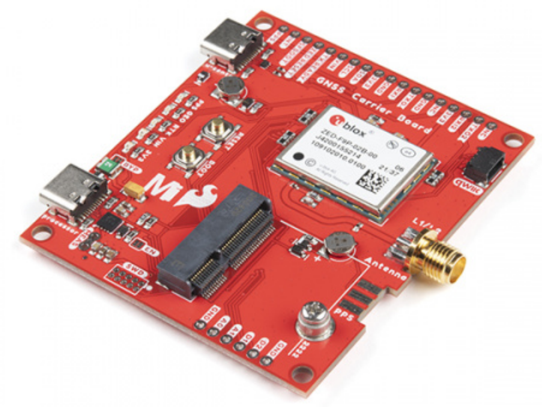
MicroMod GNSS Carrier Board (ZED-F9P) Hookup Guide

Introduction

The SparkFun MicroMod GNSS Carrier Board (ZED-F9P) combines high-precision GPS and the flexibility of MicroMod onto one board. Utilizing u-blox's ZED-F9P module, MicroMod GNSS Carrier Board is capable of 10mm 3 dimensional accuracy. Yes, you read that right, these boards can output your X, Y, and Z location that is roughly the width of your fingernail. With great power comes a few requirements: high precision GPS requires a clear view of the sky (sorry, no indoor location) and a stream of correction data from an RTCM source. We'll get into this more in a later section but as long as you have two ZED-F9P breakout boards, or access to an online correction source, your ZED-F9P can output lat, long, and altitude with centimeter grade accuracy.



SparkFun MicroMod GNSS Carrier Board (ZED-F9P)

◉ GPS-17722

Required Materials

To follow along with this tutorial, you will need the following materials at a minimum to get started. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary. To get the most out of the ZED-F9P, you will need a correction source. Depending on your setup, you may need a second ZED-F9P or access to an online correction source.

MicroMod GNSS Carrier Board (ZED-F9P) Wishlist SparkFun Wish List

SparkFun Mini Screwdriver
TOL-09146
This is just your basic reversible screwdriver - pocket sized! Both flat and phillips heads available. Comes with pin c…

GPS Antenna Ground Plate
GPS-17519

(2) USB 3.1 Cable A to C - 3 Foot
CAB-14743

MagmaX2 Active Multiband GNSS Magnetic Mount Antenna - AA.200
GPS-17108

SparkFun MicroMod GNSS Carrier Board (ZED-F9P)
GPS-17722

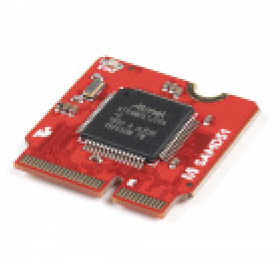
SparkFun MicroMod ESP32 Processor
WRL-16781

MicroMod Processor Board

You'll need a Processor Board with the MicroMod GNSS Carrier Board. We recommend using the MicroMod ESP32 Processor to connect to the cloud. Depending on your setup, you may need a transceiver to pass the correction data.

SparkFun MicroMod ESP32 Processor
○ WRL-16781

SparkFun MicroMod Teensy Processor
● DEV-16402



SparkFun MicroMod SAMD51 Processor
○ DEV-16791

SparkFun MicroMod Artemis Processor
◉ DEV-16401

Antenna

We recommend using a GNSS multi-band magnetic mount antenna for the full RF reception. The length of the antenna cable was also useful in mounting it.

GNSS Multi-Band Magnetic Mount Antenna - 5m (SMA)
○ GPS-15192

GNSS Multi-Band L1/L2 Surveying Antenna (TNC) - TOP106
⊖ GPS-17751

GNSS Multi-Band L1/L2 Helical Antenna (SMA) BT-560
⊖ GPS-17383

MagmaX2 Active Multiband GNSS Magnetic Mount Antenna - AA.200
◉ GPS-17108

Note: If you want to try different chip antennas, these antennas will not provide the full RF reception for the ZED-F9P.

GPS/GNSS Magnetic Mount Antenna - 3m (SMA)
◉ GPS-14986

GPS/GNSS Embedded Antenna - 1m (SMA)
⊖ GPS-14987

GPS Antenna Accessories

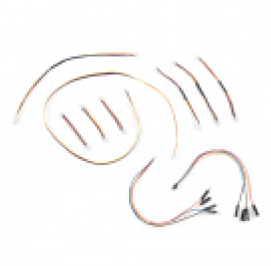
You can use the GPS antenna ground plate to improve your GPS antenna's performance. If you are using the GNSS Multi-Band L1/L2 Surveying Antenna (TNC) - TOP106, you'll need to grab the interface cable.

Interface Cable - SMA Male to TNC Male (300mm)
◉ CAB-17833

GPS Antenna Ground Plate
◉ GPS-17519

Accessories

At a minimum, you will need a USB C cable to power and program the boards. Depending on your application, you may want to grab a Qwiic cable to connect a Qwiic-enabled device.



SparkFun Qwiic Cable Kit
○ KIT-15081

USB 3.1 Cable A to C - 3 Foot
◉ CAB-14743

Tools

You will need a screw driver to tighten the screw between the processor board and carrier board.

SparkFun Mini Screwdriver
◉ TOL-09146

Suggested Reading

If you aren't familiar with the MicroMod ecosystem, we recommend reading here for an overview. We recommend reading here for an overview if you decide to take advantage of the Qwiic connector.

MicroMod	qwiic®
MicroMod Ecosystem	*Qwiic Connect System*

If you aren’t familiar with the following concepts, we recommend checking out these tutorials before continuing.

I2C

An introduction to I2C, one of the main embedded communications protocols in use today.

