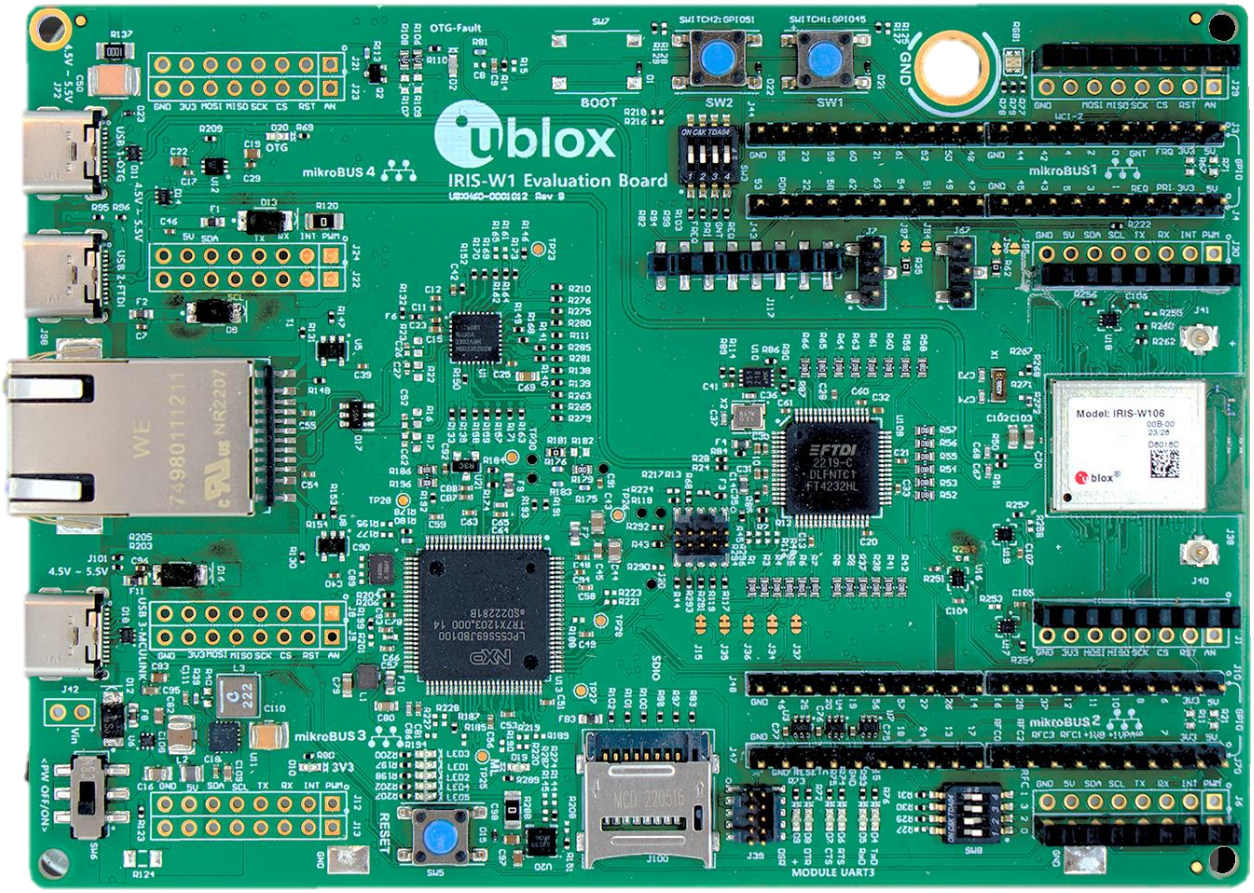
EVK-IRIS-W10

Evaluation kit for IRIS-W10 series modules

User guide



Abstract

This document describes how to set up and use the EVK-IRIS-W10 evaluation kits for prototyping the IRIS-W10 open CPU, multiradio modules. It also describes the different options for debugging and the development capabilities included in the evaluation board.

UBX-23007837 - R03
C1-Public

Document information

Title	**EVK-IRIS-W10**	
Subtitle	Evaluation kit for IRIS-W10 series modules	
Document type	User guide	
Document number	UBX-23007837	
Revision and date	R03	8-Nov-2024
Disclosure restriction	C1-Public	

This document applies to the following products:

Product name	Document status
EVK-IRIS-W101	Early production information
EVK-IRIS-W106	Early production information

Contents

1 Product description

1.1 Overview

The EVK-IRIS-W10 evaluation kit enables stand-alone use of the IRIS-W10 series module. This guide provides details about the hardware functionality of the EVK-IRIS-W10 board and includes setup instructions for starting development.

All pins and interfaces supported on IRIS-W10 series modules are easily accessible from the evaluation board. Simple USB connections serve as the physical interfaces for power, programming COM ports, debugging, and USB peripheral connectors. Additionally, the board features other interfaces like Ethernet RJ45 and an SDIO header. The EVK-IRIS-W10 board is equipped with a Reset button, Boot button, and two user-configurable buttons. Current sense resistors are incorporated for accurate current measurement within the module.

For flexible use, GPIO signals are accessible through headers and are complemented by four mikroBUS™ standard slots for convenient utilization of Click boards™. Each Click board can be seamlessly plugged into an available mikroBUS™ slot to facilitate effortless hardware expansion with a variety of standardized compact add-on boards. Click boards are designed to accommodate a diverse range of electronic modules, including sensors, transceivers, displays, encoders, motor drivers, connection ports, and more. For further information about the Click boards, visit the MIKROE website [10].

⚠ Observe that this device is for evaluation only and is not FCC approved for resale.

1.2 Kit includes

1.2.1 EVK-IRIS-W101

- EVK-IRIS-W1 evaluation board with IRIS-W101module
- USB-A to USB-C adapter cable
- Dual band PCB antenna for WLAN with 100 mm coaxial cable and U. FL connector

1.2.2 EVK-IRIS-W106

- EVK-IRIS-W1 evaluation board with IRIS-W106 module
- USB-A to USB-C adapter cable
- A dual-band integrated PCB trace antenna (external antenna not supplied)

1.3 Key features

EVK-IRIS-W10 boards provide:

- Evaluation board for IRIS-W101 or IRIS-W106 modules
- Four standard mikroBUS slots
- USB interface
- Serial communication over the FTDI USB controller
- On-board programming and debug
  - MCU-Link port via debug chip over the SWD interface
  - JTAG debugging via a four-port FTDI USB controller
- RMII/Ethernet interface (via 100 Mbit PHY circuit)
- Access to IRIS-W10 module JTAG signals over JTAG connector
- Buttons and status LEDs for user interaction
- All module GPIOs accessible from the pin headers
- Additional memory support (SRAM SOIC-08 footprint on the bottom side for manual mounting)

- Multiple Boot strap options
- Module isolation and customizable functions via solder bridges and/or resistors
- Current measurement access points from pin headers and jumpers

IRIS-W10 open CPU modules, based on the NXP RW612, support:

- Wi-Fi 6 IEEE 802.11a/b/g/n/ac/ax, Dual band WiFi 2.4/5 GHz
- IEEE 802.15.4 supporting Thread®, Matter™ over Wi-Fi, and Ethernet
- Bluetooth LE subsystem supporting Bluetooth 5.3 in 2.4 GHz band
- Bluetooth LE central, peripheral, GATT client / server roles, LE Audio
- Peripherals[1]: ADC, GPIO, I2C, Ethernet RMII, SDIO, SPI, UART

1.4 Block diagram

Figure 1 shows the block diagram and internal connections of EVK-IRIS-W10.

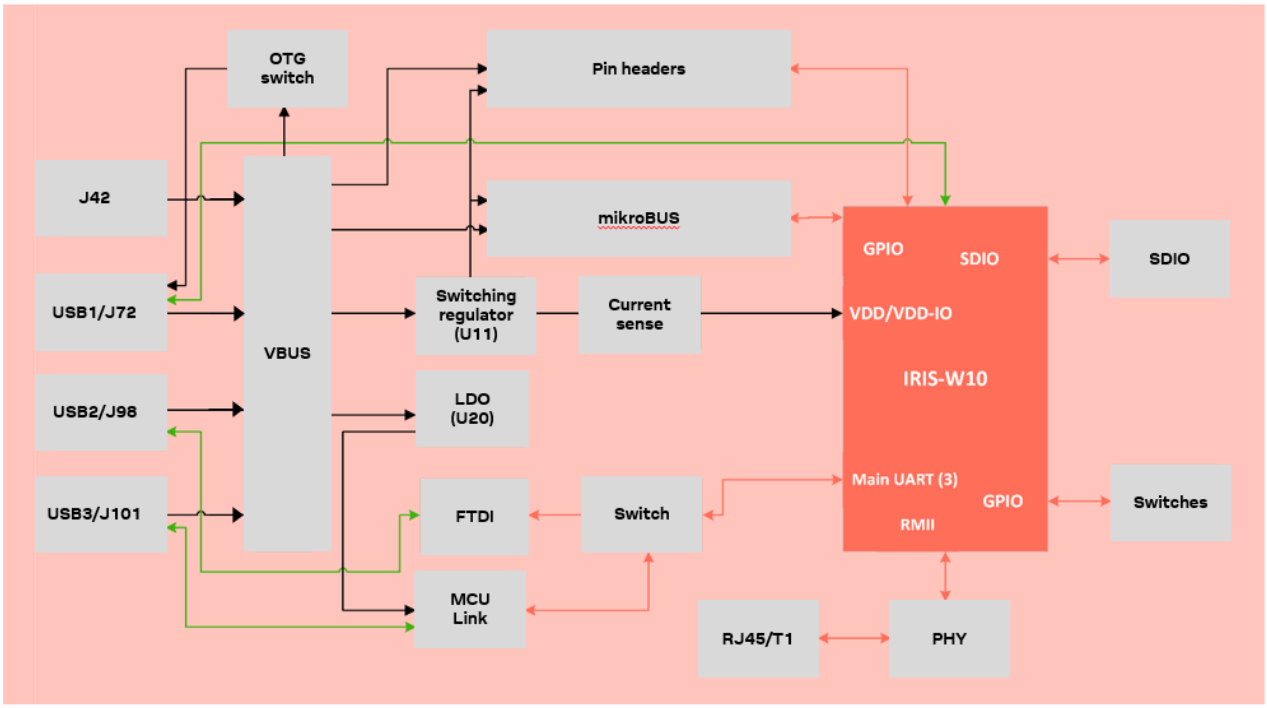


Figure 1: EVK-IRIS-W10 block diagram

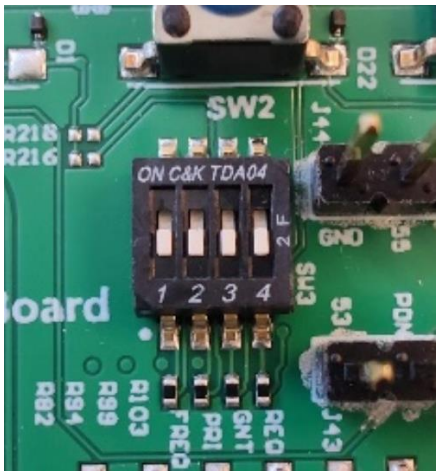
[1] Not all peripherals available simultaneously

2 Setting up the evaluation board

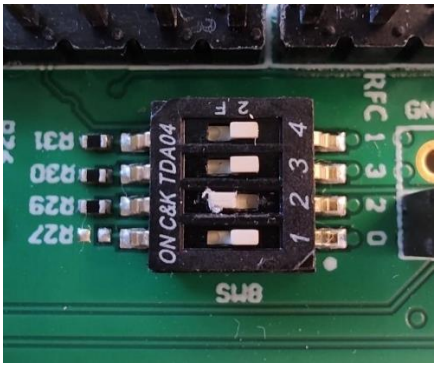
2.1 Prerequisites

Before starting up the board:

1. To configure the board to boot from QSPI flash (default), set the **SW3** switch positions as shown below . See also Automatic bootloader / strap-in.



2. To enable the SWD interface of the module, set **SW8** switch positions as shown below. See also JTAG/SWD debug interfaces.



⚠ For variants equipped for use with an external antenna (EVK-IRIS-W101), be sure to connect the antenna (or any 50 Ω RF load) to the U.FL antenna connector (J41) before powering up the board. Failing to do so can impact RF calibration.

2.2 Starting up the EVB

Attach the USB-C cable to the USB3 MCU-Link port on the EVB and connect to PC to power the module:

- **LED1** should then turn red and blink slowly.
- The green status LED (**D10**) should be lit to indicate that the internal EVK 3V3 supply is active.
- The green status LED (**D19**) should be lit to indicate that **3V3** supply to the MCU Link chip (debugger) is active.
- The Enumerated COM port "JLink CDC UART Port (COMxx)" should be shown as a port in the Windows Device Manager.

2.2.1 Wi-Fi example application

The IRIS-W10 module hosted on the EVK-IRIS-W10 evaluation board is pre-flashed with the Wi-Fi CLI example application to conveniently experience the Wi-Fi features supported in the IRIS-W10 module – without the need of downloading the SDK and compiling any firmware.

Example of supported features:

- Wi-Fi Scan
  - Wi-Fi Soft Access Point mode
  - Wi-Fi Station mode
- Throughput performance using the `iPerf` measurement tool

To run Wi-Fi_`CLI` application, open a UART console (Putty, TeraTerm, or another terminal emulator), and set the serial port to 115200 baud rate, 8-bit data, No parity, 1-bit stop, No flow and run the various feature sets available in the Wi-Fi_CLI application, as shown in Figure 2.

Figure 2: Wi-Fi CLI feature list

2.2.2 Run iperf test using the wifi_CLI example

This section assumes that the `iperf` test is run using two EVK-IRIS-W10 evaluation boards as devices – with Device A configured as an Access Point and Device B configured as a Station. Note that *iperf* measurements can also be run against other devices that support *iperf*.

Configure device A as Access Point with iperf server

To configure device A as an access point with *iperf* server, open a terminal session and enter:

```
wlan-add xyz ssid NXPAP ip:192.168.10.1,192.168.10.1,255.255.255.0 role uap channel 48
wpa2 psk 12345678

wlan-start-network xyz

iperf -s
```

⚠ The device can only operate as an Access Point (AP) on channel 1 – 11 in the 2.4 GHz band and on channels 36, 40, 44, and 48 in the 5 GHz band.

Configure device B as Station with iperf client

To configure device B as a Station with iperf server, open a terminal session and enter:

```
wlan-add test1 ssid NXPAP ip:192.168.10.3,192.168.10.1,255.255.255.0 channel 48 wpa2 psk
12345678

wlan-connect test1

iperf -c 192.168.10.1
```

☞ For information about the Wi-Fi CLI application, see also the (granted access required) *NXP Wi-Fi and Bluetooth Demo Applications for RW61x, User Manual,* UM11799 [12].

☞ To re-load Wi-Fi CLI application, see Re-loading the Wi-Fi_CLI example.

2.3 Software Development

IRIS-W10 openCPU module variants are used to develop custom applications based on the NXP MCUXpresso SDK [5], which provides all the APIs required for custom application development. Before compiling custom software, you must configure the NXP MCUXpresso SDK for use with the IRIS-W10 open CPU variant as detailed in NXP MCUXpresso SDK.

For information about the working environment setup and regulatory restrictions, see *Open CPU software* and *Qualification and approvals* chapters in the IRIS-W10 system integration guide [2].

2.3.1 NXP MCUXpresso SDK

Update the following files to compile custom applications with NXP SDK:

- flash_config
- mflash_drv
- RF `cal_data` parameters

For detailed instructions and required files, visit the u-blox short range open CPU GitHub repository [7]. For more information about how to retrieve the SDK and perform the tasks, see also the "*Software*" section of the IRIS-W10 system integration manual [2].

☞ Ensure that the `flash_config` and `mflash_drv` files are updated according to the IRIS-W1 integrated flash memory type:

- For build versions up to and including 23/45 (meaning week 45 of 2023), the module is integrated with Macronix Memory.
- For build versions from 23/46, the module is integrated with Fidelex Memory.

2.3.2 Flash and debug custom applications

EVK-IRIS-W10 supports two options for flashing and debugging custom applications, using either:

- MCU-Link port (On-board debugger)
- External debugger

2.3.2.1 MCU-Link port (On-board debugger)

1. Power the EVB via USB3 MCU-Link port.
2. Set **SW3** and **SW8** as shown in Prerequisites.
3. Use either J-Flash Lite or JLinkCommander (V7.98i and above), selecting "SWD" as the interface and "RW612" as the device to flash the compiled binary. The MCU-Link (On-board debugger) handles flashing and debugging without the need for external debuggers.

