3 C099-F9P description

3.1 Component overview

C099-F9P houses the ZED-F9P RTK high precision positioning module and an ODIN-W2 module for wireless short-range communications. An FTDI component provides dedicated COM port connections with the ZED-F9P and ODIN-W2 modules via a USB connector.

The board can be powered by USB, a DC supply socket, or by a Li-Po (lithium polymer) battery. The board has been designed using an Arduino form factor with the modules' serial ports routed to the shield headers. Note that a secondary USB power source is available via the USB-to-DC plug adapter cable.

The block diagram in Figure 4 shows the logical signal flow between the individual parts.

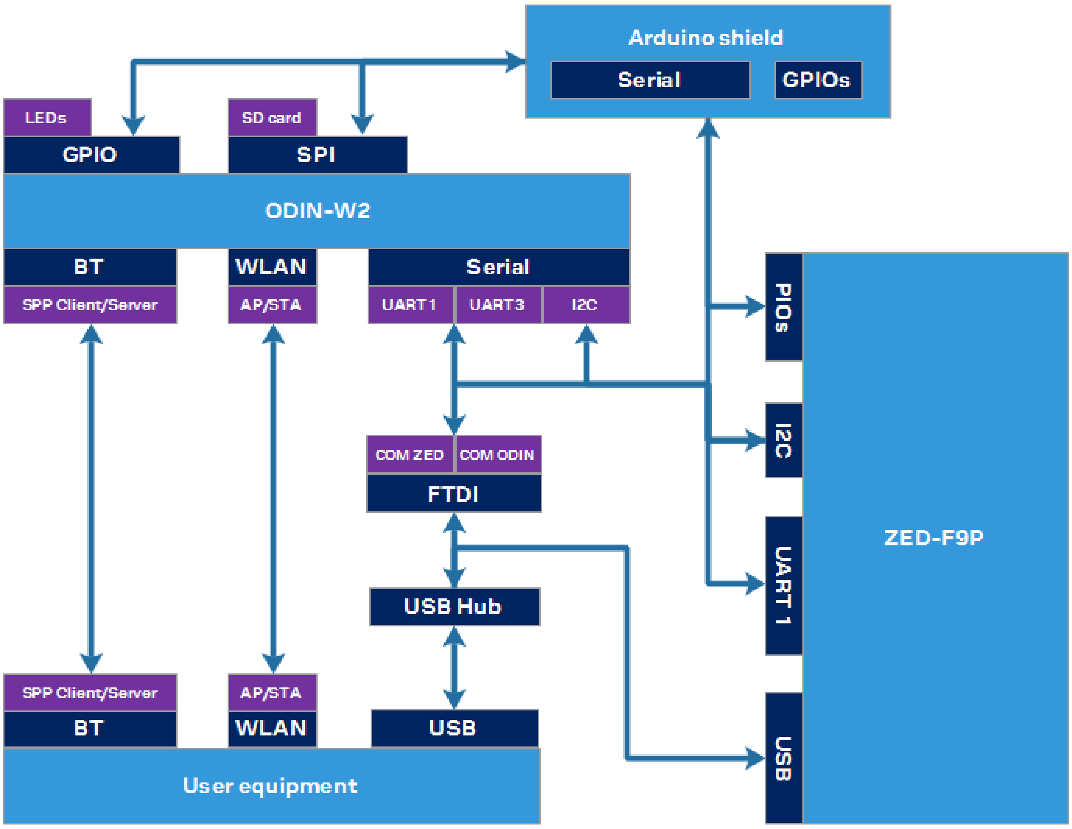


Figure 4: C099-F9P block diagram

3.2 Component identification

The following images show the position of major parts and user interfaces.

- Main components – Figure 5.
- Switches and LEDs – Figure 6.

Their functions are described later in this section.