Getting Started with U-Center for u-blox

Learn the tips and tricks to use the u-blox software tool to configure your GPS receiver.

Getting Started with MicroMod

Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!

MicroMod ESP32 Processor Board Hookup Guide

A short hookup guide to get started with the SparkFun MicroMod ESP32 Processor Board.

This tutorial is based on the GPS-RTK2's ZED-F9P. Make sure to check out the breakout boards for more information on GPS-RTK. Be sure to checkout our What is GPS RTK? tutorial.

What is GPS RTK?

SEPTEMBER 14, 2018

Learn about the latest generation of GPS and GNSS receivers to get 14mm positional accuracy!

GPS-RTK Hookup Guide

SEPTEMBER 13, 2018

Find out where you are! Use this easy hook-up guide to get up and running with the SparkFun high precision GPS-RTK NEO-M8P-2 breakout board.

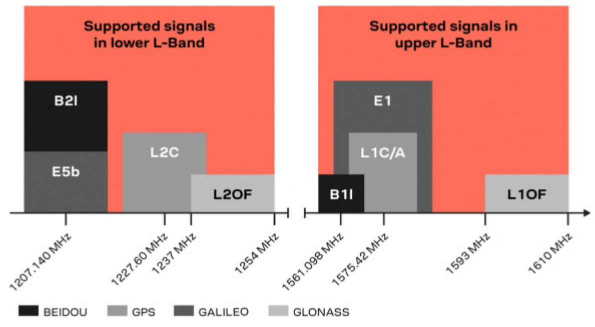
GPS-RTK2 Hookup Guide

JANUARY 14, 2019

Get precision down to the diameter of a dime with the new ZED-F9P from u-blox.

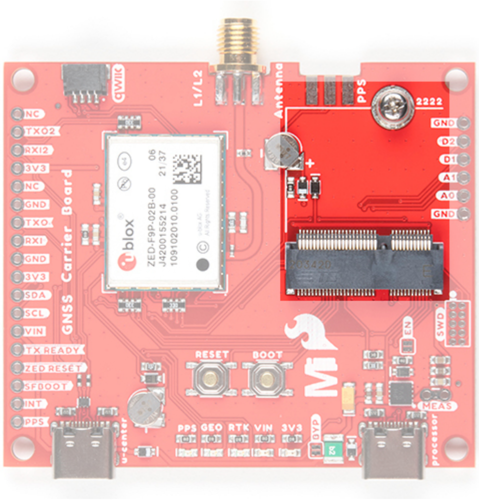
Hardware Overview

One of the key differentiators between the ZED-F9P and almost all other low-cost RTK solutions is the ZED-F9P is capable of receiving both L1 and L2 bands.



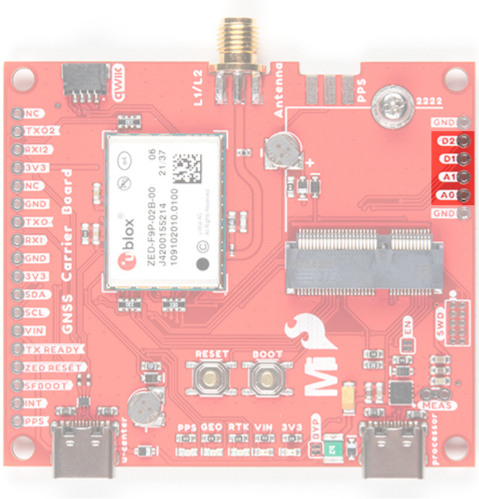
MicroMod Processor Board Socket

The MicroMod GNSS Carrier Board (ZED-F9P) includes a location for a MicroMod Processor Board. Here is where your chosen Processor Board will reside.



MicroMod Processor General Pins

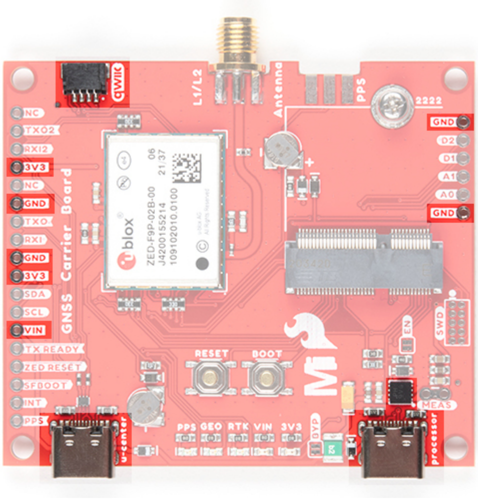
Next to the MicroMod Processor Board are extra pins if you need to use a digital or analog pin.



Power

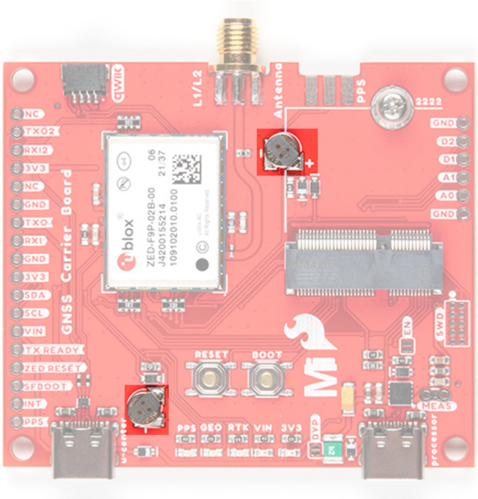
There are a few ways to power the board. Voltage is regulated down to 3.3V with the AP7361 voltage regulator. The square IC next to the USB C labeled as processor is where you will find the AP7361.

