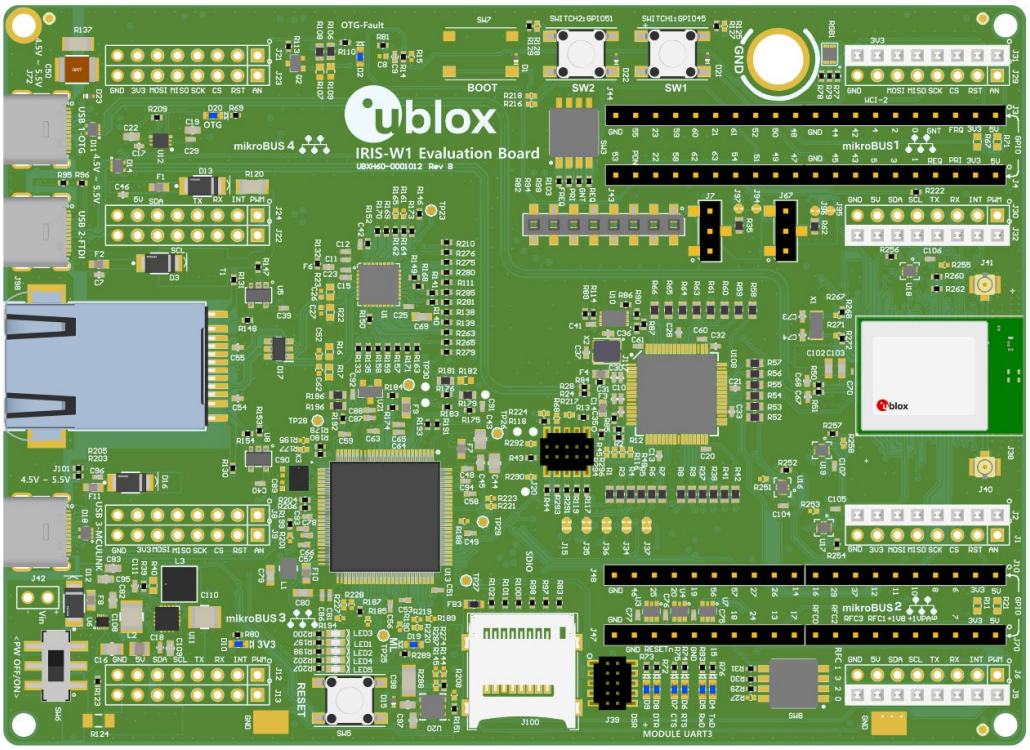
EVK-IRIS-W10

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.

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	9-Jan-2024
Disclosure restriction	C1-Public	

This document applies to the following products:

Product name
EVK-IRIS-W101
EVK-IRIS-W106

u-blox or third parties may hold intellectual property rights in the products, names, logos, and designs included in this document. Copying, reproduction, or modification of this document or any part thereof is only permitted with the express written permission of u-blox. Disclosure to third parties is permitted for clearly public documents only.
The information contained herein is provided “as is” and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability, and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.
Copyright © u-blox AG

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 the 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 [9].

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
- SDIO 3 card slot

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.

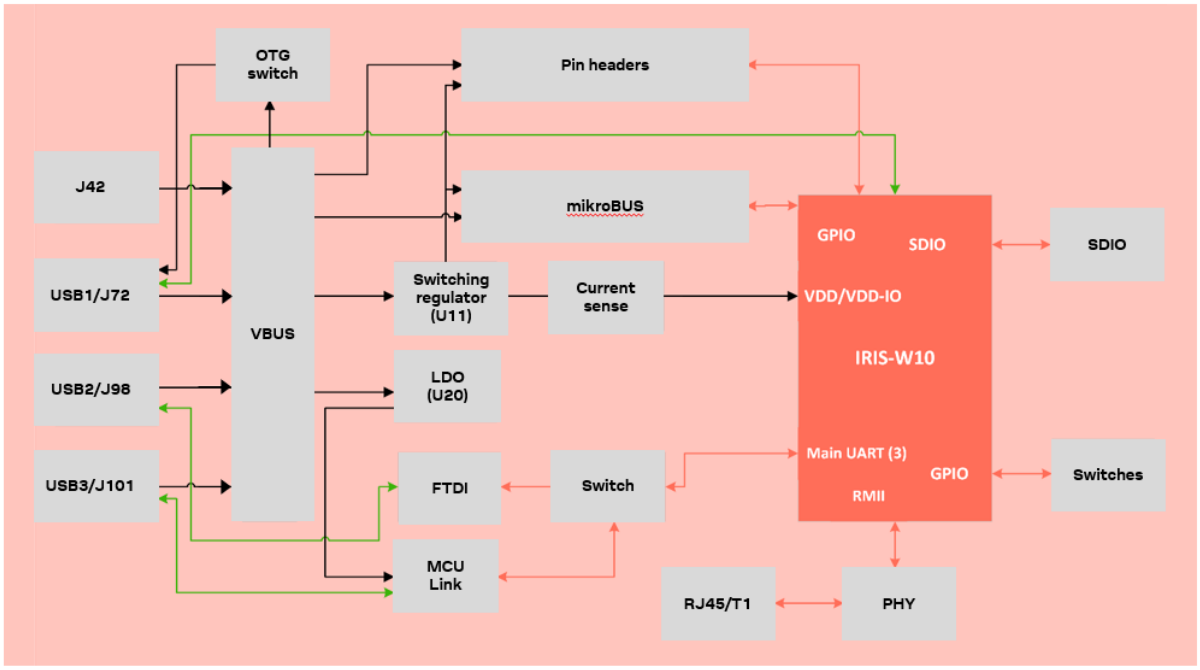


Figure 1: EVK-IRIS-W10 block diagram

[1] Not all peripherals available simultaneously

2 Setting up the evaluation board

Connect the external power supply to the EVK as described in powering the board section.

- The green status LED **(D10)** is lit when the internal EVK **3V3** supply is active.
- The green status LED **(D19)** is lit when **3V3** supply to the MCU Link chip is active.

⚠ Approvals for IRIS-W1 are currently pending. For further information about the current approval status, see the Qualifications and approvals section in the data sheet [1] .

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

The operating system installs the correct COM port drivers automatically. The drivers need to be installed only when you connect the unit to a new computer for the first time. For more information about the COM ports and their configuration, see the FTDI FT231XQ-R Datasheet [7] or LPC55S69JBD100K Datasheet [8].

Windows OS automatically assigns one COM port to the unit. To view the assigned COM ports on Windows 10:

1. Open the Control Panel and select Hardware and Sound.
2. Click **Device Manager** in **Devices and Printers**. This opens the Device Manager window where you can view the assigned COM ports.

IRIS-W10 open CPU module variants are used to develop custom software based on the NXP MCUXpresso SDK, 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. For information about for the working environment setup and regulatory restrictions, see the respective “Open CPU software” and “Qualification and approvals” sections of the IRIS-W10 system integration guide [2].

2.1 Starting up the EVB

⚠ Before powering up the EVK-IRIS-W101, be sure to connect an antenna (or any 50 Ω RF load) to the U.FL antenna connector (**J41**). Failing to do so can cause module malfunction.

2.1.1 Flash tool installation

To get the EVK ready with necessary flash tools and settings:

1. Attach the USB-C cable to the USB3 MCU-Link port on the EVB and connect to PC to power the module.
2. Set **SW3** to default configuration, as described in Automatic bootloader / strap-in
3. Download and install the latest version of the J-Link software (V7.92g onwards) from SEGGER [5]
4. Go to the NXP website [3]. Download and install the latest MCU-Link software on the PC.
5. Flash the J-Link software on the LPC chip (MCU-Link chip)
   [`MCU-LINK_INSTALLER_WIN_3.1xx`] by running the script:
   "`NXP\MCU-LINK_installer_3.1xx\scripts\program_JLINK.cmd`".
   After the successful flashing of J-Link firmware:
   - **LED1** turns red and blinks slowly.
   - Enumerated COM port "JLink CDC UART Port (COMxx)" is shown in Device Manager.

   This process is necessary only during the initial setup of a newly acquired EVK.
6. Power cycle the module/EVK after flashing the software onto the LPC chip.
7. Pull the **RFC2** pin to low on the **SW8** to enable the SWD interface of the module.

8. Replace the "`flash_config.c`" in the `SDK\boards\rdrw612bga\flash_config` folder with the updated version, which includes an updated FCB (Flash configuration block) and supports low-density Macronix 8MB QSPI Flash. The updated `flash_config.c` file is available in the u-blox short range open CPU GitHub repository [6].

2.1.2 Wi-Fi example application

The wifi_cli application is provided in binary format and can be used to quickly demonstrated Wi-Fi features without the need of installing SDK or compiling any firmware.

Example of supported features:

- Wi-Fi Scan
  - Wi-Fi Soft AP mode
  - Wi-Fi Station mode
- Throughput performance using `iPerf` measurement tool

2.1.2.1 Flash the wifi_cli example firmware

IRIS-W10 module stores the RW612 application and Wi-Fi firmware binary in different partitions of internal FlexSPI NOR flash. The application reads Wi-Fi firmware during initialization and downloads it to RW612 internal Wi-Fi MCU to run.

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

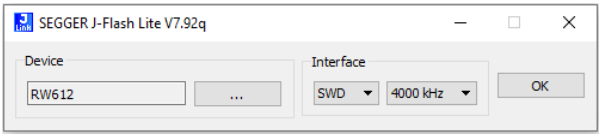


Figure 2: J-Flash Lite device config

2. Download Wi-Fi firmware `rw610_sb_wifi_v1.bin` and wifi_cli application `rw612_wifi_cli.hex` from the u-blox short range open CPU GitHub repository [6].
3. Flash Wi-Fi firmware `rw610_sb_wifi_v1.bin` at the address `0x08400000`.

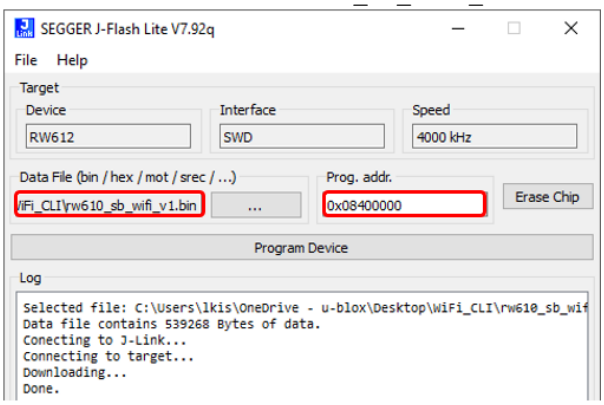


Figure 3: Flashing Wi-Fi firmware

Wi-Fi firmware must be flashed once unless it is erased. It must be flashed to the address shown in Figure 3. Ensure that the Wi-Fi firmware is flashed before running any Wi-Fi application.

4. Flash the wifi_cli application file `rw612_wifi_cli.hex`.

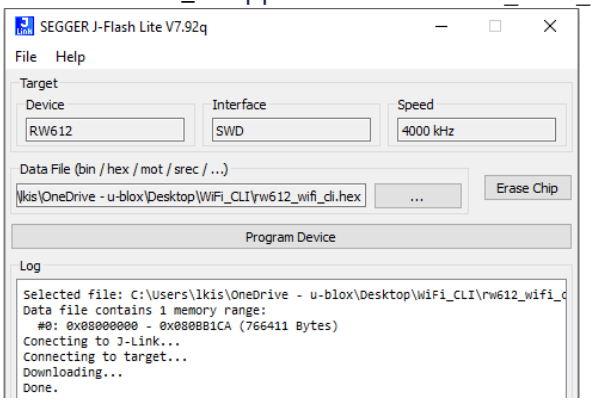


Figure 4: Flashing wifi_Cli application

5. Reset the module and open a UART console (using Putty, TeraTerm, or another terminal emulator), and set the serial port to 115200 baud rate and run different feature sets in the wifi_cli application.