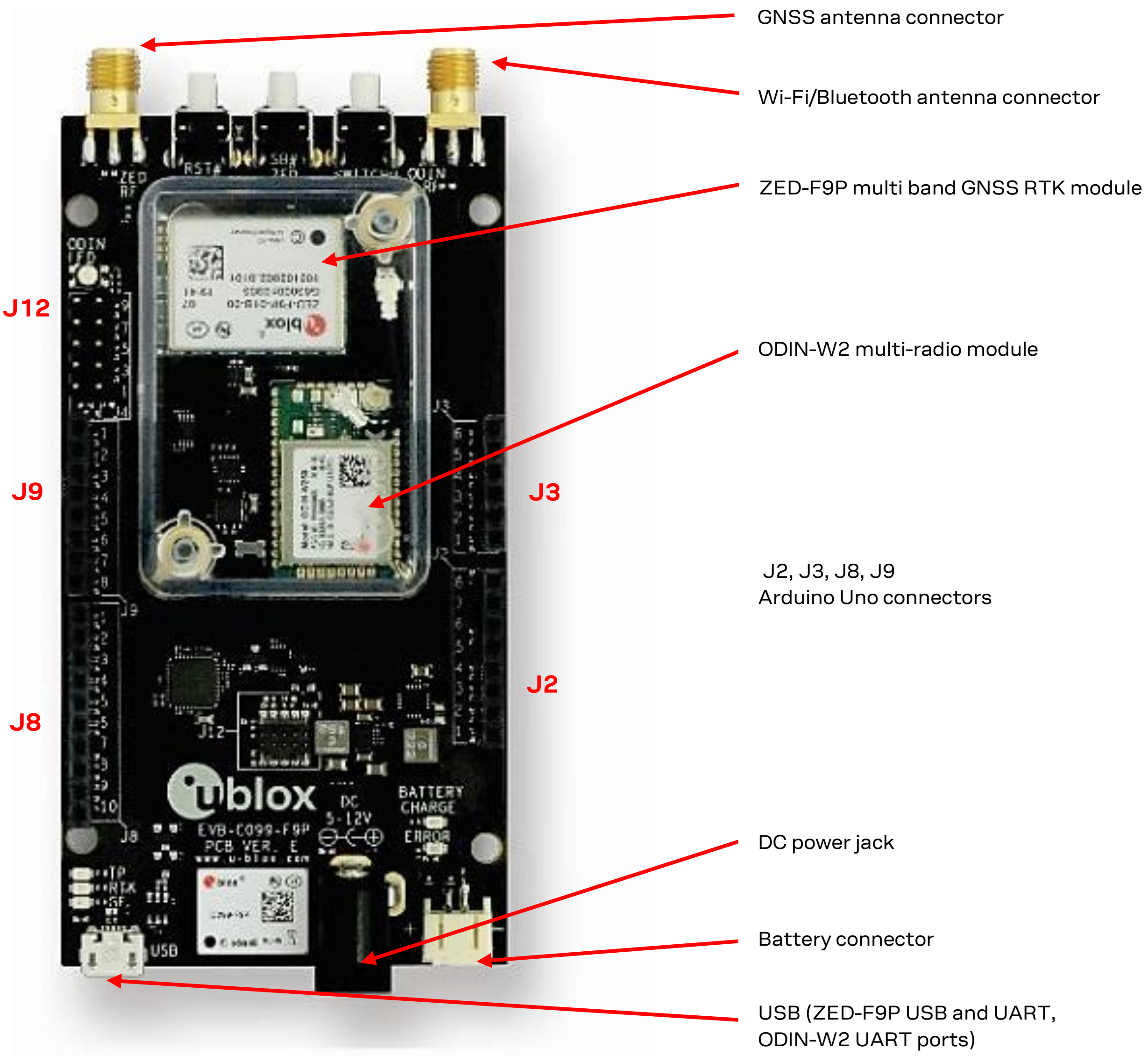
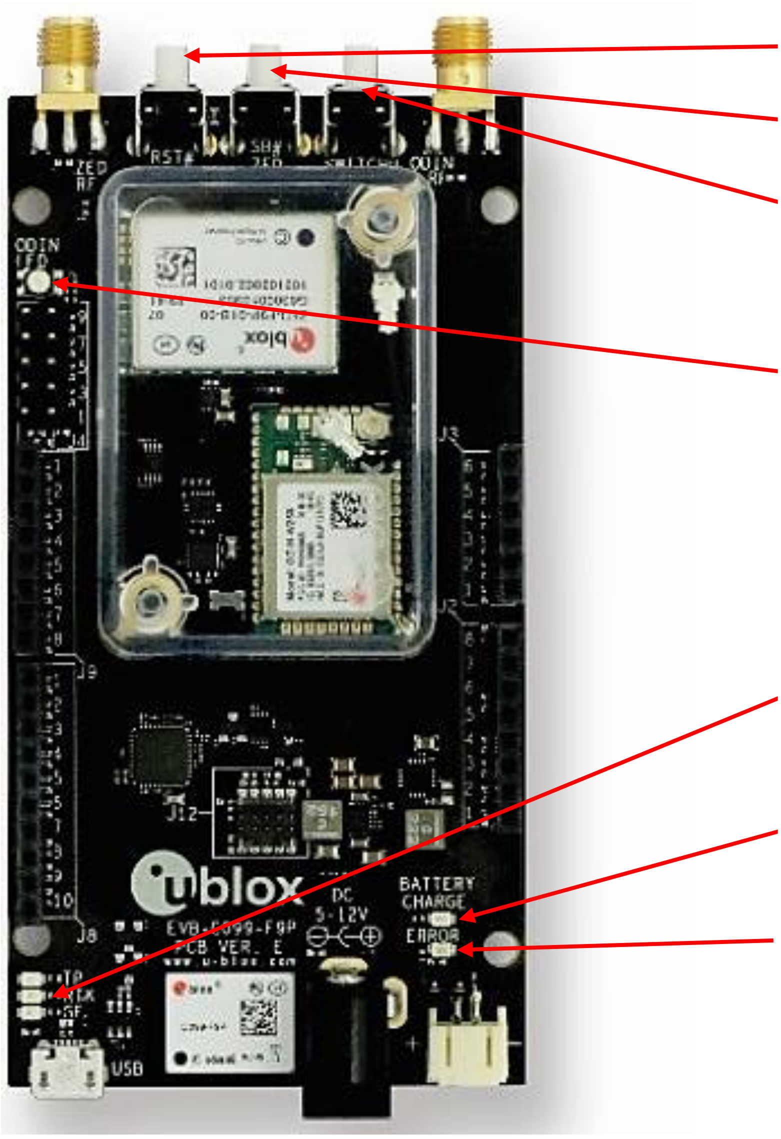

GNSS antenna connector
Wi-Fi/Bluetooth antenna connector
ZED-F9P multi band GNSS RTK module
ODIN-W2 multi-radio module
J2, J3, J8, J9
Arduino Uno connectors
DC power jack
Battery connector
USB (ZED-F9P USB and UART,
ODIN-W2 UART ports)
J12
J9
J8
J3
J2

Figure 5: Main components and USB ports



ZED-F9P and ODIN-W2 reset button

ZED-F9P safeboot button

ODIN-W2 switch 0 interrupt button

ODIN-W2 activity LED

GNSS LEDs: TP, RTK, Geofence

Battery charger LED

Battery incorrect polarity indicator LED

Figure 6: Switches and LEDs

☞ The microSD card functionality is not supported by the currently released Mbed firmware for ODIN-W2.

3.2.1 ZED-F9P status LEDs

The board provides three LEDs to show the ZED-F9P status. The location of the LEDs is shown in Figure 7 below.

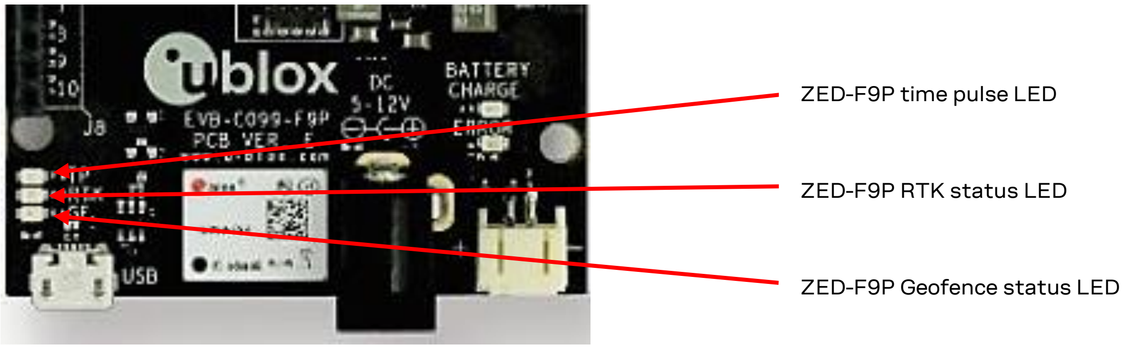
The RTK status LED provides an indication of the state of the ZED-F9P module's RTK-STAT pin.

- At startup, the LED is off.
- When a valid stream of RTCM messages is being received and utilized, but no RTK fixed mode has been achieved, the yellow LED flashes.
- When in RTK fixed mode, the yellow LED is turned on.

The blue time pulse LED flashes at the default 1-Hz rate when the time solution is valid.

If activated, the Geofence status LED indicates the current Geofence status, i.e. within or outside a designated area.

See the ZED-F9P Interface description [2] for help with configuring the time pulse output or activating the Geofence pin.