- **USB C Connector** - You can connect a USB Type C cable from your computer's USB port to the board through either of the USB Type C connectors labeled as **u-center** and **processor**. There are protection diodes connected to the voltage lines so you can connect two USB cables at the same time to power the board. The AP7361 voltage regulator will regulate the **5V** from the USB port down to 3.3V for the system voltage.
- **VIN** - If you decide to connect to the VIN pin, we recommend a voltage between **3.3V** to **6.0V**. The AP7361 voltage regulator will regulate the voltage down to 3.3V for the system voltage.
- **3V3** - If you decide to power the board through the 3.3V pin, you could connect a regulated 3.3V to this pin. Otherwise, you could use this to power any peripherals attached to the board.
- **Qwiic Connector** - The Qwiic connector connects to 3.3V and GND to power any Qwiic-enabled devices. Depending on your application, you could connect a regulated 3.3V through this port as well.
- **GND** - Of course, you'll need to connect the ground plane to your power source. This pin is available should you decide to power the board through the any of the PTH pins.



Backup Batteries

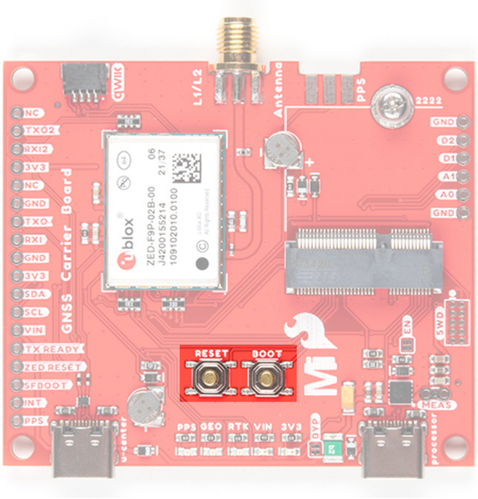
There are two built-in backup batteries (ML414H) on the board. The backup battery has a 1mAh capacity and requires 20 minute to charge. The battery near the SMA connector is for the Processor Board's RTC and helps keep the RTC running when the external power is removed. Depending on the processor, it may not be connected.



The other one that is closest to the USB C connector labeled as u-center is for the ZED-F9P module. The rechargeable battery maintains the battery backed RAM (BBR) on the GNSS module. This allows for much faster position locks (a.k.a. hot start). The BBR is also used for module configuration retention. The battery is automatically trickle charged when power is applied and should maintain settings and GNSS orbit data for up to two weeks without power.

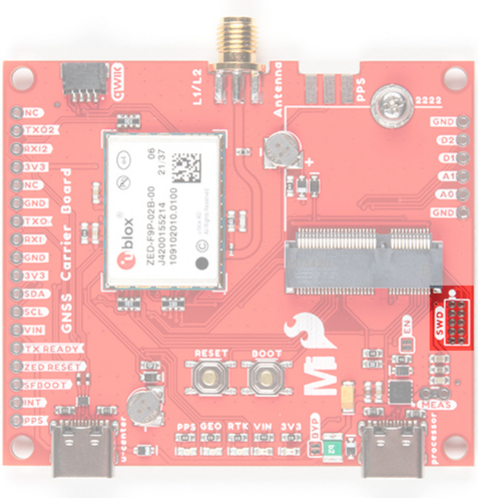
Reset and Boot Buttons

The reset button will reset the processor. The boot button will put the processor into a special boot mode. Depending on the processor board, this boot pin may not be connected.



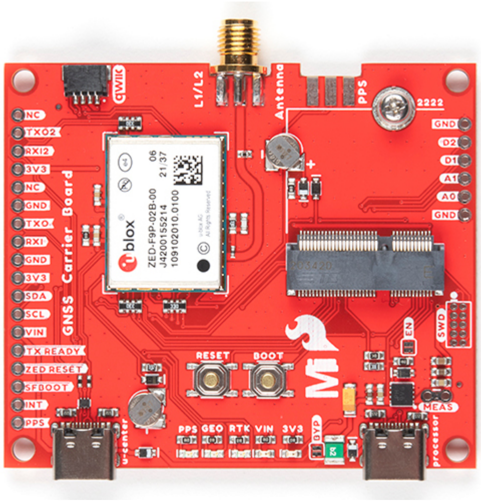
SWD Programming Pins

For advanced users, we broke out the SWD programming pins to connect to a MicroMod Processor Board. Note that this is not populated so you will need a compatible header and compatible JTAG programmer to connect.



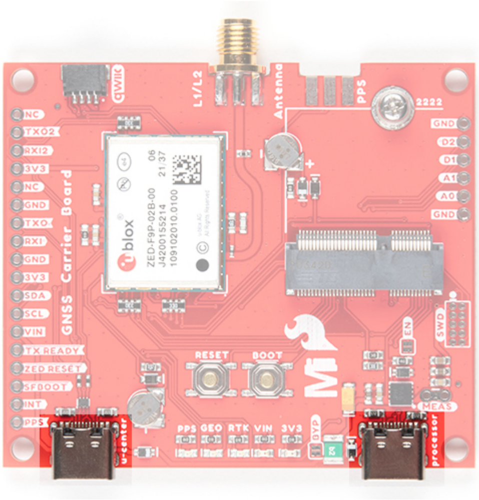
Communication Ports

The ZED-F9P is unique in that it has *five* communication ports which are all active simultaneously. You can read NMEA data over I^{2}C while you send configuration commands over the UART and vice/versa. The only limit is that the SPI pins are mapped onto the I^{2}C and UART pins so it's either SPI or I^{2}C+UART. The USB port is available at all times.