Figure 5: wifi_cli demo

2.1.2.2 Run iperf test using the wifi_cli example

This section assumes that the iperf test is run using two EVK-IRIS-W10 – with Device A configured as an Access Point and Device B configured as a Station. However, 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 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 12345678

wlan-connect test1

iperf -c 192.168.10.1
```

2.2 Software Development

2.2.1 NXP SDK

To use the EVK-IRIS-W10 together with NXP SDK:

- Create your own board file
- Adapt the examples in the NXP SDK to use this board file

For more information about how to retrieve the SDK and perform these tasks, see the "Software" section of the IRIS-W10 system integration manual [2]. See also the u-blox short range open CPU GitHub repository [6].

3 Hardware description

Design files for the EVK-IRIS-W10 PCB are available from your local u-blox support team.

Figure 6 shows the major functions provided by EVK-IRIS-W10.

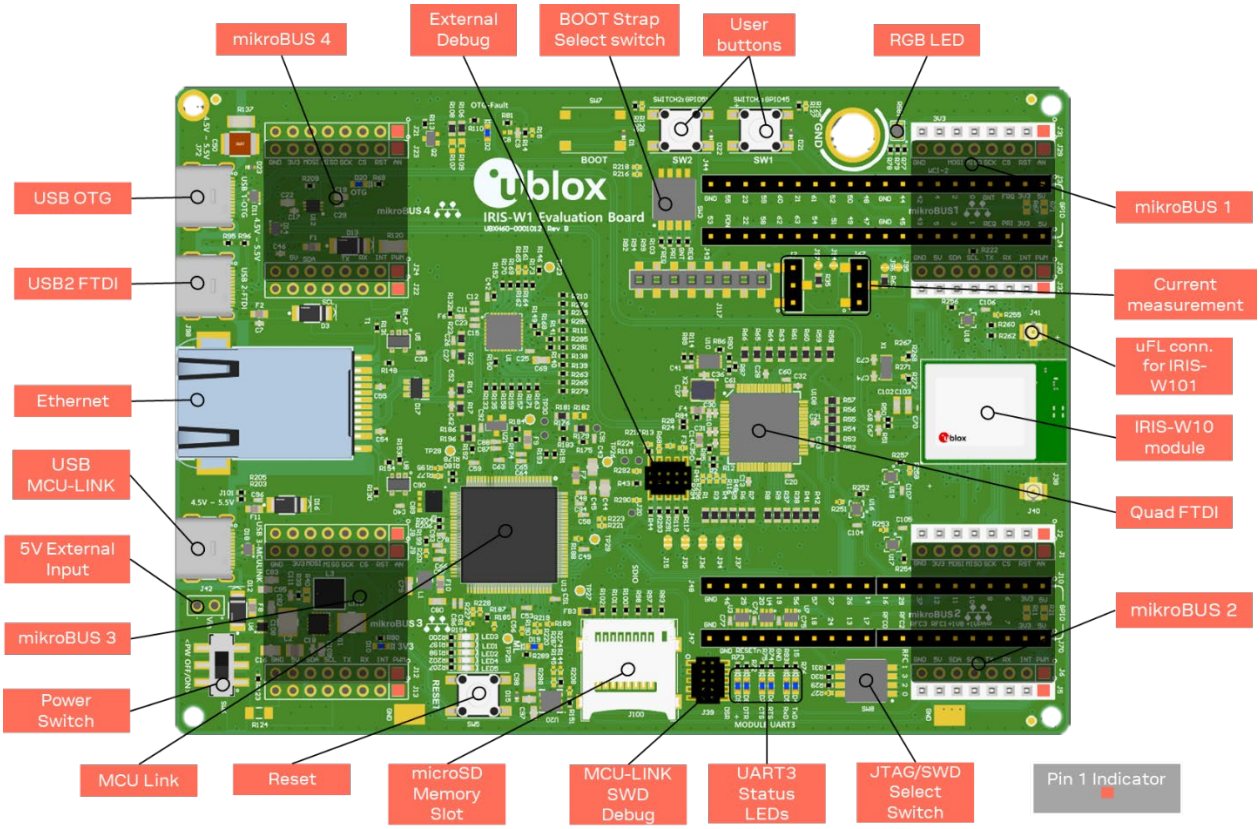


Figure 6: Header and major function locations

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

The 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.

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

3.1.1 Powering the board

After applying power to one of the sources described in Table 1, slide **SW6** shown in Figure 6 to the ON position to power-on the EVK.

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, either:

- Populate the 2.4 A power switch **U6** (MP5075GTF) and disconnect **R124,** as shown in Figure 7.
- Limit the maximum current to 600 mA by populating **R124** with a 1206 0R 2A resistor and disconnecting U6.

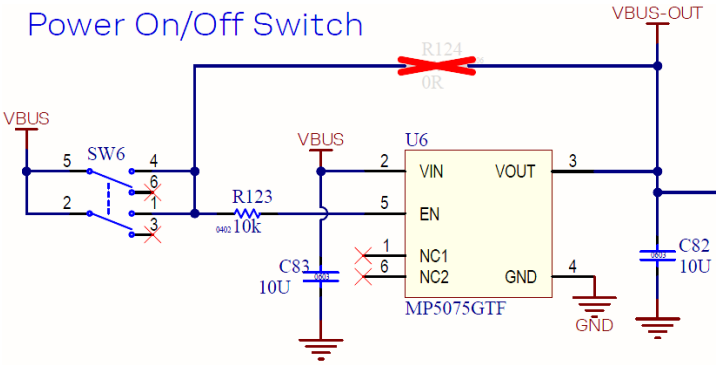


Figure 7: EVK-IRIS-W10 maximum current options

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 8. LED **D10** (green) indicates the presence of the 3.3 V output from the regulator **U11**.

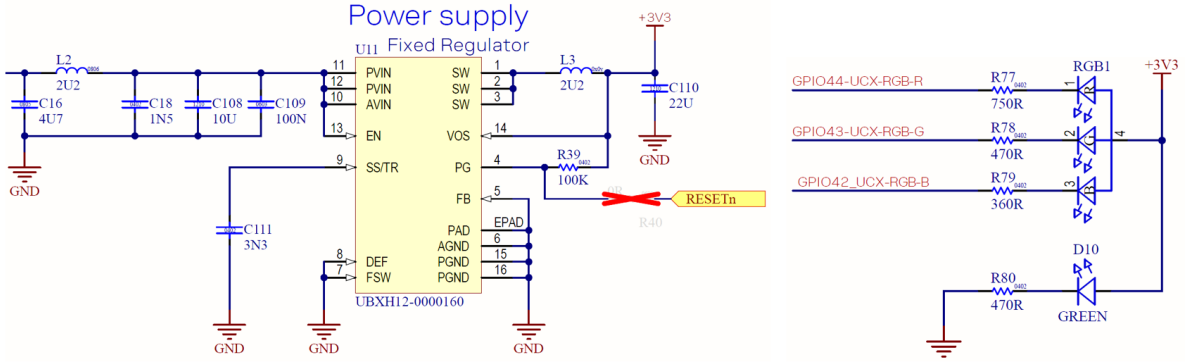


Figure 8: Switching regulator (U11)

3.1.1.2 3.3 V Linear regulator (U20)

In Figure 9, **U20** (NCP692MN33T2G is intended to supply power to the **MCU-LINK** chip with minimum interference to the IRIS-W10 radio module.

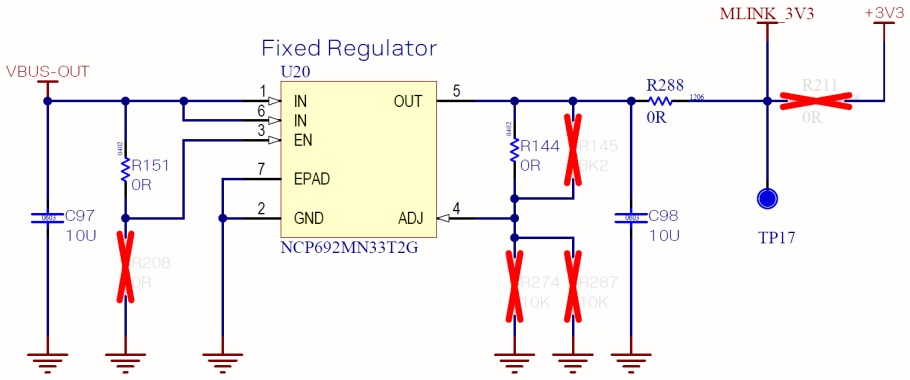


Figure 9: 3.3V linear regulator (U20)

Resistors **R151, R144**, and **R288** must be populated with **0603 0R,** and, in this case, **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 10. It is set to the OTG-device by default. Connect jumper **J14** to allow IRIS-W10 sense the presence of the MUSB-OTG-VBUS.

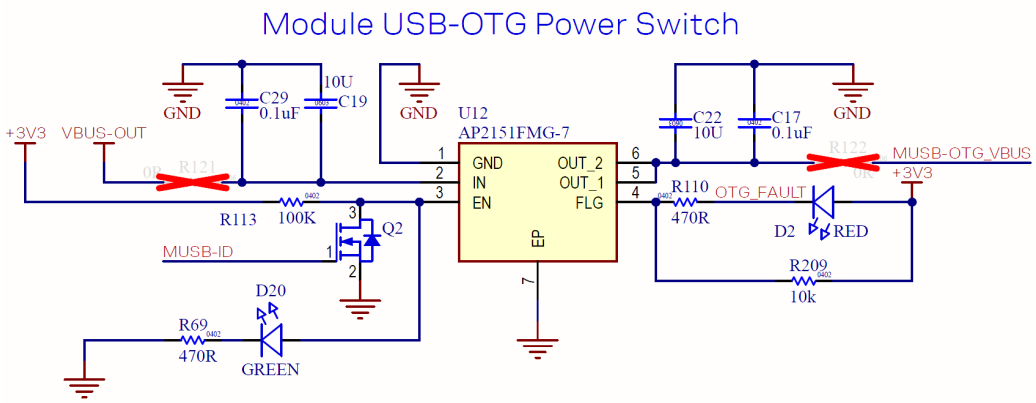


Figure 10: USB On-The-GO switch

Note that:

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

Maximum current capacity of the USB-OTG is 500 mA. LED **D2** (Red) indicates if 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 11.

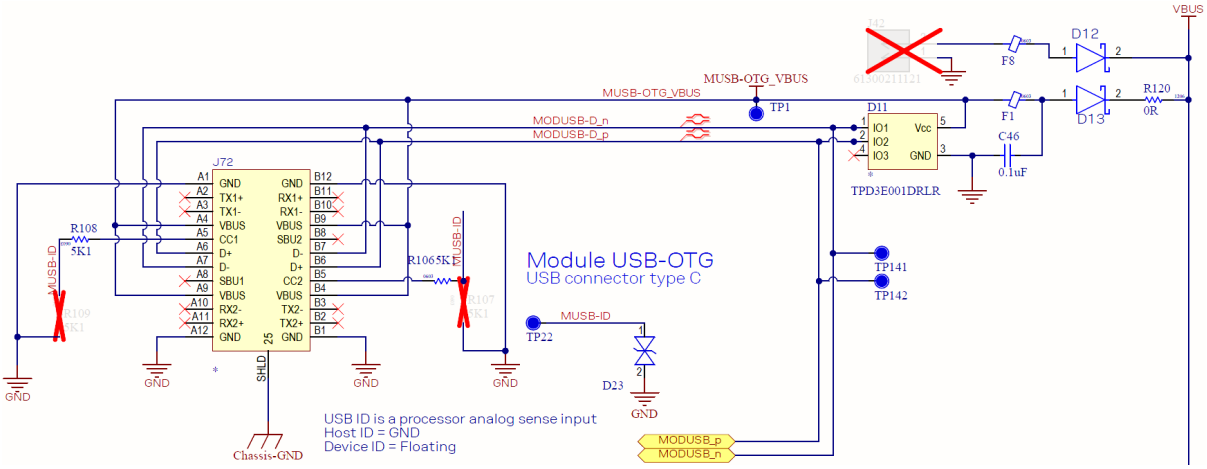


Figure 11: J72 schematic

3.1.1.4 VBUS-OUT (5V) 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. Otherwise, they can only supply a combined 500mA current and 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 12 and Figure 13**.**

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 simultaneously.

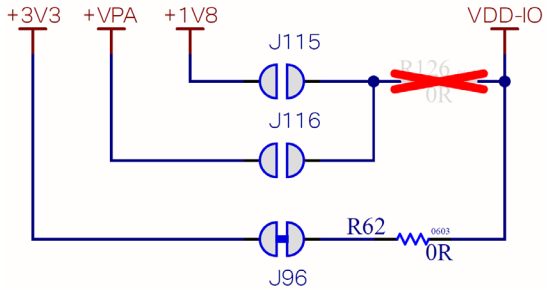


Figure 12: VDD-IO current jumpers

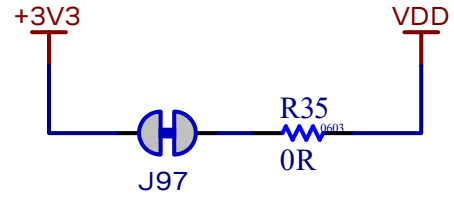


Figure 13: VDD current jumper

3.1.3 Current measurement

The evaluation board provides two current-sensing headers, as shown in Figure 14.

- **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

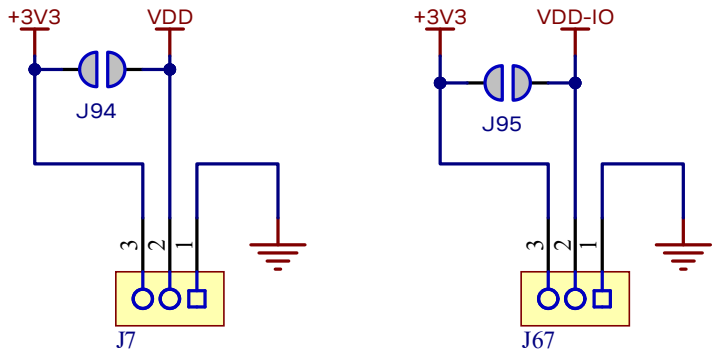


Figure 14: VDD and VDD-IO current sensing configuration

Two consecutive pins on Each 2.54 mm pitch 3-pin headers are connected across a **0R**, which can be replaced by a current-sense resistor (> **1R0** is recommended), **J97,** and **J96** respectively, as shown in Figure 12 and Figure 13. 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 12 and Figure 13. 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 the 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** is 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 for the 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 15.

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

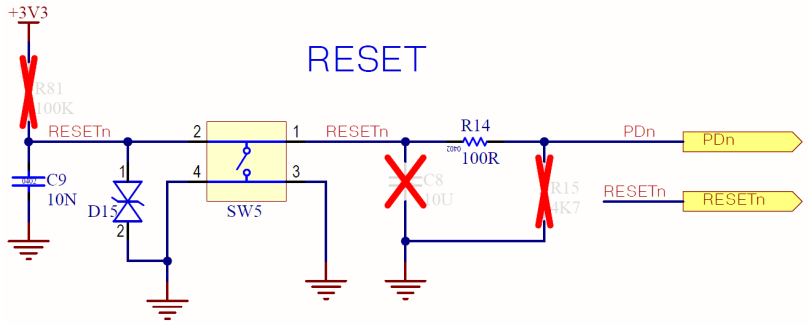