ZED-F9P time pulse LED

ZED-F9P RTK status LED

ZED-F9P Geofence status LED

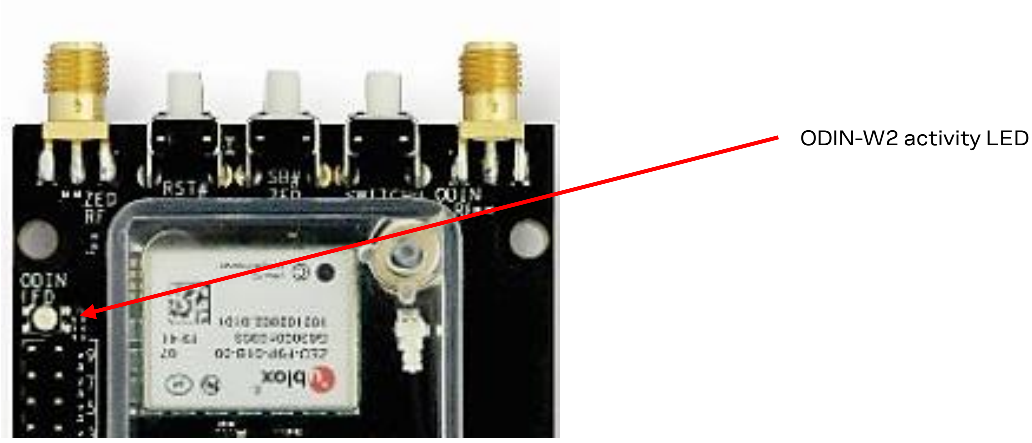
Figure 7: ZED-F9P LEDs

3.2.2 ODIN-W2 activity LED

The ODIN-W2 module uses a multi-colored LED to show a particular activity status. This is positioned adjacent to the ZED-F9P and ODIN-W2 reset switch and shown below in Figure 8. The activity status is summarized in Table 1 below.

Status	LED color	Remark
Successful startup, Bluetooth radio initialized	Green	
Bluetooth serial port profile (SPP) connection created	Blue	Connection initiated and accepted
Successful SPP data packet transmission	Blinking blue	
Failed SPP data packet transmission	Blinking red	Weak signal, Bluetooth SPP connection failure
Wi-Fi access point and station (AP and STA) ready	Yellow	Ready to accept incoming Wi-Fi STA connection
Wi-Fi STA connected to AP	Purple	Ready to accept incoming UDP client connection
Successful UDP packet transmission over Wi-Fi	Blinking purple	UDP packet reception is not indicated
ODIN-W2 in safeboot mode	LED off	Safeboot is triggered during startup. Requires the safeboot jumper to be connected.

Table 1: ODIN-W2 Mbed FW LED activity states and colors



ODIN-W2 activity LED

Figure 8: ODIN-W2 activity LED position on C099-F9P board

3.2.3 C099-F9P Jumpers

The J4 connector provides options for ZED-F9P UART multiplexing. Table2 summarizes the main jumper options.

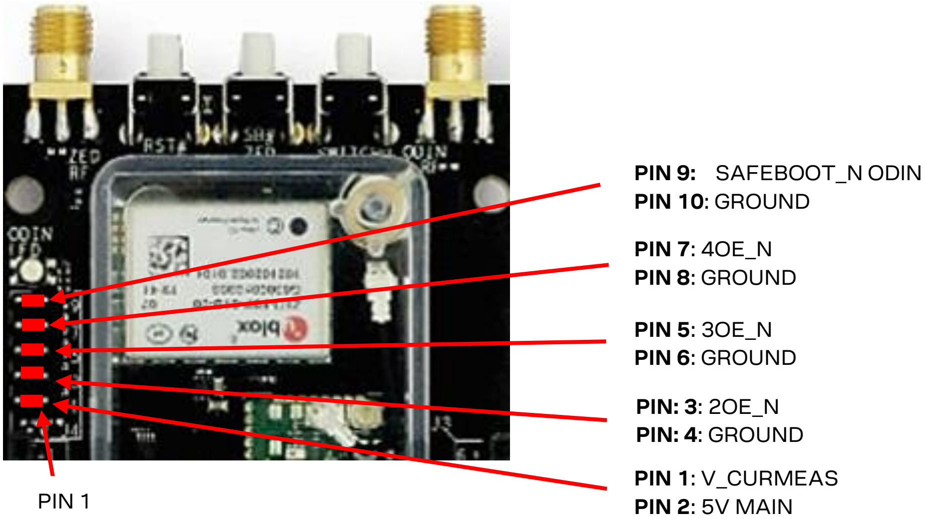


Figure 9: C099-F9P Jumper PIN details

PIN	Jumper purpose	Description
1-2	V_CURMEAS MODE	Current Measurement
3-4	ODIN UART3 MODE	Enable ZED-F9P RXD connection to ODIN UART3 (ODIN TXD3 -> ZED-F9P RXD)
5-6	UART1 MODE	Enable ZED-F9P RXD connection to ODIN UART1 (ODIN TXD -> ZED-F9P RXD)
7-8	ARDUINO MODE	Enable ZED-F9P RXD connection to Arduino shield (J9 pin1 -> ZED-F9P RXD)
9-10	SAFEBOOT_N ODIN MODE	Disable ODIN

Table 2: C099-F9P Jumpers PIN assignment

☞ Only one jumper can be placed on pins 3-4, 5-6 and 7-8 at a time. To configure the ZED-F9P, ensure there is no jumper on pins 3-4. The jumper can be replaced after configuration.

4 Using C099-F9P

The ZED-F9P module is shipped with the latest HPG firmware. Check the latest ODIN-W2 Mbed FW availability and information on the FW update procedures in section 7 Firmware update.

4.1 Powering the board

The board can be powered from a variety of sources:

- The USB connection
- A 3.7 V Li-Po battery via a JST connector
- An external 5-12 V DC source via a 2.1-mm connector; center pin V+. Also, the included USB-to-DC plug adapter cable can be used to provide an additional USB power source.