USB

As stated earlier, there are two USB ports: one for **u-center** and another for **processor**. The USB C connector labeled as **u-center** makes it easy to connect the ZED-F9P to u-center for configuration and quick viewing of NMEA sentences. It is also possible to connect a Raspberry Pi or other single board computer over USB. The ZED-F9P enumerates as a serial COM port and it is a separate serial port from the UART interface. See Getting Started with U-Center for more information about getting the USB port to be a serial COM port. The USB connector labeled as **processor** is available to program your MicroMod Processor Board. Make sure to check your respective Processor Board to install the USB-to-serial drivers.

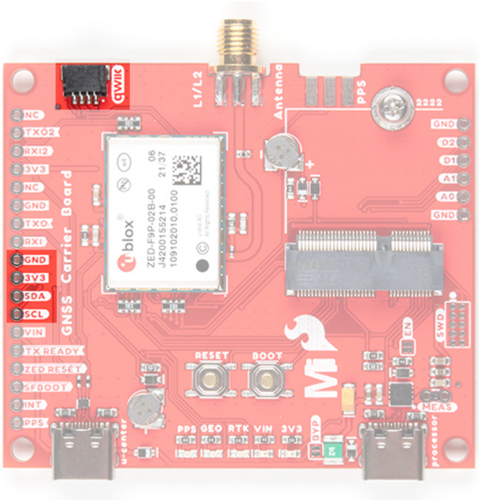


A 3.3V regulator is provided to regulate the 5V USB down to **3.3V** the module requires. External 5V can be applied or a direct feed of 3.3V can be provided. Note that if you're provide the board with 3.3V directly it should be a clean supply with minimal noise (less than 50mV VPP ripple is ideal for precision locating).

The 3.3V regulator is capable of sourcing 600mA from a 5V input and the USB C connection is capable of sourcing 2A.

I^{2}C (a.k.a DDC)

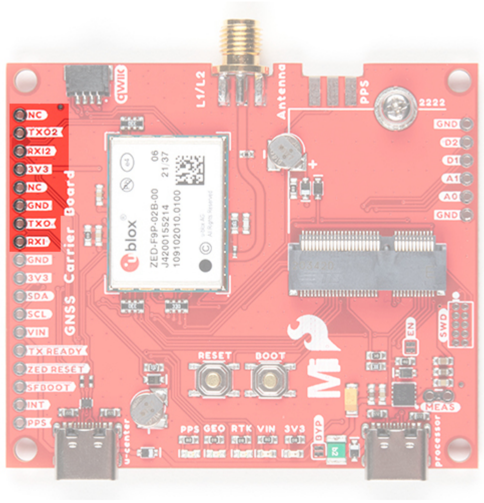
The u-blox ZED-F9P has a "DDC" port which is really just an I^{2}C port (without all the fuss of trademark issues). These pins are shared with the SPI pins. By default, the I^{2}C pins are enabled. Be sure the SPI jumper on the rear of the board is open. The MicroMod GNSS Carrier Board also includes one Qwiic connector to make daisy chaining this GPS receiver with a large variety of I^{2}C devices. Checkout Qwiic for your next project.



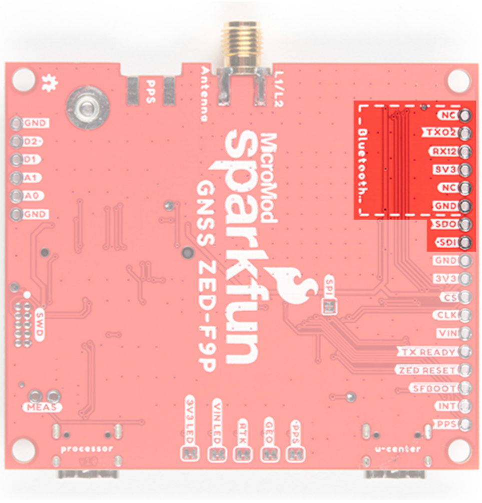
UART/Serial

The classic serial pins are available on the ZED-F9P but are shared with the SPI pins. By default, the UART pins are enabled. Be sure the SPI jumper on the rear of the board is open.

- TXO/SDO = TX out from ZED-F9P
- RXI/SDI = RX into ZED-F9P



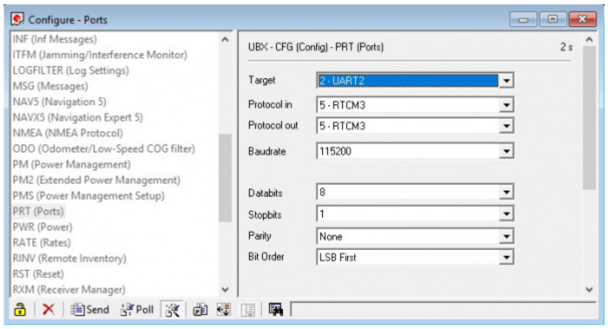
Top View



Bottom View

There is a second serial port (UART2) available on the ZED-F9P that is primarily used for RTCM3 correction data. By default, this port will automatically receive and parse incoming RTCM3 strings enabling RTK mode on the board. In addition to the TXO2/RXI2 pins we have added an additional ‘RTCM Correction’ port where we arranged the pins to match the industry standard serial connection (aka the 'FTDI' pinout). While we label this as "**Bluetooth**" on the back of the board, you can still connect any transceiver or serial-to-USB converter to this port. This pinout is compatible with our Bluetooth Mate and Serial Basic so you can send RTCM correction data from a cell phone or computer. Note that RTCM3 data can also be sent over I^{2}C, UART1, SPI, or USB if desired.