Figure 15: EVK schematic, RESETn button

3.2.1 Automatic bootloader / Bootstrap

Figure 16 shows bootloader configuration and sources available to user. Several signals and DIP switch combinations used to configure the EVK so it can boot in several scenarios.

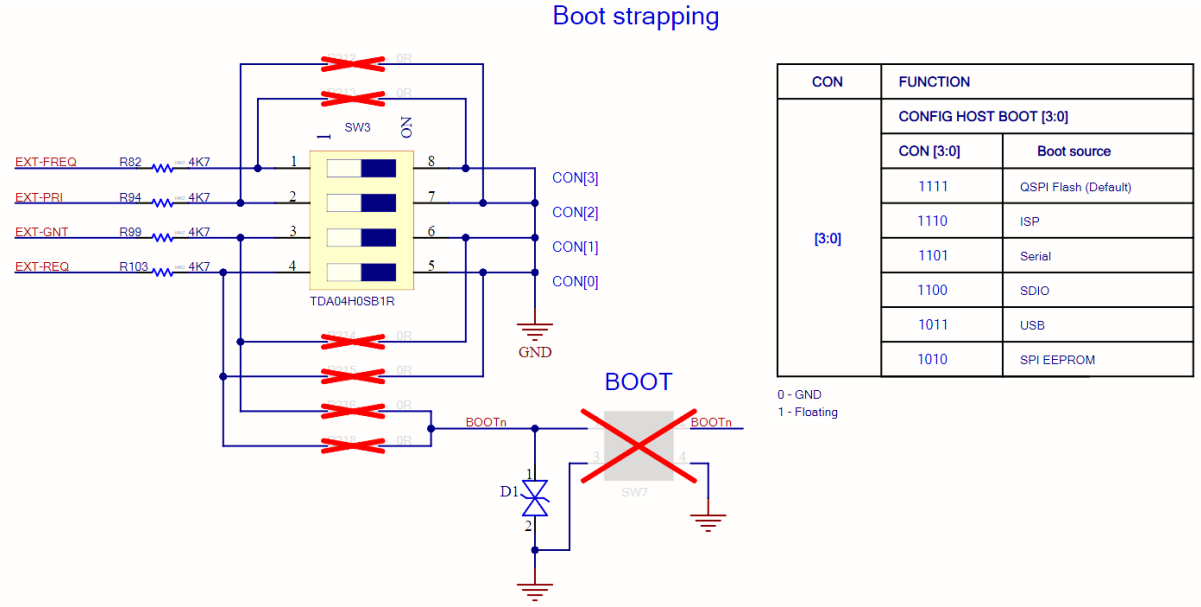


Figure 16: Bootloader schematic

EVK has two main methods to bootstrap IRIS-W10 module:

1. Default method - Using DIP switch **SW3** with resistors **R82, R94, R99** and **R103**
   - Four active-high signals (**EXT-FREQ**, **EXT-PRI**, **EXT-GNT**, **EXT-REQ**) are used to select boot source. They correspond to CON3, CON2, CON1 and CON0 respectively in the table shown in Figure 16.
   - For example, to select ISP as boot source, GND the signal **EXT-REQ** on SW3. Alternatively, other configurations are possible as described in the table data shown in Figure 16.
2. Alternate method – Combination of button **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

☞ By holding **EXT-REQ** signal to **GND,** `blhost.exe` can be used to program firmware in ISP mode. With this firmware, it is possible to perform full flash recovery if IRIS-W10 module ended in a faulty state while flashing or running an application. See also the blhost user guide [11].

3.3 Buttons

In addition to the **RESET** and **BOOT** 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 17 and Table 3.

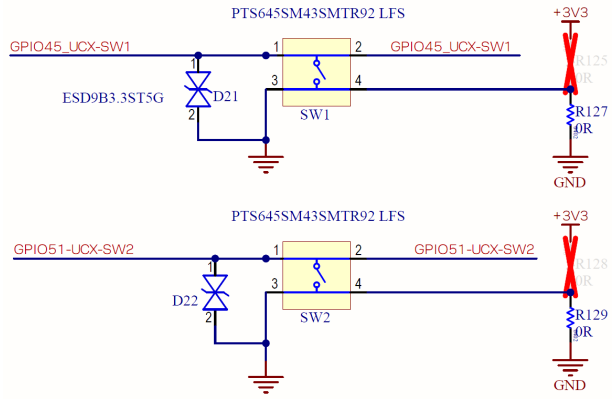


Figure 17: EVK schematic - user buttons

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 19.
- 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 18.

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

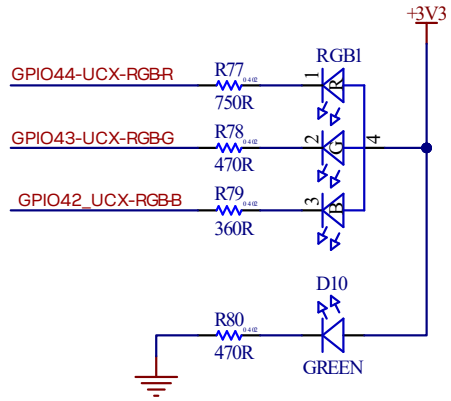


Figure 18: Schematic – RGB and power LED

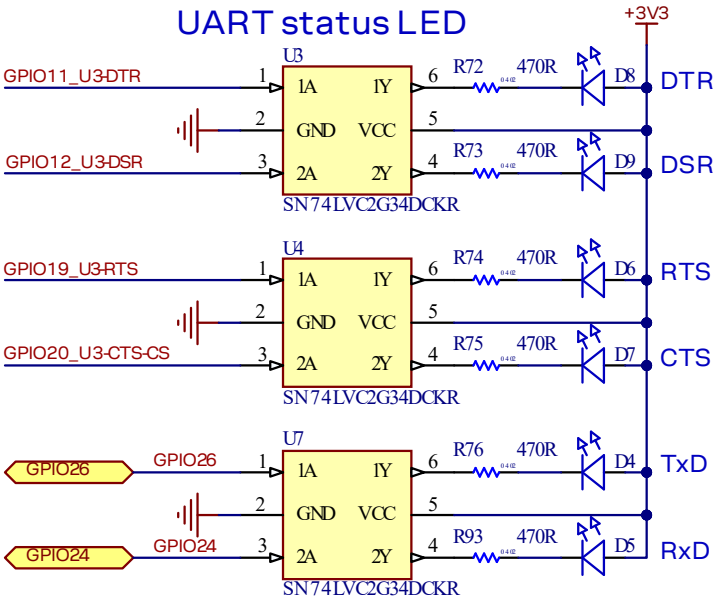


Figure 19: Schematic – UART3 status LED

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 a that seamlessly integrates with the MCUXpresso Integrated Developer Environment (IDE). It is also compatible with third-party IDEs that support 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. MCU-Link is based on the LPC55S69 microcontroller (**U13**) and features a high-speed USB interface for a high-performance debugging.

The main **UART** (**UART 3 TX, RX**) and the **SWD** interface signals of the IRIS-W10 module are connected to the MCU-Link through switches **U16** or (**R134, R136**), **U19,** or (**R142, R143**) 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 20 and further described in Table 6.

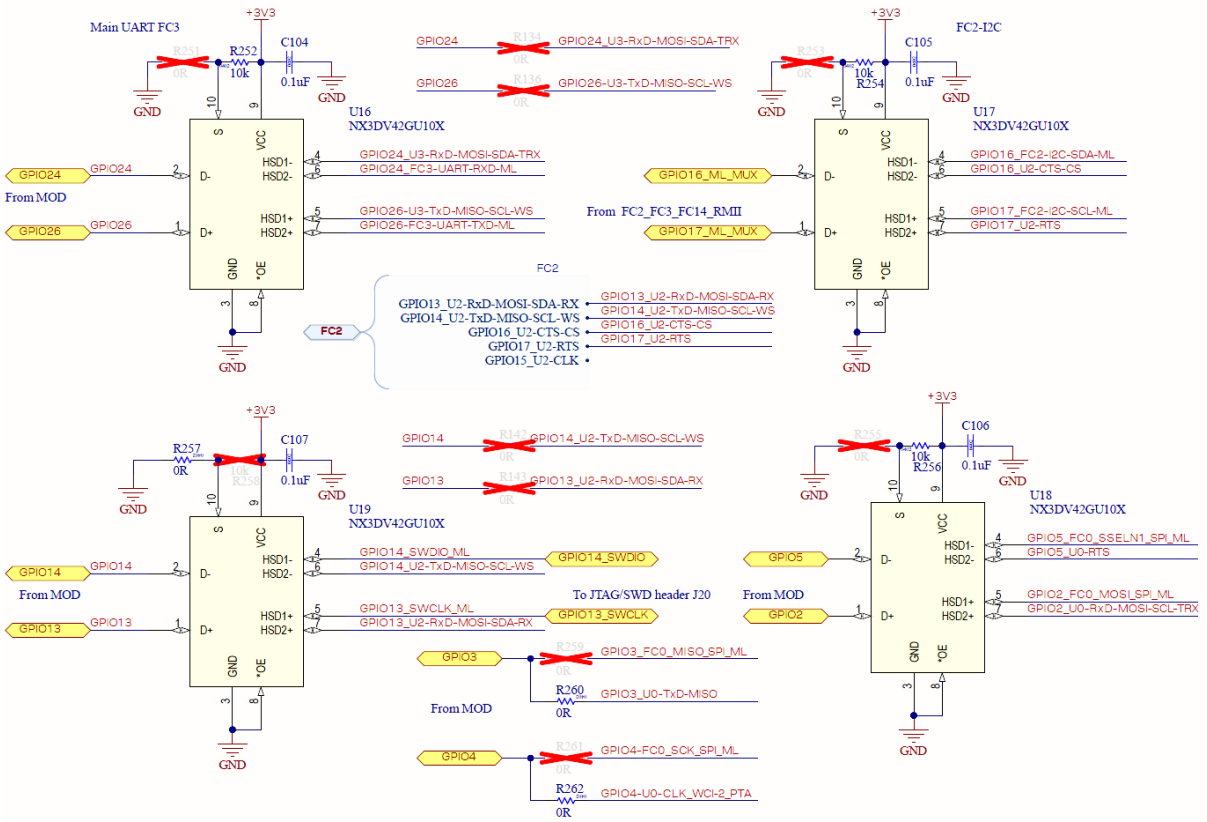


Figure 20: MCU-LINK connections

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

To use the MCU-Link as a **SWD** debugging probe for an external MCU through **J20**, populate **R13**, **R43, R68, R217, R224, R290, R292, R294, R296** with 0 Ω resitors and disconnect **resistors R44, R45, R48, R116, R117, R119, R291, R293**.

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

The SWD programming interface on the MCU-Link connects to the JTAG needle connector (**J33**) and SWD connector (**J39),** as shown in Figure 21.

R182 forces the chip to ISP mode when populated with a 0402 0R resistor.

LED1 to **LED5** indicate the connection and interfaces status of the 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 9.

Optionally, **R211** can be populated with a **1206 0R** resistor to power the MCU-Link chip (**U13**) from the **3V3** output of the DC-DC converter (**U11),** as shown in Figure 9.

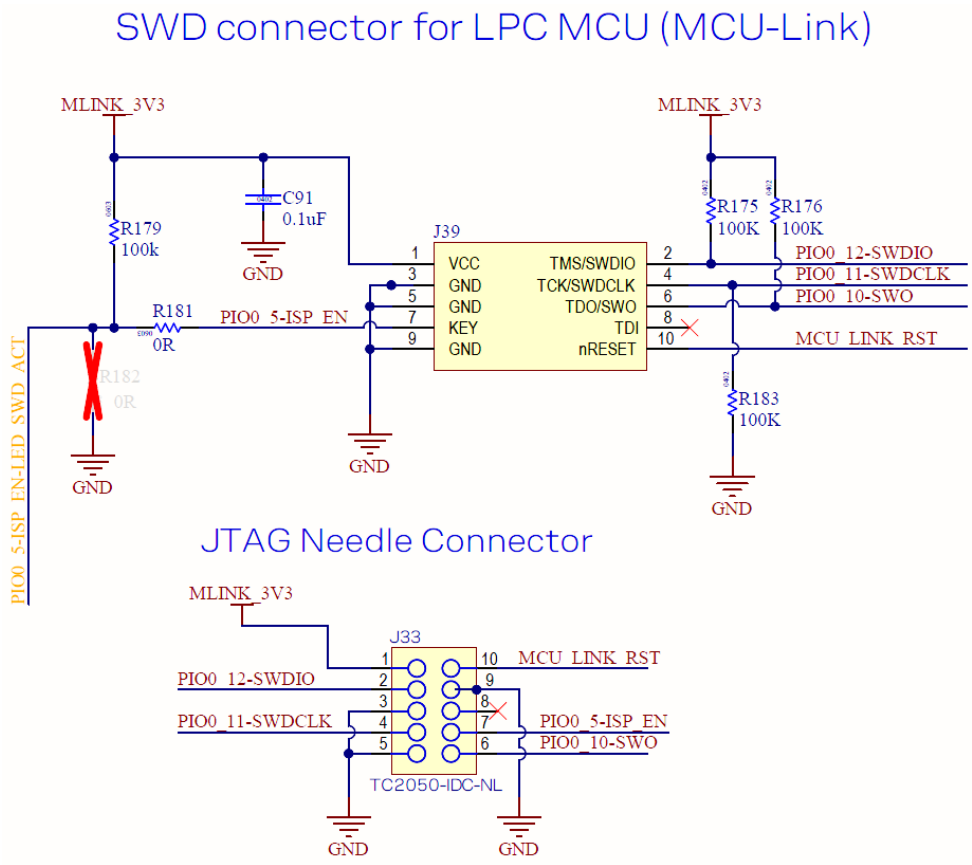


Figure 21: MCU-Link flashing options

Figure 22 shows the hardware version setting resistor group with related connections that are described in Table 7. For more information about MCU-Link hardware and firmware, see also the MCU-Link Pro documentation [10].

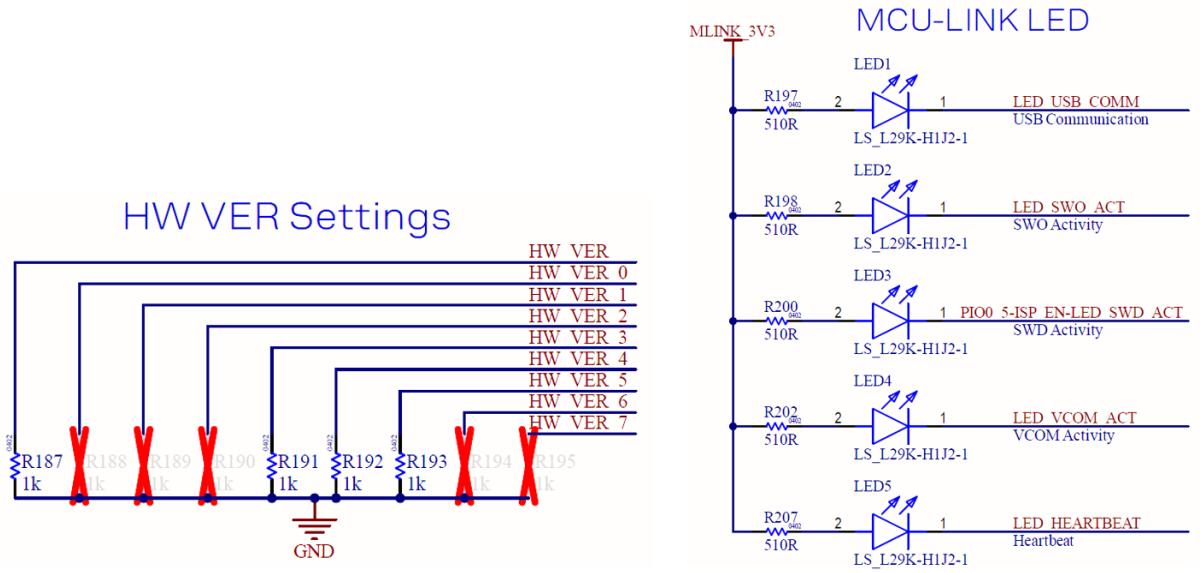


Figure 22: MCU-LINK hardware version and LEDs

MCU-Link pin	Hardware version	Function	Default status	Resistor
55	HW_VER	-	Low	R187
85	HW_VER_0	USB power negotiation when low	Float	R188
73	HW_VER_1	1-bit board id code	Float	R189
65	HW_VER_2	Disable USB-SIO bridge when low	Float	R190
1	HW_VER_3	OB(0) or Pro (1) id	Low	R191
31	HW_VER_4	Board id code is valid when low	Low	R192
5	HW_VER_5	Power measurement enabled when low	Low	R193
