# C099-F9P

## Application board (rev. E)

User guide


**Abstract**

The C099-F9P board enables customers to evaluate RTK operation with the ZED-F9P high precision GNSS receiver. The board provides short-range wireless connection via Bluetooth® or Wi-Fi for receiving correction data and logging via wireless connectivity.

# Document information

| | | |
|---|---|---|
| **Title** | **C099-F9P** | |
| **Subtitle** | Application board (rev. E) | |
| **Document type** | User guide | |
| **Document number** | UBX-18063024 | |
| **Revision and date** | R16 | 24-Nov-2022 |
| **Disclosure restriction** | C1-Public | |

This document applies to the following products:

| Product name | Type numbers | PCN reference | Product status |
|---|---|---|---|
| C099-F9P | C099-F9P-0-03<br>C099-F9P-1-03 | N/A | Mass production |

☞ It is recommended to check the ZED-F9P firmware version on the C099-F9P Application board, and update to the last released version if needed, for more details on ZED-F9P firmware update procedure see section 7.1.

# Contents

# 1 Introduction

The C099-F9P board is a convenient tool that allows customers to become familiar with the u-blox ZED-F9P high precision GNSS module. The board provides facilities for evaluating the product and demonstrating its key features. The C099-F9P application board offers:

- A ZED-F9P module for use as an RTK rover or reference station
- An ODIN-W2 short-range module with Arm® Mbed™ firmware[1] tailored for C099-F9P use to provide untethered operation using Bluetooth and Wi-Fi which are configurable via a command line interface (CLI)[2]
- Power supply options comprising a USB connection, Li-Po (lithium polymer) cell with recharging ability, and 5-12 V DC input[3]
- Small, lightweight board (110 x 55 mm) with Arduino R3/Uno shield connections for host expansion

This user guide describes the following use cases:

1. Base and rover operation via serial connectivity
2. Rover operation via Bluetooth Classic (with ODIN-W2 Mbed FW)
3. Rover operation via Wi-Fi (with ODIN-W2 Mbed FW)
4. Base and rover operation via Wi-Fi (with ODIN-W2 Mbed FW)

This user guide is split into several useful sections, including:

- Section 2: C099-F9P quick start provides information on how to get C099-F9P up and running straight out of the box.
- Section 3: C099-F9P description identifies the board's facilities.
- Section 4: Using C099-F9P provides a comprehensive guide for in-depth usage.
- Section 5: Rover operation using NTRIP shows different ways of connecting to an NTRIP service.
- Section 6: Wireless communication describes the use case of connecting base and rover boards.
- Section 7: Firmware update provides instructions for updating the firmware of the ZED-F9P high precision GNSS module as well as the ODIN-W2 short-range module.
- Section 8: Arduino header connections provide mechanical specifications for Arduino R3/Uno.
- The sections in the Appendix provide information on how to upload the ODIN-W2 firmware via JTAG, C099-F9P antenna schematics, and C099-F9P mechanical board dimensions and schematics.

[1] The Mbed FW shall be used only with a C099-F9P kit.
[2] S-center usage is not required nor supported by the CLI.
[3] The C099-F9P kit does not contain a battery or mains power adapter.

## 1.1 Package contents

The delivered package contains:

- C099-F9P board (rev. E)
- u-blox ANN-MB-00 multi-band GNSS antenna and ground plane
- Wi-Fi/Bluetooth antenna
- USB interconnect cable
- Quick start guide
- USB-to-DC plug adapter cable

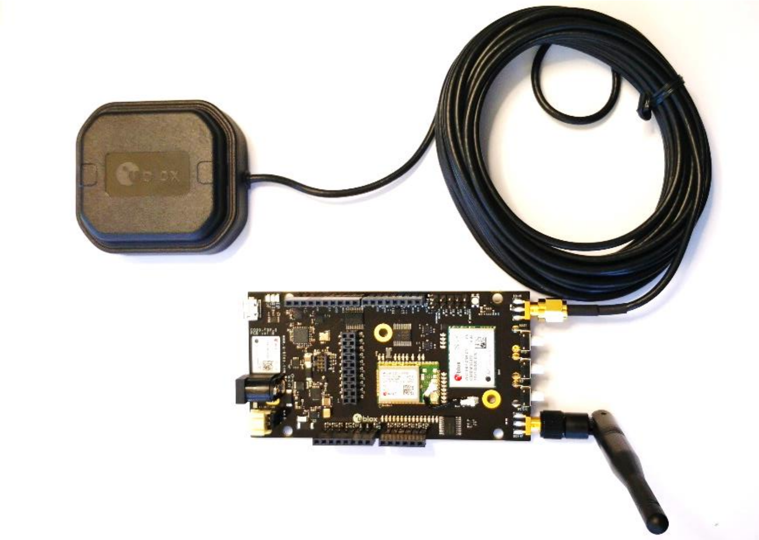

**Figure 1: C099-F9P board and antennas**

## 1.2 Additional sources of information

Prior to using the board, it is useful to download the appropriate evaluation software and keep handy the documents listed in the Related documents section.

# 2 C099-F9P quick start

This section provides some quick steps to enable ZED-F9P operation before exploring the more complex configurations described later.

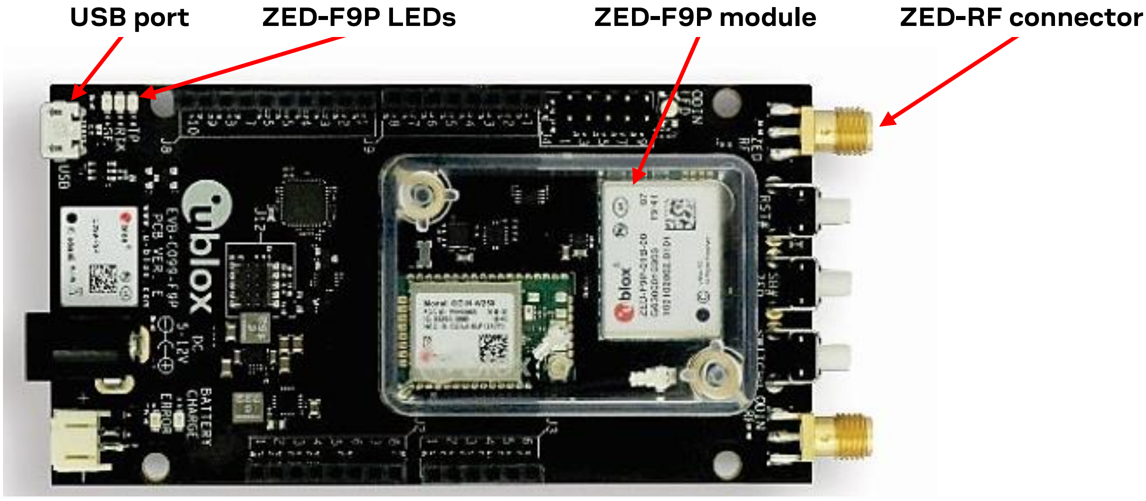


**Figure 2: Basic C099-F9P overview with details needed for quick start**

## 2.1 Starting up

- Connect the supplied multi-band GNSS antenna to the ZED-RF SMA connector. Ensure good signal reception.
- To power the board, connect the USB to a Windows PC. The FTDI and USB drivers are installed automatically[4] from Windows Update when the user connects the board for the first time. Note that the board has current limitation functionality on USB. Thus, ZED-F9P and ODIN-W2 modules power up after the drivers have been successfully installed and the USB enumeration is completed.
- Start u-center and connect to the COM port identified as **C099 application board, ZED-F9P** using Device Manager. Set the baud rate to 460800 baud. See section 4.3.1 for detailed instructions.
- The time pulse LED on the C099-F9P board blinks in blue once the ZED-F9P has obtained valid time information. Figure 3 below shows a typical u-center view with active satellite signal levels.

To operate the ZED-F9P in RTK mode, the GNSS antenna must be placed in an open environment and the unit must be connected to an RTK correction service. Where available, the evaluation kit comes with a free trial of the SmartNet correction service. Consult the leaflet included with the kit for information on how to register for the service and how to obtain mount point and user connection details before moving to the next steps.

RTK corrections can be applied using a u-center built-in NTRIP client. To use the C099-F9P board with a correction service follow these next steps:

- In u-center, click on the **Receiver** menu item.
- Select **NTRIP Client**…
- Fill in the settings for the **NTRIP caster**, **username** and **password**.
- Click **Update source table** and select the recommended NTRIP mount point.
- Click **OK** to close the dialog and connect to the service.
- In the **Data View** of u-center, the **Fix Mode** should change from **3D** to **3D/DGNSS** when RTCM corrections are received. The RTK LED blinks in green.

[4] For manual driver installation, check GNSS Sensor and VCP Device Driver User guide in Related documents

- Eventually, the status changes to **3D/DGNSS/FIXED** and the RTK LED shows a steady green light.