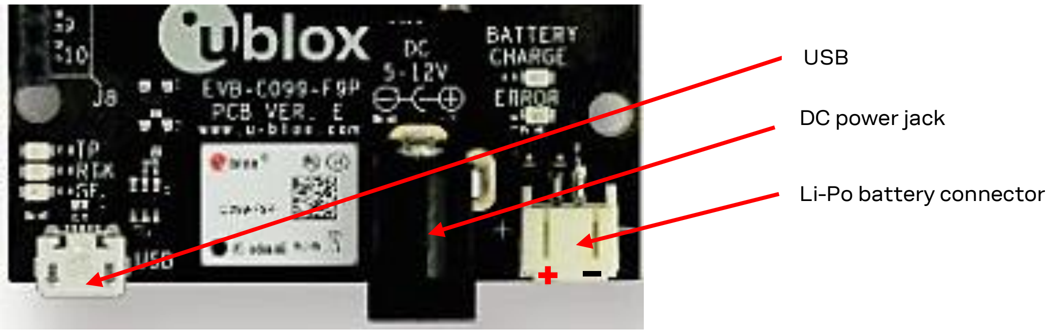


Figure 10: Power connections

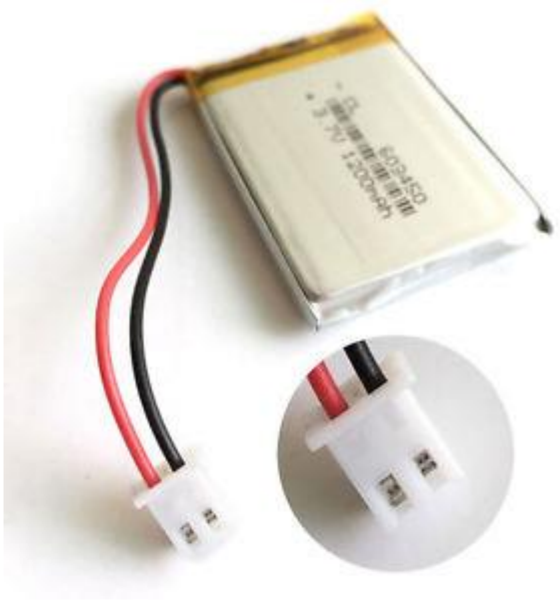


Figure 9: Typical single cell 3.7 V Li-Po battery with JST connector

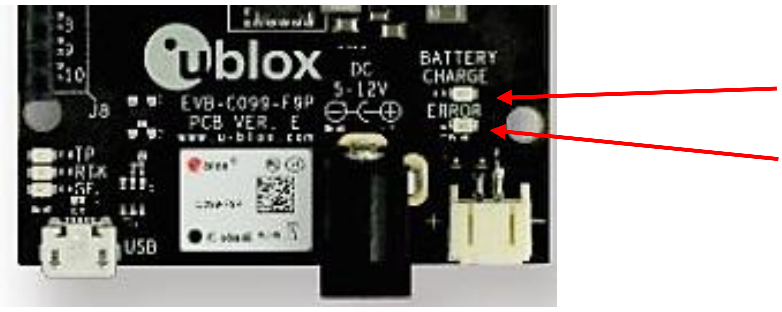
⚠ Follow all published safety advice for using bare cell Li-Po batteries while charging. Protect the batteries from mechanical damage. There is a risk of fire if the advice is not followed.

⚠ Ensure correct polarity on the JST battery connector. If the polarity is incorrect, the incorrect polarity LED is on. Due to the polarity protection feature, the supply rails are not powered.

All supply connections are fed via a Schottky diode to the main supply bus to allow multiple sources to be connected in parallel. The Li-Po battery is charged from either the DC power source or the USB power source. The charging status is indicated by a red LED which is on during charging and turned off when fully charged.

When less than 500 mA is available from the USB host, ensure sufficient extra supply via the DC power jack. Note that due to the higher current consumption caused by the battery charging it is not recommended to charge the battery via USB only.

☞ Supplying through the USB port requires the power source (USB host) to support the USB enumeration process. If the power source is not capable of enumeration, you may use the provided USB-to-DC adapter cable and connect it to the DC plug. There is no current limitation for the DC supply.



Red battery charge LED

Orange battery incorrect polarity indicator LED

Figure 10: Battery charge status LED

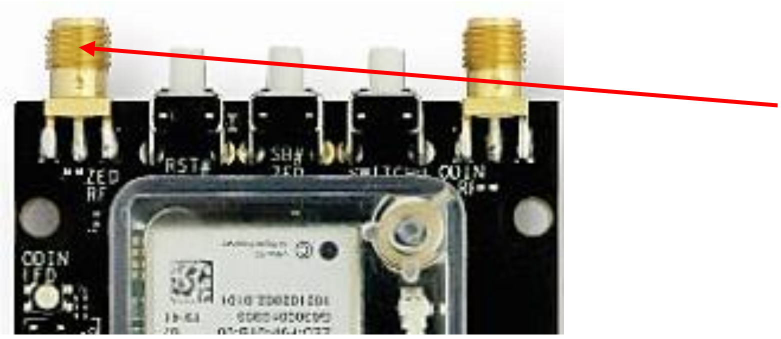
4.1.1 Non-wireless operation

For scenarios that require a non-wireless data link, ODIN-W2 can be disabled. To disable ODIN-W2, connect the safeboot jumper, which forces ODIN-W2 into safeboot mode during device startup. See Figure 6 to locate the safeboot pins.

On average, ODIN-W2 consumes less current when started in safeboot mode. In addition, safeboot mode ensures that no intentional radiation originates from the 2.4 GHz antenna connector.

4.2 GNSS RF input

The C099-F9P board should be used with the antenna supplied with the kit. If another active antenna is used, be aware that the RF input has a bias output designed to supply 3.3 V DC with a 70-mA maximum current load. A DC block is advisable if the board is connected to a signal distribution scheme or GNSS simulator to prevent any potential shorting of the antenna bias.



SMA GNSS antenna connector

Figure 11: GNSS antenna connector

When using the supplied antenna, it is advisable to use the ground plane provided. Otherwise, ensure that there is an adequate ground plane, e.g. by mounting the antenna in the center of a metallic car roof.

Figure 12: The supplied GNSS multi-band antenna

4.3 User interfaces

C099-F9P has a number of fixed connection options besides the wireless modes. There is also an additional Arduino R3 / Uno interface for external host connection.

The USB connector on the board provides connection via an on-board hub providing:

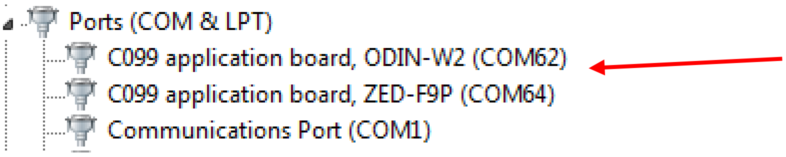
- An FTDI USB bridge to ZED-F9P UART1 and ODIN-W2 UART COM ports.
- Dedicated connection to the ZED-F9P USB port.

4.3.1 FTDI USB bridge

When the USB cable from the user's PC is connected, a driver loads and sets up two virtual serial ports, as shown below in Figure 13. Additionally, a further serial VCP is created to provide a direct connection with the ZED-F9P USB port.

☞ Ensure that the PC is connected to the internet to load the drivers from Windows Update.

The first of these is connected to the ZED-F9P serial port and should be selected with u-center. The second serial device is for the ODIN-W2 module when using s-center. In Figure 13, the ODIN-W2 connection is the first port (COM 62) and the ZED-F9P connection is the second port (COM 64). Port numbering can be different between individual PCs, but the same arrangement applies.