The RTCM correction port (UART2) defaults to **38400bps** serial but can be configured via software commands (checkout our Arduino library) or over USB using u-center. Keep in mind our Bluetooth Mate defaults to **115200bps**. If you plan to use Bluetooth for correction data (we found it to be easiest), we recommend you increase this port speed to 115200bps using u-center. Additionally, but less often needed, the UART2 can be configured for NMEA output. In general, we don't use UART2 for anything but RTCM correction data, so we recommend leaving the in/out protocols as RTCM.



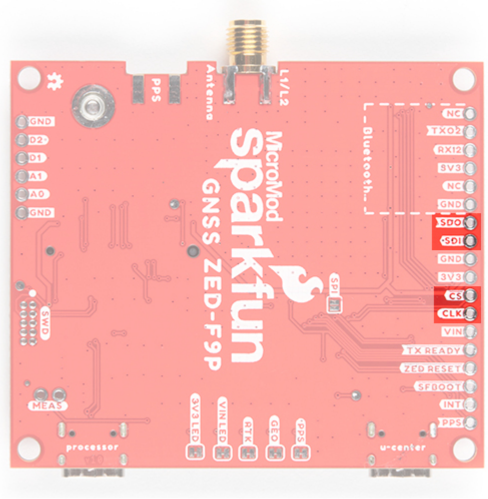
UBX - CFG (Config) - PRT (Ports) 2 s

Target: 2 - UART2
Protocol in: 5 - RTCM3
Protocol out: 5 - RTCM3
Baudrate: 115200
Databits: 8
Stopbits: 1
Parity: None
Bit Order: LSB First

If you've got the ZED-F9P setup for base station mode (also called *survey-in* mode) the UART2 will *output* RTCM3 correction data. This means you can connect a radio or wired link to UART2 and the board will automatically send *just* RTCM bytes over the link (no NMEA data taking up bandwidth).

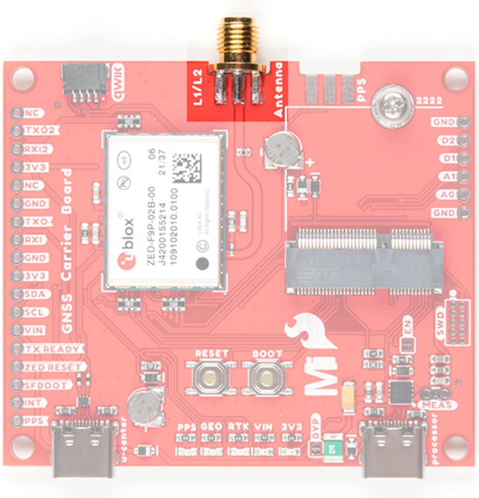
SPI

The ZED-F9P can also be configured for SPI communication. By default, the SPI port is disabled. To enable SPI close the SPI jumper on the rear of the board. Closing this jumper will disable the UART1 and I^{2}C interfaces (UART2 will continue to operate as normal).



Antenna

The ZED-F9P requires a good quality GPS or GNSS (preferred) antenna. For a secure connection, we include an SMA female connector. To make the most out of the ZED-F9P, you'll need a multi-band GNSS antenna and an SMA male connector to mate.

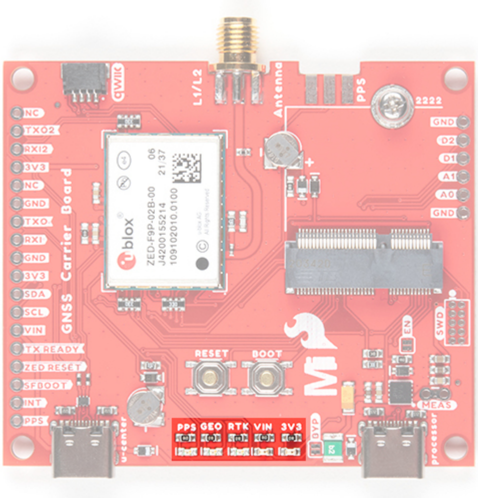


Low-cost magnetic GPS/GNSS antennas can be used (checkout the u-blox white paper) but a 4” / 10cm metal disc is required to be placed under the antenna as a metal ground plane.

LEDs

The board includes five status LEDs as indicated in the image below.

- **PPS:** The pulse per second LED will illuminate each second once a position lock has been achieved.
- **GEO:** The GEO LED can be configured to turn on/off for geofencing applications.
- **RTK:** The RTK LED will be illuminated constantly upon power up. Once RTCM data has been successfully received it will begin to blink. This is a good way to see if the ZED-F9P is getting RTCM from various sources. Once an RTK fix is obtained, the LED will turn off.
- **VIN:** The VIN LED will illuminate when there is voltage applied to the VIN pin or over USB.
- **3V3:** The power LED will illuminate when 3.3V is activated either over USB or via the Qwiic bus.

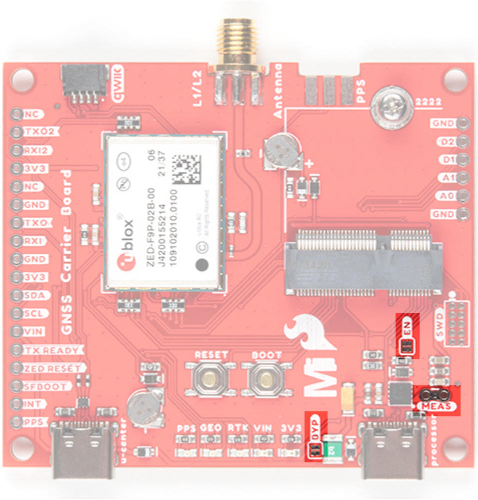


Jumpers

Note: If this is your first time working with jumpers, check out the How to Work with Jumper Pads and PCB Traces tutorial for more information.