2.3.2.2 External debugger

1. Power the EVB via USB3 MCU-Link port.
2. Enable the "JTAG" interface: Set the **SW3** switch positions as shown in Prerequisites and **SW8** as shown in JTAG/SWD debug interfaces, Table 8.
3. Connect an external debugger to J20 connector. See also JTAG/SWD debug interfaces.
4. Using either J-Flash Lite or JLinkCommander (V7.98i and above), select "JTAG" as the interface and "RW612" as the device to flash the compiled binary. In this setup, the firmware flashing and debugging are managed by the external debugger.

3 Hardware description

Design files for the EVK-IRIS-W10 are available in the u-blox short range EVK design GitHub repository [8].

Figure 3 shows the major functions provided by EVK-IRIS-W10.

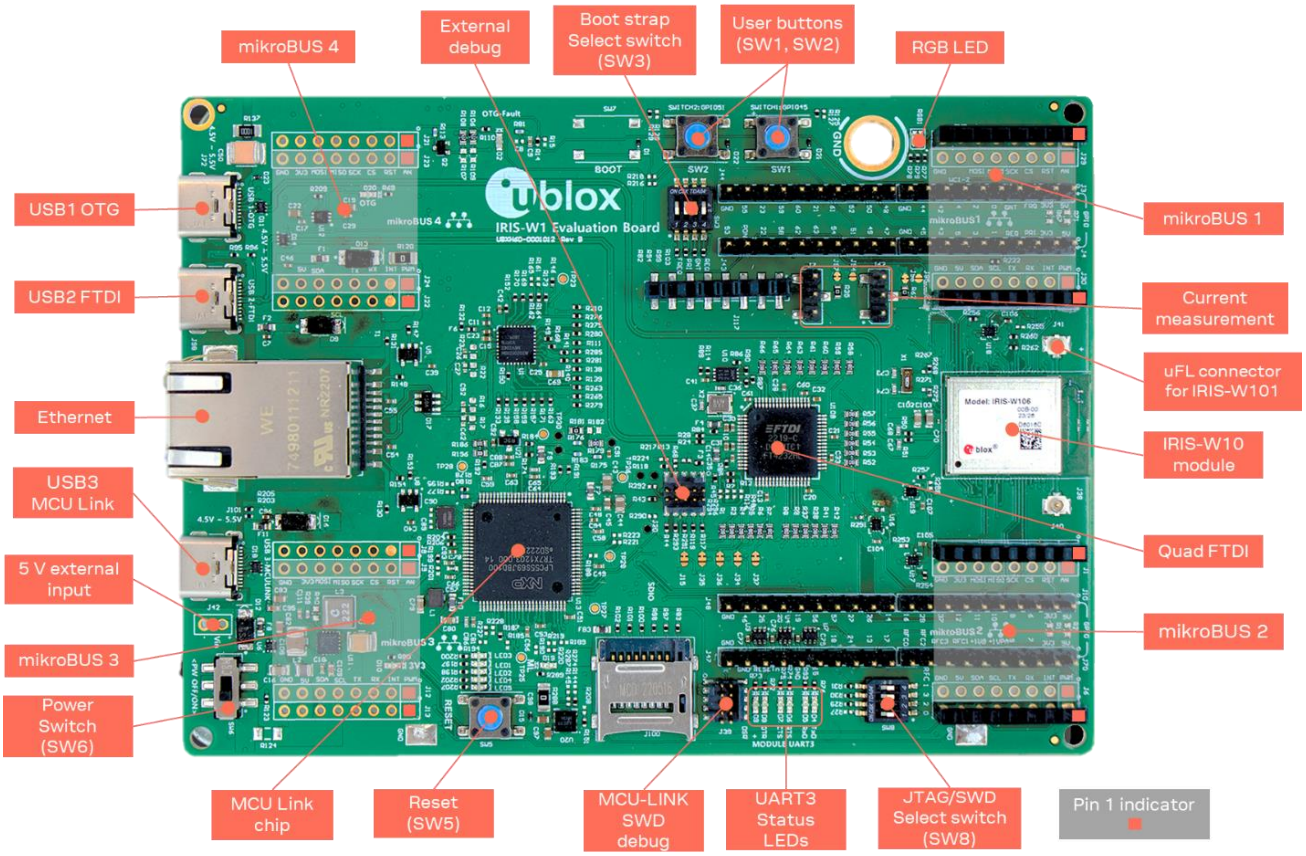


Figure 3: Header and major function locations

3.1 Power

EVK-IRIS-W10 has four potential power sources:

Source	Component / pin	Input range	Remarks
USB-C (IRIS-W10)	J72	5.0 VDC nominal	Power provided by USB peripheral on the IRIS-W10 (USB 1)
USB-C (debug/UART)	J98	5.0 VDC nominal	Power provided by debug interface (USB 2)
USB-C (MCU-LINK)	J101	5.0 VDC nominal	Power provided by MCU-LINK (USB 3)
Power header	J42, pin 1	5.0 VDC nominal (3.0 – 6.0 VDC)	2.54 mm pitch pin header

Table 1: EVK-IRIS-W1 power sources

Power sources are protected from reverse polarity by protection diodes, allowing multiple sources to be present simultaneously.

Only if the power protection circuits are left intact can the USB be safely connected at the same time as external power. This makes the programming of the module easier.

Do not connect the 12V supply. EVK USB type-C connectors are only capable of handling 5 V input.

3.1.1 Powering the board

After applying power to one of the sources described in Table 1, slide **SW6** switch positions to the ON position to power-on the EVK, as shown in Figure 3.

The input voltage **VBUS-OUT** is extended to:

- 3.3 V switching regulator input (**U11**)
- 3.3 V linear regulator input (**U20**)
- USB 2.0 On-The-Go (OTG) circuit
- **VBUS-OUT** (5 V) connections on pin headers
- mikroBUS slots 5 V positions

There are two options for the maximum current for the EVK:

- Populate the 2.4 A power switch **U6,** MP5075GTF (default), and disconnect **R124,** as shown in Figure 4.
- Limit the maximum current to 600 mA by populating **R124** with a 1206 0 Ω resistor 2A resistor and disconnecting U6.

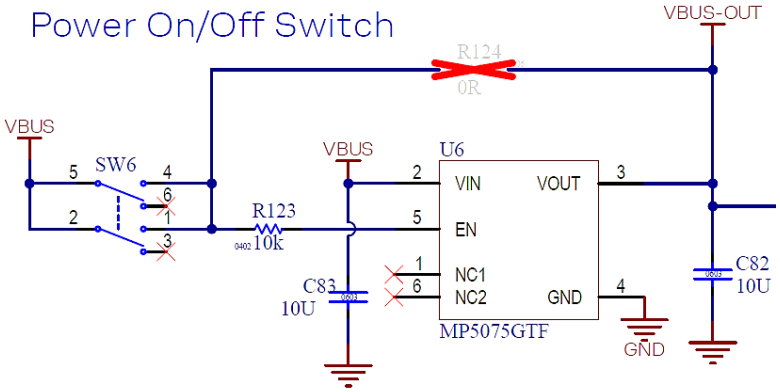


Figure 4: EVK-IRIS-W10 maximum current options

3.1.1.1 3.3 V switching regulator (U11)

The EVB is populated with a fixed 3.3 V, 3 A, switching regulator (TPS62132RGTR), as shown in Figure 5. LED **D10** (green) indicates the presence of the 3.3 V output from the regulator **U11**.

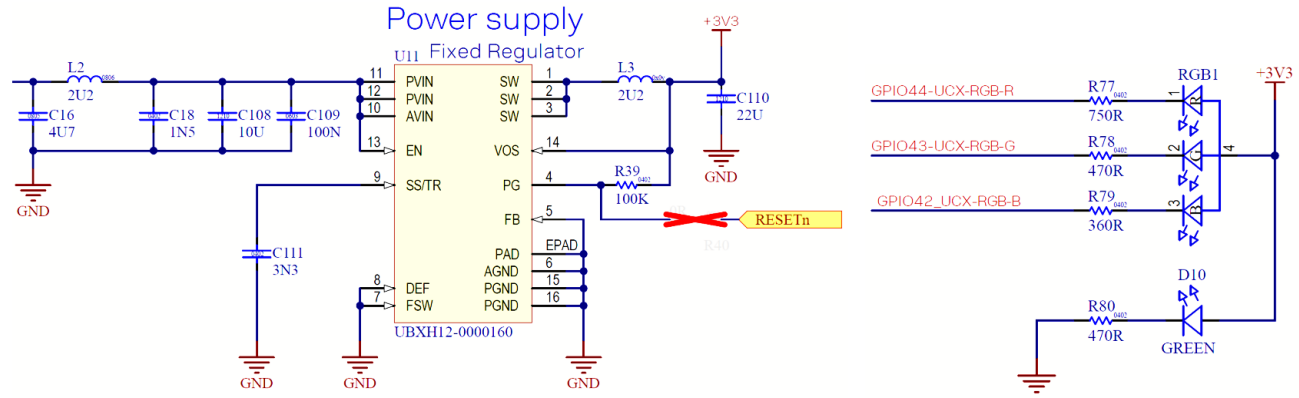


Figure 5: Switching regulator (U11)

3.1.1.2 3.3 V Linear regulator (U20)

In Figure 6, the NCP692MN33T2G regulator **(U20)** is intended to supply power to the **MCU-LINK** chip with minimum interference to the IRIS-W10 module.

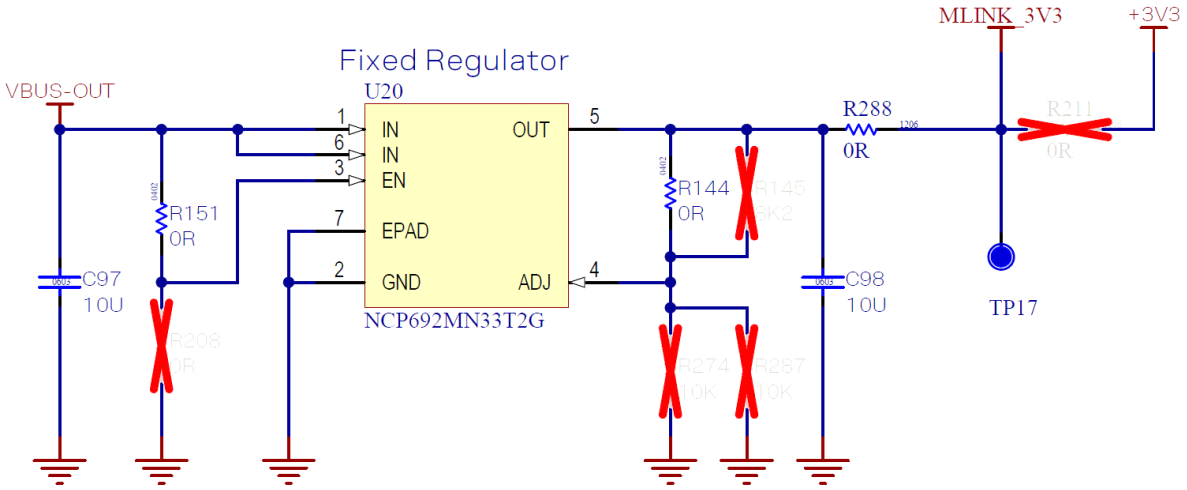


Figure 6: 3.3V linear regulator (U20)

To configure power to MCU-LINK from the regulator, resistors **R151, R144**, and **R288** must be populated with a 0 Ω, 0603 resistor**,** and **R208, R145, R274,** and **R287** must be disconnected. **LED D19** (green) indicates the presence of the 3.3 V output from **U20.**

The voltage dividers (**R145/R287**) and (**R151, R208**) provide the EVB with greater flexibility. If necessary, the EVB can then work with variable output voltage LDOs.

R288, and **R211** provide an option to supply the MCU-LINK chip from **U11** output voltage.

3.1.1.3 USB 2.0 On-The-Go (OTG)

The OTG function is only supported on the module USB-OTG power switch **(U12),** as shown in Figure 7. It is set to the OTG-device by default. Connect jumper **J14** to allow IRIS-W10 sensing on the MUSB-OTG-VBUS.

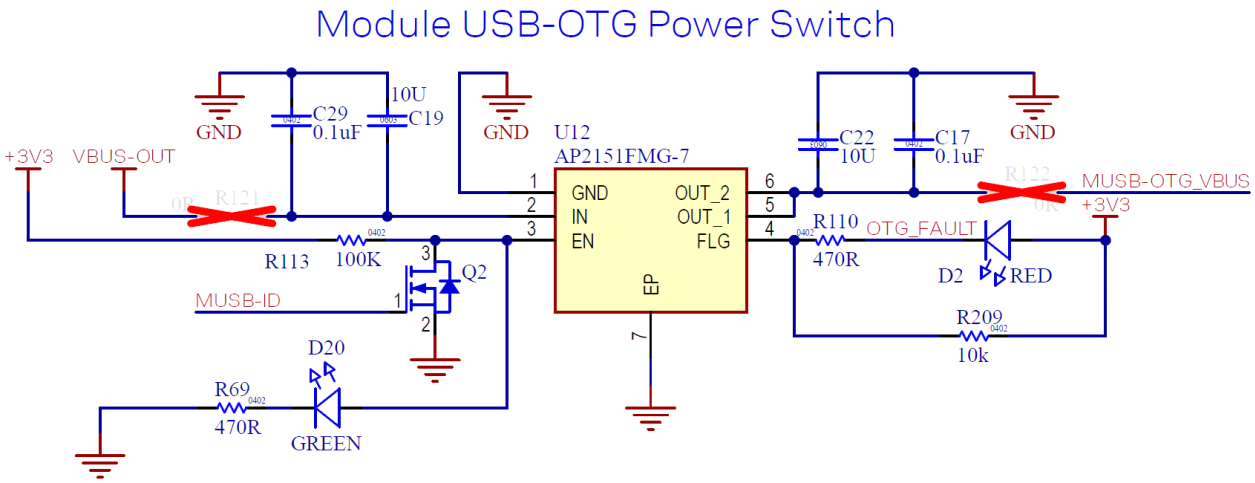


Figure 7: USB On-The-GO switch

- Use the 2.4 A maximum current option as mentioned in section Powering the board.
- Populate **R121**, and **R122** with 1206 0 Ω resistors.
- Check the status of resistors, **R106, R107, R108, R109,** required mode connection.
- LED **D20** (Green) indicates the activation of the OTG function.

The maximum current capacity of the USB-OTG is 500 mA. LED **D2** (Red) indicates whether the current to **OTG-USB** exceeds the current rating (800 mA).

In this mode, the status of the USB-ID signal defines whether the EVB or IRIS-W10 module works as either an OTG host (supply the voltage to a device board), or an OTG device. USB-ID is connected to the ground or left floating, as shown in Figure 8.

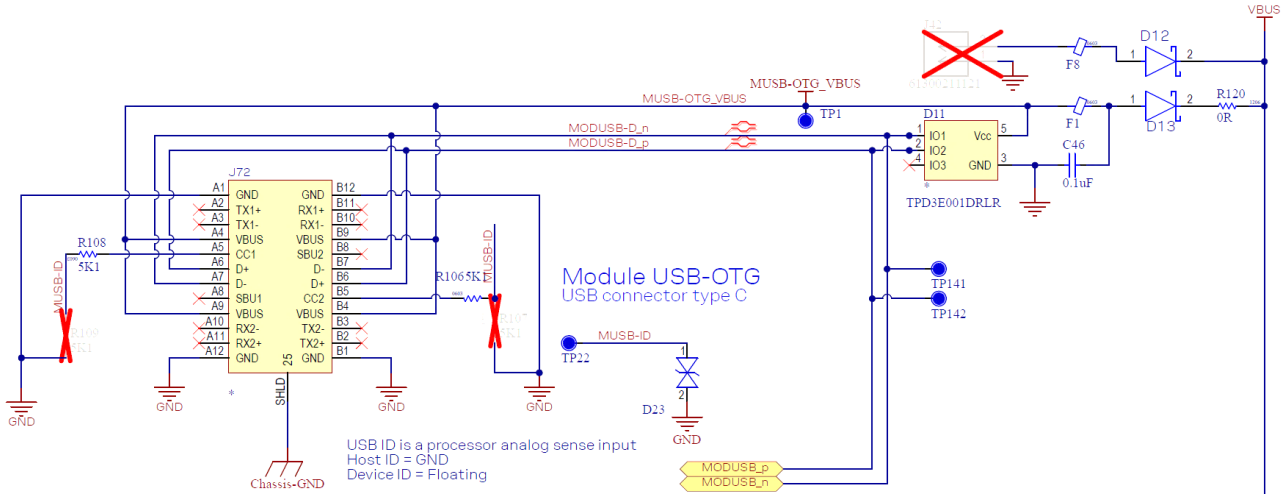


Figure 8: J72 connections

3.1.1.4 VBUS-OUT (5 V) connection on pin headers

VBUS-OUT (5 V) connections on the **J3**, **J4**, **J10**, and **J70** pin headers should only be used for reference or verification. As these connections can only supply a combined 500 mA current, the connections should not be utilized in conjunction with USB-OTG Host mode.