56	HW_VER_6	VCOM disabled when low	Float	R194
27	HW_VER_7	SWD debug disabled when low	Float	R195

Table 7: MCU-LINK hardware version functions

3.5.2 USB-to-UART FTDI

EVK has two USB-to-UART FTDI chips to connect with IRIS-W10 module over serial communication.

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

Table 8 shows the inter connections between IRIS-W10 module, **U108**, and **U109**, where the main COM port3 on FTDI chip is connected to UART3 of IRIS-W10 module through 1 kΩ resistors.

To enable this option of communication, module main **UART (UART 3)** should be switched to be connected to the **FTDI** chip through USB switch **U16** by disconnecting **R252, R134, R136** and populate **R251** with **0R,** or in case of **U16** is not populated, then populating **R134, R136** with **0R.**

IRIS-W10 pin name	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 8: Main COM port connections

UART3 clock signal from IRIS-W10 pin **F3, GPIO 25** is multiplexed by RMII clock signal. **R282/R279** are intended to demultiplex the two signals **R279** is populated by default and connects to RMII clock, by disconnecting **R279** and populating **R282** with **0R, GPIO25** will be connected to pin **8** on header **J48.**

3.6 JTAG/SWD debug interfaces

There are two interfaces for debugging IRIS-W10 either **JTAG** or **SWD**. When powering up or resetting the EVK-IRIS-W10, the status of the **RF-Control** strapping pins defines the active interface as described in Table 9.

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

Table 9: RF-CTRL strap-in (F=Float)

SW8 is used to set the RF-CNTL signals to the desired status as shown in Figure 23. An external target hardware can be attached to the **J20** header connector for firmware programming and debug. **J20** is implemented with a 2x5 header with 1.27 mm pitch.

The IRIS-W10 module **SWD** interface, **GPIO13-SWDCLK** and **GPIO14-SWDIO**, are connected to the MCU-LINK interface and **J45** needles connector simultaneously by default through **0R** resistors **R2 and R12** those **GPIOs** could be also connected to **J20** by disconnecting the last two resistors and populate resistors **R48, R116** with **0 R.** See connections in Figure 24.

GPIOs **13** and **14** are also accessible through pin headers **J47 (**pin 2) and **J48 (**pin 1)**,** respectively. By toggling the USB switch **U19** by disconnecting **R257, R142, R143** and populating **R258** with **0R** (or in case of **U19** is not populated), previous pins can be accessed from the pin headers by populating **R142, R143** with **0R.** Note that **U19** and **R142, R143** should not be populated simultaneously**.**

The **JTAG** interface of the **IRIS-W10** module is connected to **J20 and FTDI** chip through resistors, and pin headers simultaneously by default, as shown in Table 10.

Jumpers grouped in Table 10 and Figure 26 should be set to their default connections, as shown in Figure 25.

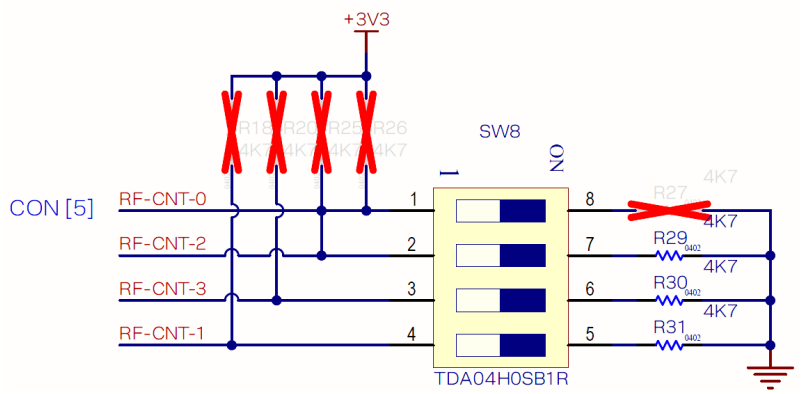


Figure 23: RF-CNTRL strap-in SW8

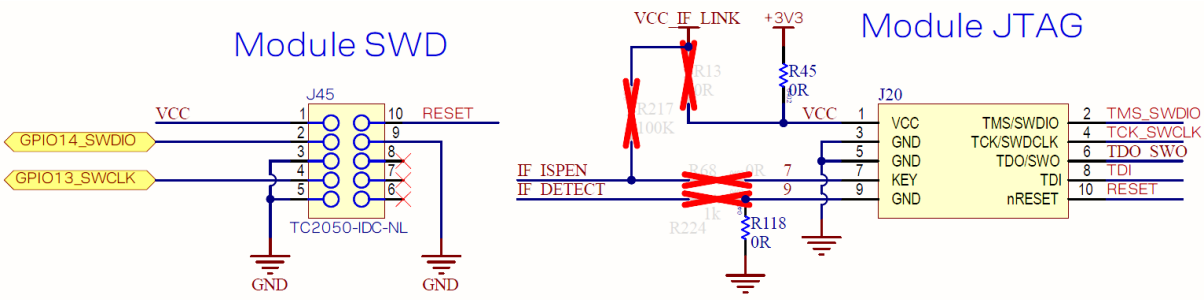


Figure 24: IRIS module SWD/JTAG connection

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

Table 10: JTAG connections

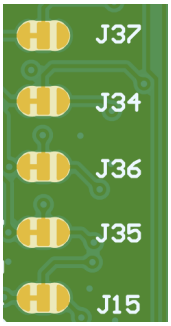


Figure 25: jumpers default connections

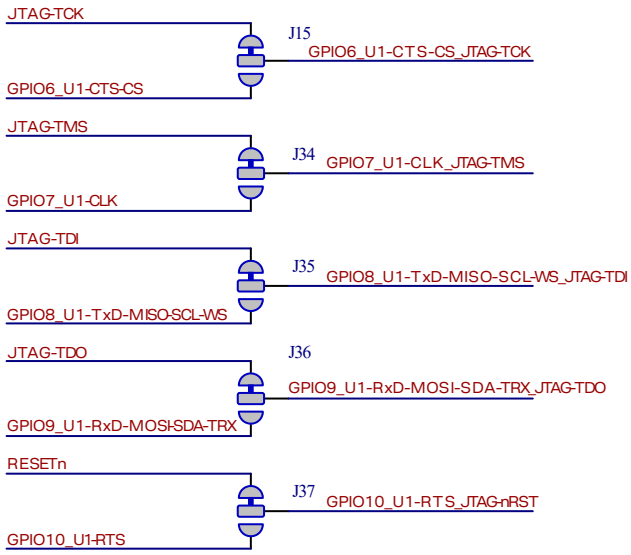


Figure 26: External JTAG debug interface demultiplexing

3.7 32.768 kHz low frequency clock

The evaluation board has a **32.768 kHz** crystal connected to the IRIS-W10 module that allows use of the external crystal oscillator option to source the RTC clock which is **not** enabled by default.

GPIO21 and **GPIO23** multiplex some of the RMII signals and the external crystal oscillator option. Figure 27 shows how to demultiplex those GPIOs.

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

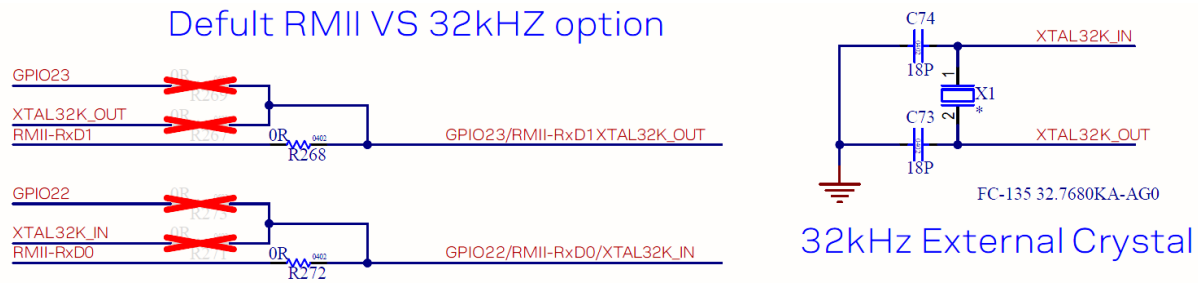


Figure 27: Schematic – 32 kHz crystal

3.8 mikroBUS slots

EVK-IRIS-W10 has four mikroBUS standard compatible slots with multiple choices of **RESET** signal for each slot. All slots are not enabled to use by default, and that is mainly due to the multiplexing of the Flexcomm interfaces pins of the NXP RW612 chip as shown in Table 11.