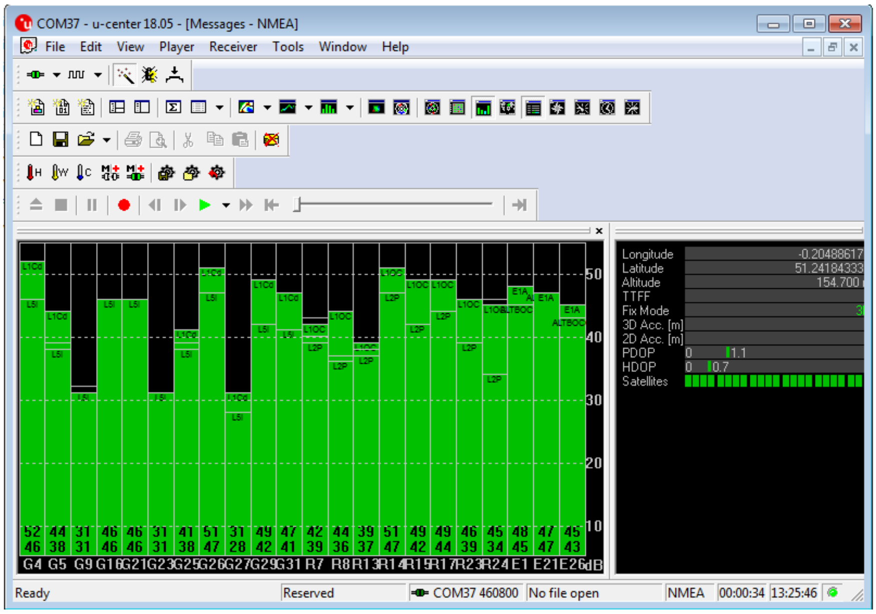
Ports (COM & LPT)
C099 application board, ODIN-W2 (COM62)
C099 application board, ZED-F9P (COM64)
Communications Port (COM1)

ZED-F9P and ODIN-W2
COM ports installed

Figure 13: Windows Device Manager COM port view

In addition, a third VCP is created corresponding to the ZED-F9P USB port. Windows 10 users see a new VCP device in the Device Manager window when it loads a built-in driver. With older Windows installations, a driver is loaded via Windows Update. In this case the device is identified as a u-blox GNSS device in the Device Manager window.

Open u-center (V18.12 or later), select the ZED-F9P serial port, and set the baud rate to 460800 to match the ZED-F9P default UART setting. Once connected, u-center shows typical received signal levels from multiple GNSS bands, see Figure 14 below.



Longitude -0.20488617
Latitude 51.24184333
Altitude 154.700
TTFF
Fix Mode 3
3D Acc. [m]
2D Acc. [m]
PDOP 0 1.1
HDOP 0 0.7
Satellites

Figure 14: u-center view with ZED-F9P connected

Additional UBX protocol messages can be enabled to view additional information in u-center. For example, the following are typical messages the user can poll or enable for periodic update.

- UBX-NAV-HPPOSLLH
- UBX-NAV-RELPOSNED
- UBX-NAV-SIG
- UBX-NAV-PVT
- UBX-NAV-STATUS
- UBX-NAV-SVIN

☞ For help with the Message view, see u-center User guide [3].

4.3.2 Command line interface of ODIN-W2

The user controls the ODIN-W2 module through a command line interface (CLI) that supports Remote Procedure Call syntax, as described below:

```
/<function_name>/run <argument 1> <argument 2> …
```

To access the ODIN-W2 CLI, use the following default serial settings:

- Baud rate: 460800
- Serial frame: 8 bits, 1 stop bit, no parity
- Flow control: None

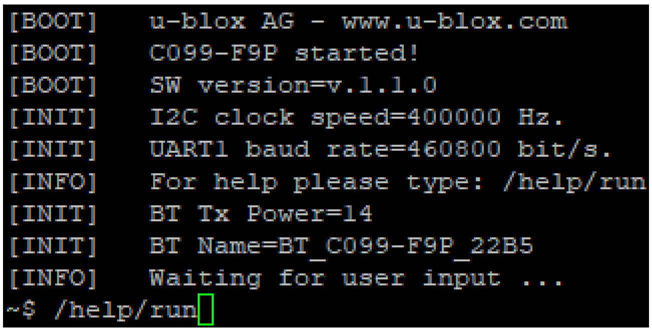
Prior to connecting to the ODIN-W2 CLI, check the following terminal settings:

☞ Putty (**Settings – Terminal**)

  - local echo force off
  - implicit CR in every LF off
  - implicit LF in every CR off

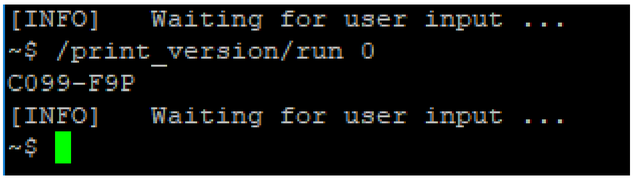
☞ Tera Term (**Setup – Terminal**)

  - newline receive CR and transmit CR
  - local echo disabled
  - terminal ID VT100



```
[BOOT]    u-blox AG - www.u-blox.com
[BOOT]    C099-F9P started!
[BOOT]    SW version=v.1.1.0
[INIT]    I2C clock speed=400000 Hz.
[INIT]    UART1 baud rate=460800 bit/s.
[INFO]    For help please type: /help/run
[INIT]    BT Tx Power=14
[INIT]    BT Name=BT_C099-F9P_22B5
[INFO]    Waiting for user input ...
~$ /help/run
```

Figure 15: CLI help command



```
[INFO]    Waiting for user input ...
~$ /print_version/run 0
C099-F9P
[INFO]    Waiting for user input ...
~$
```

Figure 16: Example RPC syntax

By typing the help command as in Figure 15, ODIN-W2 displays all available user commands with a short description. The CLI embodies character echo with limited text edit functions. Misspelled commands are replied to with a list of supported commands. Note that ODIN-W2 features I/O-related functions for diagnostic purposes. These functions are listed by the CLI but are not documented in this user guide.

4.4 Persistent ODIN-W2 settings

By default, ODIN-W2 starts in Bluetooth initiator role, and the ODIN-W2 UART1 is configured to use a 460800 baud rate. However, some user settings can be stored in the non-volatile data storage (flash) in ODIN-W2 and applied after a power cycle.

The user settings are saved into the flash memory via the following CLI command:

```
/mem_store/run <argument 1> <argument 2>
```

4.4.1 Revert to factory default

Factory default settings can be set by one of the two methods:

1. `/mem_erase/run` (via CLI)
2. Press down the SW0 button for more than 3 seconds.

During the next restart of ODIN-W2, the factory default settings will be applied.

1. Configure C099-F9P to Wi-Fi AP mode by using the CLI command in terminal:

   ```
   /mem_store/run wifi_ap
   ```

2. Set the C099-F9P Wi-Fi and I2C interfaces to support base operation[5]:

   ```
   /mem_store/run base
   ```

3. Restart C099-F9P to apply the Wi-Fi AP settings.

   ☞ ODIN-W2 waits until a Wi-Fi STA (rover) connects to it before streaming any data over the wireless link. See section 6.3.2.1 Rover operation in Wi-Fi STA mode for rover configurations.

4. Apply ZED-F9P base settings through u-center by connecting to the ZED-F9P UART 1 port.

   ☞ Any RTCM messages sent over the Wi-Fi link shall be configured for the ZED-F9P I2C interface as depicted in Figure 23.