3.1.1.5 mikroBUS slots 5 V positions

Intended for mikroBUS standard 5 V pins.

3.1.2 IRIS-W10 module power

+3V3 is supplied to IRIS-W10 through **VDD-IO** (**J96**) and **VDD** (**J97**), with resistors **R62** and **R35** intended for current measurement, as shown in Figure 9.

VDD-IO can be supplied from the **+1V8** or **+VPA** module outputs through jumpers **J115** and **J116** (bottom side of the EVB). **R126** is needed for the current consumption sensing process.

⚠ Jumpers **J96, J115, and J116** should not be connected at the same time.

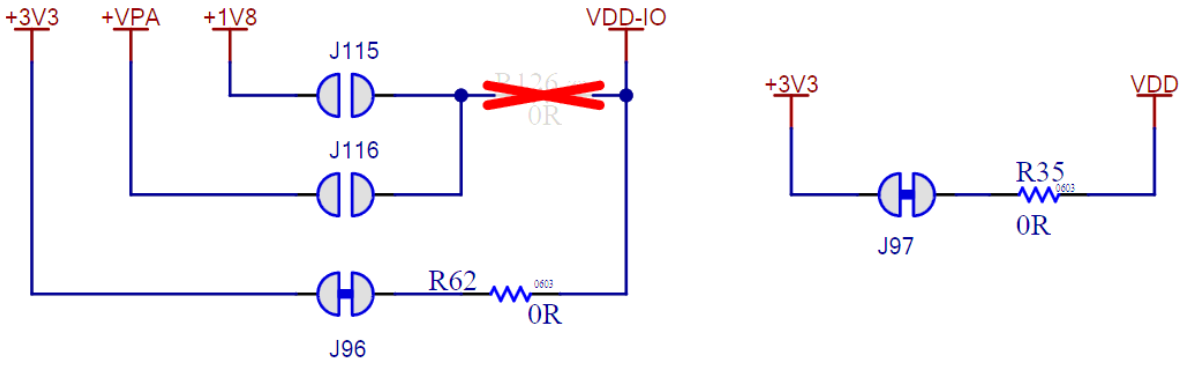


Figure 9: VDD-IO and VDD supply jumpers

3.1.3 Current measurement

The evaluation board provides two current-sensing headers, as shown in Figure 10.

- **J7** for current measurement of the **VDD** module supply
- **J67** for current measurement of the **VDD-IO** module supply

Module Current Measurement Module IO Current Measurement

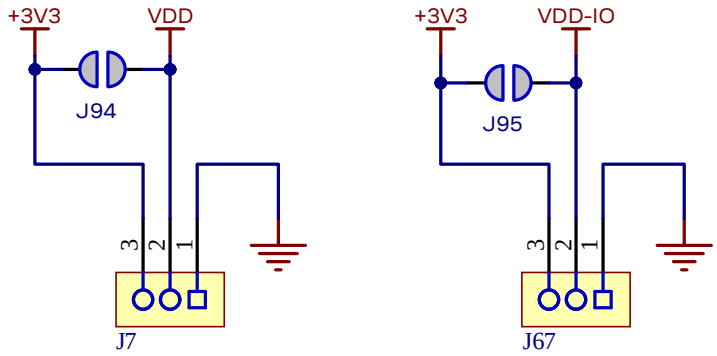


Figure 10: VDD and VDD-IO current sensing configuration

Two consecutive pins on Each 2.54 mm pitch 3-pin headers are connected across a 0 Ω resistor, which can be replaced by a current-sense resistor (> 1 Ω is recommended), **J97,** and **J96** respectively, as shown in Figure 9. The third pin on **J7** and **J67** are connected to **GND**. The **VDD** and **VDD-IO** module supplies are sourced through the resistors shown in Figure 9. To measure current consumption, use a multimeter or other precise voltage measurement device and measure the voltage drop across pins 2 and 3. If the current sensor is removed from the circuit, current can also be measured directly by opening **J97** or **J96**. Use an ammeter in-series with two voltage pins.

If **VDD-IO** is supplied by **+1V8** or **+VPA**, **R126** can be replaced by a current-sense resistor.

⚠ Pin 1 of **J7** and **J67** are connected to **GND**.

To bypass the current sense resistors, **R35** and **R62**, solder the respective jumpers **J94** and **J95.**

☞ The default hardware configuration doesn't require any modification of the current-sense headers on EVK-IRIS-W10 to perform properly.

3.2 Reset

The active-low reset signal, **RESETn**, is connected to the module **PDn** pin, the FTDI reset pin, the four mikroBUS slots, and a momentary button switch (**SW5),** as shown in Figure 11.

Some Click boards need different **RESET** signals, and mikroBUS slots have extra independent configurable **RESETn/RESET** options through **GPIO50**. See also mikroBUS slots.

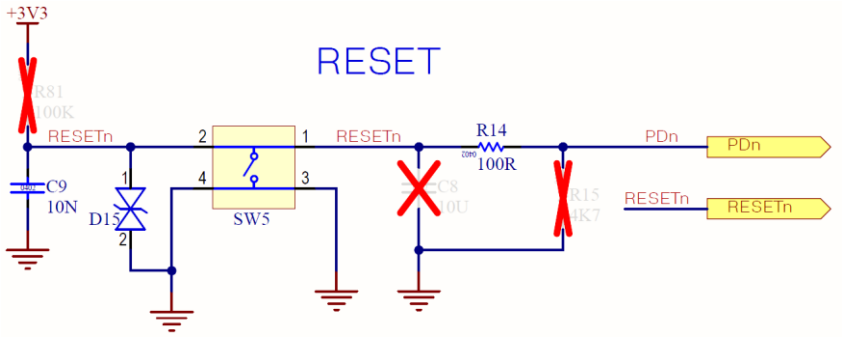
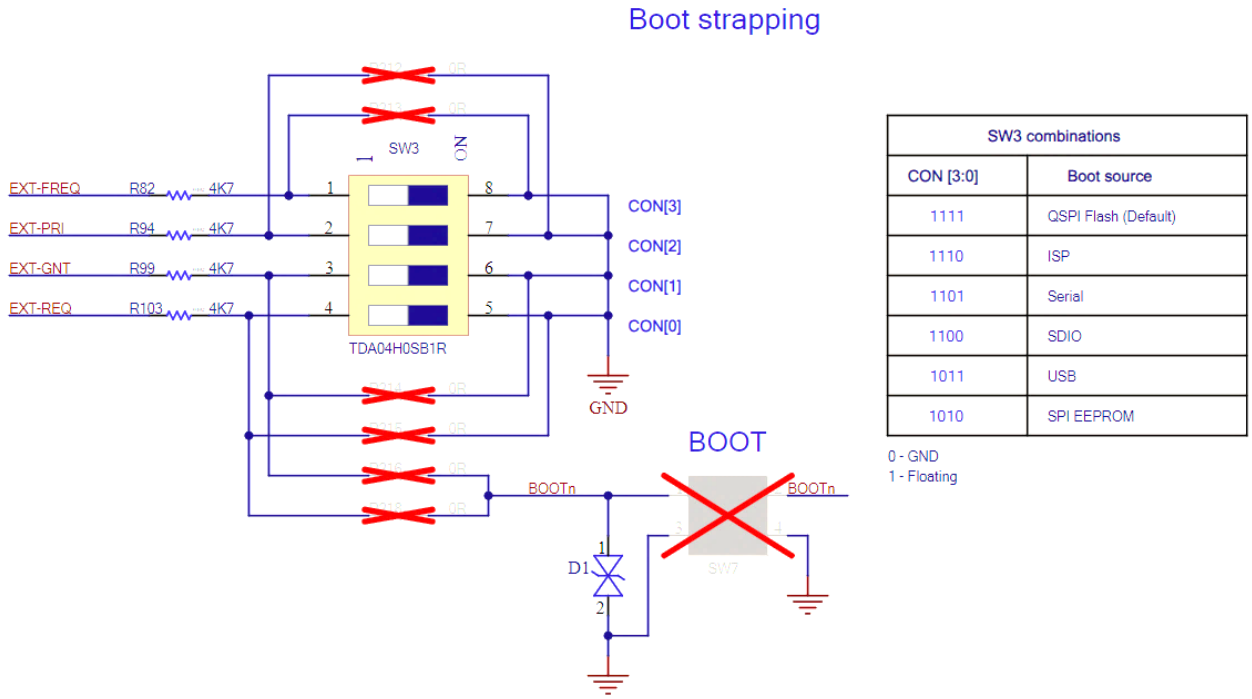


Figure 11: EVK schematic, RESET button

3.2.1 Automatic bootloader / Bootstrap

Figure 12 shows bootloader configuration and the boot sources available to user. Several signals and the DIP switch **SW3** positions determine the bootstrap method for booting the IRIS-W1 module.



SW3 combinations	
CON [3:0]	Boot source
1111	QSPI Flash (Default)
1110	ISP
1101	Serial
1100	SDIO
1011	USB
1010	SPI EEPROM

0 - GND
1 - Floating

Figure 12: Bootloader schematic

EVK-IRIS-W10 supports two main methods for bootstrapping the module:

- Default method: This method uses DIP switch **SW3** along with resistors **R82**, **R94**, **R99**, and **R103**. Four active-high signals (**EXT-FREQ**, **EXT-PRI**, **EXT-GNT**, **EXT-REQ**) are used to select the boot source. These signals correspond to the logic states for **CON3**, **CON2**, **CON1**, and **CON0**, respectively, as shown in Figure 12. For example, to select ISP as boot source, GND the signal **EXT-REQ** on **SW3**. Other configurations are also possible.
- Alternative method: This method uses a combination of buttons (**SW7, R216, R218**, **R212, R213, R214,** and **R215)** as shown in Table 2.

BOOT button	Installed resistor	Boot source
SW7	R216	Serial
	R218	ISP
	R216, R218	SDIO
-	R212	USB
	R212, R213	SDIO
	R212, R215	SPI EEPROM
	R214	Serial
	R215	ISP

Table 2: Boot/strap-in configuration

☞ Hold **EXT-REQ** signal to **GND** to set the EVK in a mode for In-System programming (ISP) over a supported interface, using the blhost application [15]. If the IRIS-W10 module enters a faulty state when flashing or running an application, you can perform full flash recovery using the blhost application. See also the blhost user guide [14] and blhost application [15] .

3.3 Buttons

EVK-IRIS-W1 has two more momentary push-button switches. These switches are active-low or high and connect to ground or **+3V3** when pressed. The buttons and the associated GPIO signals are shown in Figure 13 and Table 3.

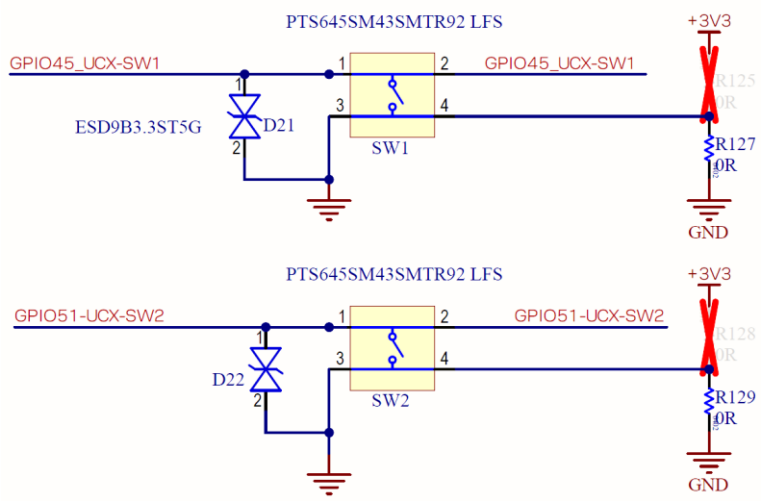


Figure 13: EVK schematic - user buttons

Table 3 describes the various user buttons and their relationship with the corresponding GPIO signals.

Button function	Reference designator	GPIO	Function
SW_1	SW1	GPIO45	No predefined function (software controlled)
SW_2	SW2	GPIO51	No predefined function (software controlled)
BOOT	SW7	EXT-GNT EXT-REQ	Pressing SW7 when R216 or R218 are installed starts the bootloader in ISP or Serial mode respectively. See also Automatic bootloader / Bootstrap.
RESET	SW5	PDn	Resets the module, FTDI chip, and mikroBUS slots

Table 3: User button definitions

3.4 User LEDs

EVK-IRIS-W10 supports 16 LEDs:

- Power status (**D10**): Indicates **3V3** on the board when lit (green) dependent on **SW6**
- MCU-LINK Power status (**D19**): Indicates **3V3** on MCU-LINK when lit (green), dependent on **SW6.**
- OTG (**D20**): Indicates OTG circuit is enabled lit (green).
- OTG-Fault (**D2**): Indicates over current of the OTG circuit when lit (red).
- MCU-LINK (**LED1 - LED5**): Indicate MCU-LINK connection status. See also MCU-LINK.
- UART3 status (**D4–D9**): Indicate UART3 signal status under GPIO control, as shown in Figure 15.
- System status (**RGB1**): Powered by **+3V3** and turned on by pulling the associated GPIO low. Associated GPIOs can be used as ADCs after disconnecting each LED by removing resistors **R77–R79**, as shown in Table 4 and Figure 14.

RGB LED	Associated GPIO	Disable option
Red (pin1)	GPIO44	Remove R77 to disconnect
Green (pin2)	GPIO43	Remove R78 to disconnect
Blue (pin3)	GPIO42	Remove R79 to disconnect

Table 4: RGB LED-associated signals

Figure 14 shows the schematic for the RGB status and power LEDs, and the configuration resistors (**R77-R79**).

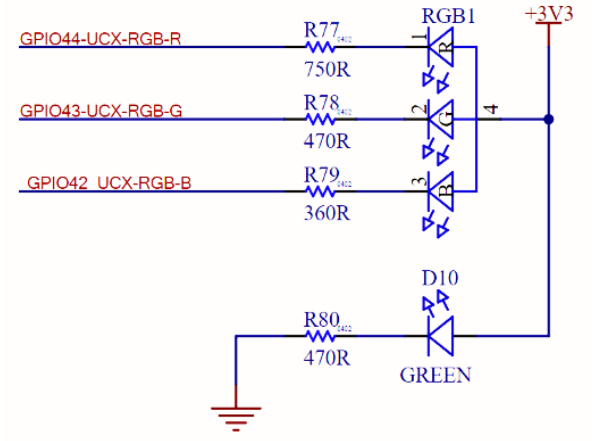


Figure 14: Schematic – RGB and power LED

Figure 15 shows the UART3 status LEDs (**D4–D9**) that indicate UART3 signal status under GPIO control.

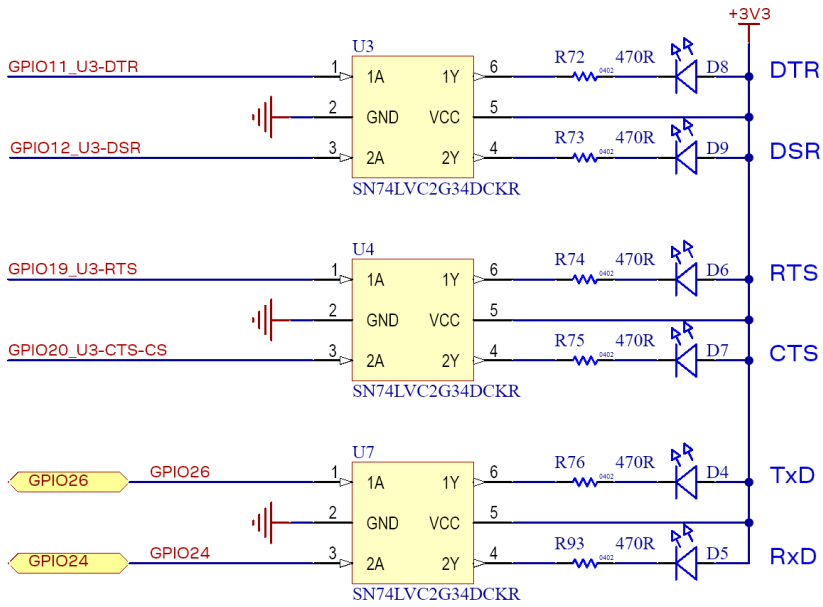


Figure 15: Schematic – UART3 status LED

Table 5 describes each of the **UART3** LEDs and their relationship with GPIO and serial **UART3** signals. Disconnect each LED from the GPIO by disconnecting **U3, U4,** and **U7**.