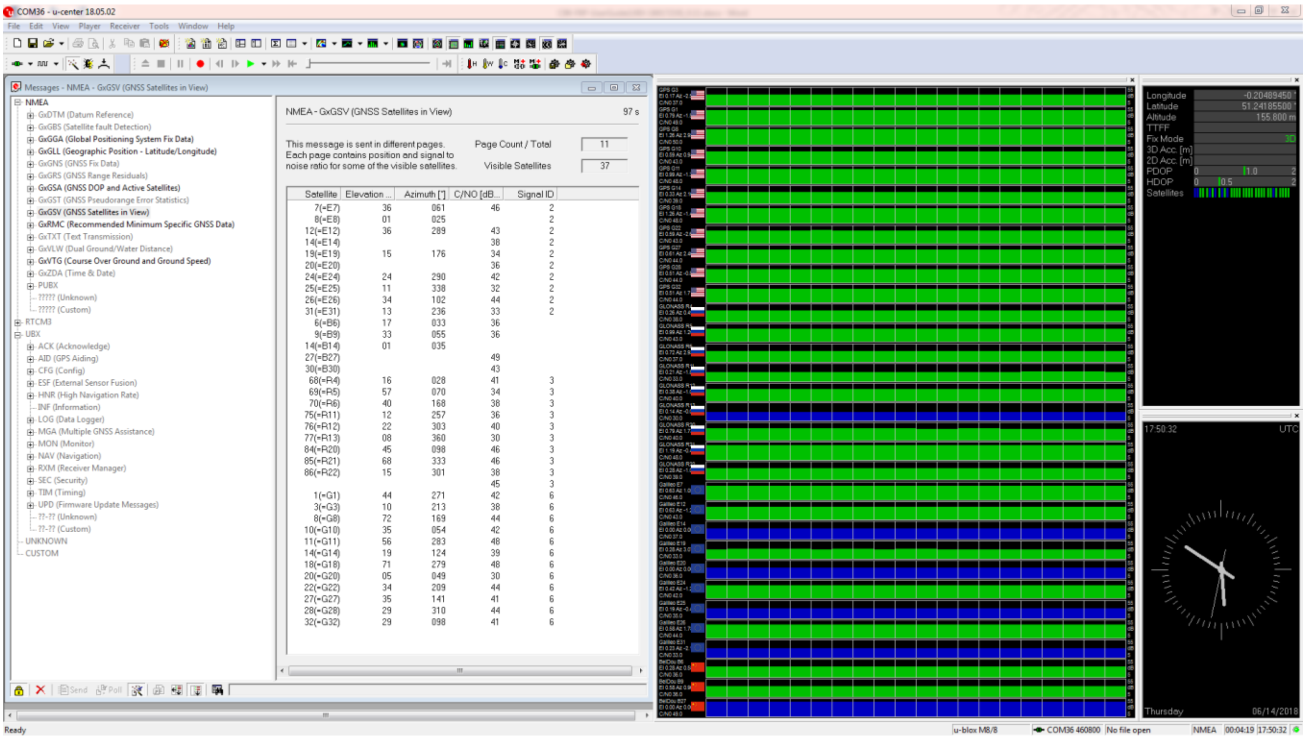

**Figure 3: u-center showing a view of the ZED-F9P default operation**

# 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.

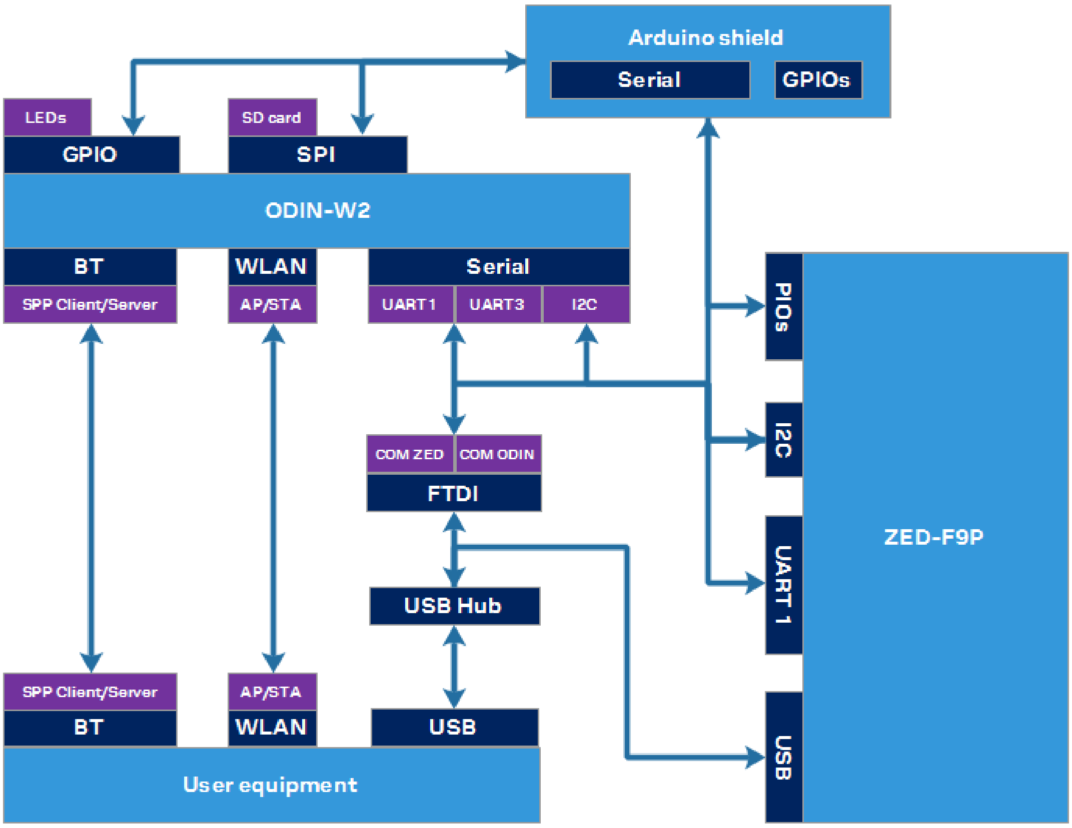


**Figure 4: C099-F9P block diagram**

## 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.

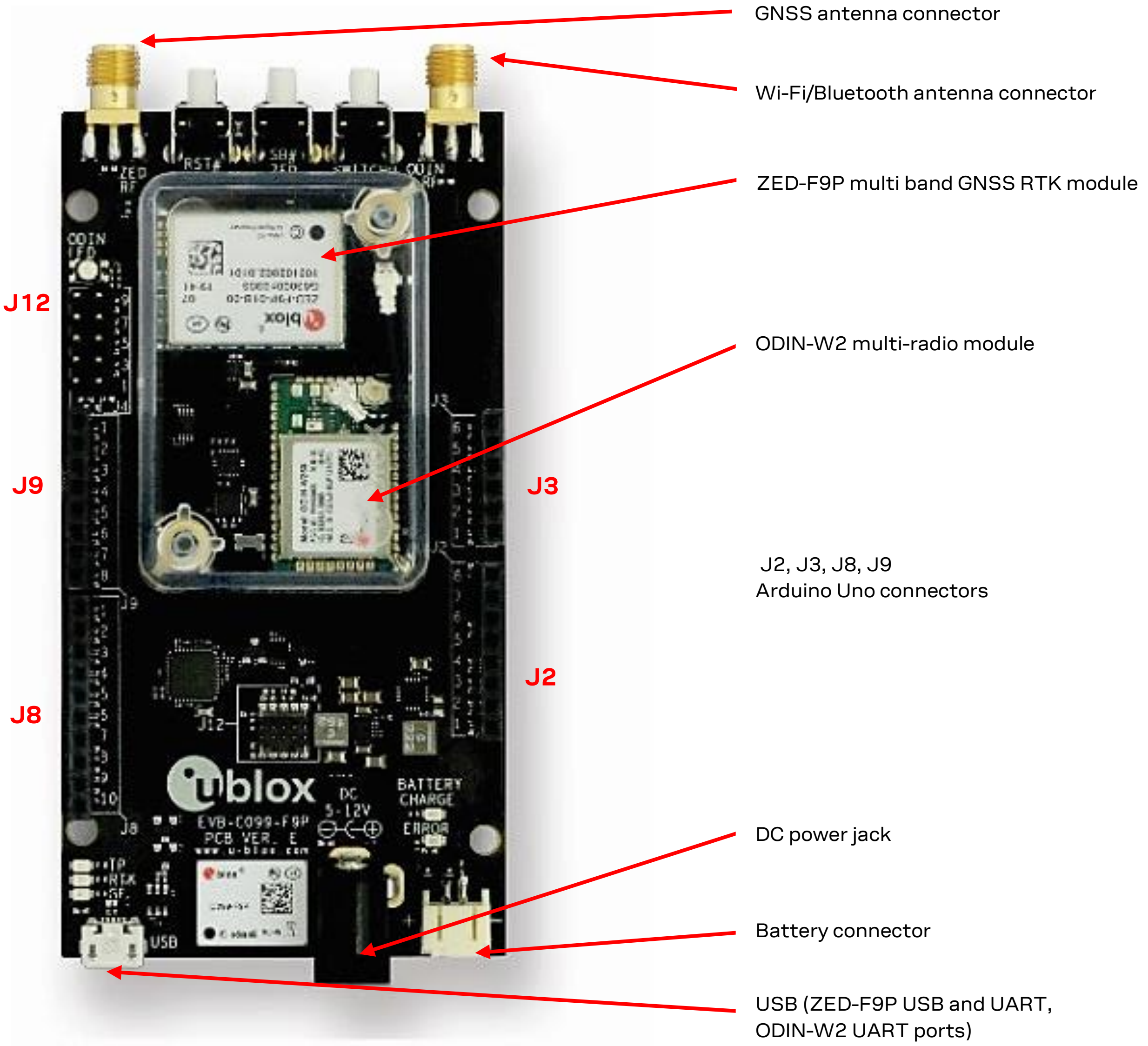


**Figure 5: Main components and USB ports**

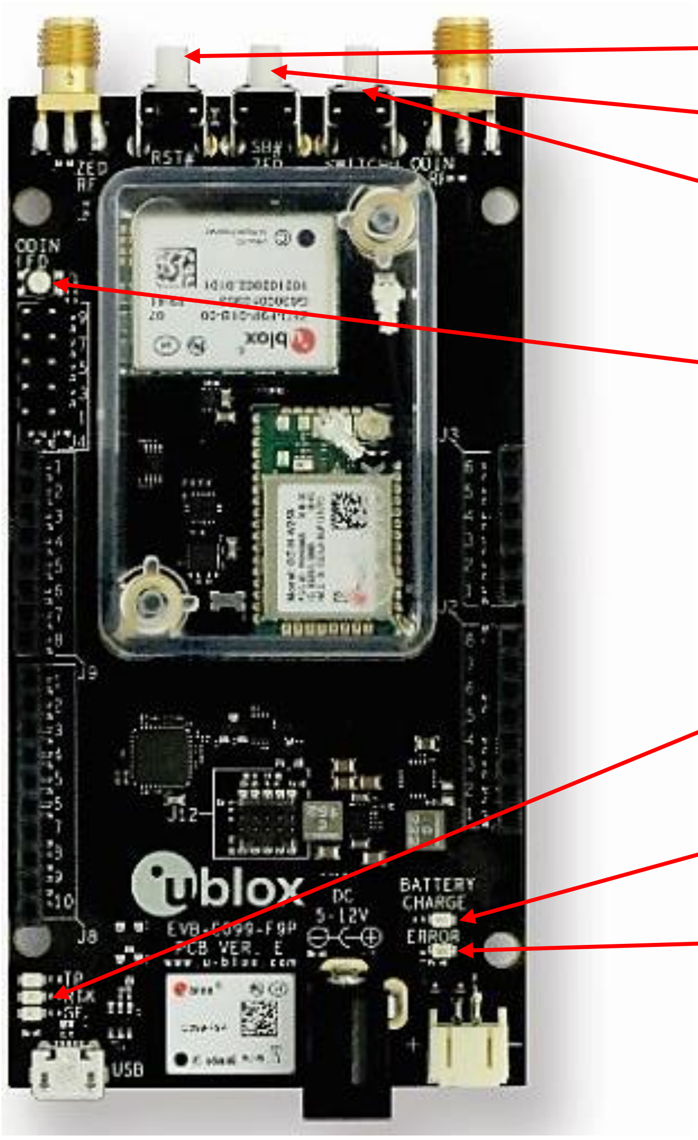


**Figure 6: Switches and LEDs**

☞ 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.