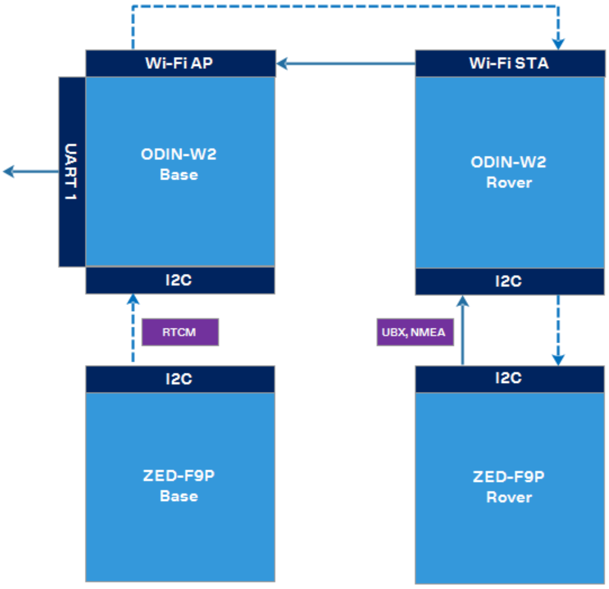


Figure 23: Wi-Fi base and rover setup

6.3.1.2 Rover operation in Wi-Fi AP mode

To connect to a C099-F9P rover via a Wi-Fi link, follow the configuration steps below:

1. Configure C099-F9P to Wi-Fi AP mode by using the CLI command in terminal:

   ```
   /mem_store/run wifi_ap
   ```

2. Set C099-F9P to operate as a rover:

   ```
   /mem_store/run rover
   ```

3. Restart C099-F9P to apply the Wi-Fi AP settings.
4. Connect the host PC’s Wi-Fi to the Wi-Fi AP of C099-F9P:

   “C099-F9P” is the default SSID
   “123456789” is the default WPA2 passphrase

[5] ZED-F9P I2C output port is enabled for RTCM messages. Disabled for UBX and NMEA protocols.

8 Arduino header connections

The board size and the four connectors comply with the Arduino R3/Uno mechanical specification. The functions of each I/O align as much as possible to the Arduino-specified functions. Check the pin functions and electrical compatibility before using the product with an Arduino R3/Uno - see Figure 39 below. All the pin functions besides power are 3.3 V compliant.

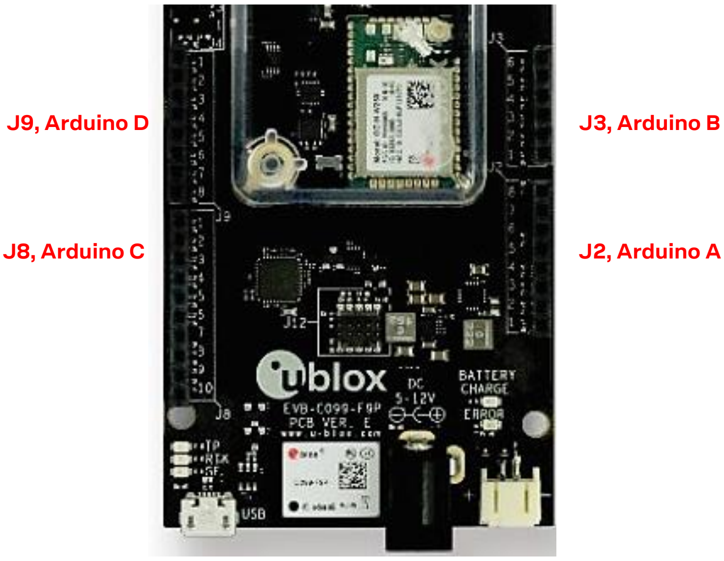


Figure 38: C099-F9P Arduino connectors

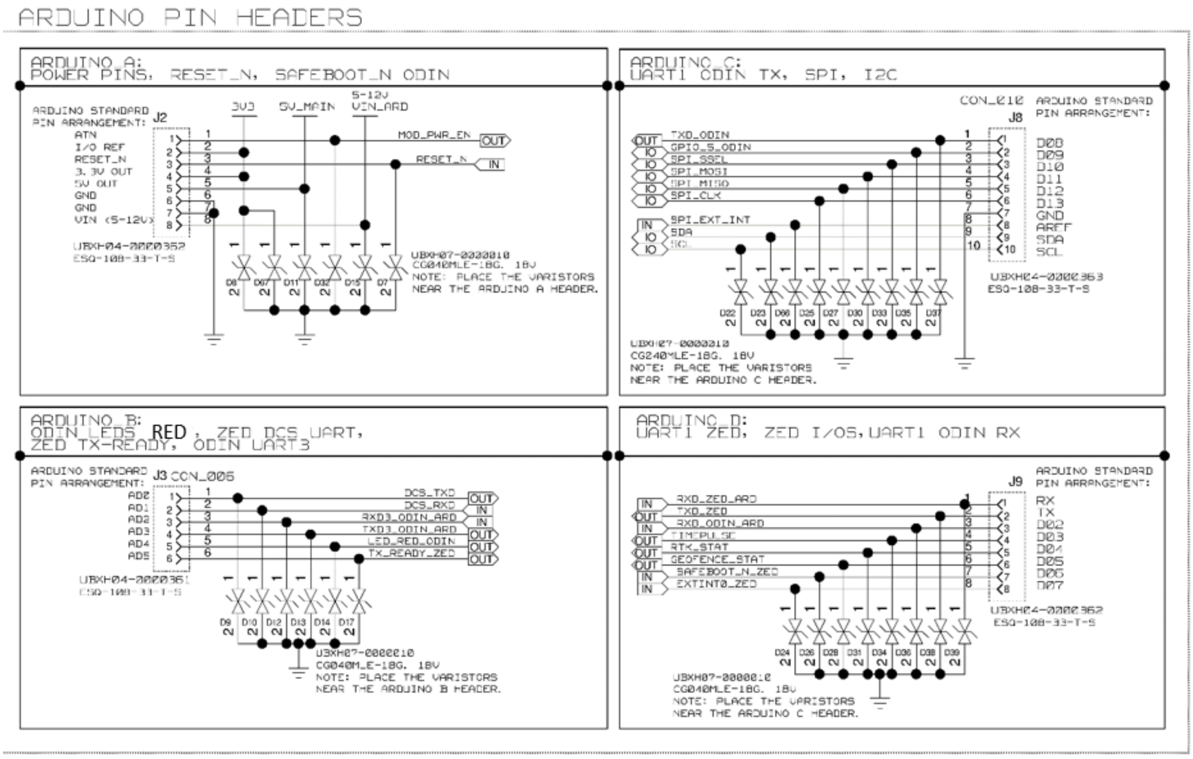


Figure 39: C099-F9P Arduino R3 connections

Appendix

A Glossary

Abbreviation	Definition
CLI	Command-line interface
FW	Firmware
LiPo	Lithium polymer
NTRIP	Networked transport of RTCM via internet protocol
NVDS	Non-volatile data storage
RTK	Real time kinematic
UART	Universal asynchronous receiver transmitter
UDP	User datagram protocol
USB	Universal serial bus
UTC	Coordinated universal time
VCP	Virtual COM port
Wi-Fi AP	Wi-Fi access point
Wi-Fi STA	Wi-Fi station