There are jumpers located throughout the board. Below are the jumpers on the top side of the board.

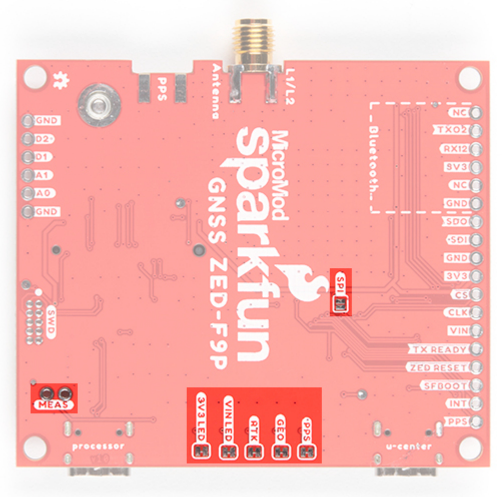
- **Bypass (BYP):** By default, the BYP is left open. Adding a solder jumper bypasses the 2A resettable fuse on the back of the board should you decide to pull more than 2A from your USB source. Proceed with caution should you decide to bypass the jumper.
- **Enable (EN):** By default, the EN jumper is left open. This jumper is connected to a processor board's GPIO pin. The processor board can control the ATP's voltage regulator. Depending on the processor, this may not be connected.
- **Current Measurement (MEAS):** By default, the MEAS is closed. Cutting the jumper and soldering to the PTH pads will allows you to insert a current meter and precisely monitor the how much current your application is consuming.



Below are the jumpers on the bottom side of the board.

- **3V3 LED:** By default, the 3V3 LED is closed. Cutting this jumper will disable the LED when there is 3.3V.
- **VIN LED:** By default, the VIN LED is closed. Cutting this jumper will disable the LED whenever there is an input voltage.

- **RTK:** By default, the RTK is closed. Cutting this jumper will disable the RTK LED. This LED indicates when there is an RTK fix.
- **GEO:** By default, the GEO is closed. Cutting this jumper will disable the GEO LED. This LED is used for geofencing applications.
- **PPS:** By default, the PPS is closed. Cutting this jumper will disable the PPS LED whenever a position lock has been achieved.
- **SPI:** By default, the SPI is open. Closing SPI with solder enables the SPI interface and disables the UART and I^{2}C interfaces. USB will still function.

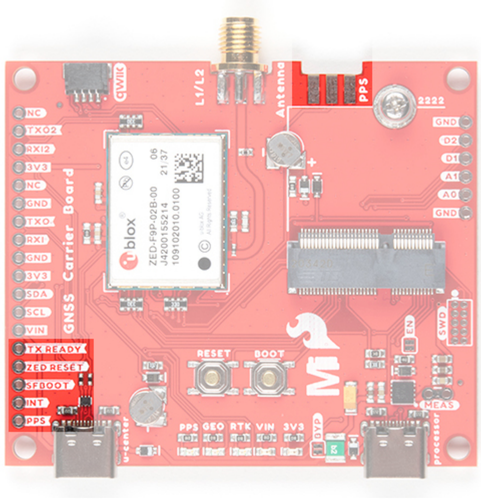


Note: The MEAS jumper has PTH pins. You will need to disconnect the PTH pads on the top side of the board before you are able to measure the current from the bottom of the board effectively.

Control Pins

These pins are used for various extra control of the ZED-F9P:

- **TX READY:** Transmit ready output pin. Can be configured using U-Center to indicate that the transmit buffer is full and ready to be transmitted. This is connected to D0 on the MicroMod Processor Board.
- **ZED RESET:** Reset input pin. Pull this line low to reset the module.
- **SFBOOT:** Safeboot input pin. This is required for firmware updates to the module and generally should not be used or connected.
- **INT:** Interrupt input/output pin. Can be configured using U-Center to bring the module out of deep sleep or to output an interrupt for various module states.
- **PPS:** Pulse-per-second output pin. Begins blinking at 1Hz when module gets basic GPS/GNSS position lock.



Note: For those that need to connect a SMA connector to the PPS pin, the RTK-SMA includes a footprint so that you can manually solder the connector to the board for your application. The PPS output is helpful as a clock source correction when synchronizing equipment (not 1 Hz but in many MHz). The PPS output can be configured to output a very accurate clock which scientists use to correct less accurate, but much faster clocks. To configure, you can use the u-center to adjust The NEO-F9P's setting under **View** > **Configuration View** > **TP (TimePulse)**.

SMA Connector
● WRL-00593
$1.95

Hardware Pinout

GNSS ZED-F9P CARRIER PINOUT TABLE

MICROMOD GENERAL PINOUT TABLE

MICROMOD GENERAL PIN DESCRIPTIONS

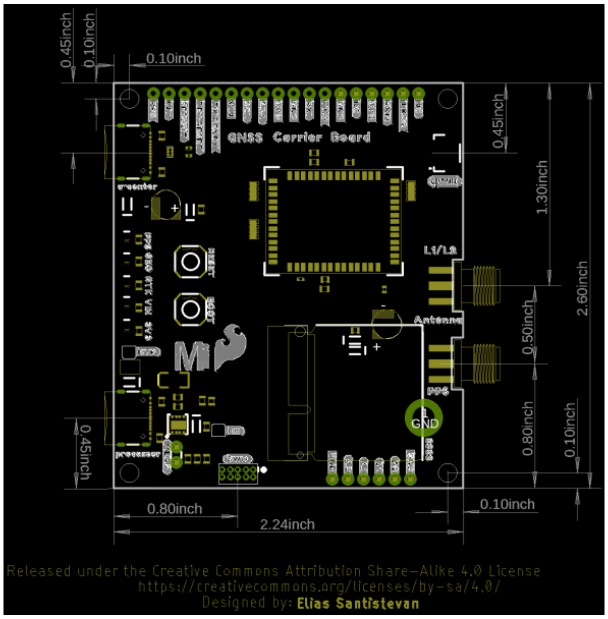
AUDIO	UART	GPIO/BUS	I^{2}C	SDIO	SPI	Dedicated