**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**


**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.

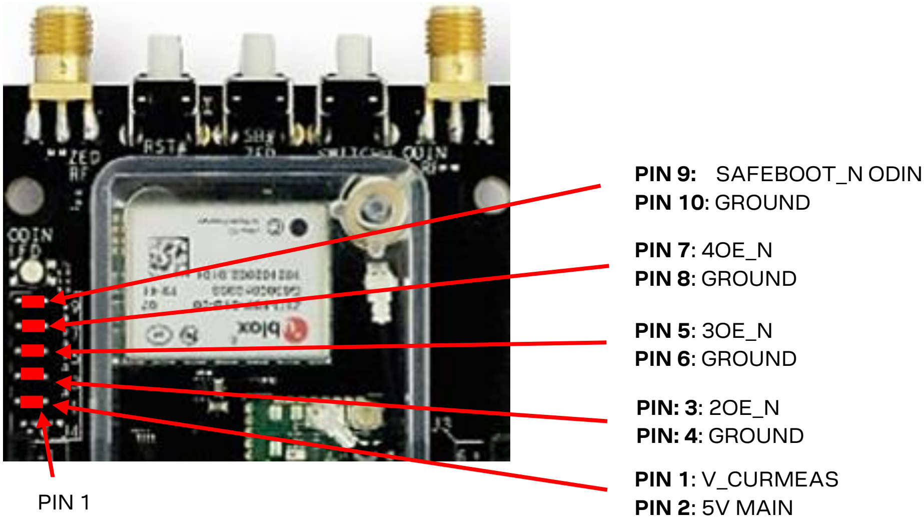


**Figure 9: C099-F9P Jumper PIN details**

| 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.

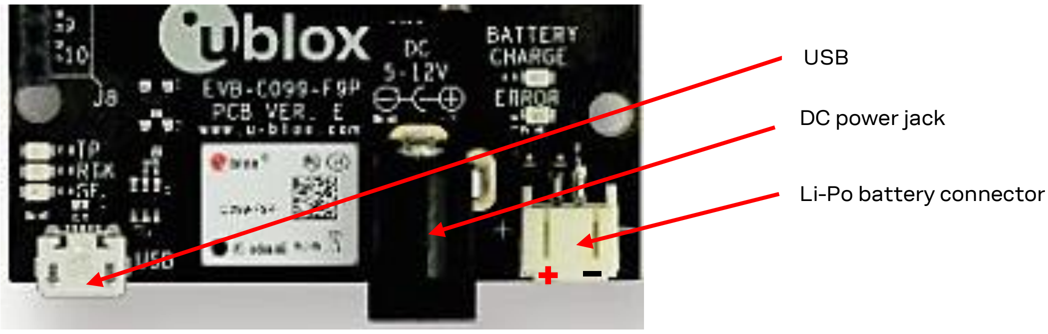


**Figure 10: Power connections**

**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.

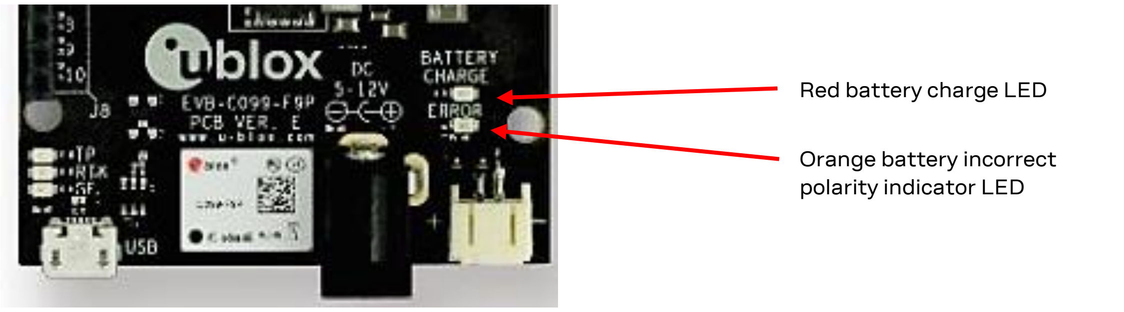


**Figure 10: Battery charge status LED**

### 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.

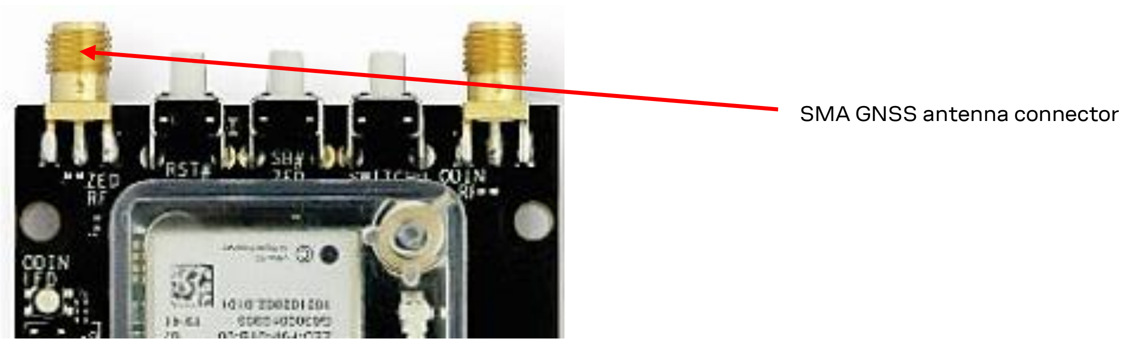


**Figure 11: GNSS antenna connector**

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.

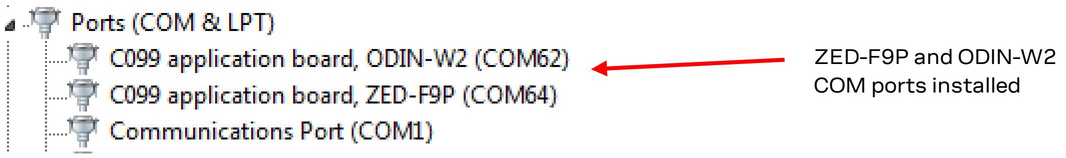


**Figure 13: Windows Device Manager COM port view**

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.

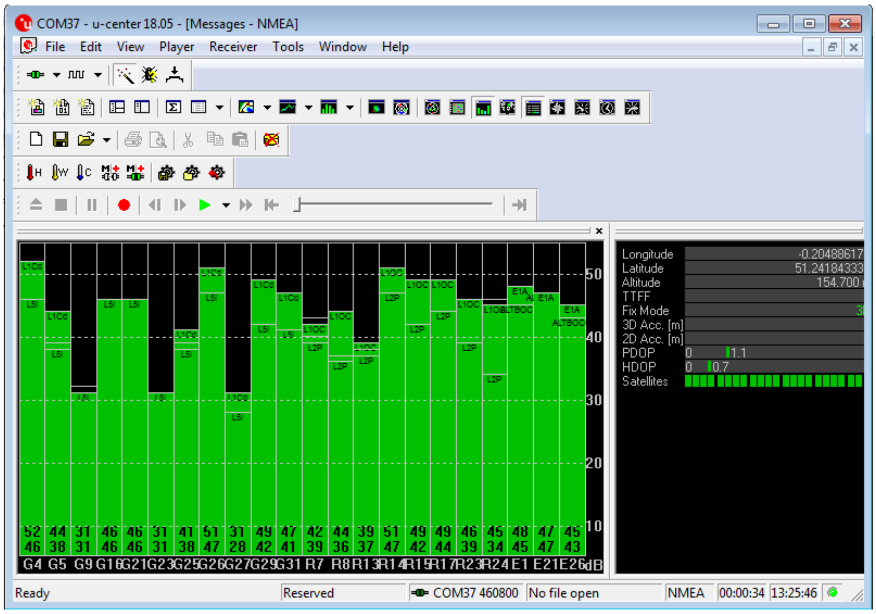


**Figure 14: u-center view with ZED-F9P connected**

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

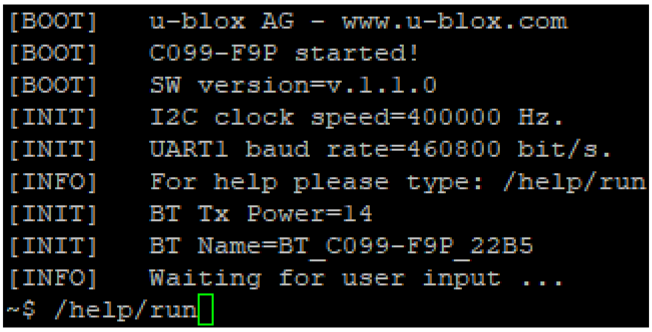


**Figure 15: CLI help command**

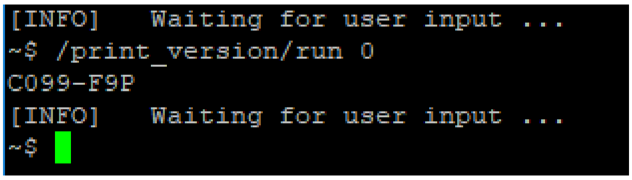


**Figure 16: Example RPC syntax**

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.

# 5 Rover operation using NTRIP

This section shows how ZED-F9P is used as a rover using correction information provided over the internet using NTRIP. This is usually provided by a host from a single reference station or as a Network RTK Virtual Reference Service (VRS).

A suitable host is a PC with internet access. A host runs an NTRIP client and streams RTCM corrections to C099-F9P through a UART or Bluetooth connection. Messages transmitted through a Bluetooth or Wi-Fi link are forwarded to I2C bus and vice versa. The user is advised to enable desired messages in both UART and I2C interfaces in ZED-F9P.

## 5.1 PC hosting via u-center

The u-center application includes an NTRIP client for PC hosting. The u-center User guide [3] provides help when setting NTRIP service connections. Users can connect via Bluetooth for wireless operation or directly via a serial COM port. Once the service is active, RTCM corrections are sent over the connection and data can be logged as usual with u-center.

The u-center User guide [3] provides more information concerning NTRIP connections. Enter the required connection settings using the client setting window shown below.

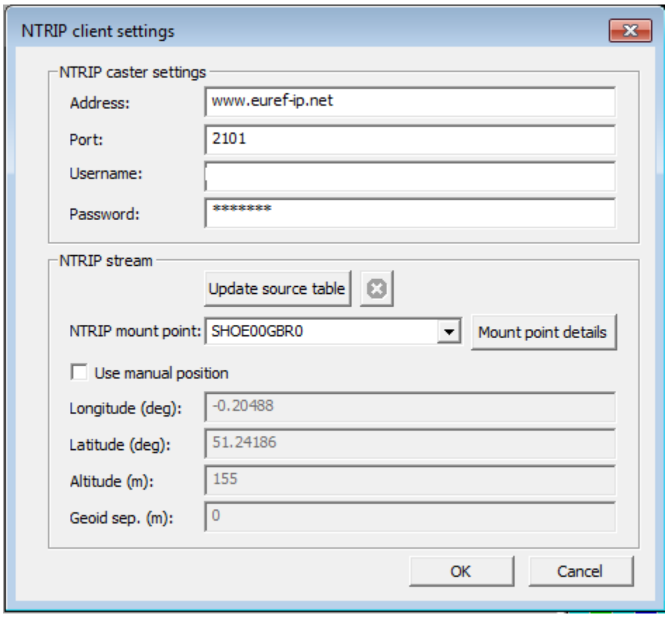


**Figure 17: u-center NTRIP client view**

Ensure that the NTRIP client connection icon is green. This indicates a successful NTRIP connection and that RTCM data is transferred to C099-F9P.