Table 4: Explanation of the abbreviations and terms used

B Resources

Applicable configuration files are available in u-blox Github:

https://github.com/u-blox/ublox-C099_F9P-uCS

C u-blox ODIN-W2 BT Rover.txt

Copy all the text below this line into a text file named “u-blox ODIN-W2 BT Rover.txt”.

```
AT+UBTLN="ODIN-W2-xxxx"
AT+UBTLC=000000
AT+UBTCM=2
AT+UBTDM=3
AT+UBTPM=2
AT+UBTMSP=1
AT+UBTLE=0
AT+UBTSM=1
AT+UNHN="ODIN-W2-200036001551373333393539"
AT+UDDRP=0,"",0
AT+UDDRP=1,"",0
AT+UDDRP=2,"",0
AT+UDDRP=3,"",0
AT+UDDRP=4,"",0
AT+UDDRP=5,"",0
AT+UDDRP=6,"",0
AT+UWSCA=0,4
AT+UWSC=0,0,0
AT+UWAPC=0,4
AT+UWAPC=0,0,0
AT+UWAPC=0,2,""
AT+UWAPC=0,4,1
AT+UWAPC=0,5,2,2
AT+UWAPC=0,100,1
AT+UWAPC=0,106,1
AT+UWSC=0,0,0
ATS2=43
```

```
ATS3=13
ATS4=10
ATS5=8
AT+UDCFG=0,1
AT&S1
AT&D0
ATE1
AT+UBTCFG=1,1
AT+UBTCFG=2,1
AT+UBTCFG=3,56602
AT+UBTCFG=4,127
AT+UBTCFG=5,0
AT+UBTCFG=6,0
AT+UBTCFG=7,2000
AT+UBTCFG=8,0
AT+UBTCFG=9,0
AT+UBTLECFG=1,1600
AT+UBTLECFG=2,2000
AT+UBTLECFG=3,7
AT+UBTLECFG=4,24
AT+UBTLECFG=5,40
AT+UBTLECFG=6,0
AT+UBTLECFG=7,2000
AT+UBTLECFG=8,5000
AT+UBTLECFG=9,48
AT+UBTLECFG=10,48
AT+UBTLECFG=11,24
AT+UBTLECFG=12,40
AT+UBTLECFG=13,0
AT+UBTLECFG=14,2000
AT+UBTLECFG=15,5000
AT+UBTLECFG=16,48
AT+UBTLECFG=17,48
AT+UBTLECFG=18,24
AT+UBTLECFG=19,40
AT+UBTLECFG=20,0
AT+UBTLECFG=21,2000
AT+UBTLECFG=22,5000
AT+UBTLECFG=23,48
AT+UBTLECFG=24,48
AT+UBTLECFG=25,0
AT+UMSM=1
AT+UMRS=460800,2,8,1,1,1
AT&W
AT+CPWROFF
```

D Rover ODIN-W2 Access Point UDP Server.txt

Copy all the text below this line into a text file named “Rover ODIN-W2 Access Point UDP Server .txt”.

```
AT+UWAPCA=0,4
AT+UWAPC=0,0,1
AT+UWAPC=0,2,UBXWifi
AT+UWAPC=0,4,1
AT+UWAPC=0,5,1,1
AT+UWAPC=0,100,1
AT+UWAPC=0,101,192.168.0.10
AT+UWAPC=0,102,255.255.0.0
AT+UWAPC=0,103,192.168.0.1
AT+UWAPC=0,104,0.0.0.0
AT+UWAPC=0,105,0.0.0.0
AT+UWAPC=0,106,1
AT+UWAPCA=0,1
AT+UWAPCA=0,3
AT+UWCFG=1,0
AT+UDSC=1,2,5003,1
AT+UMSM=1
```

```
AT+UMRS=460800,2,8,1,1,0
AT&D0
AT&W
AT+CPWROFF
```

E Reference station ODIN-W2 UDP client.txt

Copy all the text below this line into a text file named “Base ODIN-W2 Station UDP client.txt”.

```
AT+UWSCA=0,4
AT+UWSC=0,0,1
AT+UWSC=0,2,"UBXWifi"
AT+UWSC=0,5,1
AT+UWSC=0,100,2
AT+UWSCA=0,1
AT+UWSCA=0,3
AT+UWCFG=1,0
AT+UMSM=1
AT+UDDRP=0,"udp://192.168.0.10:5003",2
AT+UMRS=460800,2,8,1,1,0
AT&D0
AT&W
AT+CPWROFF
```

F F9P Stationary Base config C99.txt

Note: The receiver will output messages upon configuration below setting, however, for stationary base configuration, RTCM 1005 will only be output once the survey-in is completed, or the fixed coordinates are entered for the base antenna. Use the u-center "Packet Console View" to verify message output. Once surveyed in correctly, it will indicate a TIME solution mode in the u-center Data view.