Function		Bottom Pin	Top Pin	Function	
	(Not Connected)		**75**	GND	
	3.3V	**74**	**73**		
	RTC_3V	**72**	**71**		
		70	**69**		
		66	**65**		
		64	**63**		
		62	**61**	TXO	SDO
		60	**59**	RXI	SDI
		58	**57**	SCL	SCK
		56	**55**	SDA	$\overline{CS}$
		54	**53**		
		52	**51**		
		50	**49**	VIN/3	
		48	**47**		
		46	**45**	GND	
		44	**43**		
		42	**41**		
D2	G0	**40**	**39**	GND	
	A1	**38**	**37**		
	GND	**36**	**35**		
	A0	**34**	**33**	GND	
		32	**31**		
		30	**29**		
		28	**27**		

		26	**25**		
		24	**23**	SWDIO	
	RXI2	**22**	**21**	SWDCK	
	TXO2	**20**	**19**	TXO	SDO
	D1	**18**	**17**	RXI	SDI
	INT	**16**	**15**		
SCK	SCL	**14**	**13**		
$\overline{\text{CS}}$	SDA	**12**	**11**	$\overline{\text{BOOT}}$	
	TX_READY	**10**	**9**	VIN	
		8	**7**	GND	
	RESET	**6**	**5**	USB_D-	
	3.3V_EN	**4**	**3**	USB_D+	
	3.3V	**2**	**1**	GND	

Board Dimension

The board is about 2.24"x2.60" and includes four mounting holes on each corner. If you include the length of the connectors sticking out from the edge of the board, the overall size of the board is about 2.52"x2.60".



Designed by: Elias Santistevan

Hardware Assembly

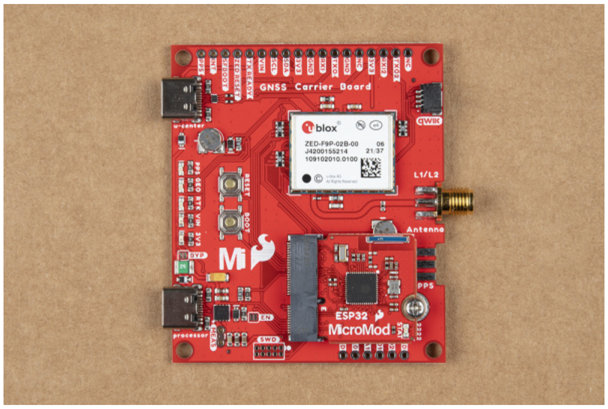
If you have not already, make sure to check out the Getting Started with MicroMod: Hardware Hookup for information on inserting your Processor Board to your Carrier Board. Just insert the MicroMod Processor Board at an angle of about 25° into the M.2 socket, push down, and secure with the screw.

Getting Started with MicroMod

OCTOBER 21, 2020

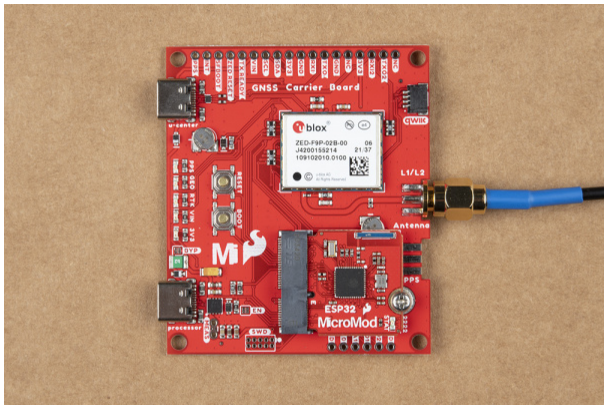
Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!

At a minimum, your setup should look like the image below. In this case, we had the MicroMod ESP32 Processor Board secured in the M.2 connector.



GNSS Multi-band Antenna

As stated earlier, you'll need a multi-band antenna and a metal ground plate to make the best use of the ZED-F9P. Connect the two SMA connectors together and tighten the nut. You'll simply need the nut to be finger tight.

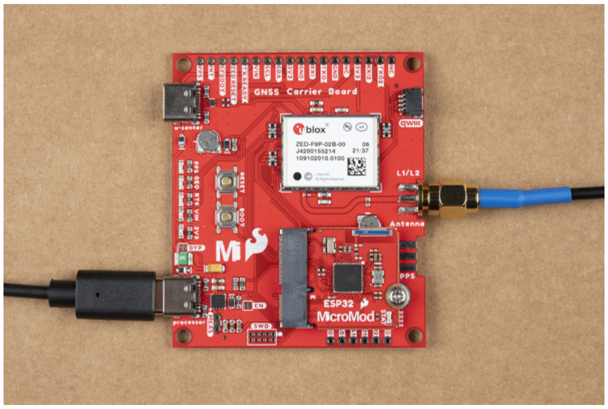


If you’re indoors you *must* run a SMA extension cable long enough to locate the antenna where it has a clear view of the sky. That means no trees, buildings, walls, vehicles, or concrete metally things between the antenna and the sky. Be sure to mount the antenna on a 4”/10cm metal ground plate to increase reception.