When utilizing the mikroBUS slots, consider:

- mikroBUS1 cannot be used as SPI interface simultaneously with **WCI-2** function or FTDI-SPI, which shares **GPIO 0, GPIO 1, GPIO 2, GPIO 3 and GPIO 4**. However, mikroBUS1 can still be used a UART or I2C interface.
- mikroBUS2 cannot be used as SPI interface simultaneously with **SDIO** function, which shares **GPIO 15 and GPIO 16.** However, mikroBUS2 can still be used as a UART or I2C interface.
- mikroBUS4 cannot be used totally with **RMII** interface, which shares **GPIO 56 and GPIO 57.**
- mikroBUS3 cannot be used totally with **JTAG** interface, which shares all GPIOs on this slot.
- There is a dedicated GPIO for the analog pin in each mikroBUS slot, as shown in Table 12.
- All four mikroBUS slots share **GPIO 27** as the **PWM** pin, so this GPIO can only be used in a single slot. **GPIO 27** is enabled on slot 2 by default, as shown in Table 12 and Table 13.
- All four mikroBUS slots share **GPIO 18** as **INT** pin, so this can be used in a single slot – but not simultaneously with **SDIO. GPIO 18** is enabled on slot 2 by default, as shown in Table 12 and Table 13.
- All four mikroBUS slots share the **RESET** signal options – either to the global **RESETn** signal of the EVB (active low) or to **GPIO 50**, which is intended as user-defined RESET. This is mainly because the Click boards have different reset options, as shown in Table 14.
- The inner row of each slot is easily accessible – even with the use of add-on Click boards.
- The top-position slots are for extended use or include antenna add-on Click boards, as shown in Table 11.

		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 11: mikroBUS slots pin configuration

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

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

Table 13: mikroBUS PWM/INT jumpers, (*) default connection

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

Table 14: mikroBUS RESET configuration resistors, (*) default connection

3.8.1 mikroBUS 1 slot

This slot is connected to the Flexcomm interface **0**, cannot be used as **SPI** standard simultaneously with FTDI-SPI function, which shares **GPIO4** as shown in Figure 28. However, mikroBUS1 can still be used as UART or I2C standards with 2WCI-2 function.

To enable this slot:

- The USB switch **U18** (default connection) disconnect **R255** and populate **R256** with **0R.**
- Disconnect R8, R9, R38, R37, R259, R261 and populate R260, R262 with 0R.
- Check jumpers **J25, J27, J80, J83, J84, J85, J89, J92** and resistors **R67, R71** to match the required operation, as described in Table 12, Table 13, Table 14 and Table 15.

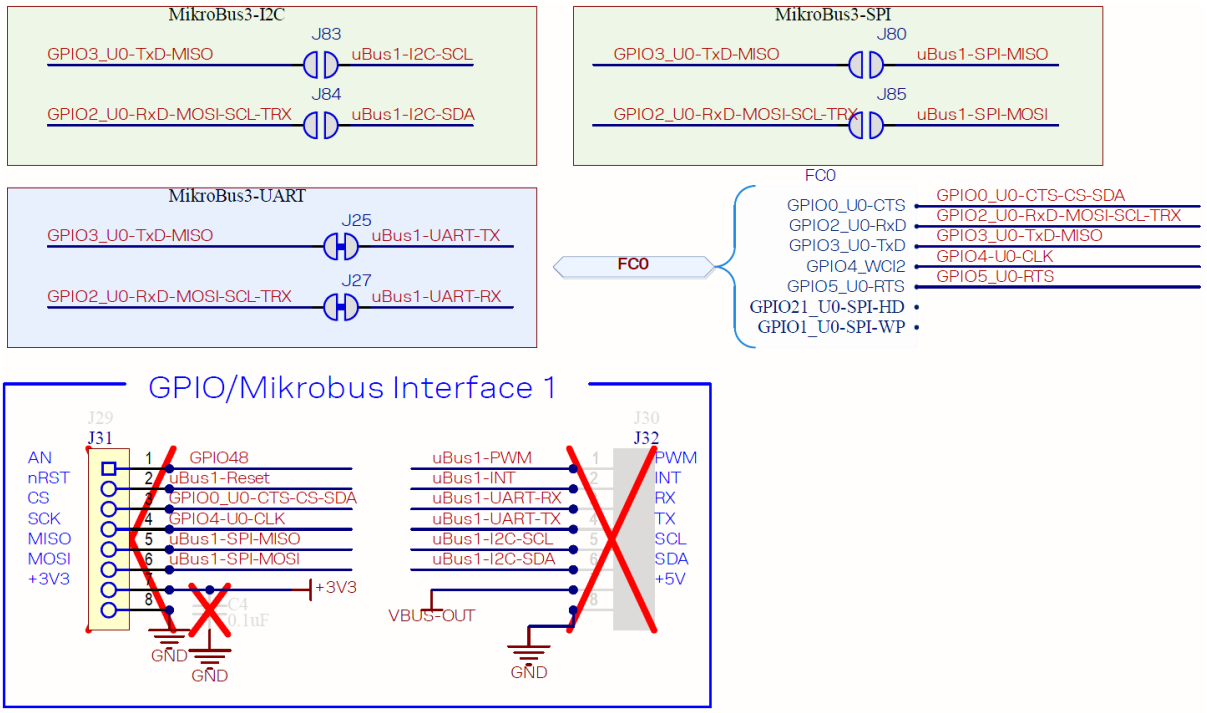


Figure 28: mikroBUS 1 schematic

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

Table 15: mikroBUS 1 working standard options, (*) default connection

3.8.2 mikroBUS 2 slot

This slot is connected to the Flexcomm interface 2, cannot be used either as SPI standard simultaneously with SDIO function, which shares **GPIO 15** and **GPIO 16**. or as UART with SWD interface enabled (**U19** default connection, disconnect **R257** and populate **R258** with **0R**) as shown in Figure 30.

To **enable** this slot:

- In case of USB switch **U19** is populated, toggle the default connection: Disconnect **R142, R142, R258,** and populate **R257** with **0R.** Otherwise populate **R142, R143** with **0R**.
- Jumpers **J55** and **J56** are set to their default connection, as shown in Figure 29.
- Check that jumpers, **J26, J28, J81, J82, J86, J87, J88, J91,** and resistors **R11, R21,** match the required operation, as described in Table 12, Table 13, Table 14, and Table 16.

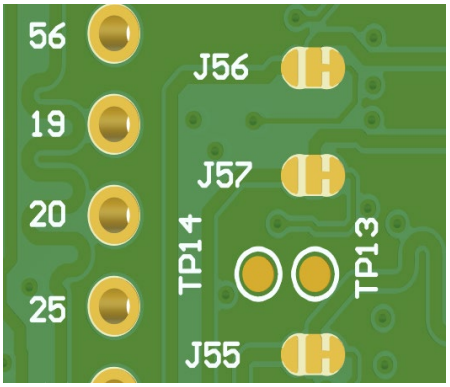


Figure 29: mikroBUS 2 jumpers J55, J56 default connection

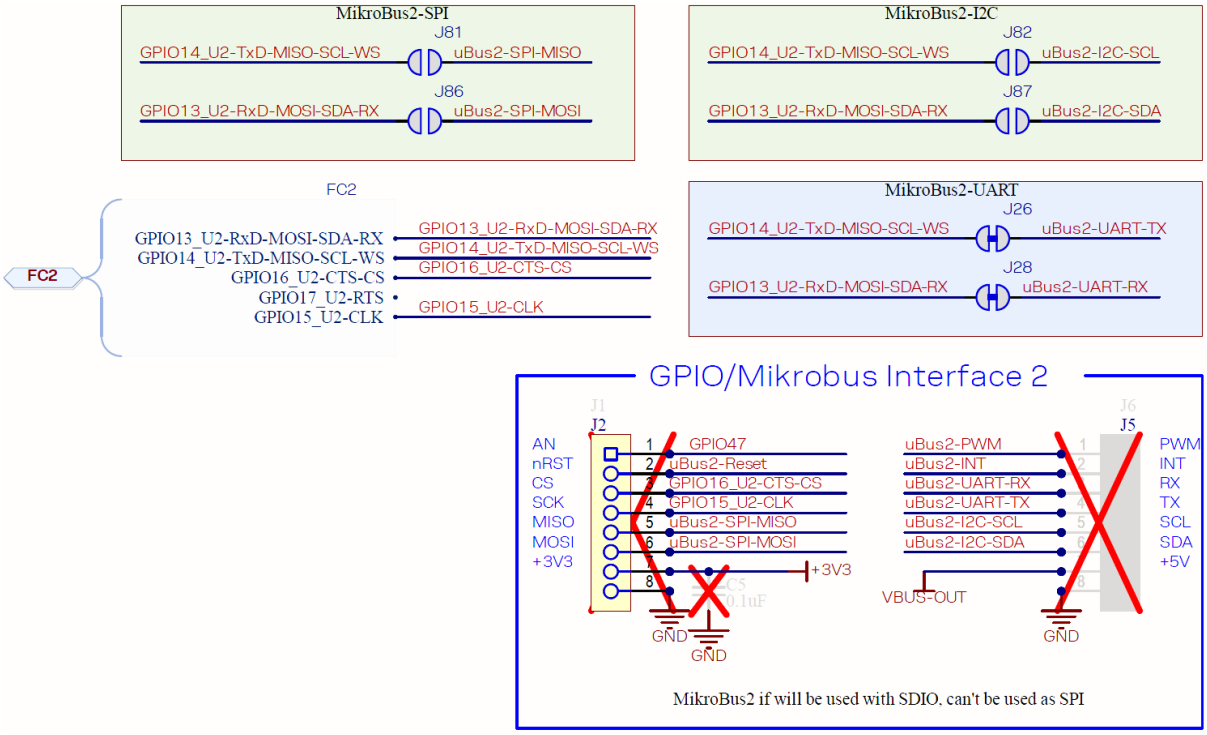


Figure 30: mikroBUS 2 schematic

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

Table 16: mikroBUS 2 working standard options, (*) default connection

3.8.3 mikroBUS 3 slot

This slot is connected to the Flexcomm interface 1 that cannot be used with the **JTAG** interface that shares all GPIOs on this slot, as shown in Figure 32.

To enable this slot:

Invert the default connection of **J34, J36, J35, J15,** as shown in Figure 31**.**

Check that jumpers **J11, J16, J18, J73, J76, J79, J61, J62** and resistors **R46, R70,** match the required operation, as described in Table 12, Table 13, Table 14, and Table 17 .

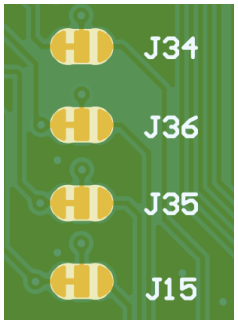


Figure 31: mikroBUS 3 jumpers J34, J36, J35, J15 default connection

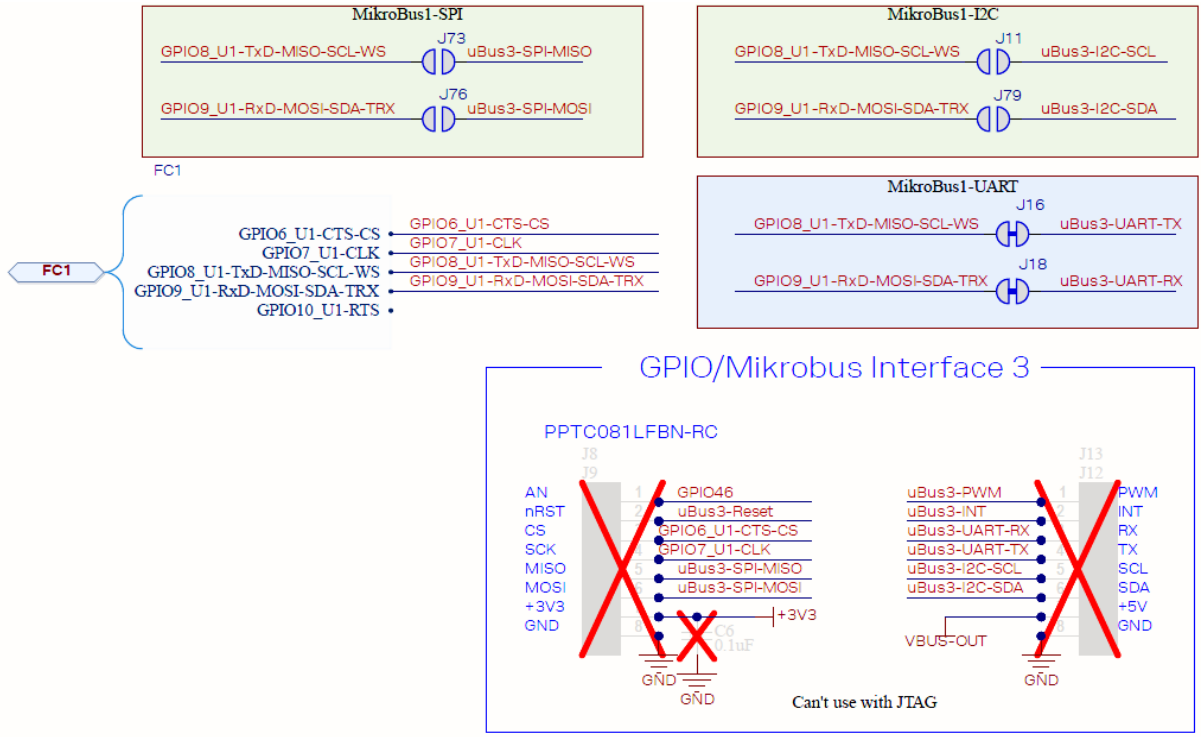


Figure 32: mikroBUS3 schematic

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

Table 17: mikroBUS 3 working standard options, (*) default connection

3.8.4 mikroBUS 4 slot

This slot is connected to the Flexcomm interface 1 that cannot be used totally with RMII function, which shares **GPIO55, GPIO 56** and **GPIO 57** as shown in Figure 33.

To enable this slot:

- Disconnect R111, R263, R265 and populate R115, R264, R266 with 0R
- Check that jumpers **J17, J19, J74, J75, J77, J78, J90, J93** and resistors **R10, R19,** match the required operation, as described in Table 12, Table 13, Table 14, and Table 18.

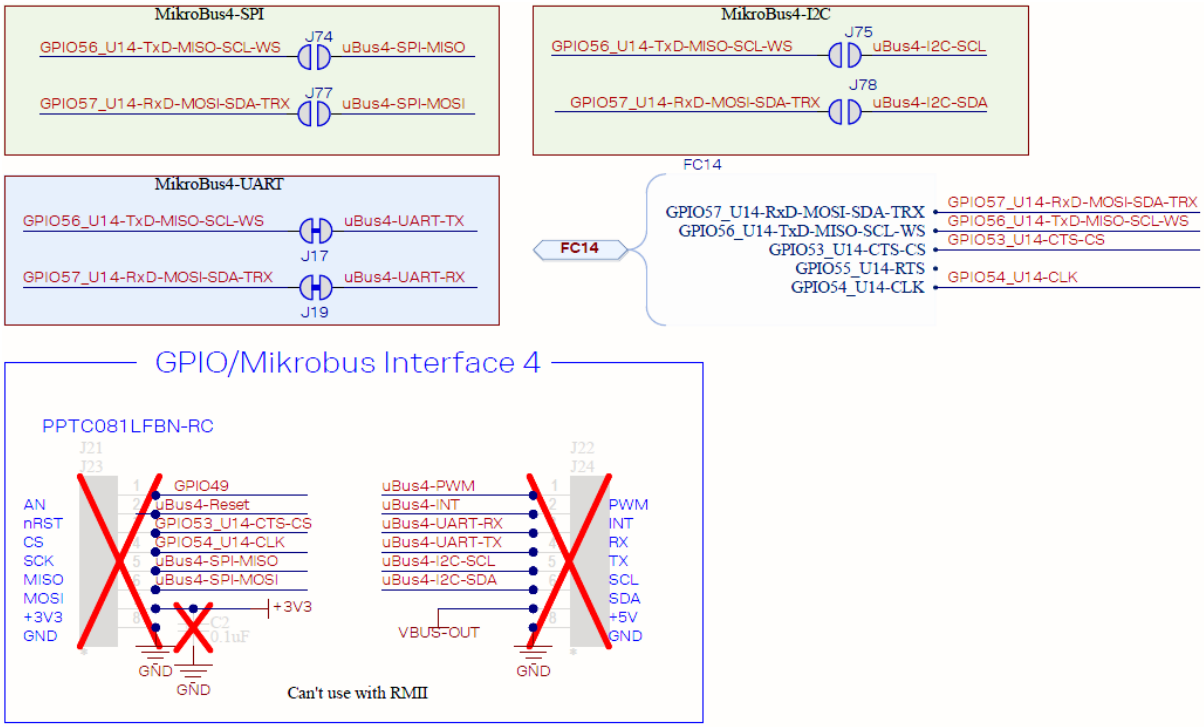


Figure 33: mikroBUS4 schematic

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

Table 18: mikroBUS 4 working standard options, (*) default connection

3.9 RMII

The IRIS-W10 EVB 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 (described in Table 19) from other functions that cannot be used simultaneously with RMII, as shown in Figure 34.

- Group 1, Flexcomm **F14 RX, TX, RTS** signals – populate **R111, R263, R265** with **0R** and disconnect **R115, R264, R266**.
- Group 2, Flexcomm **F3** Clock signal – populate **R279** with **0R** and disconnect **R282.**
- Group 3, 32.768 kHz external crystal oscillator – populate **R268, R272** with **0R** and disconnect **R269, R267, R271, R272.**
- Group 4, pin headers and reset signals – populate **R210, R277, R278, R283, R284, R286** with **0R** and disconnect **R270, R275, R276, R280, R281, R285**.

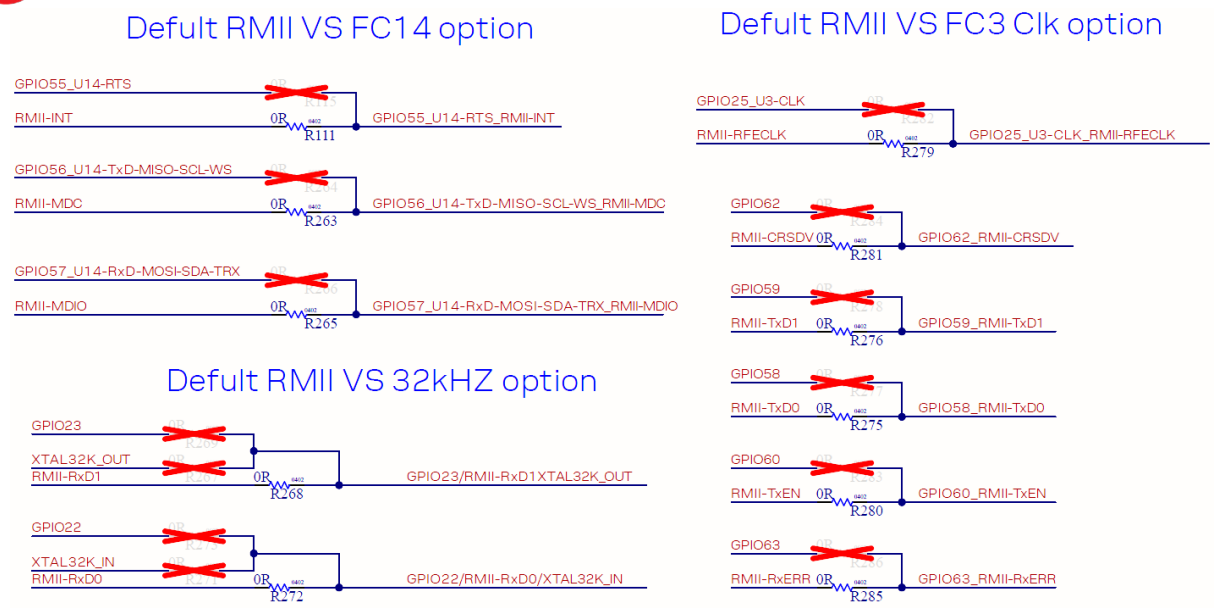


Figure 34: RMII demultiplexing resistors schematic

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 19: RMII signals and pins

3.9.1 RMII Strap-in options

The strap-in pins are latched at the de-assertion of reset. the MAC RMII receive input pins may drive high/low during power-up or reset, and consequently cause the PHY strap-in pins on the RMII signals to be latched to unintended high/low states. In this case, external pull-ups (4.7 kΩ) or pull-downs (1.0 kΩ) should be added to these PHY strap-in pins to ensure that the intended values are strapped-in correctly, as shown in Figure 35 and Table 20.

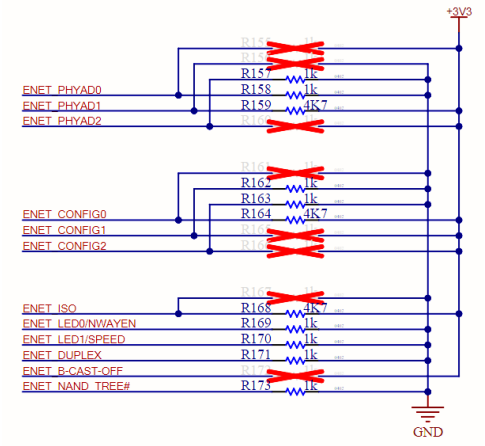


Figure 35: PHY circuit strapping options

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 20: RMII strap-in

3.10 SDIO 3

SDIO 3 function is not enabled by default on **J100**. All SDIO signals shown in Table 21 are series terminated with 33 Ω resistors for better signal quality, as shown in Figure 36.

To enable the SDIO 3 function:

- Invert default connections for **J55, J56, J57, J58, J59, J60**, as shown in Figure 37 and Figure 38.
- Connect jumper **J99**
- Populate capacitors **C71** 10 uF, **C72** 100 nF
- Populate TVS DIODE **U9** (CDDFN10-0516P)

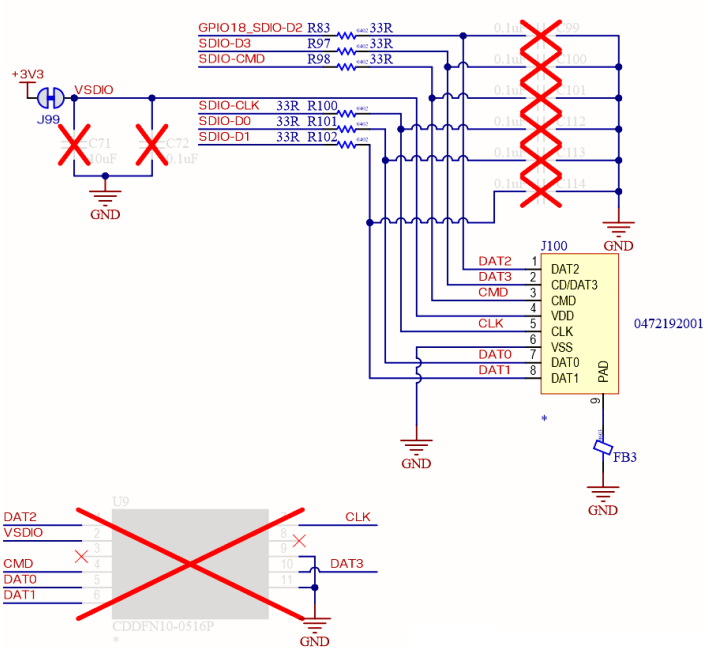


Figure 36: SDIO schematic

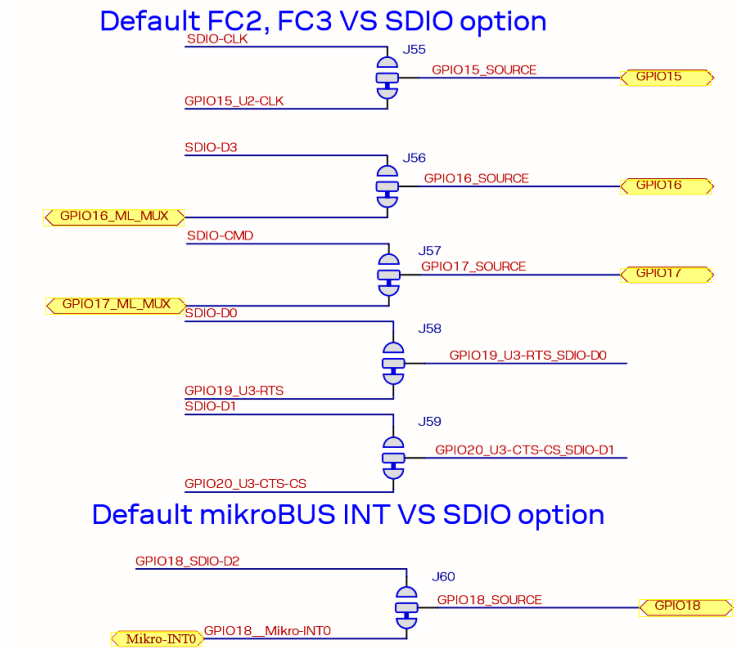


Figure 37: SDIO jumpers schematic

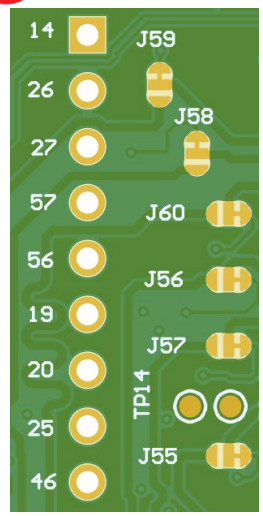


Figure 38: Jumpers J55, J56, J57, J58, J59, J60 default connection

IRIS-W10 pin name	IRIS-W10 function	Resistor/Jumper enable	Interface IC function
A8	GPIO15/SDIO-CLK	R100/J55	SDIO-CLK
A7	GPIO16/SDIO-D3	R97/J56	SDIO-D3
B7	GPIO17/SDIO-CMD	R98/J57	SDIO-CMD
B8	GPIO19/SDIO-D0	R101/J58	SDIO-D0
A9	GPIO20/SDIO-D1	R102/J59	SDIO-D1
B6	GPIO18/SDIO-D2	R83/J60	SDIO-D2

Table 21: SDIO signals and pins

3.11 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 39.

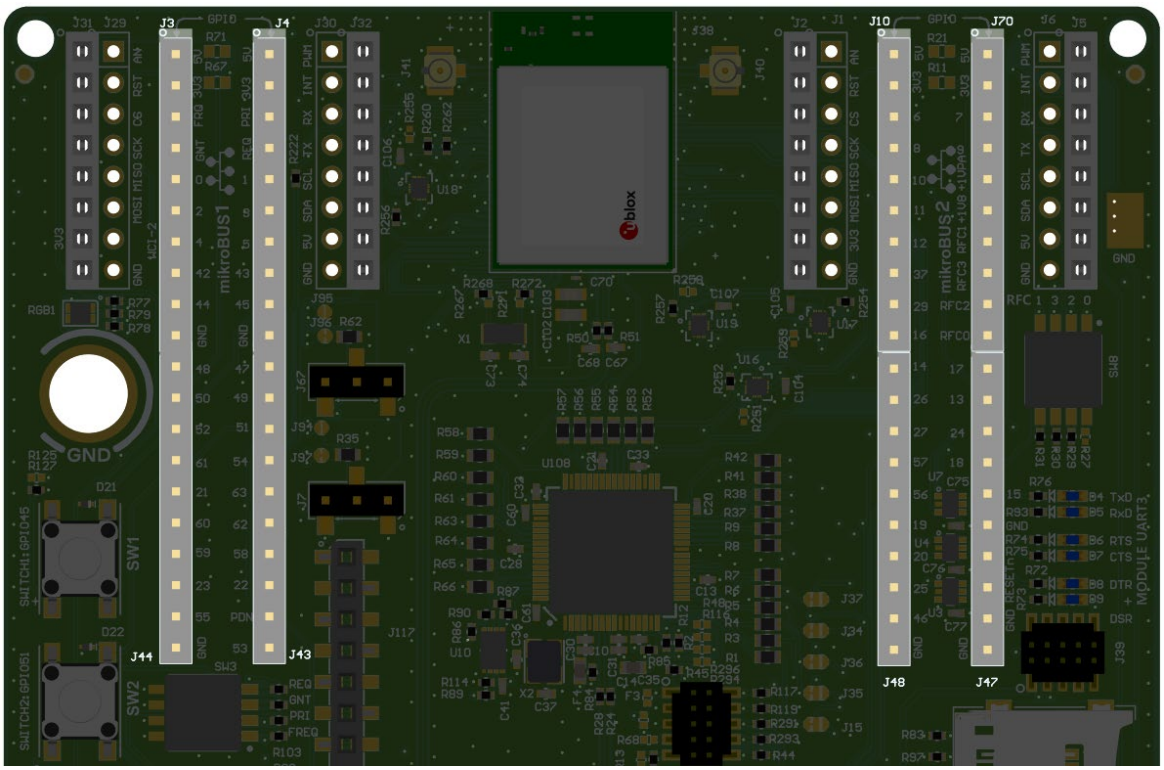


Figure 39: Pin headers

Several GPIOs described in Table 22, Table 23, and Table 24:

- are connected to the pin headers by default and can't be disconnected
- are connected to the pin headers by default but can be disconnected
- are not connected to the pin headers by default but can be connected
- may be connected to the pin headers and other locations simultaneously