Copy all the text below this line into a text file named “F9P Base config C99.txt.

```
# Config changes format version 1.0
# created by u-center version 18.11 at 11:37:53 on Tuesday, 08 Jan 2019
[del]
[set]
  RAM CFG-UART1INPROT-NMEA 0        # write value 0                  to item id 10730002
Flash CFG-UART1INPROT-NMEA 0        # write value 0                  to item id 10730002
  RAM CFG-UART1INPROT-RTCM3X 0      # write value 0                  to item id 10730004
Flash CFG-UART1INPROT-RTCM3X 0      # write value 0                  to item id 10730004
  RAM CFG-UART1OUTPROT-UBX 0        # write value 0                  to item id 10740001
Flash CFG-UART1OUTPROT-UBX 0        # write value 0                  to item id 10740001
  RAM CFG-UART1OUTPROT-NMEA 0       # write value 0                  to item id 10740002
Flash CFG-UART1OUTPROT-NMEA 0       # write value 0                  to item id 10740002
  RAM CFG-UART1OUTPROT-RTCM3X 1     # write value 1                  to item id 10740004
Flash CFG-UART1OUTPROT-RTCM3X 1     # write value 1                  to item id 10740004
Flash CFG-UART1INPROT-UBX  0        # write value 0                  to item id 10730001
  RAM CFG-UART1INPROT-UBX  0        # write value 0                  to item id 10730001
  RAM CFG-MSGOUT-RTCM_3X_TYPE1005_UART1 0x1  # write value 1  0x1   to item id 209102be
Flash CFG-MSGOUT-RTCM_3X_TYPE1005_UART1 0x1  # write value 1  0x1   to item id 209102be
  RAM CFG-MSGOUT-RTCM_3X_TYPE1074_UART1 0x1  # write value 1  0x1   to item id 2091035f
Flash CFG-MSGOUT-RTCM_3X_TYPE1074_UART1 0x1  # write value 1  0x1   to item id 2091035f
  RAM CFG-MSGOUT-RTCM_3X_TYPE1084_UART1 0x1  # write value 1  0x1   to item id 20910364
Flash CFG-MSGOUT-RTCM_3X_TYPE1084_UART1 0x1  # write value 1  0x1   to item id 20910364
  RAM CFG-MSGOUT-RTCM_3X_TYPE1124_UART1 0x1  # write value 1  0x1   to item id 2091036e
Flash CFG-MSGOUT-RTCM_3X_TYPE1124_UART1 0x1  # write value 1  0x1   to item id 2091036e
  RAM CFG-MSGOUT-RTCM_3X_TYPE1230_UART1 0x5  # write value 5  0x5   to item id 20910304
Flash CFG-MSGOUT-RTCM_3X_TYPE1230_UART1 0x5  # write value 5  0x5   to item id 20910304
  RAM CFG-MSGOUT-RTCM_3X_TYPE1005_USB 0x1    # write value 1  0x1   to item id 209102c0
Flash CFG-MSGOUT-RTCM_3X_TYPE1005_USB 0x1    # write value 1  0x1   to item id 209102c0
  RAM CFG-MSGOUT-RTCM_3X_TYPE1074_USB 0x1    # write value 1  0x1   to item id 20910361
Flash CFG-MSGOUT-RTCM_3X_TYPE1074_USB 0x1    # write value 1  0x1   to item id 20910361
  RAM CFG-MSGOUT-RTCM_3X_TYPE1084_USB 0x1    # write value 1  0x1   to item id 20910366
Flash CFG-MSGOUT-RTCM_3X_TYPE1084_USB 0x1    # write value 1  0x1   to item id 20910366
```

```
  RAM CFG-MSGOUT-RTCM_3X_TYPE1124_USB 0x1    # write value 1  0x1   to item id 20910370
Flash CFG-MSGOUT-RTCM_3X_TYPE1124_USB 0x1    # write value 1  0x1   to item id 20910370
  RAM CFG-MSGOUT-RTCM_3X_TYPE1230_USB 0x5    # write value 5  0x5   to item id 20910306
Flash CFG-MSGOUT-RTCM_3X_TYPE1230_USB 0x5    # write value 5  0x5   to item id 20910306
RAM CFG-MSGOUT-RTCM_3X_TYPE1094_UART1 0x1    # write value 1  0x1   to item id 20910369
Flash CFG-MSGOUT-RTCM_3X_TYPE1094_UART1 0x1  # write value 1  0x1   to item id 20910369
  RAM CFG-MSGOUT-RTCM_3X_TYPE1094_USB 0x1    # write value 1  0x1   to item id 2091036b
Flash CFG-MSGOUT-RTCM_3X_TYPE1094_USB 0x1    # write value 1  0x1   to item id 2091036b
  RAM CFG-MSGOUT-UBX_NAV_PVT_USB 0x1         # write value 1  0x1    to item id 20910009
Flash CFG-MSGOUT-UBX_NAV_PVT_USB 0x1         # write value 1  0x1    to item id 20910009
  RAM CFG-MSGOUT-UBX_NAV_SVIN_USB 0x1        # write value 1  0x1    to item id 2091008b
Flash CFG-MSGOUT-UBX_NAV_SVIN_USB 0x1        # write value 1  0x1    to item id 2091008b
Flash CFG-UART1-BAUDRATE   0x70800     # write value 460800  0x70800 to item id 40520001
  RAM CFG-UART1-BAUDRATE   0x70800     # write value 460800  0x70800 to item id 40520001
```

G F9P Rover config C99.txt

Copy all the text below this line into a text file named “F9P Rover config C99.txt.

```
# Config changes format version 1.0
# created by u-center version 18.11 at 11:16:51 on Tuesday, 27 Nov 2018
[del]
[set]
  RAM CFG-UART1INPROT-UBX   1        # write value 1                  to item id 10730001
Flash CFG-UART1INPROT-UBX   1        # write value 1                  to item id 10730001
  RAM CFG-UART1INPROT-NMEA 0         # write value 0                  to item id 10730002
Flash CFG-UART1INPROT-NMEA 0         # write value 0                  to item id 10730002
  RAM CFG-UART1INPROT-RTCM3X 1       # write value 1                  to item id 10730004
Flash CFG-UART1INPROT-RTCM3X 1       # write value 1                  to item id 10730004
  RAM CFG-UART1OUTPROT-UBX 1         # write value 1                  to item id 10740001
Flash CFG-UART1OUTPROT-UBX 1         # write value 1                  to item id 10740001
  RAM CFG-UART1OUTPROT-NMEA 1        # write value 1                  to item id 10740002
Flash CFG-UART1OUTPROT-NMEA 1        # write value 1                  to item id 10740002
  RAM CFG-UART1OUTPROT-RTCM3X 0      # write value 0                  to item id 10740004
Flash CFG-UART1OUTPROT-RTCM3X 0      # write value 0                  to item id 10740004
  RAM CFG-USBINPROT-UBX    1         # write value 1                  to item id 10770001
Flash CFG-USBINPROT-UBX    1         # write value 1                  to item id 10770001
  RAM CFG-USBINPROT-NMEA   1         # write value 1                  to item id 10770002
Flash CFG-USBINPROT-NMEA   1         # write value 1                  to item id 10770002
  RAM CFG-USBINPROT-RTCM3X 1         # write value 1                  to item id 10770004
Flash CFG-USBINPROT-RTCM3X 1         # write value 1                  to item id 10770004
  RAM CFG-USBOUTPROT-UBX   1         # write value 1                  to item id 10780001
Flash CFG-USBOUTPROT-UBX   1         # write value 1                  to item id 10780001
Flash CFG-USBOUTPROT-NMEA  1         # write value 1                  to item id 10780002
  RAM CFG-USBOUTPROT-RTCM3X 0        # write value 0                  to item id 10780004
Flash CFG-USBOUTPROT-RTCM3X 0        # write value 0                  to item id 10780004
Flash CFG-UART1-BAUDRATE   0x70800  # write value 460800  0x70800    to item id 40520001
  RAM CFG-UART1-BAUDRATE   0x70800  # write value 460800  0x70800    to item id 40520001
```

H C099-F9P antenna specification

H.1 Wi-Fi/Bluetooth antenna specification

EX-IT WLAN RPSMA / Ex-IT WLAN SMA	
Manufacturer	ProAnt
Type	½ wave dipole dual-band antenna
Polarization	Vertical
Gain	+3 dBi
Impedance	50 Ω
Size	107 mm (straight)
Type	Monopole
Connector	• Reverse polarity SMA plug (inner thread and pin receptacle)

	• SMA plug (inner thread and pin)
Comment	To be mounted on the U.FL to SMA or reverse polarity SMA adapter cable
Approval	FCC, IC, RED, MIC, NCC, KCC*, ANATEL, and ICASA

Table 5: Wi-Fi/Bluetooth antenna

☞ The variant included in the C099-F9P kit has an SMA connector and has to be mounted on the corresponding antenna connector of the C099-F9P board if you wish to use Wi-Fi or Bluetooth connectivity.

I ODIN-W2 firmware upload via JTAG

ODIN-W2 firmware can be uploaded through the 10-pin JTAG connector by using the STM Link Utility software and ST LINK V2 debugger device. STM Link Utility software can be found on

https://www.st.com/en/development-tools/stsw-link004.html

Check the availability of the ST LINK V2 debugger device with local STM distributors.