LED	Color	GPIO	Comments
D4	Green	GPIO26/UART3-TxD	UART3-TxD activity indicator
D5	Green	GPIO24/UART3-RxD	UART3-RxD activity indicator
D6	Green	GPIO19/UART3-RTS	UART3-RTS activity indicator
D7	Green	GPIO20/UART3-CTS	UART3-CTS activity indicator
D8	Green	GPIO11/UART3-DTR	UART3-DTR activity indicator
D9	Green	GPIO12/UART3-DSR	UART3-DSR activity indicator

Table 5: UART3 LEDs and associated signals

3.5 Serial communication

EVK-IRIS-W1 provides two options for serial communication and debugging:

- MCU-Link (default) chip
- FTDI chip

3.5.1 MCU-Link

The MCU-Link is a powerful and cost-effective debug probe that seamlessly integrates with the MCUXpresso IDE. It is also compatible with third-party IDEs that support the CMSIS-DAP protocol. The USB3 MCU-Link port on the EVB provides a USB-to-UART bridge feature (VCOM) that can be used to provide a serial connection between the IRIS module and a host computer. It is based on the LPC55S69 microcontroller (**U13**) and features a high-speed USB interface for a high-performance debug.

The main **UART** (**UART3 TX, RX**) and the **SWD** interface signals of the IRIS-W10 module are connected to the MCU-Link through switches **U16** or **U19** by default. Optional connections to the Flexcomm 0 SPI interface through switch **U18** and resistors **R259, R260, R261, R262**, and connections to the I2C Flexcomm 2 interface through switch **U17**, are also possible. The connection options are shown in Figure 16 and further described in Table 6.

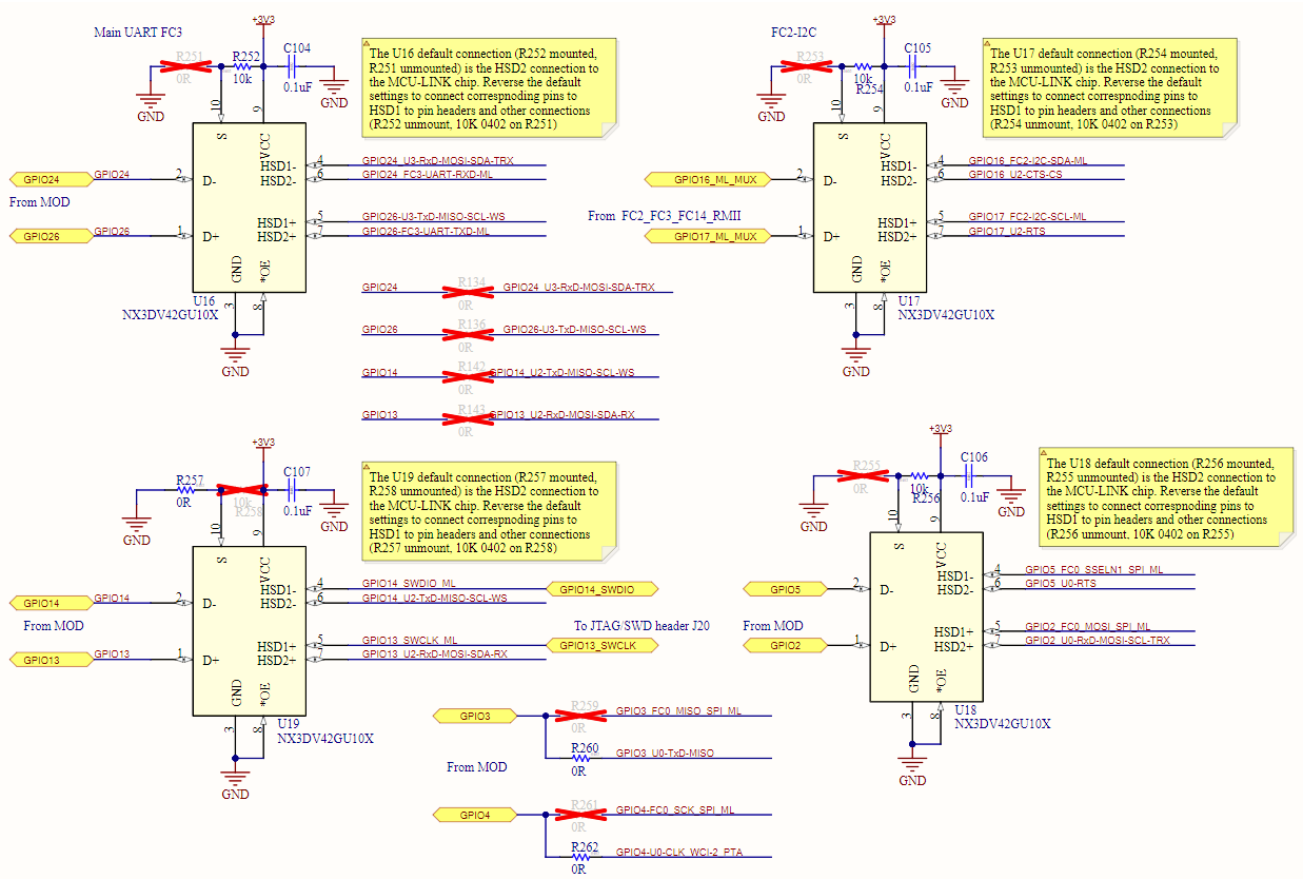


Figure 16: MCU-LINK connections

The MCU-Link JTAG/SWD Debug Probe [3], a free utility from NXP, is compatible with Windows 10, MacOS, and Linux. The utility provides an easy way to install firmware updates.

IRIS-W10 GPIO/Function	Connection	MCU-Link pin
GPIO 2/SPI0-MOSI	Not default	60
GPIO 3/SPI0-MISO	Not default	62
GPIO 4/SPI0-CLK	Not default	61
GPIO 5/ SPI0-SSELN1	Not default	74
GPIO 13/SWD-CLK	Default	54
GPIO 14/SWD-SWDIO	Default	81
GPIO 16/FC2-I2C_SDA	Not default	86
GPIO 17/ FC2-I2C_SCL	Not default	76
GPIO 24/UART3-RXD	Default	79
GPIO 26/UART3-TXD	Default	70

Table 6: MCU-Link connections

Figure 17 shows how the SWD programming interface on the MCU-Link connects to the JTAG needle connector (**J33**) or SWD connector (**J39**), where:

- **R182** forces the chip to ISP mode when populated with a 0402 0 Ω resistor.
- **LED1** to **LED5** indicate the connection and state of each interface chip.
- For minimum interference with the IRIS-W10 radio modules, the MCU-Link (**U13**) is powered using a separate linear voltage regulator **U20,** as shown in Figure 6.

Optionally, **R211** can be populated with a 1206 a 0 Ω resistor to power the MCU-Link chip (**U13**) from the **3V3** output of the DC-DC converter (**U11),** as shown in Figure 5.

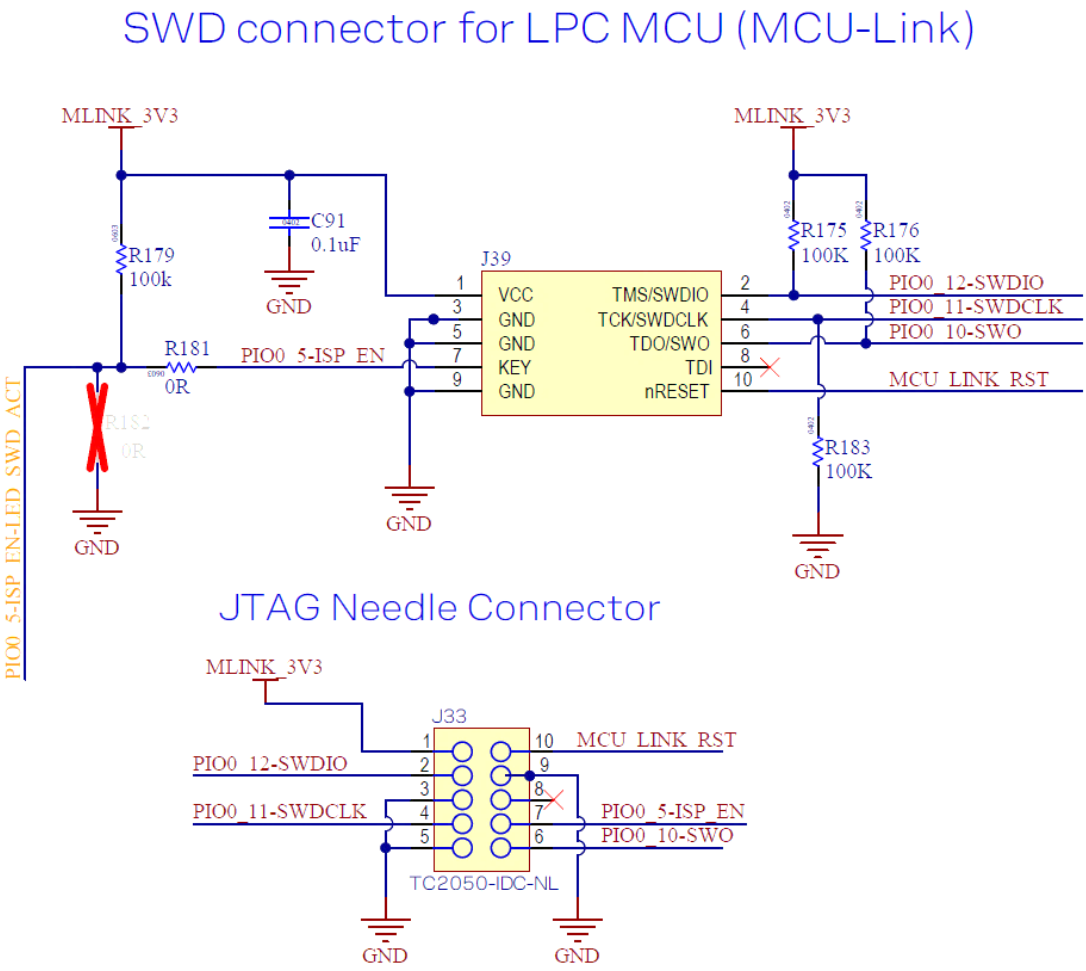


Figure 17: MCU-Link flashing options

3.5.2 USB-to-UART FTDI

EVK-IRIS-W1has two USB-to-UART FTDI chips that connect to the IRIS-W10 module over the serial interface:

- Quad channel USB-to-UART IC **(U108)**
- Single channel USB-to-UART IC **(U109)**

Table 7 describes the connections to the IRIS-W10 module through **U108**, and **U109**. The main COM port 3 on the FTDI chip connects to UART3 on the module via 1 kΩ resistors.

To enable this communication option:

- Connect the module's main UART (**UART3**) to the FTDI chip through USB switch **U16** by disconnecting **R252**, **R134**, and **R136** and adding a 0 Ω resistor at **R251**.
- If **U16** is unpopulated, install 0 Ω resistors at **R134** and **R136**.

IRIS-W10 pin	IRIS-W10 function	Resistor/Jumper enable	Interface IC function
B14	GPIO6/JTAG-TCK	R1/J15	FTDI-JTAG-TCK Pin 16, U108
B12	GPIO8/JTAG-TDI	R3/J36	FTDI- JTAG-TDI Pin 17, U108
A12	GPIO9/JTAG-TDO	R4/J34	FTDI- JTAG-TDO Pin 18, U108
A14	GPIO7/JTAG-TMS	R5/J35	FTDI- JTAG-TMS Pin 19, U108
B11	GPIO10/JTAG-RESETn	R6/J37	FTDI- JTAG-RESETn Pin 22, U108
-	JTAG-Bootn	R7	FTDI- Pin 23, U108
M9	GPIO 4/SPI-CLK	R8	FTDI-SPI-CLK, Pin 26, U108
M10	GPIO 2/SPI-MOSI	R9	FTDI-SPI-MOSI, Pin 27, U108
N10	GPIO 3/SPI-MISO	R37	FTDI-SPI-MISO, Pin 28, U108
M11	GPIO 0/SPI-CS	R38	FTDI-SPI-CS, Pin 29, U108
N11	GPIO 1/SPI-WP	R41	FTDI-SPI-WP, Pin 30, U108
M2	GPIO 21/SPI-HD	R42	FTDI-SPI-HD, Pin 32, U108
A10	GPIO 26/UART-TXD	R52	FTDI-UART-RXD, Pin 39, U108, Pin 17, U109
B9	GPIO 24/UART-RXD	R53	FTDI-UART-TXD, Pin 38, U108, Pin 1, U109
B8	GPIO 19/UART-RTS	R54	FTDI-UART-RTS, Pin 40, U108, Pin 19, U109
A9	GPIO 20/UART-CTS	R55	FTDI-UART-CTS, Pin41, U108, Pin 6, U109
A11	GPIO 11/UART-DTR	R56	FTDI-UART-DTR, Pin 43, U108, Pin 18, U109
B10	GPIO 12/UART-DSR	R57	FTDI-UART-DSR, Pin 44, U108, Pin 4, U109
-	-	R58	Pin 48, U108
-	-	R59	Pin 52, U108
-	-	R60	Pin 53, U108
-	-	R61	Pin 54, U108
-	-	R63	Pin 55, U108
-	-	R64	Pin 57, U108
-	-	R65	Pin 58, U108
-	-	R66	Pin 59, U108

Table 7: Main COM port connections

3.6 JTAG/SWD debug interfaces

There are two interfaces for debugging and programming IRIS-W10: **SWD** and **JTAG**. When powering up or resetting the EVK-IRIS-W10, the status of the **RF-CNTL** strapping pins determine the active interface, as described in Table 8.

	RF-CNTL0	RF-CNTL1	RF-CNTL2	RF-CNTL3
SWD	F	F	0	F
JTAG	F	F	F	F

Table 8: RF-CTRL strap-in (F=Float)

Use **SW8** to set the RF-CNTL signals to the desired status as shown in Figure 18.

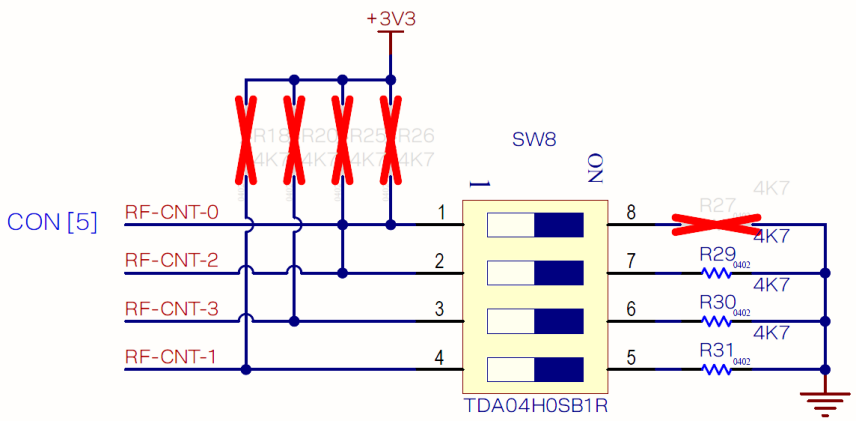


Figure 18: RF-CNTRL strap-in SW8

- By default, the **SWD** interface (**GPIO13_SWCLK** and **GPIO14_SWDIO**) is enabled on both the USB3 MCU-Link port (On-board debugger) and **J45** needle connector at the same time, via resistors **R2, R12**.
  - To enable the **SWD** interface on the **J20** JTAG connector, remove resistors **R2** and **R12**, and install 0 Ω, 0402 resistors on **R48** and **R116**, as shown in Figure 19.
- To use **JTAG** interface on the EVK, disable **SWD** interface by setting **RF_CTRL2** to the rest position on DIP switch **SW8.** The **JTAG** interface of the IRIS-W10 module is directly connected to **J20** and **FTDI** chip through resistor network, and pin headers simultaneously, as shown in Table 9.
  - External target hardware can be attached to the **J20** JTAG connector for firmware programming and debug. **J20** is implemented with a 2x5 header with 1.27 mm pitch.