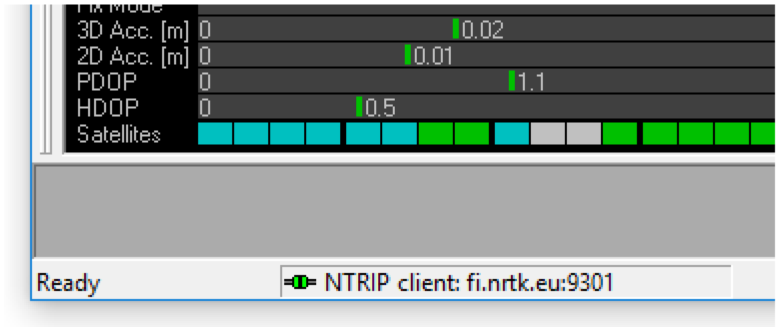


**Figure 20: u-center NTRIP Client connection icon in the status bar of u-center**

Confirm that the rover has obtained RTK fixed mode in the u-center Data view:

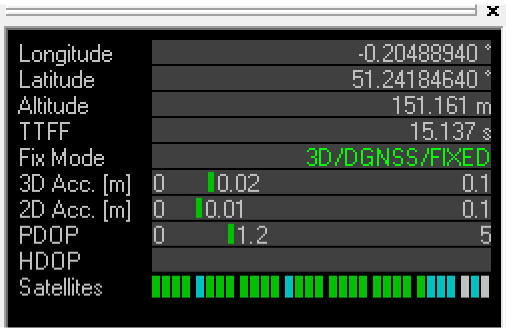


**Figure 18: u-center Data view RTK FIXED indication**

## 5.2 Mobile hosting

A portable rover option is offered by an Android application that utilizes a Bluetooth connection to a single C099-F9P. An example application is provided by Lefebure and it is available from Google Play Store: https://play.google.com/store/apps/details?id=com.lefebure.ntripclient.

The application integrates an NTRIP client that forwards corrections received from a cellular or a wireless network to the Bluetooth interface. In addition, the application logs the incoming NMEA messages from C099-F9P into the phone memory.

Prior to usage, the following steps are required:

1. Pair an Android phone with C099-F9P (see section 6.1.1 ODIN-W2 as the pairing responder).
2. Insert the necessary credentials for the NTRIP service through the application settings.
3. Configure the desired NMEA messages to I2C interface in ZED-F9P via the UART 1 interface.


**Figure 19: Lefebure Android NTRIP client**

# 6 Wireless communication

## 6.1 Bluetooth pairing

Prior to operation, the user is requested to pair ODIN-W2 with a host device. Pairing is the process for creating one or more shared secret keys and is required only once for a pair of devices. ODIN-W2 can be paired with one of two alternatives:

1. The host initiates, ODIN-W2 responds.
2. ODIN-W2 initiates, the host responds.

### 6.1.1 ODIN-W2 as the pairing responder

- Once you have verified that the terminal connection is available, use the following command to make the ODIN-W2 visible and connectable: `/bt_visible/run`
- ODIN-W2 acknowledges a successful reception of the command and informs once it is ready to respond to a pairing request.
- Next, perform a Bluetooth scan to find the C099-F9P board. Every C099-F9P has a predefined, unique Bluetooth name in BT_C099-F9P_XYZW format as shown in Figure 20.

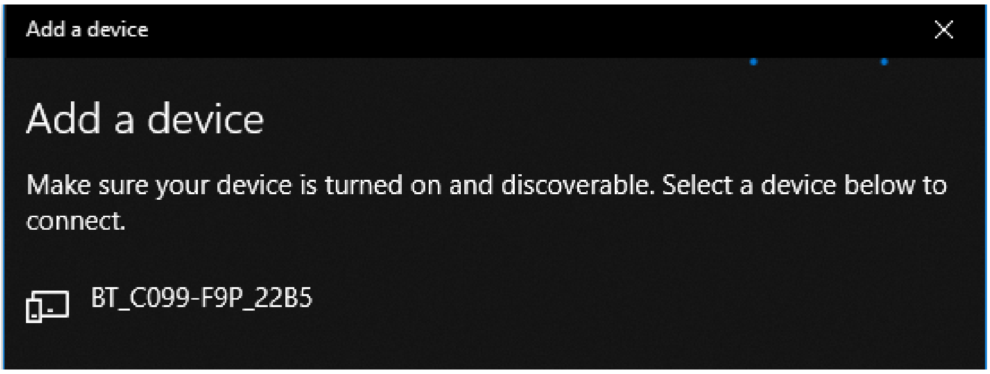


**Figure 20: Windows 10 menu for adding a Bluetooth device**

### 6.1.2 ODIN-W2 as the pairing initiator

- Enable Bluetooth visibility on the host device.
- Once you have verified a working CLI connection, use the following command to scan any nearby Bluetooth devices: `/bt_inquiry/run`
- C099-F9P lists the nearby Bluetooth devices, their Bluetooth addresses ("MAC") and the corresponding RSSI values. To ensure sufficient radio link quality, check that the RSSI level of the host device is well above -80 dBm. Weak signal levels can result in connection losses and limited range.
- Once C099-F9P has found the host device, the following command starts the pairing process: `/bt_bond/run <MAC address>`
- ODIN-W2 waits until the user has accepted the pairing request on the host device. Note that the pairing request fails if an internal timeout is reached. Typically, you can accept an incoming pairing request in the host Bluetooth menu.
- Finally, the host and ODIN-W2 permanently store their exchanged link keys for future connections.

## 6.2 Bluetooth serial port

C099-F9P supports incoming and outgoing Bluetooth serial connections. To find the corresponding Bluetooth COM ports, refer to Bluetooth options indicated in Figure 21.

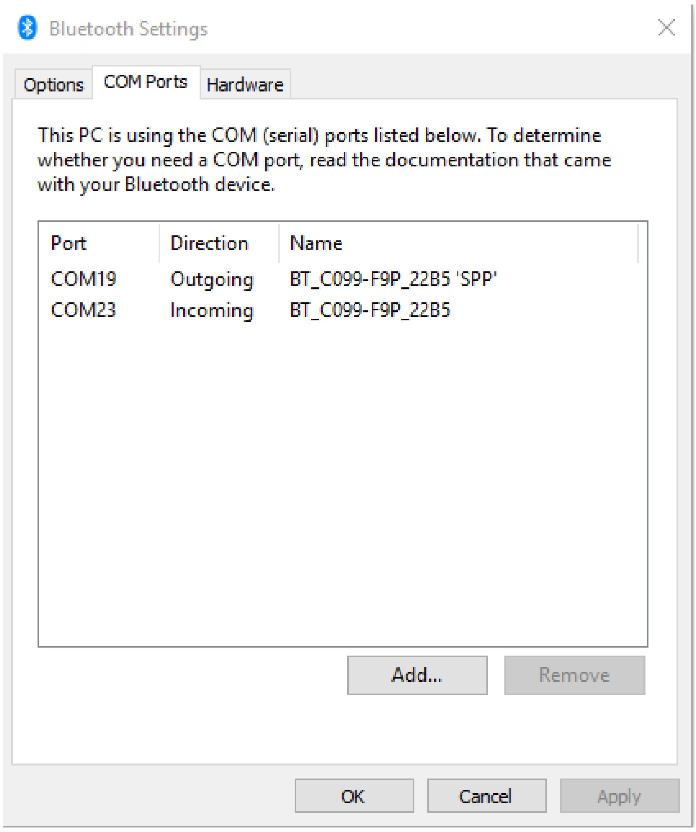


**Figure 21: Bluetooth COM ports**

Typically, Windows hosts automatically set the corresponding COM ports if the pairing process was initiated at the host, as described in 6.1.1 ODIN-W2 as the pairing responder. Often, the user is requested to add incoming and outgoing ports manually if the pairing process was initiated at C099-F9P, as described in 6.1.2 ODIN-W2 as the pairing initiator.

### 6.2.1 Server SPP connection

- To use the server port (incoming port) on the host PC, connect to the incoming COM port in u-center prior to the CLI command on the C099-F9P: /bt_sppcli/run <MAC address>
- After a successful connection, C099-F9P starts to stream data from ZED-F9P to the Bluetooth COM port. Note that you can ignore the baud rate of the Bluetooth serial port on the host PC.

### 6.2.2 Client SPP connection

- To use the outgoing port (client port) on the host PC, set C099-F9P to server mode by issuing the following command: /bt_visible/run
- After selecting the client port (outgoing port) in u-center, C099-F9P will be automatically requested to open a data stream between ZED-F9P and the Bluetooth COM port. Note that u-center has default COM port behavior resulting in connection failures or non-listed outgoing Bluetooth COM ports. As a workaround, it is recommended to change the default COM port enumeration in u-center as shown in Figure 22.

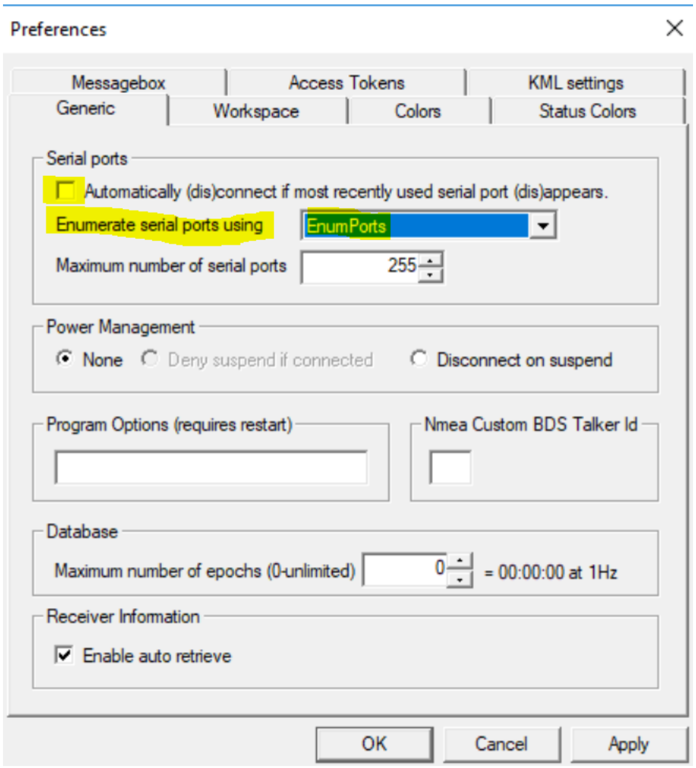


**Figure 22: u-center COM port enumeration**

- To force C099-F9P to start in Bluetooth SPP server mode on the next device restart, use the following CLI command: `/mem_store/run bt 1`

You can later revert to the default startup settings by erasing the memory content, which is described in section 4.4.1 Revert to factory default.