The right columns in Table 22, and Table 23 show how to connect, disconnect, or disconnect other locations related to each GPIO or function.

GPIO	Header	Pin	Connection	How to connect, disconnect, 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 **10K.**
GPIO3	J4	6	Default but can disconnect	Disconnect **R37**, **R255**, **J25, J80, J83** populate **R260** with **0R.**
GPIO4	J3	7	Default (WCI-2)	Disconnect **R8** and MikroBUS 1- **SCK and** populate **R262** with **0R.**
GPIO5	J4	7	Default but can disconnect	Disconnect **R255** and populate **U18** and **R256** with **10K.**
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**
GPIO13	J47	2	Not default	Disconnect **R257** and populate **U19** and **R258** with **10K.** Or populate **R143** with **0R.**
GPIO14	J48	1	Not default	Disconnect **R257** and populate **U19** and **R258** with **10K.** Or populate **R142** with **0R.**
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 **10K.** Disconnect MikroBUS 2- **CS**
GPIO17	J47	1	Default but can disconnect	**J57** default connection, disconnect **R253** and populate **U17** and **R254** with **10K.**
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 **0R.**
GPIO22	J43	8	**Not** default	Disconnect **R272** and populate **R273** with **0R.**
GPIO23	J44	8	**Not** default	Disconnect **R271** and populate **R269** with **0R.**
GPIO24	J47	3	**Not** default	Disconnect **R252** and populate **U19** and **R251** with **10K.** Or populate **R134** with **0R.**
GPIO25	J48	8	**Not** default	Disconnect **R279** and populate **R282** with **0R.**
GPIO26	J48	2	**Not** default	Disconnect **R252** and populate **U19** and **R251** with **10K.** Or populate **R136** with **0R.**
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**

GPIO	Header	Pin	Connection	How to connect, disconnect, disconnect other locations
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 **0R.**
GPIO56	J48	5	**Not** default	Disconnect **R263, J17, J74, J75** and populate **R264** with **0R.**
GPIO57	J48	4	**Not** default	Disconnect **R265, J19, J77, J78** and populate **R266** with **0R.**
GPIO58	J43	7	**Not** default	Disconnect **R275** and populate **R277** with **0R.**
GPIO59	J44	7	**Not** default	Disconnect **R276** and populate **R278** with **0R.**
GPIO60	J44	6	**Not** default	Disconnect **R280** and populate **R283** with **0R.**
GPIO61	J44	4	Default and can't disconnect	
GPIO62	J43	6	**Not** default	Disconnect **R281** and populate **R284** with **0R.**
GPIO63	J43	5	**Not** default	Disconnect **R285** and populate **R286** with **0R.**

Table 22: GPIO mapping to the pin headers

IRIS-W10 function	Header	Pin	Default	How to connect, disconnect, 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 **0R.**
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 **0R** 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 23: Other signals mapping to the pin headers

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

Table 24: Power signals mapping to the pin headers

3.12 QSPI memory

The EVK-IRIS-W10 provides an option for external memory. A Quad SPI PSRAM (**U2**), shown in Figure 40 and Table 25, can be optionally mounted on the rear side of EVK board.

- The memory is as powered by **+3V3.**
- **C24** and **C47** are intended as decoupling capacitors for the PSRAM.

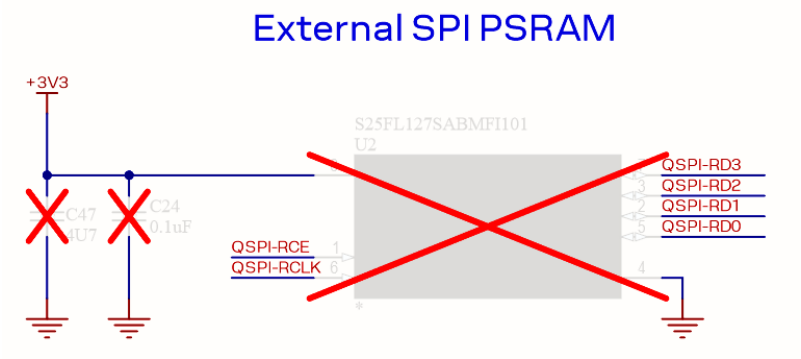


Figure 40: QSPI memory schematic

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 25: QSPI memory signals and pins

3.13 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 26 shows all jumper types and locations on the EVK, while Table 27 shows the default function and alternate function of each jumper – if applicable.

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 26: solder-bridge jumper types

Jumper	Default connection	Alternate function
J11	Connect GPIO8 to mikroBUS4 SCL	Cannot 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	Cannot be connected simultaneously with J73, J11
J17	Connect GPIO 56 to mikroBUS4-UART_TX	Cannot be connected simultaneously with J74, J75
J18	Connect GPIO 9 to mikroBUS3-UART_RX	Cannot be connected simultaneously with J76, J79
J19	Connect GPIO 57 to mikroBUS4-UART_RX	Cannot be connected simultaneously with J77, J78
J25	Connect GPIO 3 to mikroBUS1-UART_TX	Cannot be connected simultaneously with J80, J83
J26	Connect GPIO 14 to mikroBUS2-UART_TX	Cannot be connected simultaneously with J81, J82
J27	Connect GPIO 2 to mikroBUS1-UART_RX	Cannot be connected simultaneously with J84, J85
J28	Connect GPIO 13 to mikroBUS2-UART_RX	Cannot 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	Connect SDIO-CLK

Jumper	Default connection	Alternate function
	J47-15	
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	Cannot be connected simultaneously with J88, J89, J90
J62	Enable mikroBUS3-INT	Cannot be connected simultaneously with J91, J92, J93
J73	Connect GPIO 8 to mikroBUS3-SPI_MISO	Cannot be connected simultaneously with J11, J16
J74	Connect GPIO 56 to mikroBUS4-SPI_MISO	Cannot be connected simultaneously with J17, J75
J75	Connect GPIO 56 to mikroBUS4-I2C_SCL	Cannot be connected simultaneously with J17, J74
J76	Connect GPIO 9 to mikroBUS3-SPI_MOSI	Cannot be connected simultaneously with J18, J79
J77	Connect GPIO 57 to mikroBUS43-SPI_MOSI	Cannot be connected simultaneously with J19, J78
J78	Connect GPIO 57 to mikroBUS4-I2C_SDA	Cannot be connected simultaneously with J19, J77
J79	Connect GPIO 9 to mikroBUS3- I2C_SDA	Cannot be connected simultaneously with J18, J76
J80	Connect GPIO 3 to mikroBUS1-SPI_MISO	Cannot be connected simultaneously with J25, J83
J81	Connect GPIO 14 to mikroBUS2-SPI_MISO	Cannot be connected simultaneously with J26, J82
J82	Connect GPIO 14 to mikroBUS2-I2C_SCL	Cannot be connected simultaneously with J26, J81
J83	Connect GPIO 3 to mikroBUS1- I2C_SCL	Cannot be connected simultaneously with J25, J80
J84	Connect GPIO 2 to mikroBUS1- I2C_SDA	Cannot be connected simultaneously with J27, J85
J85	Connect GPIO 2 to mikroBUS1- SPI_MOSI	Cannot be connected simultaneously with J27, J84
J86	Connect GPIO 13 to mikroBUS2-SPI-MOSI	Cannot be connected simultaneously with J28, J87
J87	Connect GPIO 13 to mikroBUS2-I2C_SDA	Cannot be connected simultaneously with J28, J86
J88	Connect GPIO 27 to mikroBUS2-PWM	Cannot be connected simultaneously with J61, J89, J90
J89	Connect GPIO 27 to mikroBUS1-PWM	Cannot be connected simultaneously with J61, J88, J90
J90	Connect GPIO 27 to mikroBUS4-PWM	Cannot be connected simultaneously with J61, J88, J89
J91	Enable mikroBUS2-INT	Cannot be connected simultaneously with J62, J92, J93
J92	Enable mikroBUS1-INT	Cannot be connected simultaneously with J62, J91, J93
J93	Enable mikroBUS4-INT	Cannot be connected simultaneously with J62, J91, J92
J94	Bypass R35, VDD current sense resistor	Cannot be used with J97
J95	Bypass R62, VDD-IO current sense resistor	Cannot be used with J96, J115, J116
J96	Connect R62, VDD-IO current sense resistor to +3.3V	Cannot be used with J95, J115, J116
J97	Connect R35, VDD current sense resistor	Cannot be used with J94
J115	Connect R62, VDD-IO current sense resistor to +1V8	Cannot be used with J95, J96, J116
J116	Connect R62, VDD-IO current sense resistor to +1VPA	Cannot be used with J95, J96, J115

Table 27: solder-bridge jumpers showing default connections and alternate functions

3.14 Test points

Table 28 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 28: Test point description

Appendix

A 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
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 26: 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] https://www.segger.com/downloads/jlink/
[6] Github repo u-blox-sho-OpenCPU
[7] FTDI FT231XQ-R Datasheet, FT231X (ftdichip.com)
[8] https://www.nxp.com/docs/en/nxp/data-sheets/LPC55S6x_DS.pdf
[9] https://www.mikroe.com/click
[10] https://www.nxp.com/docs/en/user-manual/UM11673.pdf
[11] https://www.nxp.com/downloads/en/schematics/MCU-LINK-PRO-SCH.pdf
[12] https://www.nxp.com/docs/en/user-guide/MCUBLHOSTUG.pdf

☞ 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	09-Jan-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. Replaced "Hello world" example with "Wi-Fi example application". Revised "Hardware description" chapter, including the addition of function and header descriptions in Figure 6. Updated Figure 7 to Figure 40.

Contact

u-blox AG

Address: Zürcherstrasse 68
8800 Thalwil
Switzerland

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