The SWD/JTAG multiplex connections and default jumper positions are shown in Figure 19.

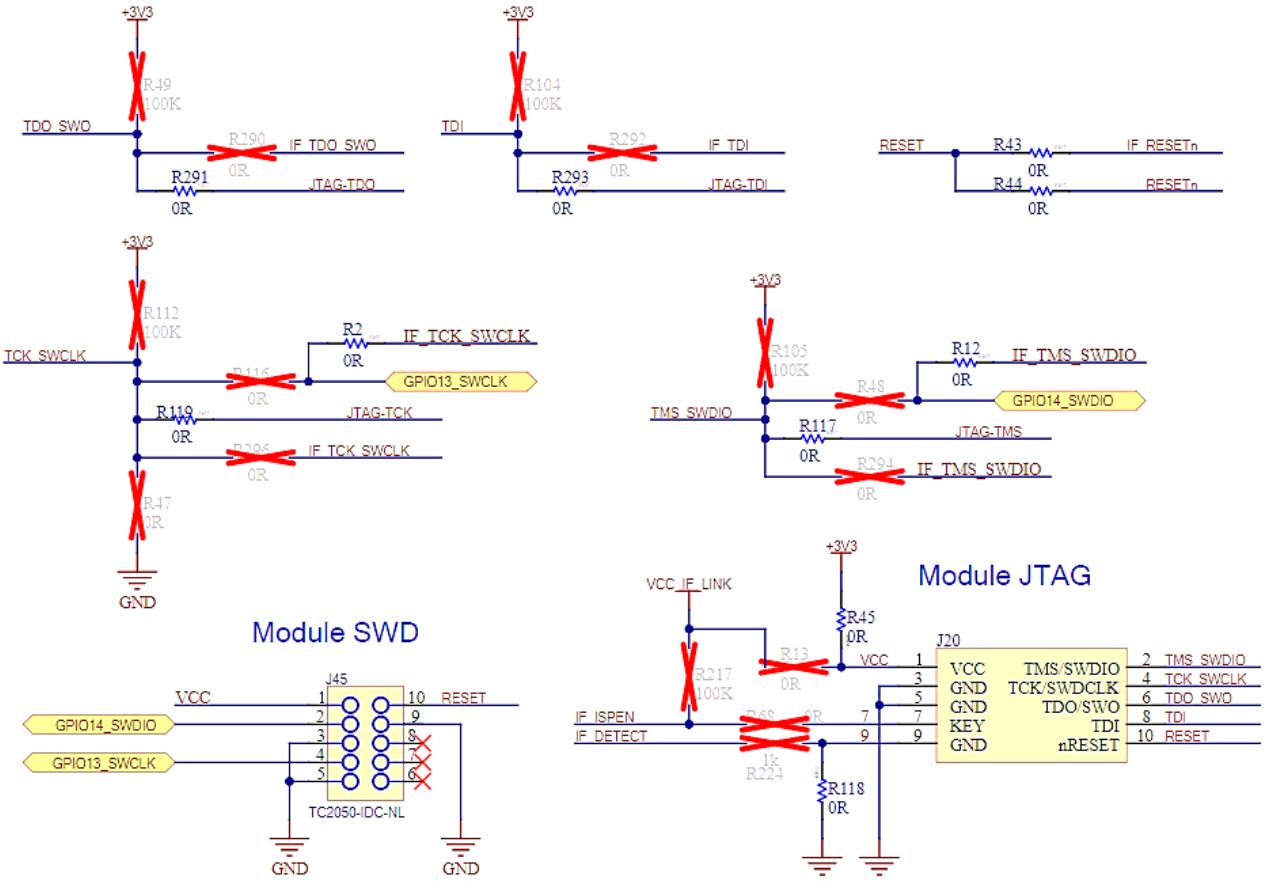


Figure 19: SWD/JTAG multiplex connections

Jumpers described in Table 9 and shown Figure 20 should be set to their default connections. The default positions on the board are shown in Figure 21.

IRIS-W10 pin	IRIS-W10 function	Resistor/Jumper	FTDI pin	Header/Pin
A14	GPIO7/JTAG-TMS	R5/J34	19	J70/3
B14	GPIO6/JTAG-TCK	R1/J15	16	J10/3
A12	GPIO9/JTAG-TDO	R4/J36	18	J70/4
B12	GPIO8/JTAG-TDI	R3/J35	17	J10/4
B11	GPIO10/JTAG-RESETn	R6/J37	22	J47/8

Table 9: JTAG connections

A schematic for JTAG multiplex connections through J5 and J34–J37 is shown in Figure 20.

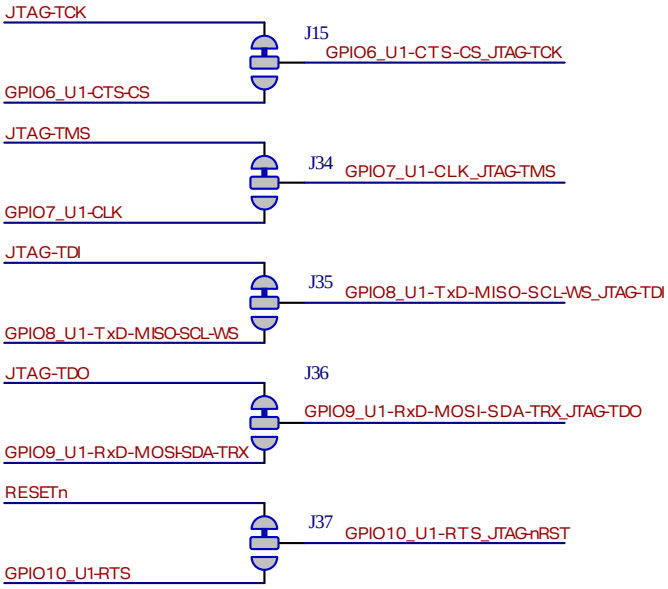


Figure 20: External JTAG debug interface demultiplexing

The default connections for J5 and J34–J37 as seen on the EVK board are shown in Figure 21.

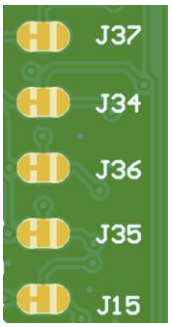


Figure 21: Jumpers default connections

3.7 32.768 kHz low frequency clock

EVK-IRIS-W10 is equipped with a **32.768 kHz** crystal that can be used to supply the IRIS-W1 module as an external RTC clock source.

GPIO21 and **GPIO23** multiplex some of the RMII signals and the external crystal oscillator option. Figure 22 shows how to demultiplex those GPIOs.

To enable low frequency clock, populate **R267, R271** with 0 Ω resistors and disconnect **R268, R269, R272, R273**.

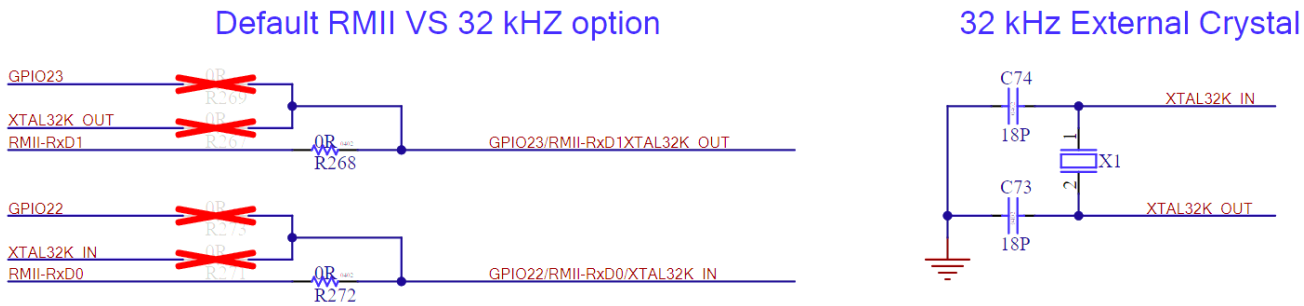


Figure 22: Schematic – 32 kHz crystal

3.8 mikroBUS slots

The EVK-IRIS-W10 features four mikroBUS standard-compatible slots, each offering multiple options for the RESET signal. By default, none of the slots are enabled for use. This is primarily because the Flexcomm interface pins in the NXP RW612 chip are multiplexed.

Table 10 outlines the pin configuration for each mikroBUS slots.

		GPIO	UART	SPI	I2C	SDIO	JTAG	WCI-2	RMII	FTDI
mikroBUS 1	**Flexcomm 0**	GPIO0		CS						SPI-RX
		GPIO2	RX	MOSI	SDA					SPI-CS
		GPIO3	TX	MISO	SCL					SPI-TX
		GPIO4		CLK				WCI-2		SPI-Clk
mikroBUS 2	**Flexcomm 2**	GPIO13	RXD	MOSI	SDA					
		GPIO14	TXD	MISO	SCL					
		GPIO15		CLK		SDIO-CLK				
		GPIO16		CS		SDIO-D3				
mikroBUS 3	**Flexcomm 1**	GPIO6		CS			JTAG			
		GPIO7		CLK			JTAG			
		GPIO8	TXD	MISO	SCL		JTAG			
		GPIO9	RXD	MOSI	SDA		JTAG			
mikroBUS 4	**Flexcomm 14**	GPIO53		CS						
		GPIO54		CLK						
		GPIO56	TXD	MISO	SCL				MDC	
		GPIO57	RXD	MOSI	SDA				MDIO	

Table 10: mikroBUS slots pin configuration

When using the mikroBUS slots, keep in mind the following considerations:

- WCI-2 and FTDI-SPI functions on mikroBUS1 cannot operate at the same time as SPI, since they share GPIO0, GPIO2, GPIO3, and GPIO4. However, WCI-2 can be used simultaneously with UART or I2C.
- SDIO function on mikroBUS2 cannot operate at the same time as SPI, since they share GPIO 15 and GPIO 16. However, SDIO can be used simultaneously with UART or I2C.
- mikroBUS3 cannot be fully utilized when JTAG function is active in the EVK, as it shares all GPIOs.
- mikroBUS4 cannot be fully utilized when RMII function is active in the EVK, as it shares GPIO 56 and GPIO 57.
- Each mikroBUS slot has a dedicated GPIO for the analog pin (AN), as shown in Table 11.
- All four mikroBUS slots share GPIO 27 as the PWM pin, which can only be used in one slot at a time. By default, it is enabled on mikroBUS2, as indicated in Table 11 and Table 12.
- GPIO 18 is shared as the INT pin across all four mikroBUS slots, meaning it can only be used in one slot at a time and not simultaneously with SDIO. By default, it is enabled on mikroBUS2, as shown in Table 11 and Table 12.
- All four mikroBUS slots share the same RESET options: either the global **RESETn** signal of the EVB (active low) or GPIO 50, intended as a user-defined RESET. This is due to the varying reset options of Click boards, as detailed in Table 13.

Table 11 outlines the default GPIO assignments for the analog, INT, and PWM signals on each mikroBUS slot.

	mikroBUS 1	mikroBUS 2	mikroBUS 3	mikroBUS 4	SDIO
GPIO18	INT	INT*	INT	INT	SDIO-D2
GPIO27	PWM	PWM*	PWM	PWM	
GPIO46			AN		
GPIO47		AN			
GPIO48	AN				
GPIO49				AN	
GPIO50	RESET	RESET	RESET	RESET	

Table 11: mikroBUS slots other pin configuration, (*) default connection

Table 12 shows the default jumper connection for the INT, and PWM signals on each mikroBUS slot.

	GPIO 18 INT	GPIO 27 PWM
mikroBUS 1	J92	J89
mikroBUS 2	J91*	J88*
mikroBUS 3	J62	J61
mikroBUS 4	J93	J90

Table 12: mikroBUS PWM/INT jumpers, (*) default connection

Table 13 shows the resistor configuration of both reset signals on each mikroBUS slot.

	RESETn	GPIO 50
mikroBUS 1	R67	R71
mikroBUS 2	R11	R21
mikroBUS 3	R46	R70
mikroBUS 4	R10	R19

Table 13: mikroBUS RESET configuration resistors, (*) default connection

3.8.1 mikroBUS 1 slot

- UART is enabled by default through jumpers **J25** and **J27**, as shown in Table 14.
- To enable I2C or SPI, adjust the state of jumpers **J83** and **J84** for SPI, or **J80** and **J85** for I2C, as shown in Figure 23. Also, toggle jumpers **J25** and **J27** to disable UART.
- For jumper configurations of other signals in the mikroBUS slots, refer to Table 11, Table 12, and Table 13.
- Install 1KΩ, 0603 on **R67** or **R71** to select RESET source.

		UART		SPI		I2C	
		RX	**TX**	**MOSI**	**MISO**	**SDA**	**SCL**
mikroBUS 1	GPIO 2	J27*		J84		J85	
	GPIO 3		J25*		J83		J80

Table 14: mikroBUS 1 interface options, (*) default connection

Figure 23 shows the connection of mikroBUS 1 to the Flexcomm 0 interface.

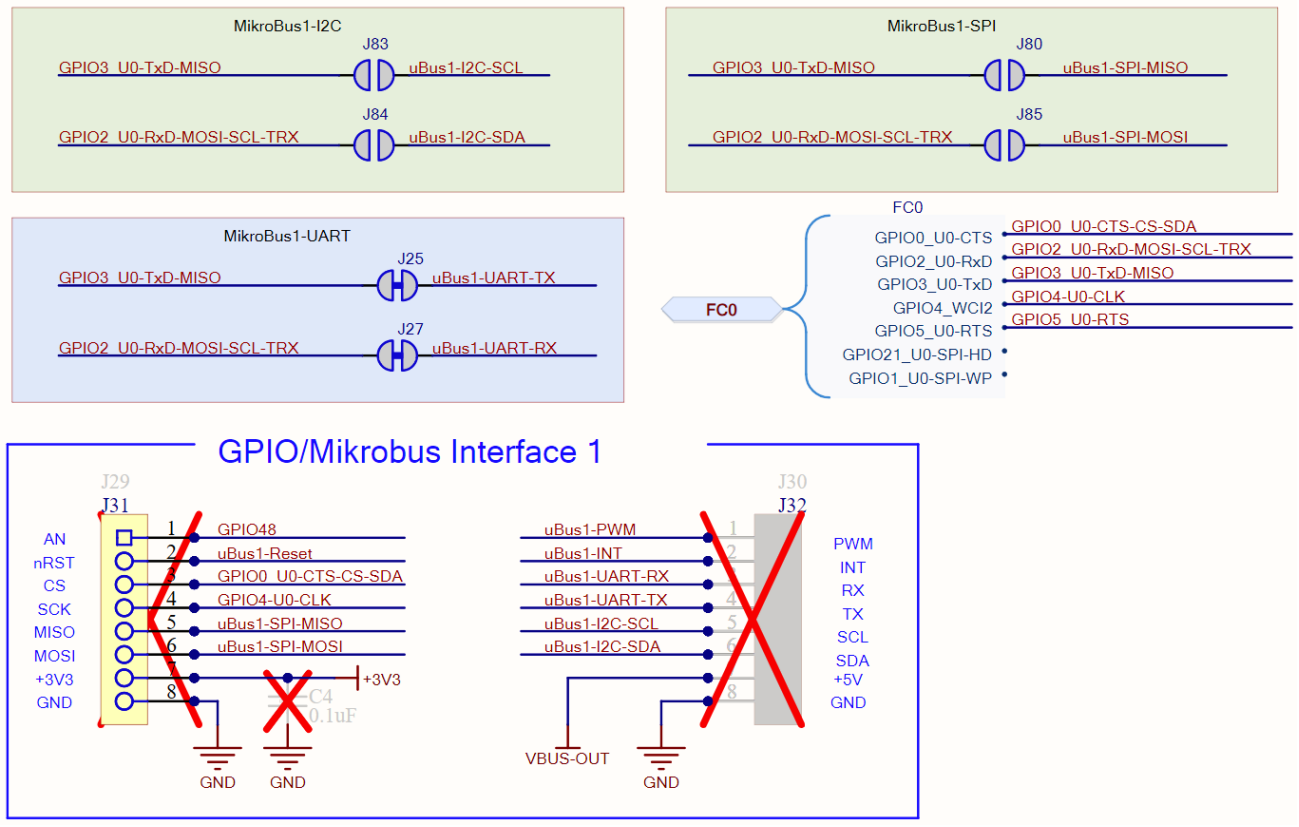


Figure 23: mikroBUS 1 slot signals and Jumper configuration

3.8.2 mikroBUS 2 slot

- UART is enabled by default through jumpers J26 and J28, as shown in Table 15.
- To enable I2C or SPI, adjust the state of jumpers J81 and J86 for SPI, or J82 and J87 for I2C, as shown in Figure 24. Also, toggle jumpers J26 and J28 to disable UART.
- Install 1KΩ, 0603 on R11 or R21 to select RESET source.
- For jumper configurations of other signals in the mikroBUS slots, refer to Table 11, Table 12, and Table 13.