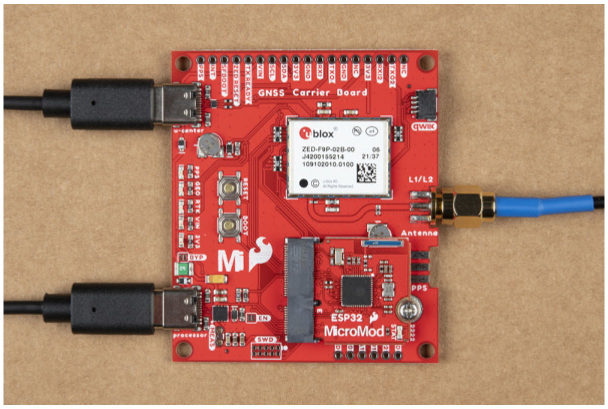
Of course, you could also attach the ground plate on a camera tripod. Just make sure to secure it with weights if there are heavy winds when using the ZED-F9P.

USB-C Cable

To program and power the microcontroller, insert the USB-C cable into the USB-C connector labeled as **Processor**.



To update the ZED-F9P's firmware or configure the module, insert the USB-C cable into the USB-C connector labeled as **u-center**.



You can have both USB cables connected at the same time since there are protection diodes connected on the voltage lines.

Software Installation

Note: This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you have not previously installed an Arduino library, please check out our installation guide.

The SparkFun u-blox Arduino library enables the reading of all positional datums as well as sending binary UBX configuration commands over I^{2}C. This is helpful for configuring advanced modules like the ZED-F9P but also the NEO-M8P-2, SAM-M8Q and any other u-blox module that use the u-blox binary protocol.

The SparkFun u-blox Arduino library can be downloaded with the Arduino library manager by searching '**SparkFun u-blox GNSS**' or you can grab the zip here from the GitHub repository:

SPARKFUN U-BLOX ARDUINO LIBRARY (ZIP)

Once you have the library installed checkout the various examples.

- **Example1:** Read NMEA sentences over I^{2}C using u-blox module SAM-M8Q, NEO-M8P, etc
- **Example2:** Parse NMEA sentences using MicroNMEA library. This example also demonstrates how to overwrite the `processNMEA` function so that you can direct the incoming NMEA characters from the u-blox module to any library, display, radio, etc that you prefer.

- **Example3:** Get latitude, longitude, altitude, and satellites in view (SIV). This example also demonstrates how to turn off NMEA messages being sent out of the I^{2}C port. You'll still see NMEA on UART1 and USB, but not on I^{2}C. Using only UBX binary messages helps reduce I^{2}C traffic and is a *much* lighter weight protocol.
- **Example4:** Displays what type of a fix you have the two most common being none and a full 3D fix. This sketch also shows how to find out if you have an RTK fix and what type (floating vs. fixed).
- **Example5:** Shows how to get the current speed, heading, and dilution of precision.
- **Example6:** Demonstrates how to increase the output rate from the default 1 per second to many per second; up to 30Hz on some modules!
- **Example7:** Older modules like the SAM-M8Q utilize an older protocol (version 18) whereas the newer modules like the ZED-F9P depricate some commands using the latest protocol (version 27). This sketch shows how to query the module to get the protocol version.
- **Example8:** u-blox modules use I^{2}C address 0x42 but this is configurable via software. This sketch will allow you to change the module's I^{2}C address.
- **Example9:** Altitude is not a simple measurement. This sketch shows how to get both the ellipsoid based altitude and the MSL (mean sea level) based altitude readings.
- **Example10:** Sometimes you just need to do a hard reset of the hardware. This sketch shows how to set your u-blox module back to factory default settings.
- **NEO-M8P**
  - **NEO-M8P Example1:** Send UBX binary commands to enable RTCM sentences on u-blox NEO-M8P-2 module. This example is one of the steps required to setup the NEO-M8P as a base station. For more information have a look at the u-blox manual for setting up an RTK link.
  - **NEO-M8P Example2:** This example extends the previous example sending all the commands to the NEO-M8P-2 to have it operate as a base. Additionally the `processRTCM` function is exposed. This allows the user to overwrite the function to direct the RTCM bytes to whatever connection the user would like (radio, serial, etc).
  - **NEO-M8P Example3:** This is the same example as NEO-M8P's Example2. However, the data is sent to a serial LCD via I^{2}C.
- **ZED-F9P**
  - **ZED-F9P Example1:** This module is capable of high precision solutions. This sketch shows how to inspect the accuracy of the solution. It's fun to watch our location accuracy drop into the millimeter scale.
  - **ZED-F9P Example2:** The ZED-F9P uses a new u-blox configuration system of VALGET/VALSET/VALDEL. This sketch demonstrates the basics of these methods.
  - **ZED-F9P Example3:** Setting up the ZED-F9P as a base station and outputting RTCM data.
  - **ZED-F9P Example4:** This is the same example as ZED-F9P's Example3. However, the data is sent to a serial LCD via I^{2}C.

This SparkFun u-blox library really focuses on I^{2}C because it's faster than serial and supports daisy-chaining. The library also uses the UBX protocol because it requires far less overhead than NMEA parsing and does not have the precision limitations that NMEA has.

Arduino Examples

Example 1: Positional Accuracy

As a quick test, we will be using the first example in the **ZED-F9P** folder (located in **File Examples > SparkFun u-blox GNSS Arduino Library** > **ZED-F9P** > **Example1_GetPositionAccuracy**).

If you have not already, select your Board (in this case the **SparkFun ESP32 MicroMod**), and associated COM port. Upload the code to the board and and set the Arduino Serial Monitor to **115200 baud.** