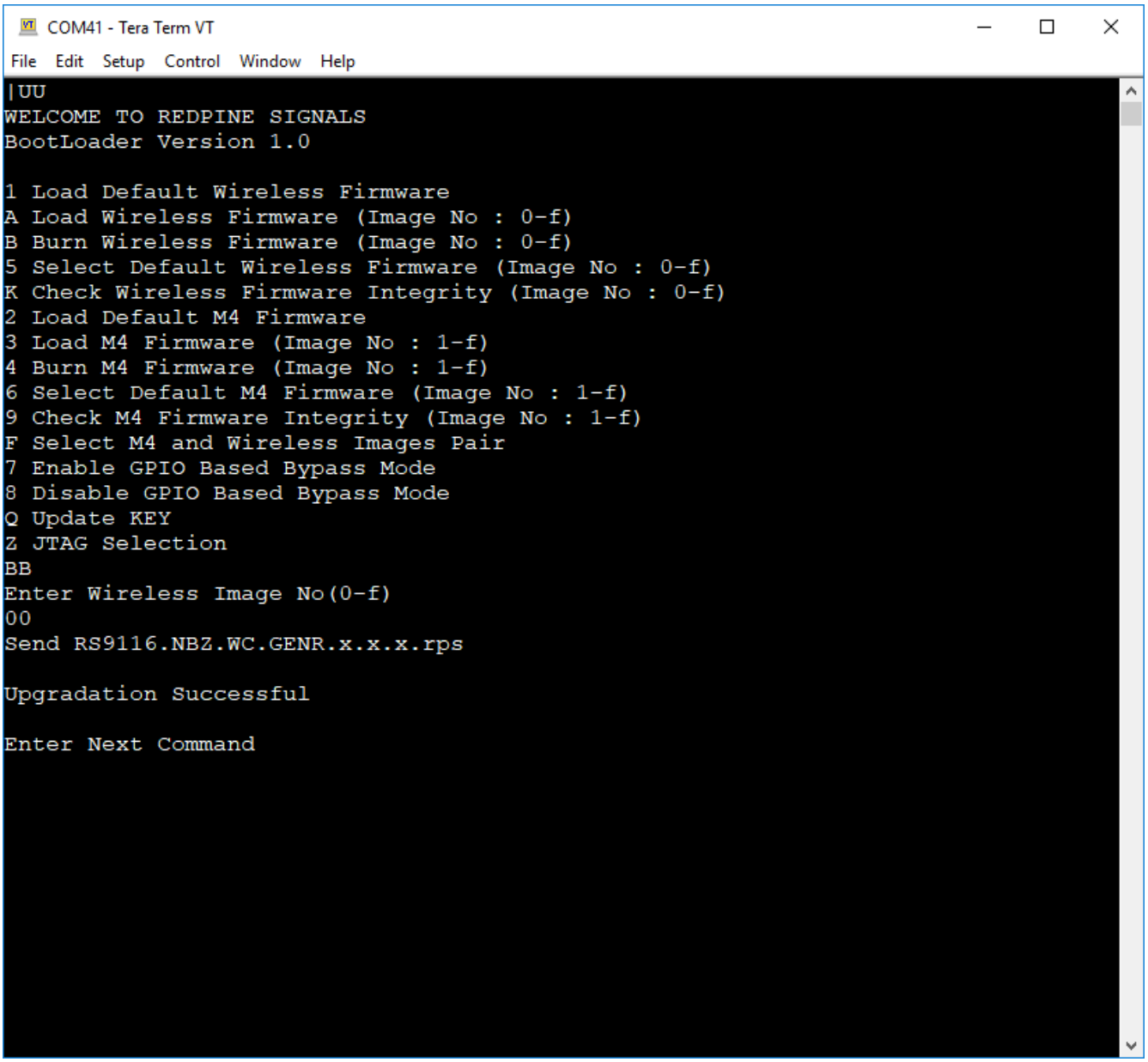


Figure 18: Flashing Firmware to the RS14100. Note that your image will be named "RS14100...rps" Not "RS9116...rps"

7.2 Flashing M4 Firmware

It is possible to flash application firmware to the M4 via the same USB-CDC interface.

1. Follow the steps 1-4 as indicated in chapter **Flashing NWP Firmware.**
2. Choose option 4 and select image 1.
3. Go to File0-> Transfer-> Kermit-> Send.
4. Select the image **RS14100_EVK_DEMO.bin** in <Package>\Examples\EVK_Demo. Tera Term will begin sending this image.
5. RS14100 will send the message **"Flash Download Successful"** once the flashing process is completed.
6. Turn off the device and switch ISP back to OFF. Your device is now flashed with the EVK demo. Please see the EVK Demo Guide for details on this demo.

8 Troubleshooting

Below are some issues you may run into while evaluating the platform.
Q1: The device cannot be detected. Keil gives me the error Debug Unit Not Found.
A1: There are several reasons why this may occur:

1. The device is not powered on.
2. CMIS-DAP adapter is not seated properly. Remove it and reseat it.
3. CMIS-DAP adapter is not plugged into the PC.
4. Debug settings in options do not match step 4 in **Running the AP Example.**
5. A previously flashed program is shutting off the debugger. In this case, turn ISP ON to halt the program, reflash with Keil, and turn the device off. Turn ISP back OFF and power the device on.

Q2: Keil is not flashing the latest code to the RS14100.
A2: Build or rebuild the project and try debugging again.

Q3: Flash is giving a timeout error.
A3: This occurs occasionally, especially after power cycling. Try debugging again.

Q4: Keil gives the error Flash Algorithm Not Found.
A4: The latest Redpine DFP may not be installed. Install the DFP and try again.

Q5: The USB-CDC interface is not showing up as a COM port on the PC.
A5: Ensure that ISP is set to ON. Then plug the USB-CDC port. Finally, plug the POWER port. POWER port must be plugged in last.

Q6: I flashed the code but do not see an AP.
A6: Try stepping through the program with the debugger and see if any of the below problems occur.

Q7: Code is flashing and the debugger is running, but I cannot step through the code.
A7: Verify that ISP is OFF and try again.

Q8: Code appears to run but fails at rsi_driver_init() or gets stuck at rsi_wireless_init().
A8: Reflash the NWP firmware as described in the chapter **Flashing NWP Firmware**. Make sure to use the firmware provided by the same package as the example project. (e.g. if using the access_point example from 0.9.3 package, use 0.9.3 firmware). Re-flash and rerun the application.

Q9: I can proceed past rsi_wireless_init() but some other API is failing.
A9: There may be a configuration issue. Use an unmodified version of the project and try again.

Q10: Is it possible to use JTAG debugging in the low power demos?
A10: No. When the device enters sleep mode, it turns off the JTAG interface to save power. It is not possible to reinitialize the debug session after this point