		UART		SPI		I2C	
		RX	TX	MOSI	MISO	SDA	SCL
mikroBUS 2	GPIO 13	J28*		J86		J87	
	GPIO 14		J26*		J81		J82

Table 15: mikroBUS 2 interface options, (*) default connection

Figure 24 shows the connection of mikroBUS 2 to the Flexcomm 2 interface.

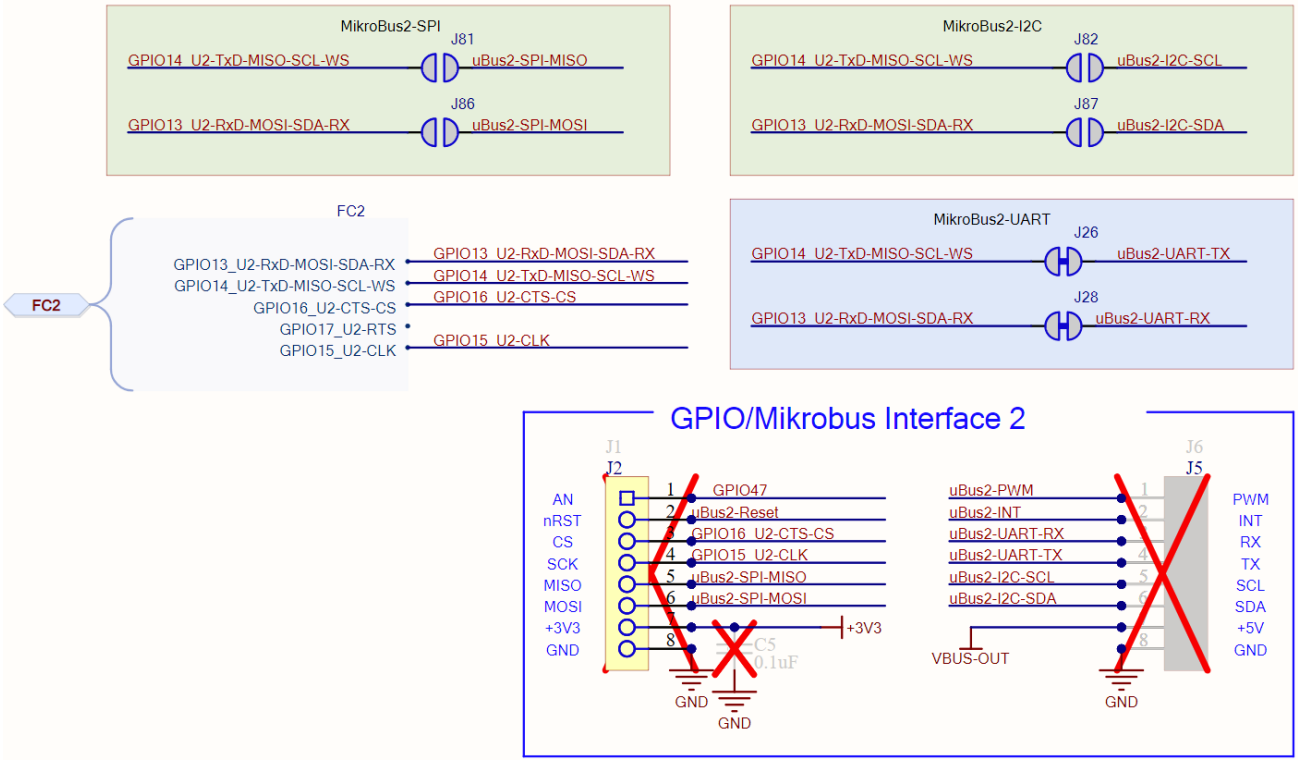


Figure 24: mikroBUS 2 slot signals and Jumper configuration

3.8.3 mikroBUS 3 slot

- UART is enabled by default through jumpers J16 and J18, as shown in Table 16.
- To enable I2C or SPI, adjust the state of jumpers J73 and J76 for SPI, or J11 and J79 for I2C, as shown in Figure 26. Also, toggle jumpers J16 and J18 to disable UART.
- For jumper configurations of other signals in the mikroBUS slots, refer to Table 11, Table 12, and Table 13.
- Toggle the state of jumpers **J15, J34, J35,** and **J36** shown in Figure 25 to access Flexcomm1 interface in mikroBUS.
- Install 1KΩ, 0603 on R46 or R70 to select RESET source.

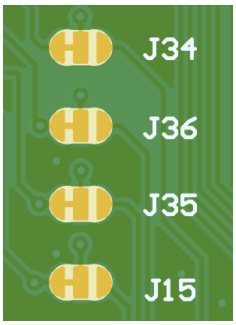


Figure 25: Jumpers J15, J34, J35, and J36 default connection

		UART		SPI		I2C	
		RX	TX	MOSI	MISO	SDA	SCL
mikroBUS 3	GPIO 9	J18*		J76		J79	
	GPIO 8		J16*		J73		J11

Table 16: mikroBUS 3 interface options, (*) default connection

Figure 26 shows the connection of mikroBUS 3 to the Flexcomm 1 interface.

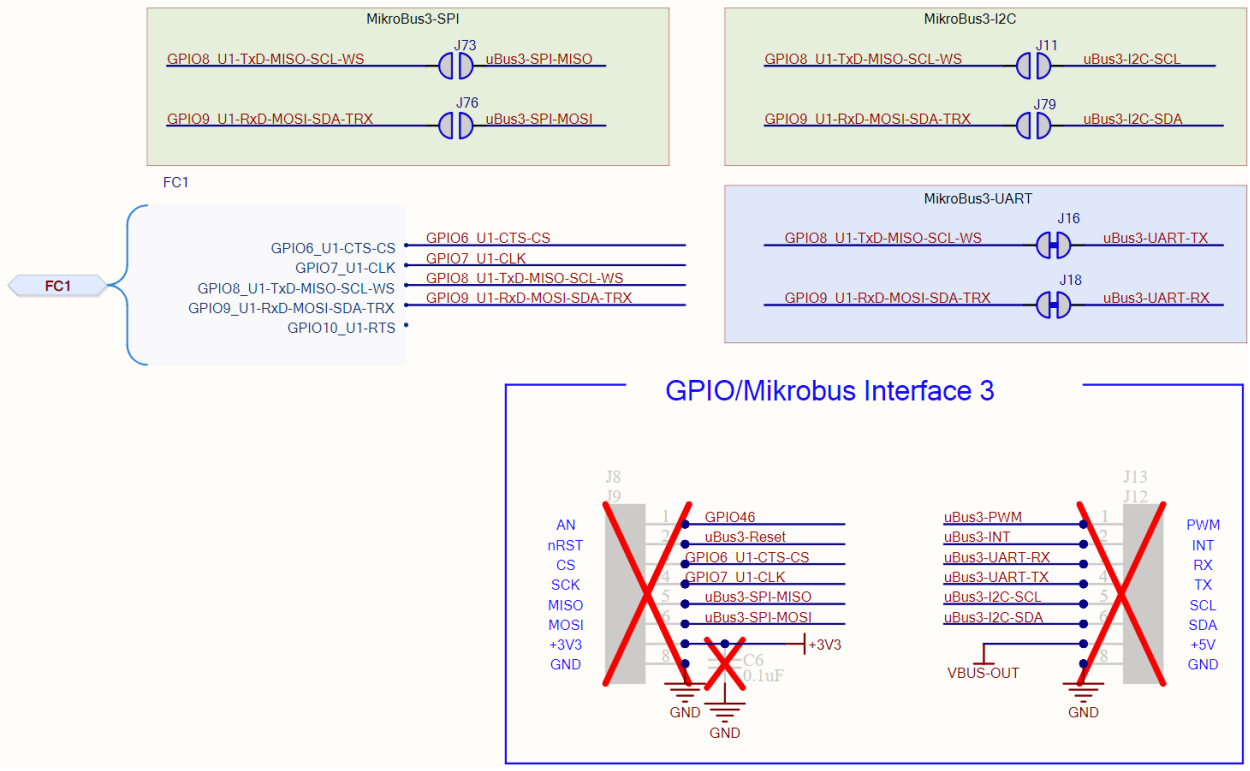


Figure 26: mikroBUS 3 slot signals and jumper configuration

3.8.4 mikroBUS 4 slot

- UART is enabled by default through jumpers **J17** and **J19**, as shown in Table 17.
- To enable I2C or SPI, adjust the state of jumpers **J74** and **J77** for SPI, or **J75** and **J78** for I2C, as shown in Figure 27. Also, toggle jumpers **J17** and **J19** to disable UART.
- For jumper configurations of other signals in the mikroBUS slots, refer to Table 11, Table 12, and Table 13.
- Install 0 Ω, 0402 on **R115**, **R264**, **R266** and remove **R111**, **R263**, **R265** to access Flexcomm 14 interface in mikroBUS.
- Install 1KΩ, 0603 on **R10** or **R19** to select RESET source.

		UART		SPI		I2C	
		RX	**TX**	**MOSI**	**MISO**	**SDA**	**SCL**
mikroBUS 4	GPIO 57	J19*		J77		J78	
	GPIO 56		J17*		J74		J75

Table 17: mikroBUS 4 interface options, (*) default connection

Figure 27 shows the connection of mikroBUS 4 to the Flexcomm 14.

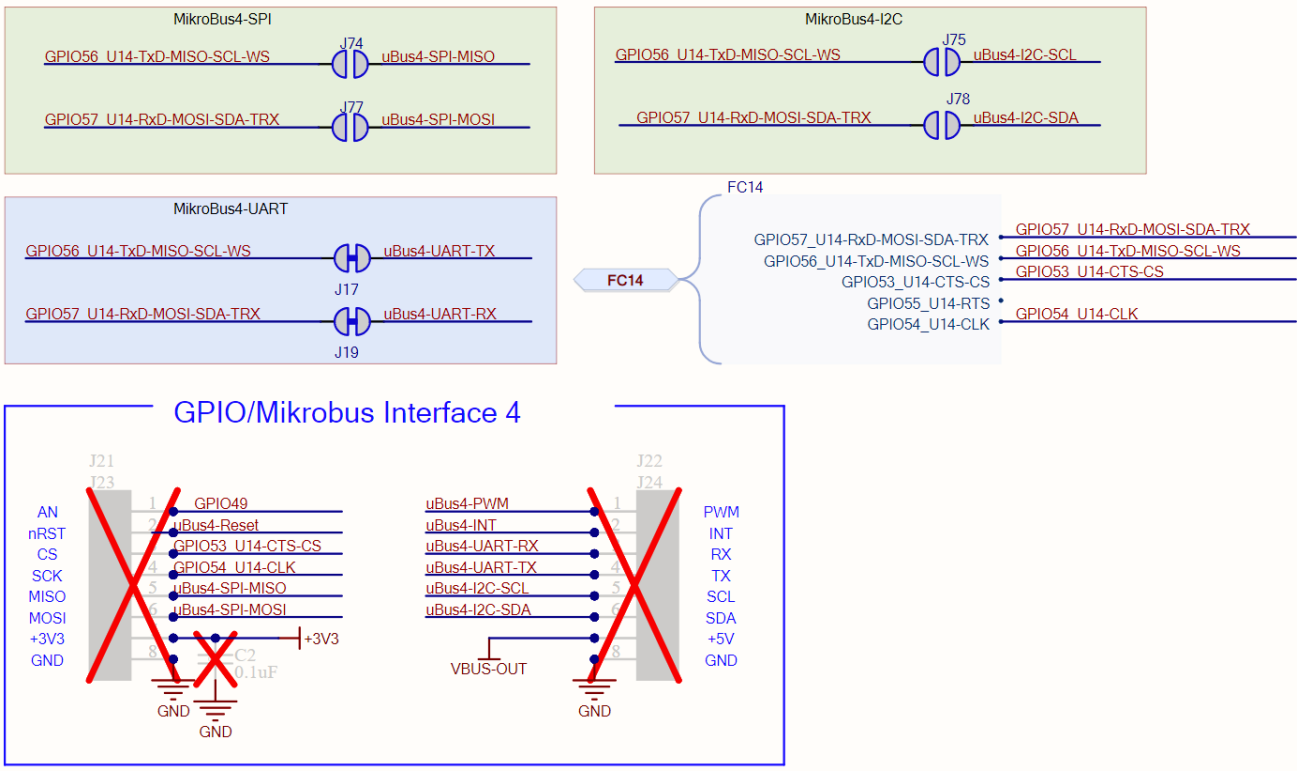


Figure 27: mikroBUS 4 slot signals and jumper configuration

3.9 RMII

The EVK-IRIS-W10 supports RMII standard through 10BASE-T/100BASE-TX Physical Layer Transceiver **U1** (KSZ8081MNXRNB) and 1 Port RJ45 Surface Mount 10/100 Base-T, AutoMDIX **T1** (74980111211).

RMII is enabled by default. However, there are four groups of resistors that demultiplex the RMII signals from other functions that cannot be used simultaneously with RMII, as described in Table 18 and illustrated in Figure 28.

- Group 1, Flexcomm 14 **RX, TX, RTS** signals – populate **R111, R263, R265** with 0 Ω, 0402 resistors and disconnect **R115, R264, R266**.
- Group 2, Flexcomm 3 Clock signal – populate **R279** with a 0 Ω, 0402 resistor and disconnect **R282.**
- Group 3, 32.768 kHz external crystal oscillator – populate **R268, R272** with 0 Ω, 0402 resistors and disconnect **R269, R267, R271, R272.**
- Group 4, pin headers and reset signals – populate **R210, R277, R278, R283, R284, R286** with 0 Ω, 0402 resistors and disconnect **R270, R275, R276, R280, R281, R285**.

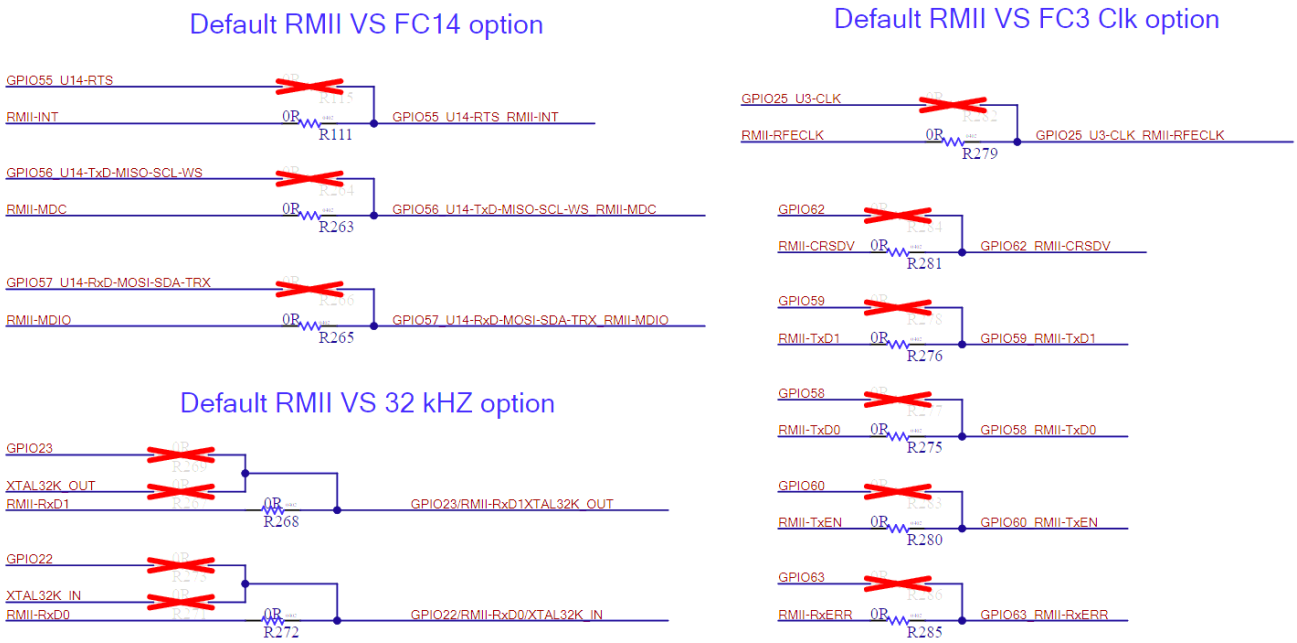


Figure 28: RMII demultiplexing resistors schematic

IRIS-W10 pin name	IRIS-W10 function	Related resistors	Interface	Pin on U1
N1	GPIO63/RMII-RXER	R285/R141	PHY-RXER	Pin 20
J3	GPIO60/RMII-TXEN	R280	PHY-TXEN	Pin 23
M1	GPIO62RMII-CRSDV	R281/R140	PHY-CRSDV	Pin 18
L2	GPIO56/RMII-MDC	R263	PHY-DMC	Pin 12
L1	GPIO57/RMII-MDO	R265	PHY-MDIO	Pin 11
K2	GPIO59/RMII-TXD1	R276	PHY-TXD1	Pin 25
K1	GPIO58/RMII-TXD0	R275/R138	PHY-TXD0	Pin 24
J2	GPIO23/RMII-RXD1	R268/R139	PHY-RXD1	Pin 15
J1	GPIO22/RMII-RXD0	R272	PHY-RXD0	Pin 16
L4	GPIO55/RMII-INT	R111/R149	PHY-INT	Pin 21
F3	GPIO25/RMII-RXCLK	R279	PHY-RXCLK	Pin 9
M2	GPIO21/RMII-RESET	R210	PHY-RESET	Pin 32

Table 18: RMII signals and pins

3.9.1 RMII Strap-in options

The strap-in pins are latched at the de-assertion of reset. During power-up or reset, the MAC RMII input pins can briefly drive high or low, which can unintentionally latch the RMII PHY strap-in pins into incorrect high or low states. To reduce the risk, add external pull-up (4.7 kΩ) or pull-down (1.0 kΩ) resistors to the PHY strap-in pins. To ensure that the intended values are strapped-in correctly, implement the connection configuration shown in Figure 29 and described in Table 19.

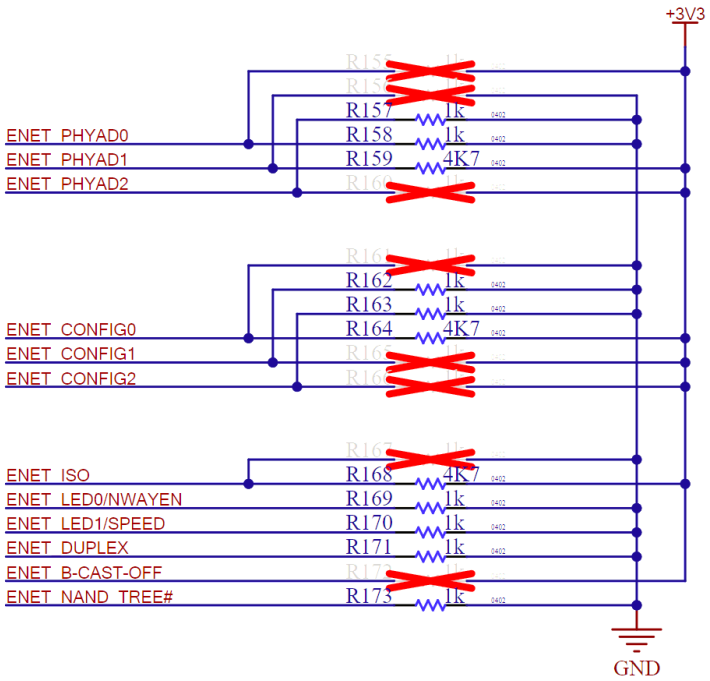


Figure 29: PHY circuit strapping options

Configuration	Description
PHYAD [2:0]	PHY ADDR 00-XXX (00010 DEFAULT)
CONFIG [2:0]	IF MODE 001 RMII 101 RMII Back-to-Back (not supported) xxx Reserved-not used
ISO	ISOLATE mode Pull-up = Enable Pull-down (default) = Disable
SPEED	SPEED mode SPEED Pull-up (default) = 100Mbps Pull-down = 10Mbps
DUPLEX	DUPLEX mode Pull-up (default) = Half Duplex Pull-down = Full Duplex
NWAYEN	Nway Auto-Negotiation Pull-up (default) = Enable Pull-down = Disable
B_CAST_OFF	Broadcast Off - for PHY Address 0 Pull-up = PHY Address 0 set as unique PHY address Pull-down (default) = PHY Address 0 set as broadcast PHY address
NAND_TREE#	NAND Tree Mode Pull-up (default) = Disable Pull-down = Enable

Table 19: RMII strap-in

3.10 Pin headers

All GPIOS on IRIS-W10, except PSRAM-related GPIOs, are accessible on through-hole (TH) pin headers **J3**, **J4**, **J10**, **J43**, **J44**, **J47**, **J48, and J70**, as shown in Figure 30.

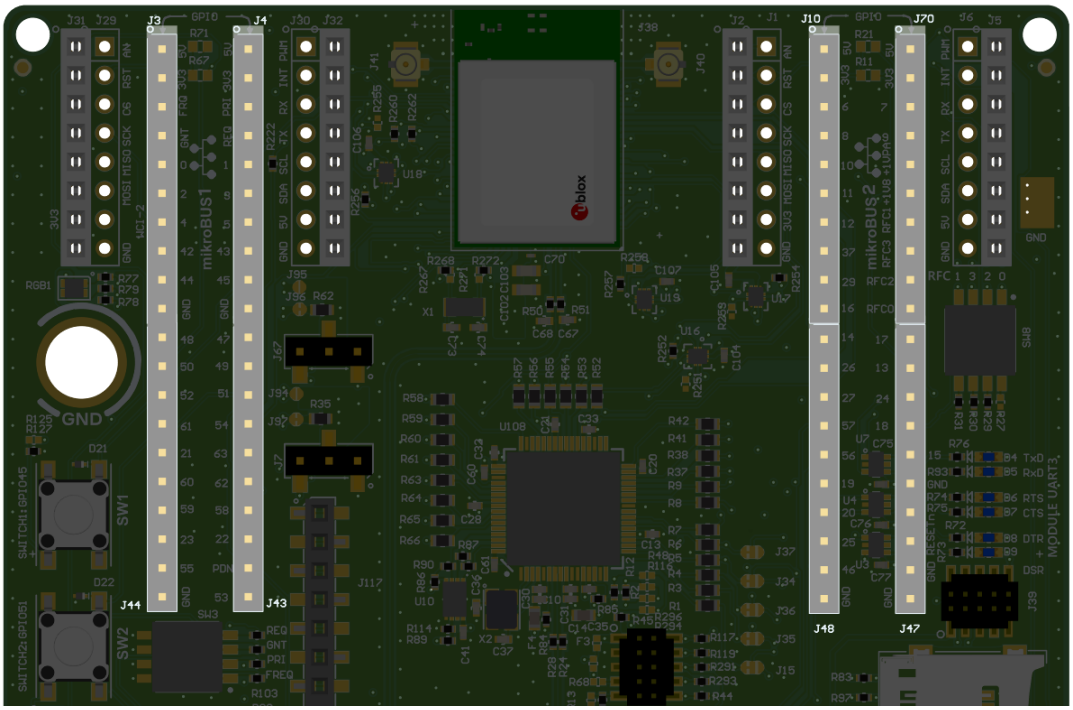


Figure 30: Pin headers

GPIOs described in Table 20, Table 21, and Table 22 are either:

- Connected to the pin headers by default and can't be disconnected
- Connected to the pin headers by default but can be disconnected
- Not connected to the pin headers by default but can be connected
- Possibly connected to the pin headers and other locations simultaneously

The right columns in Table 20, and Table 21 show how to connect, disconnect, or disconnect other locations related to each GPIO or function.

GPIO	Header	Pin	Connection	How to connect/disconnect other locations
GPIO0	J3	5	Default and can't disconnect	Disconnect **R38** and MikroBUS 1- **CS**
GPIO1	J4	5	Default and can't disconnect	Disconnect **R41**
GPIO2	J3	6	Default but can disconnect	Disconnect **R9**, **R255**, **J27, J84, J85** populate **U18** and **R256** with 10 kΩ resistor.
GPIO3	J4	6	Default but can disconnect	Disconnect **R37**, **R255**, **J25, J80, J83** populate **R260** with a 0 Ω resistor.
GPIO4	J3	7	Default (WCI-2)	Disconnect **R8** and MikroBUS 1- **SCK** and populate **R262** with a 0 Ω resistor.
GPIO5	J4	7	Default but can disconnect	Disconnect **R255** and populate **U18** and **R256** with 10 kΩ resistor.
GPIO6	J10	3	Default and can't disconnect	Disconnect **J15**
GPIO7	J70	3	Default and can't disconnect	Disconnect **J34**
GPIO8	J10	4	Default and can't disconnect	Disconnect **J35**
GPIO9	J70	4	Default and can't disconnect	Disconnect **J36**
GPIO10	J10	5	Default and can't disconnect	Disconnect **J37**
GPIO11	J10	6	Default and can't disconnect	Disconnect **R257** and **U3**
GPIO12	J10	7	Default and can't disconnect	Disconnect **R256** and **U3**

GPIO	Header	Pin	Connection	How to connect/disconnect other locations
GPIO13	J47	2	Not default	Disconnect **R257** and populate **U19** and **R258** with 10 kΩ resistor**.** Or populate **R143** with **a 0 Ω resistor.**
GPIO14	J48	1	Not default	Disconnect **R257** and populate **U19** and **R258** with 10 kΩ resistor. Or populate **R142** with a 0 Ω resistor.
GPIO15	J47	5	Default but can disconnect	**J55** default connection, disconnect MikroBUS 2- **SCK**
GPIO16	J10	10	Default but can disconnect	**J56** default connection, disconnect **R253** and populate **U17** and **R254** with 10 kΩ resistor**.** Disconnect MikroBUS 2- **CS**
GPIO17	J47	1	Default but can disconnect	**J57** default connection, disconnect **R253** and populate **U17** and **R254** with 10 kΩ resistor**.**
GPIO18	J47	4	Default but can disconnect	**J60** default connection, disconnect **J62, J91, J92, J93**
GPIO19	J48	6	Default but can disconnect	**J58** default connection, disconnect **R55** and **U4**
GPIO20	J48	7	Default but can disconnect	**J59** default connection, disconnect **R54** and **U4**
GPIO21	J44	5	Not default	Disconnect **R210, R42** and populate **R270** with a 0 Ω resistor.
GPIO22	J43	8	Not default	Disconnect **R272** and populate **R273** with **a** 0 Ω resistor.
GPIO23	J44	8	Not default	Disconnect **R271** and populate **R269** with **a** 0 Ω resistor.
GPIO24	J47	3	Not default	Disconnect **R252** and populate **U19** and **R251** with 10 kΩ resistor. Or populate **R134** with a 0 Ω resistor.
GPIO25	J48	8	Not default	Disconnect **R279** and populate **R282** with a 0 Ω resistor.
GPIO26	J48	2	Not default	Disconnect **R252** and populate **U19** and **R251** with 10K. Or populate **R136** with a 0 Ω resistor.
GPIO27	J48	3	Default and can't disconnect	Disconnect **J61, J88, J89, J90**
GPIO29	J10	9	Default and can't disconnect	
GPIO37	J10	8	Default and can't disconnect	
GPIO42	J3	8	Default and can't disconnect	Disconnect **R79**
GPIO43	J4	8	Default and can't disconnect	Disconnect **R78**
GPIO44	J3	9	Default and can't disconnect	Disconnect **R77**
GPIO45	J4	9	Default and can't disconnect	Release **SW1**
GPIO46	J48	9	Default and can't disconnect	Disconnect MikroBUS 3- **AN**
GPIO47	J43	1	Default and can't disconnect	Disconnect MikroBUS 2- **AN**
GPIO48	J44	1	Default and can't disconnect	Disconnect MikroBUS 1- **AN**
GPIO49	J43	2	Default and can't disconnect	Disconnect MikroBUS 4- **AN**
GPIO50	J44	2	Default and can't disconnect	Disconnect **R19, R21, R70, R71**
GPIO51	J43	3	Default and can't disconnect	Release **SW2**
GPIO52	J44	3	Default and can't disconnect	
GPIO53	J43	10	Default and can't disconnect	Disconnect MikroBUS 4- **CS**
GPIO54	J43	4	Default and can't disconnect	Disconnect MikroBUS 4- **SCK**
GPIO55	J44	9	Not default	Disconnect **R111** and populate **R115** with a 0 Ω resistor.
GPIO56	J48	5	Not default	Disconnect **R263, J17, J74, J75** and populate **R264** with a 0 Ω resistor.
GPIO57	J48	4	Not default	Disconnect **R265, J19, J77, J78** and populate **R266** with a 0 Ω resistor.
GPIO58	J43	7	Not default	Disconnect **R275** and populate **R277** with a 0 Ω resistor.
GPIO59	J44	7	Not default	Disconnect **R276** and populate **R278** with a 0 Ω resistor.
GPIO60	J44	6	Not default	Disconnect **R280** and populate **R283** with a 0 Ω resistor.
GPIO61	J44	4	Default and can't disconnect	
GPIO62	J43	6	Not default	Disconnect **R281** and populate R284 with a 0 Ω resistor.
GPIO63	J43	5	Not default	Disconnect **R285** and populate R286 with a 0 Ω resistor.

Table 20: GPIO mapping to the pin headers

IRIS-W10 function	Header	Pin	Default	How to connect/disconnect other locations
EXT-FRQ	J3	3	Default and can't disconnect	Disconnect **R82**
EXT-GNT	J3	4	Default and can't disconnect	Disconnect **R99**
EXT-PRI	J4	3	Default and can't disconnect	Disconnect **R94**
EXT-REQ	J4	4	Default and can't disconnect	Disconnect **103**
WCI-2	J3	7	Default (GPIO4)	Disconnect **R8** and MikroBUS 1- **SCK and** populate **R262** with a 0 Ω resistor.
nRESET	J47	8	Default (GPIO10)	**J37** default connection, disconnect **R6, R10, R11, R46, R67** release **SW5**
RF-CNT-0	J70	10	Default and can't disconnect	Release **SW8-1,** disconnect **R26**
RF-CNT-1	J70	7	Default and can't disconnect	Release **SW8-4,** disconnect **R18**
RF-CNT-2	J70	9	Default and can't disconnect	Release **SW8-2,** disconnect **R25**
RF-CNT-3	J70	8	Default and can't disconnect	Release **SW8-3,** disconnect **R20**
PDn	J43	9	Default	Populate **R222** with a 0 Ω resistor and disconnect **R14, R15**
+1V8	J70	6	Default and can't disconnect	Disconnect **J115**
+VPA	J70	5	Default and can't disconnect	Disconnect **J116**

Table 21: Other signals mapping to the pin headers

Header	GND	+3.3 V	+5 V
J3	10	2	1
J4	10	2	1
J44	10		
J47	6,8,9,10		
J48	10		
J10		2	1
J70		2	1

Table 22: Power signals mapping to the pin headers

3.11 QSPI memory

EVK-IRIS-W10 provides an option for external memory. A Quad SPI PSRAM (**U2**), shown in Figure 31 and described in Table 23, can be optionally mounted on the rear side of EVK board.

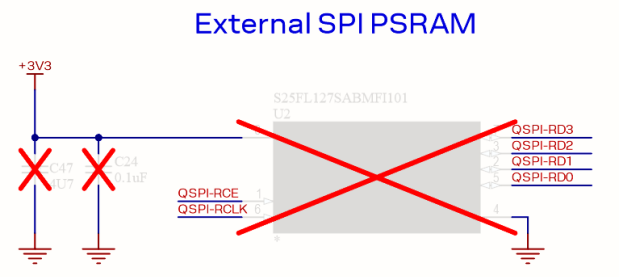


Figure 31: QSPI memory

Interface function	IRIS-W10 pin	Interface IC function
QSPI-RCE	A5	QSPI SRAM slave select 0
QSPI-RCLK	A2	QSPI SRAM interface clock 0
QSPI-RD0	B4	Data bit 0 for QSPI SRAM interface
QSPI-RD1	A4	Data bit 1 for QSPI SRAM interface
QSPI-RD2	B3	Data bit 2 for QSPI SRAM interface
QSPI-RD3	A3	Data bit 3 for QSPI SRAM interface

Table 23: QSPI memory signals and pins

3.12 Jumpers

When configuring GPIO functionality, EVK-IRIS-W10 supports several solder-bridge jumpers. Read the instructions carefully before altering any jumper to determine how each function is configured. Several jumpers are wired in series to demultiplex GPIOs that have multiple functions.

Table 24 shows all jumper types and locations on the EVK.

Jumper	Jumper type	Location	Jumper	Jumper type	Location
J11	Solder-bridge open	Bottom	J76	Solder-bridge open	Bottom
J14	Solder-bridge short	Top	J77	Solder-bridge open	Bottom
J15	Solder-bridge option	Top	J78	Solder-bridge open	Bottom
J16	Solder-bridge short	Bottom	J79	Solder-bridge open	Bottom
J17	Solder-bridge short	Bottom	J80	Solder-bridge open	Bottom
J18	Solder-bridge short	Bottom	J81	Solder-bridge open	Bottom
J19	Solder-bridge short	Bottom	J82	Solder-bridge open	Bottom
J25	Solder-bridge short	Bottom	J83	Solder-bridge open	Bottom
J26	Solder-bridge short	Bottom	J84	Solder-bridge open	Bottom
J27	Solder-bridge short	Bottom	J85	Solder-bridge open	Bottom
J28	Solder-bridge short	Bottom	J86	Solder-bridge open	Bottom
J34	Solder-bridge option	Top	J87	Solder-bridge open	Bottom
J35	Solder-bridge option	Top	J88	Solder-bridge short	Bottom
J36	Solder-bridge option	Top	J89	Solder-bridge open	Bottom
J37	Solder-bridge option	Top	J90	Solder-bridge open	Bottom
J55	Solder-bridge option	Bottom	J91	Solder-bridge short	Bottom
J56	Solder-bridge option	Bottom	J94	Solder-bridge open	Top
J57	Solder-bridge option	Bottom	J95	Solder-bridge open	Top
J58	Solder-bridge option	Top	J96	Solder-bridge short	Top
J60	Solder-bridge option	Bottom	J97	Solder-bridge short	Top
J61	Solder-bridge open	Bottom	J99	Solder-bridge short	Top
J62	Solder-bridge open	Bottom	J115	Solder-bridge open	Bottom
J73	Solder-bridge open	Bottom	J116	Solder-bridge open	Bottom
J74	Solder-bridge open	Bottom			
J75	Solder-bridge open	Bottom			

Table 24: solder-bridge jumper types

Table 25 shows the default function and alternate function of each jumper – if applicable.

Jumper	Default connection	Alternate function
J11	Connect GPIO8 to mikroBUS4 SCL	Can't be connected simultaneously with J73, J16
J14	Connect USB-VBUS to IRIS-W10 VBUS pin	
J15	GPIO 6 to JTAG-TCK	Connect GPIO6 to mikroBUS3-CS
J16	Connect GPIO 8 to mikroBUS3-UART_TX	Can't be connected simultaneously with J73, J11
J17	Connect GPIO 56 to mikroBUS4-UART_TX	Can't be connected simultaneously with J74, J75
J18	Connect GPIO 9 to mikroBUS3-UART_RX	Can't be connected simultaneously with J76, J79
J19	Connect GPIO 57 to mikroBUS4-UART_RX	Can't be connected simultaneously with J77, J78
J25	Connect GPIO 3 to mikroBUS1-UART_TX	Can't be connected simultaneously with J80, J83
J26	Connect GPIO 14 to mikroBUS2-UART_TX	Can't be connected simultaneously with J81, J82
J27	Connect GPIO 2 to mikroBUS1-UART_RX	Can't be connected simultaneously with J84, J85
J28	Connect GPIO 13 to mikroBUS2-UART_RX	Can't be connected simultaneously with J86, J87
J34	Connect GPIO 7 to JTAG-TMS.	Connect mikroBUS3 SPI clock
J35	Connect GPIO 8 to JTAG-TDI	MikroBUS3-UART-TX SPI-MISO I2C
J36	Connect GPIO 9 to JTAG-TDO.	MikroBUS3-UART-RX SPI-MOSI I2C-SDA
J37	Connect GPIO 10 to JTAG-RESET	NC
J55	Connect GPIO 15 to mikroBUS2 SPI-Clk and J47-15	Connect SDIO-CLK
J56	Connect GPIO 16 to U17	Connect SDIO-D3
J57	Connect GPIO 17 to U17	Connect SDIO-CMD
J58	Connect GPIO 19 to U3-RTS	Connect SDIO-D0
J59	Connect GPIO 20 to U3-CTS	Connect SDIO-D1
J60	Connect GPIO 18 to mikroBUS-INT	Connect SDIO-D2
J61	Connect GPIO 27 to mikroBUS3-PWM	Can't be connected simultaneously with J88, J89, J90
J62	Enable mikroBUS3-INT	Can't be connected simultaneously with J91, J92, J93
J73	Connect GPIO 8 to mikroBUS3-SPI_MISO	Can't be connected simultaneously with J11, J16
J74	Connect GPIO 56 to mikroBUS4-SPI_MISO	Can't be connected simultaneously with J17, J75
J75	Connect GPIO 56 to mikroBUS4-I2C_SCL	Can't be connected simultaneously with J17, J74
J76	Connect GPIO 9 to mikroBUS3-SPI_MOSI	Can't be connected simultaneously with J18, J79
J77	Connect GPIO 57 to mikroBUS43-SPI_MOSI	Can't be connected simultaneously with J19, J78
J78	Connect GPIO 57 to mikroBUS4-I2C_SDA	Can't be connected simultaneously with J19, J77
J79	Connect GPIO 9 to mikroBUS3- I2C_SDA	Can't be connected simultaneously with J18, J76
J80	Connect GPIO 3 to mikroBUS1-SPI_MISO	Can't be connected simultaneously with J25, J83
J81	Connect GPIO 14 to mikroBUS2-SPI_MISO	Can't be connected simultaneously with J26, J82
J82	Connect GPIO 14 to mikroBUS2-I2C_SCL	Can't be connected simultaneously with J26, J81
J83	Connect GPIO 3 to mikroBUS1- I2C_SCL	Can't be connected simultaneously with J25, J80
J84	Connect GPIO 2 to mikroBUS1- I2C_SDA	Can't be connected simultaneously with J27, J85
J85	Connect GPIO 2 to mikroBUS1- SPI_MOSI	Can't be connected simultaneously with J27, J84
J86	Connect GPIO 13 to mikroBUS2-SPI-MOSI	Can't be connected simultaneously with J28, J87
J87	Connect GPIO 13 to mikroBUS2-I2C_SDA	Can't be connected simultaneously with J28, J86
J88	Connect GPIO 27 to mikroBUS2-PWM	Can't be connected simultaneously with J61, J89, J90
J89	Connect GPIO 27 to mikroBUS1-PWM	Can't be connected simultaneously with J61, J88, J90
J90	Connect GPIO 27 to mikroBUS4-PWM	Can't be connected simultaneously with J61, J88, J89
J91	Enable mikroBUS2-INT	Can't be connected simultaneously with J62, J92, J93
J92	Enable mikroBUS1-INT	Can't be connected simultaneously with J62, J91, J93
J93	Enable mikroBUS4-INT	Can't be connected simultaneously with J62, J91, J92

Jumper	Default connection	Alternate function
J94	Bypass R35, VDD current sense resistor	Can't be used with J97
J95	Bypass R62, VDD-IO current sense resistor	Can't be used with J96, J115, J116
J96	Connect R62, VDD-IO current sense resistor to +3.3V	Can't be used with J95, J115, J116
J97	Connect R35, VDD current sense resistor	Can't be used with J94
J115	Connect R62, VDD-IO current sense resistor to +1V8	Can't be used with J95, J96, J116
J116	Connect R62, VDD-IO current sense resistor to +1VPA	Can't be used with J95, J96, J115

Table 25: Solder-bridge jumpers showing default connections and alternate functions

3.13 Test points

Table 26 describes the function of each test point on the EVK.

Test point	Function	Test point	Function
TP1	MUSB-OTG-VBUS	TP22	MUSB-OTG-ID
TP2	GND	TP23	ENET_B-CAST-OFF
TP3	Chassis-GND	TP24	GND
TP4	GND	TP25	P1_1-MCULINK
TP5	Chassis-GND	TP26	P1_9-MCULINK
TP6	GND	TP27	P1_20-MCULINK
TP7	GND	TP28	P1_21-MCULINK
TP8	GND	TP29	P1_31-MCULINK
TP9	GND	TP30	P1_7-MCULINK
TP10	GND	TP31	FL_USB1_D_n
TP11	GND	TP32	FL_USB1_D_p
TP12	GND	TP33	GND
TP13	GND	TP101	GND
TP14	GND	TP102	GND
TP15	GND	TP103	GND
TP16	GND	TP104	GND
TP17	MCU-LINK +3.3V	TP141	MOD-USB-N
TP18	GND	TP142	MOD-USB-P
TP19	GND	TP111	USB-N
TP20	GND	TP112	USB-P
TP21	GND		

Table 26: Test point description

Appendix

A Re-loading the Wi-Fi_CLI example

The IRIS-W10 module hosted on the EVK-IRIS-W10 evaluation board is pre-flashed with the Wi-Fi CLI example application. If the application is overwritten or otherwise deleted or corrupted from where it resides in the flash memory, this appendix information describes how to re-load this application.

A.1 Prerequisite

1. Download the J-Link software (V7.98i and above) from SEGGER [6].
2. Using an external debugger connected to **J20** connector, as described in JTAG/SWD debug interfaces, flash the IRIS-W1 module with a pre-compiled Wi-Fi_CLI application.
3. Identify the IRIS-W10 integrated flash memory type as mentioned in NXP MCUXpresso SDK and download relevant files from the u-blox short range open CPU GitHub repository [7].

A.2 Flashing the firmware and application

1. Open J-Flash Lite or JLinkCommander and connect the RW612 device using SWD interface at 4000 kHz speed.

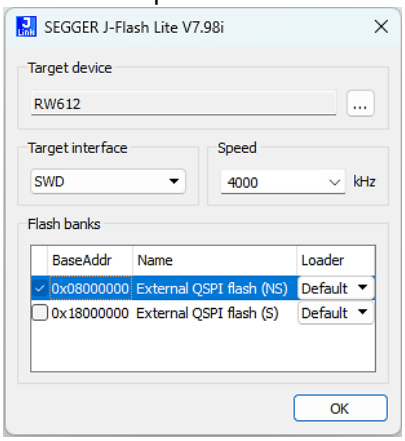


Figure 32: J-Flash Lite device configuration

2. Download Wi-Fi firmware `rw61x_sb_wifi_a2.bin` and wifi_cli application `rw612_wifi_cli_v16_fi8.hex` from the u-blox short range open CPU GitHub repository [7].
3. Flash Wi-Fi firmware `rw61x_sb_wifi_a2.bin` at the address `0x08400000`.

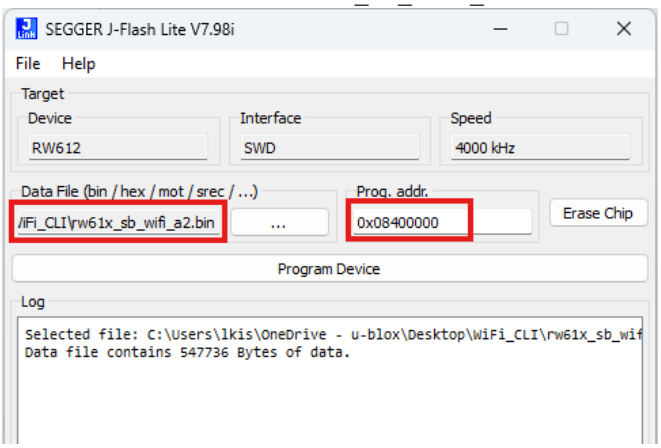


Figure 33: Flashing Wi-Fi firmware

☞ Ensure that the Wi-Fi firmware is flashed before running any Wi-Fi application. To prevent the firmware from being deleted, it must be flashed at least once to the address shown in Figure 33. Otherwise, it is erased.

4. Flash the wifi_cli application `rw612_wifi_cli_v16_fi8.hex`.

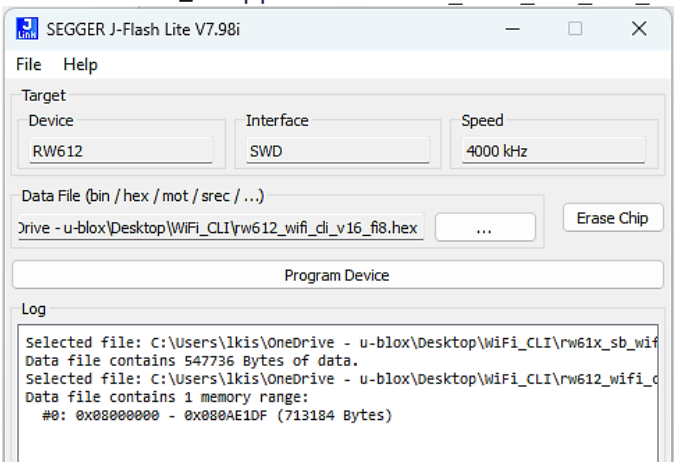


Figure 34: Flashing wifi_CLI application

B Glossary

Abbreviation	Definition
CLK	Clock
CPU	Central Processing Unit
CTS	Clear To Send
DC	Direct Current
DC-DC	DC to DC converter
DFU	Device Firmware Update
EVK	Evaluation Kit
FCB	Flash Configuration Block
FICR	Factory Information Configuration Register
GPIO	General Purpose Input / Output
ISP	In-System Programming
LDO	Low Drop-Out voltage regulator
LE	Low Energy
LED	Light Emitting Diode
LF	Low Frequency
LiPo	Lithium-Polymer battery
NFC	Near-Field Communications
QSPI	Quad Serial Peripheral Interface
RC	Resistor-Capacitor network
RTS	Request To Send
RXD	Receive data signal
SES	SEGGER Embedded Studio
SIG	Special Interest Group
SoC	System on Chip
SPI	Serial Peripheral Interface
TH	Through Hole
TXD	Transmit data signal
UICR	User Information Configuration Register
USB	Universal Serial Bus

Table 27: Explanation of the abbreviations and terms used

Related documentation

[1] IRIS-W10 data sheet, UBX-23002331
[2] IRIS-W10 system integration manual, UBX-23003263
[3] MCU-Link JTAG/SWD Debug Probe | NXP Semiconductors
[4] MCUXpresso Integrated Development Environment (IDE)
[5] MCUXpresso Software Development Kit
[6] SEGGER J-Link software
[7] Open CPU Github repository u-blox-sho-OpenCPU
[8] EVK design Github repository, evk_designs_sho_altium
[9] NXP LPC55S6x Product data sheet
[10] Mikro Click boards
[11] MCU-Link Pro standalone debug probe, User Manual, UM11673
[12] NXP Wi-Fi and Bluetooth Demo Applications for RW61x, User Manual UM11799 (Access req.)
[13] NXP MCU-Link, schematics
[14] Blhost, User's Guide
[15] Blhost, download

For product change notifications and regular updates of u-blox documentation, register on our website, www.u-blox.com.

Revision history

Revision	Date	Name	Comments
R01	04-Sep-2023	habd	Initial release for EVK-IRIS-W10 PT1
R02	24-Nov-2023	habd	Updated for changes in new prototype spin of the board. Revised ch2 in Setting up the evaluation board. Added Starting up the EV section.
R03	08-Nov-2024	lkis	Included changes to module and EVK key features. Revised "Setting up the evaluation board" chapter with two new sections, Starting up the EVB and Software Development, and other changes. Removed "Hello world" example with "Wi-Fi example application". Revised "Hardware description" chapter, including the addition of function and header descriptions in Figure 3. Included minor updates in the Hardware description chapter. Added appendix describing the Re-loading the Wi-Fi_CLI example. Added Flash and debug custom applications.

Contact

u-blox AG

Address: Zürcherstrasse 68
8800 Thalwil
Switzerland

For further support and contact information, visit us at www.u-blox.com/support.