# 6.3 Wi-Fi connectivity

C099-F9P can be operated in Wi-Fi mode to enable longer communication range, higher wireless link throughput, and interconnection between base and rover boards. The on-board ZED-F9P and ODIN-W2 modules are interconnected via I2C bus, as in the Bluetooth operation. Hence, ensure that the desired ZED-F9P messages are enabled for the I2C interface.

| | Base | Rover |
|---|---|---|
| Wi-Fi access point / UDP server | ✓ | ✓ |
| Wi-Fi station / UDP client | N/A | ✓ |

**Table 3: ODIN-W2 Wi-Fi modes**

## 6.3.1 Wi-Fi access point and UDP server

The C099-F9P RTK base can be set to operate as a Wi-Fi access point and UDP server to deliver RTCM corrections via a Wi-Fi link. For rover operation, C099-F9P can be configured either to Wi-Fi STA or Wi-Fi AP mode. The latter configuration is suitable for a single rover connected to a u-center UDP client. Refer to section 4.3.2 Command line interface of ODIN-W2 to see the required terminal settings for the command line interface.

### 6.3.1.1 Base operation in Wi-Fi AP mode

Follow the steps below to configure ODIN-W2 in Wi-Fi AP mode and to redirect incoming rover data to the ODIN-W2 UART1 port (remote logging) as depicted in Figure 23. The UART1 and USB ports on the rover ZED-F9P remain as optional logging interfaces.

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.

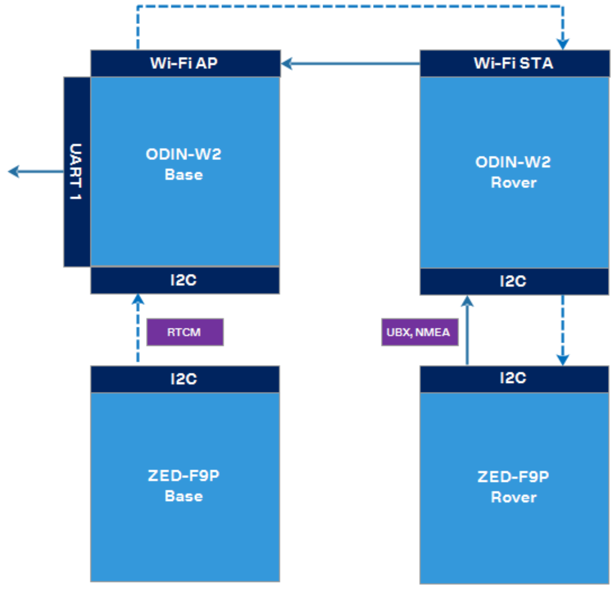


**Figure 23: Wi-Fi base and rover setup**

### 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.

### 6.3.2 Wi-Fi Station and UDP Client

#### 6.3.2.1 Rover operation in Wi-Fi STA mode

Typically, the Wi-Fi STA mode is applicable when two C099-F9Ps (base and rover) interconnect via a Wi-Fi link. Firstly, it is recommended to configure the base as instructed in section 6.3.1.1 Base operation in Wi-Fi AP mode. Secondly, the rover C099-F9P is set up to function in Wi-Fi STA and rover mode:

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

   ```
   /mem_store/run wifi_sta
   ```

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

   ```
   /mem_store/run rover
   ```

3. Restart C099-F9P to apply the Wi-Fi STA settings.
4. The rover C099-F9P automatically connects to the C099-F9P base.

The AP and STA use the default pre-stored SSID “C099-F9P”. You can set a new SSID and read the current SSID with the following commands:

1. Read the current SSID setting:

   ```
   /wifi_getssid/run
   ```

2. Set and store a new SSID:

   ```
   /wifi_setssid/run <your_SSID>
   ```

☞ Wi-Fi connectivity between base and rover requires a matching SSID.

## 6.4 Host UDP client

### 6.4.1 Client UDP connection

Follow these steps to start monitoring the ZED-F9P output and to feed in RTCM correction data:

1. Navigate to **Receiver** > **Connection**> **Network Connection** menu in u-center and connect to C099-F9P via a UDP client socket:

   ```
   udp://192.168.0.1:5555
   ```

2. After a successful UDP connection, the NTRIP connection can be started as described in section 5.1. RTCM messages are automatically forwarded to the active UDP socket by selecting the **Current connection** option in the NTRIP menu.

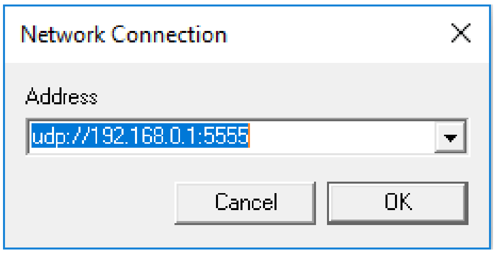


**Figure 24: UDP client connection**

## 6.5 Wireless link limitations

### 6.5.1 Data throughput

The system throughput of the Bluetooth and Wi-Fi links is dominated by the effective I2C and Bluetooth SPP or Wi-Fi data rates, respectively. It is recommended to limit the average byte load from ZED-F9P to 17 kB/sec. The following examples approximate the output load of the default configuration of the ZED-F9P firmware:

- 1 Hz navigation rate: NMEA, UBX-NAV-RELPOSNED, UBX-NAV-PVT enabled < 2.4 kB/sec
- 5 Hz navigation rate: NMEA, UBX-NAV-RELPOSNED, UBX-RXM-RTCM enabled < 17 kB/sec
- 10 Hz navigation rate: NMEA, UBX-RXM-RTCM enabled < 14 kB/sec

ODIN-W2 outputs an error message when the I2C bus load is too high. In such situations, some messages may get dropped. To avoid this, it is recommended to adjust the enabled messages in the ZED-F9P I2C interface.

### 6.5.2 Link loss

#### 6.5.2.1 Bluetooth Classic

During a Bluetooth transmission failure (red LED blinking), check the system for typical root causes:

- Bluetooth SPP COM port on the host device stalled or disconnected.
- Insufficient signal quality between the host device and C099-F9P.

Recover the system by restarting ODIN-W2. To do this, press the reset button.

If ZED-F9P is configured through a Bluetooth link, e.g. using UBX-CFG messages, it is recommended to apply all configurations manually in u-center. Uploading a large configuration file may fail due to the limited link bandwidth.

#### 6.5.2.2 Wi-Fi 2.4 GHz

A Wi-Fi disconnection is reported by a CLI message from the ODIN-W2 UART1 interface and the LED indication (yellow LED). If the disconnection is not intended, check the Wi-Fi interface on the host PC.

Wi-Fi channel congestion can be avoided by changing the Wi-Fi AP channel on C099-F9P. Typically, channel congestion is experienced when the Wi-Fi connection indicators (e.g. LEDs) are OK but no data is received. Use the following commands to set and read the current channel stored in the ODIN-W2 RAM:

1. `/wifi_setch/run <integer number 1-11>`
2. `/wifi_getch/run`

Then, continue with the Wi-Fi connectivity settings, refer to section 6.3 Wi-Fi connectivity. Ensure C099-F9P is restarted to apply the new channel.

### 6.5.3 Windows OS issues with Bluetooth SPP

There are some known issues with the Windows Bluetooth serial port profile (SPP) implementation for Windows 7-10. The symptoms include the Bluetooth Virtual COM port not installing or applications not connecting to the Bluetooth virtual COM port. In other cases, Windows might crash or become unresponsive. This is not related to the ODIN-W2 Bluetooth implementation that uses the Bluetooth standard SPP.

**Figure 25: ASUS USB-BT400**

A known industry fix is to not use the Windows Bluetooth stack and PC Bluetooth hardware. This is done by using a USB Bluetooth adapter that uses its own Bluetooth stack. A device that is known to work is the ASUS USB-BT400 (USB 2.0). Once installed, use the Bluetooth virtual COM port assigned to this device and not the built-in Bluetooth interface.

# 7 Firmware update

This section shows how to update the GNSS and Wi-Fi/Bluetooth modules' firmware, if required.

The board is delivered with the latest versions of firmware running on the ZED-F9P and ODIN-W2 modules. However, newer versions may become available during the lifetime of the product.

## 7.1 ZED-F9P firmware update

This section shows how to update the firmware and re-enable the configuration settings required for C099-F9P. There are two possible serial communication channels to update ZED-F9P: the UART1 and USB2.0 ports.

1. To update ZED-F9P, connect to u-center via USB to the COM port identified as ZED-F9P and poll MON-VER to view the installed firmware: see Figure 13 for the Device Manager COM port view. To download new firmware, follow the sequence detailed below.

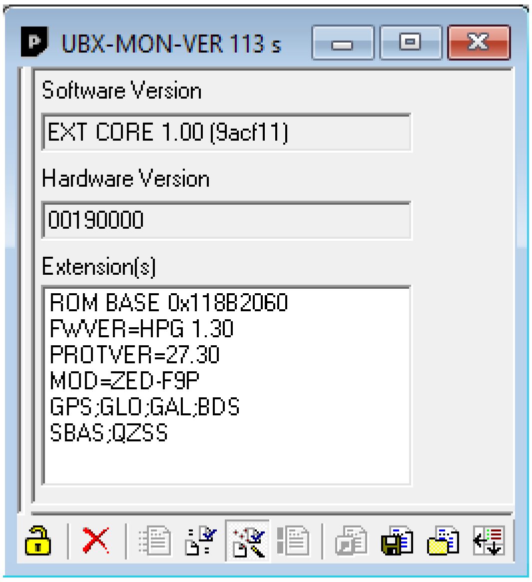


**Figure 26: MON-VER poll response for a ZED-F9P**

2. To begin updating the firmware, select **Tools** > **Firmware Update**…

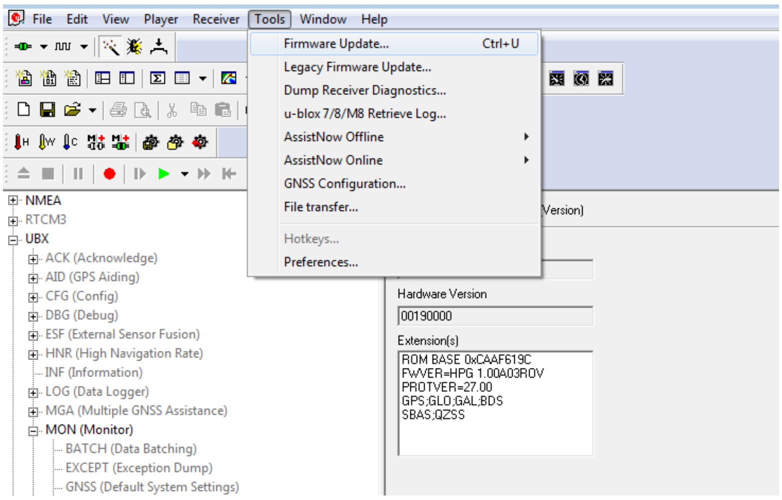


**Figure 30: Selecting u-center Firmware Update mode**

The following **Firmware image** update window appears:

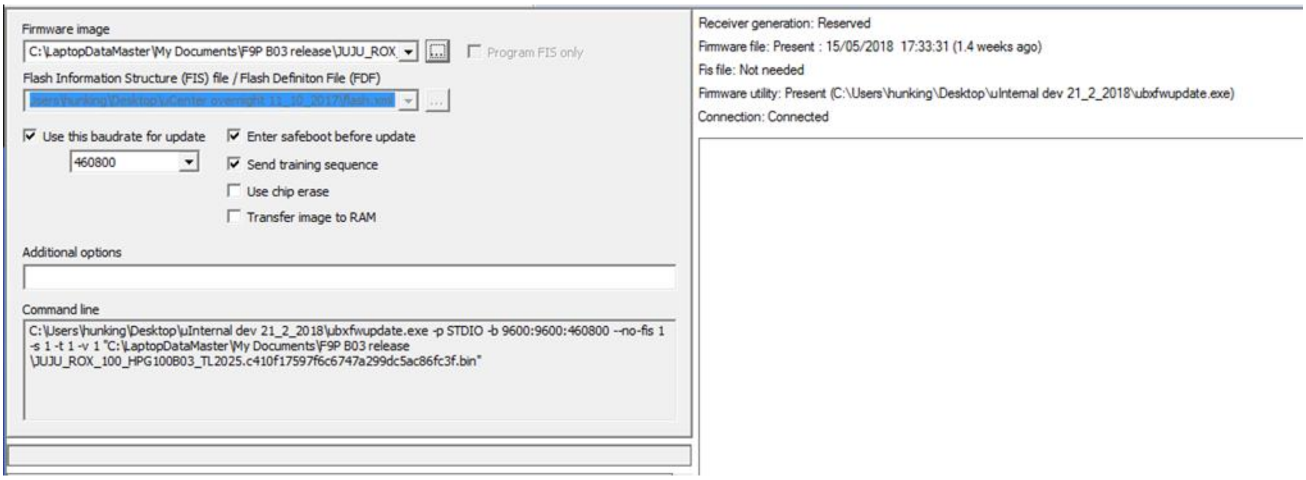


**Figure 27: Selecting u-center Firmware image folder**

3. At the top is the **Firmware image** file selection window. Click the button on the right of the window. This allows you to select the folder and file. Select the new firmware image bin file.
4. Select **Enter safeboot before update** and **Send training sequence**. Select the **Use this baudrate for update** option and select e.g. 460800 from the drop-down list. This is shown in Figure  below.

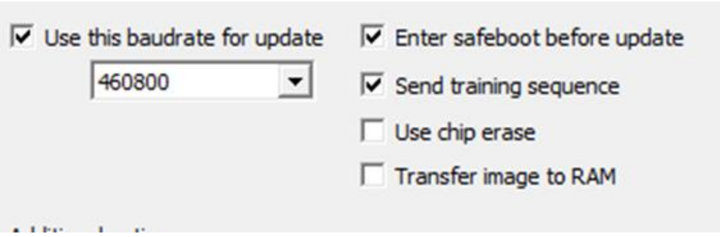


**Figure 28: Setting the required baud rate, safeboot and training sequence options**

5. Click the **GO** button in the bottom-left corner of the window to begin the download.

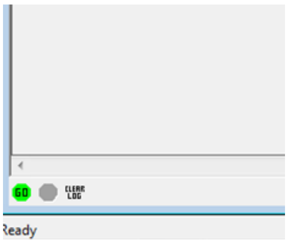


**Figure 29: Click GO for firmware update**

The firmware update progress is shown adjacent to the input window.

When programming is complete, the module starts up in a default configuration, in which the ZED-F9P serial port is set to 38400 baud. Change this to 460800 baud to provide sufficient data bandwidth and work correctly with the ODIN-W2 module. To make the baud rate change persistent, make the selections shown in Figure .

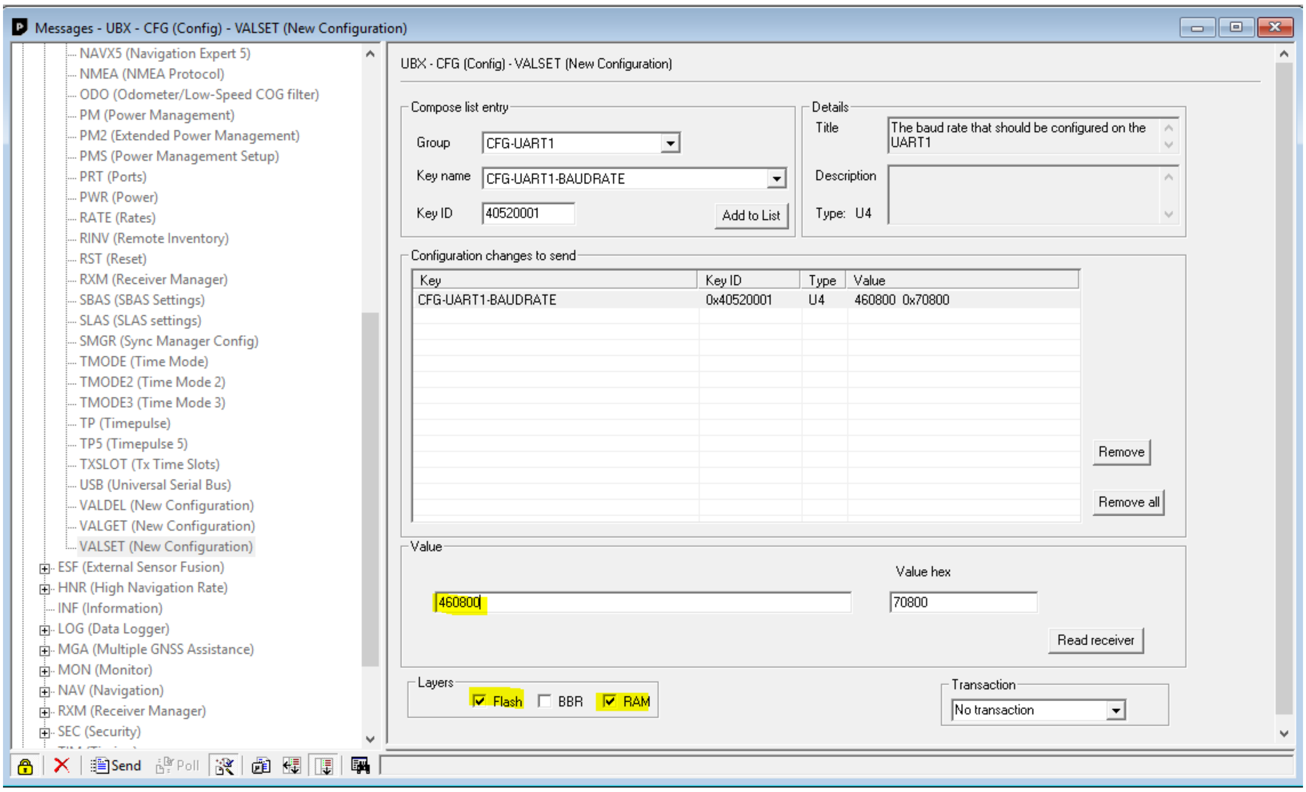


**Figure 30: Setting ZED-F9P UART1 back to 460800 baud and saving it to flash memory**

## 7.2 ODIN-W2 firmware update

Users have a choice to run two distinct firmware variants in ODIN-W2. By factory default, the ODIN-W2 module in C099-F9P runs dedicated Mbed application firmware.

### 7.2.1 Mbed OS 3 application firmware

The latest released binary is available via the u-blox GitHub repository:

https://github.com/u-blox/ublox-C099_F9P-mbed-3

You can update the firmware on ODIN-W2 with the following toolset:

- Through ODIN-W2 UART1 by using stm32flash.exe

1. It is recommended to download the stm32flash.exe command-line tool from the STM website or from Sourceforge: https://sourceforge.net/projects/stm32flash/

2. Place the downloaded stm32flash executable in the same folder with the FW binary and check for the correct ODIN-W2 COM port number in the Device Manager (Windows users).

3. To ensure no settings persist over the firmware versions, it is recommended to revert to factory default before uploading new firmware. See section 4.4.1 Revert to factory default for instructions.

4. Prior to firmware upload, ODIN-W2 must be started in safeboot mode. Proceed by placing a safeboot jumper and reboot C099-F9P. The locations of the safeboot pin header and the reset button are depicted in Figure 6. To confirm that ODIN-W2 started in safeboot mode, check that the ODIN-W2 activity LED remains off. Use the following command structure in PowerShell or a command prompt to start the firmware upload:

```
.\stm32flash.exe -b 115200 -w <c099mbed3.bin> -S 0x8000000 COM<port number>
```

5. To confirm a successful firmware upload, remove the safeboot jumper and restart the device. The ODIN-W2 activity LED lights up.

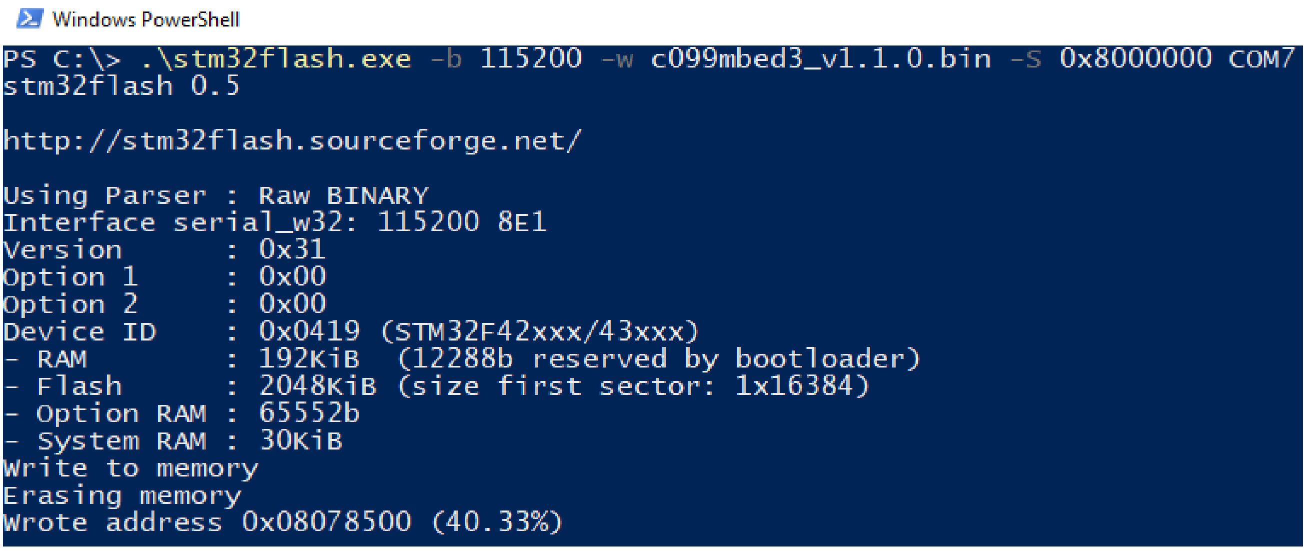


**Figure 31: Power shell capture of FW upload**

### 7.2.2 u-connectXpress software

To utilize the standard ODIN-W2 connectivity stack, a firmware update is required.

The latest u-blox u-connectXpress software and related documentation is available via u-blox.com: https://www.u-blox.com/en/product/odin-w2-series

It is recommended to download the stm32flash.exe command line tool from STM website or from Sourceforge: https://sourceforge.net/projects/stm32flash/

The software upload procedure consists of two consecutive phases. First, you must upload a bootloader, but prior to the bootloader upload, you must restart the ODIN-W2 in safe boot mode.

Proceed by placing a safe boot jumper and reboot the C099-F9P. The location of the safe boot pin header and the reset button is depicted in Figure 6.

Continue with the bootloader upload:

```
.\stm32flash.exe -b 115200 -w <ODIN-W2-BOOT.bin> -S 0x8000000 COM<port number>
```

The actual connectivity software shall be uploaded while the ODIN-W2 is still in safe boot mode. Ensure correct memory indexing by incrementing the memory argument as shown below:

```
.\stm32flash.exe -b 115200 -w <ODIN-W26X-SW.bin> -S 0x8010000 COM<port number>
```

Windows PowerShell

```
PS C:\> .\stm32flash.exe -b 115200 -w ODIN-W2-BOOT-v0.8.2.bin -S 0x8000000 COM7
stm32flash 0.5

http://stm32flash.sourceforge.net/

Using Parser : Raw BINARY
Interface serial_w32: 115200 8E1
Version      : 0x31
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0419 (STM32F42xxx/43xxx)
- RAM        : 192KiB  (12288b reserved by bootloader)
- Flash      : 2048KiB (size first sector: 1x16384)
- Option RAM : 65552b
- System RAM : 30KiB
Write to memory
Erasing memory
Wrote address 0x08002914 (100.00%) Done.

PS C:\>
```

**Figure 32: Power shell capture of bootloader upload**

Windows PowerShell

```
PS C:\> .\stm32flash.exe -b 115200 -w ODIN-W26X_FW5.0.1-002.bin -S 0x8010000 COM7
stm32flash 0.5

http://stm32flash.sourceforge.net/

Using Parser : Raw BINARY
Interface serial_w32: 115200 8E1
Version      : 0x31
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0419 (STM32F42xxx/43xxx)
- RAM        : 192KiB  (12288b reserved by bootloader)
- Flash      : 2048KiB (size first sector: 1x16384)
- Option RAM : 65552b
- System RAM : 30KiB
Write to memory
Erasing memory
Wrote address 0x0804a100 (15.78%)
```

**Figure 33: Power shell capture of connectivity software upload**

Once the connectivity software is uploaded successfully, the ODIN-W2 will be set with a baud rate of 115200 baud with hardware flow control enabled. You must change this to no hardware flow control and set the ODIN-W2 UART baud rate to 460800 to ensure sufficient link bandwidth.

To proceed, restart the ODIN-W2 in a normal boot mode by removing the safeboot jumper and pressing the RESET button.

- Set s-center to 115200 baud and no flow control.
- Open the COM port.

Select **EVK-ODIN-W2 via ST-LINK** button to disable flow control as shown in Figure 34.

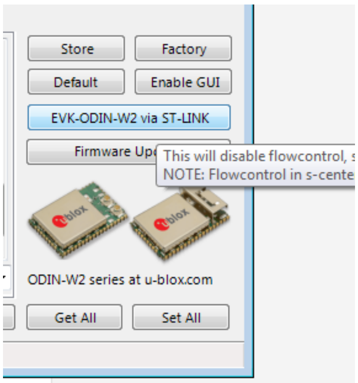


**Figure 34: Resetting ODIN-W2 to no flow control**

Navigate to "User Defines" AT command tab, as shown in Figure 35.

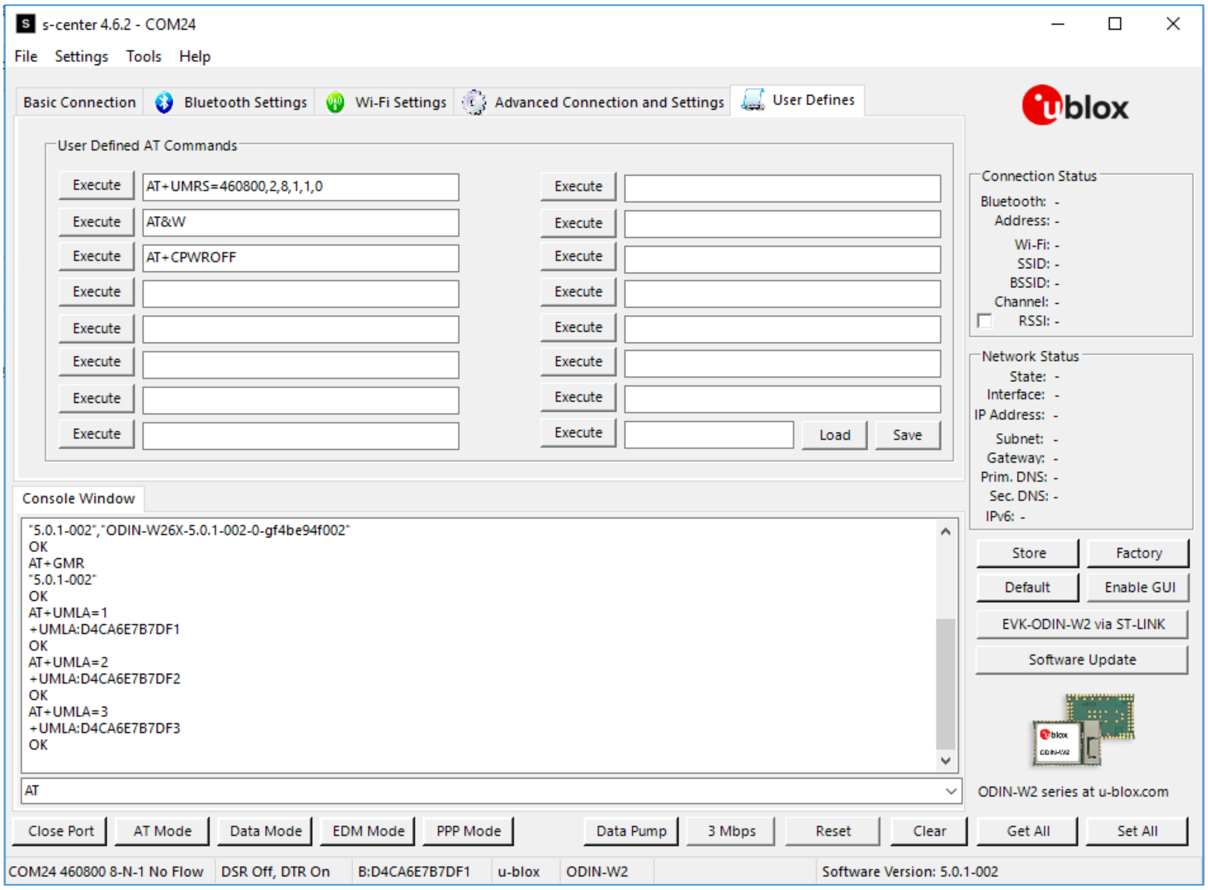


**Figure 35: s-center**

Execute the following command set sequentially:

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

Finally, adjust s-center baud rate to match 460800 by closing and opening the UART port.

Click the **AT Mode** button to ensure it is responding correctly. You will see it respond with AT commands if communication is OK.

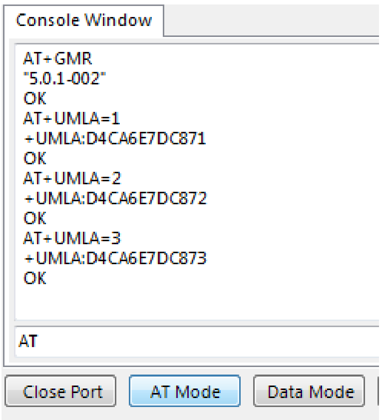


**Figure 40: Clicking AT Mode button**

Download a u-blox configuration file for the ODIN-W2 module. The "u-blox ODIN-W2 BT Rover.txt" file is the default configuration file shipped with the C099-F9P. See Appendix **Error! Reference source not found.** for configuration file resources.

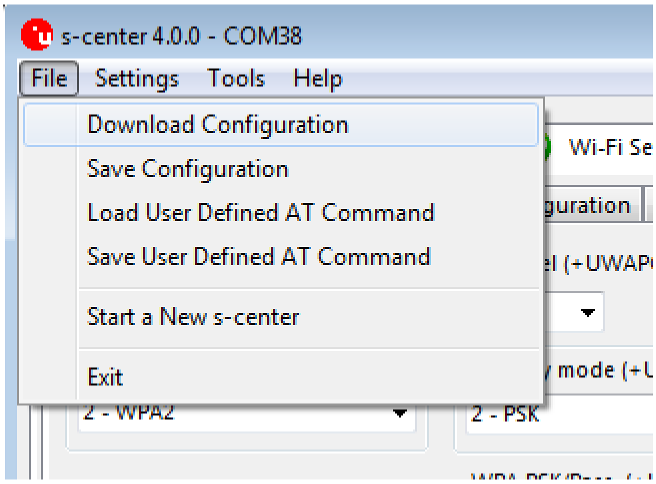


**Figure 36: Selecting File > Download Configuration**

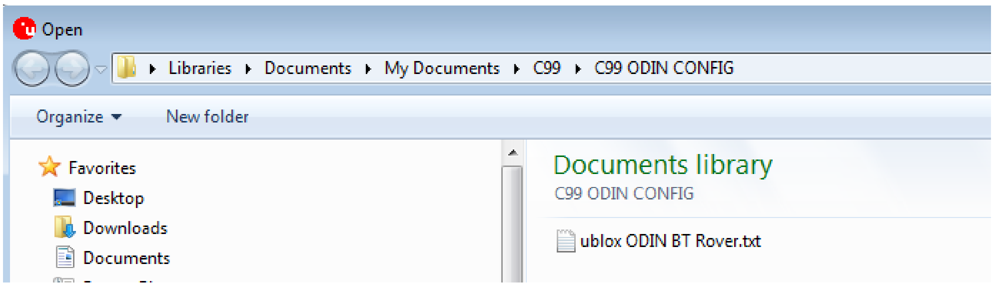


**Figure 37: Selecting "u-blox ODIN-W2 BT Rover.txt" file**

Select the file and click **Open**. It will download the file and write it to flash.

☞ The ODIN-W2 UART will now be set to 460800 baud in Data default mode. It will be ready for use again.

Disconnect s-center from the ODIN-W2 port and power the C099-F9P off and on to ensure it will be using the new configurations as default.

Position a jumper at “OE3” for Bluetooth operation.

The rover is now ready to connect to PC or Mobile via Bluetooth SPP.

The board is now ready to use for the wireless connection examples described in the earlier sections.

When untethered operation is not required, the ZED-F9P dedicated USB connection on the C099-F9P can be used for supplying corrections and monitoring or logging purposes with u-center.

# 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.

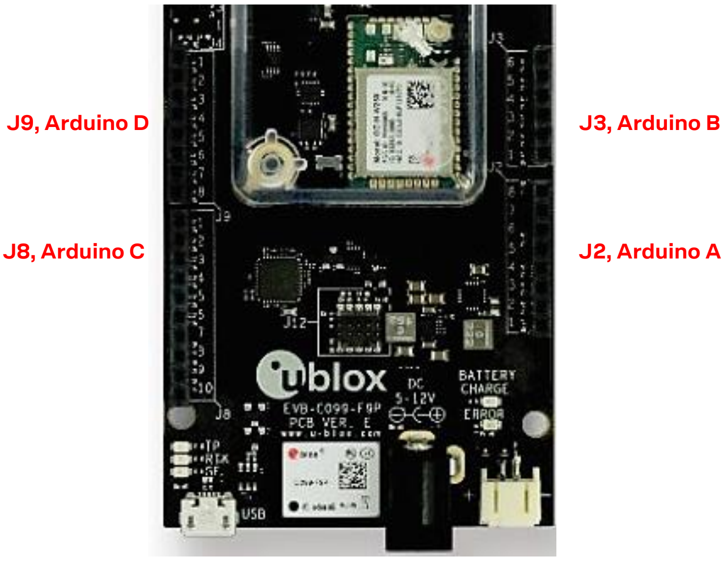


**Figure 38: C099-F9P Arduino connectors**

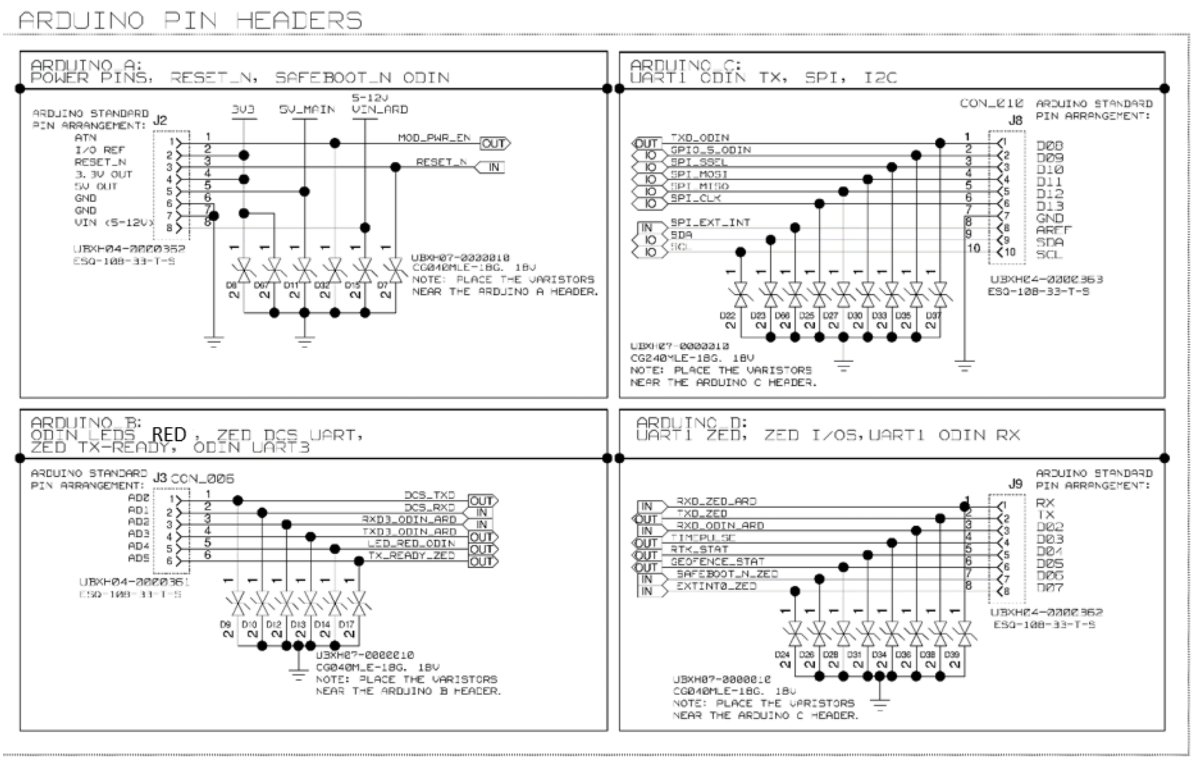


**Figure 39: C099-F9P Arduino R3 connections**

# 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-W2-20003600155137333393539"
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.

# J Mechanical board dimensions

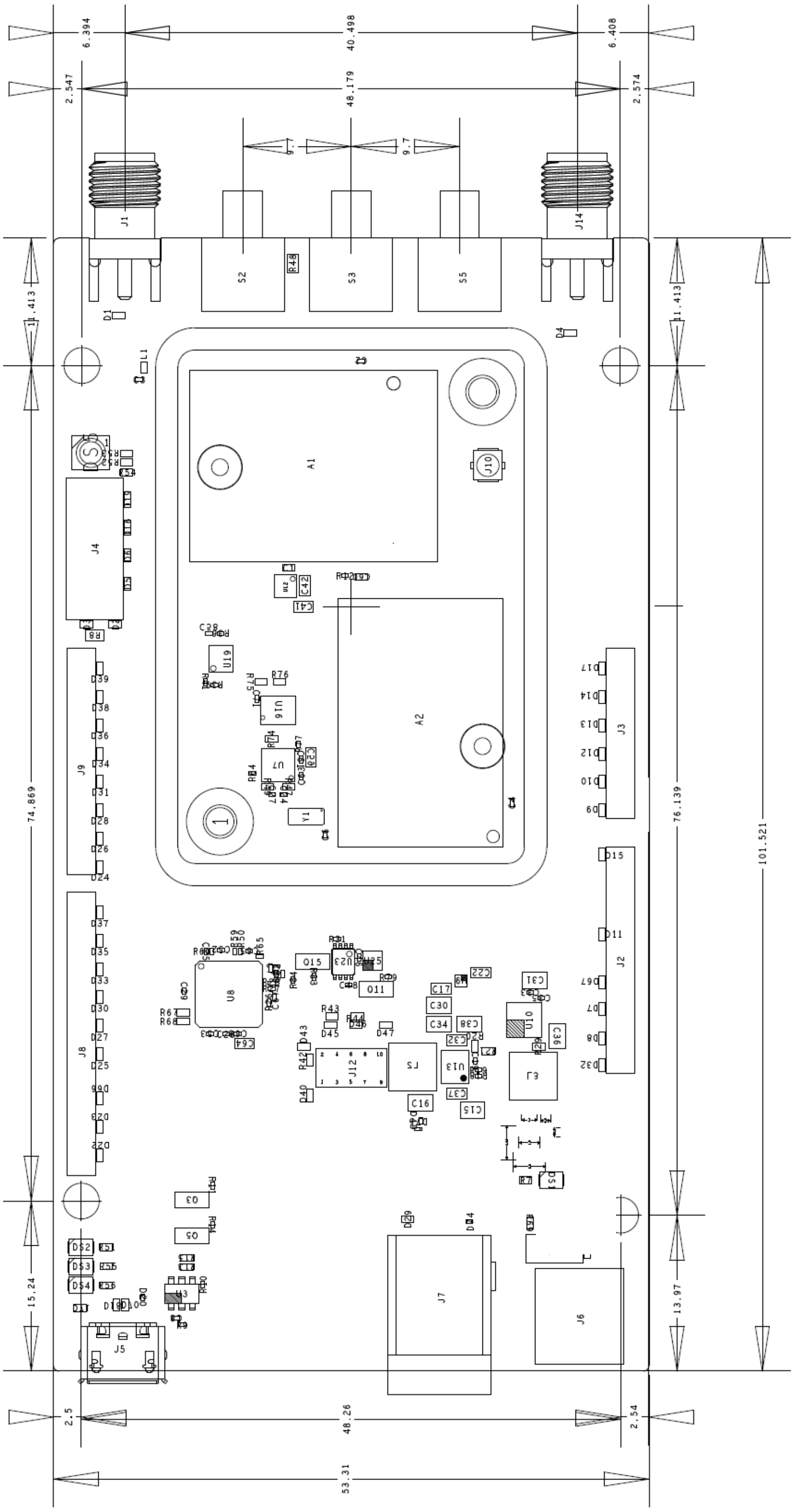


**Figure 40: C099-F9P rev. E dimensions**

# K C099-F9P schematics

C099-F9P evaluation board schematics, UBX-22008807

# Related documents


[1] ZED-F9P Integration manual, UBX-18010802
[2] HPG 1.32 Interface description, UBX-22008968
[3] u-center User guide, UBX-13005250
[4] u-blox GNSS Sensor and VCP Device Driver User guide, UBX-15022397
[5] ANN-MB series multi-band GNSS antennas Data sheet, UBX-18049862


☞ For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (www.u-blox.com).

# Revision history

| Revision | Date | Name | Comments |
|---|---|---|---|
| R01 | 10-Jul-2018 | ghun/byou | Initial release |
| R02 | 19-Oct-2018 | byou | Updates for the C099-F9P rev. B board revision. |
| R03 | 8-Nov-2018 | olep | Updates for Mbed3 FW in ODIN-W2 |
| R04 | 1-Feb-2019 | olep | Updates for Wi-Fi and NVDS features in ODIN-W2 |
| R05 | 21-Feb-2019 | olep | Updated Arduino J9 schematics. Polarity requirement of the battery connector. |
| R06 | 29-Mar-2019 | olep | Updates for Wi-Fi AP and STA operation |
| R07 | 23-May-2019 | olep | Editorial changes |
| R08 | 11-June-2019 | olep | Updates on Wi-Fi base instructions. |
| R09 | 25-Sep-2019 | jhak/jjus | Added instructions for non-wireless operation.<br>Updates for the C099-F9P rev. C board revision |
| R10 | 12-Nov-2019 | jhak | Updates for the C099-F9P rev. E board revision.<br>Connectivity SW renamed to u-connectXpress |
| R11 | 5-Dec-2019 | mala | Improved the quality of the schematic drawings. |
| R12 | 29-June 2020 | ghun | Update for HPG 1.13 |
| R13 | 13-Aug-2021 | dbhu | Changed DC source from 6-12 V to 5-12 V. |
| R14 | 23-Mar-2022 | dbhu | Corrected figure 6 for ODIN-W2 safeboot pins.<br>Updates for HPG 1.30 and schematics. |
| R15 | 28-Apr-2022 | angi | Update title and Document information section |
| R16 | 24-Nov-2022 | dbhu/vema | Added 3.2.3 C099-F9P Jumper section<br>Updated the appendix section |

# Contact


Address: u-blox AG
Zürcherstrasse 68
8800 Thalwil
Switzerland


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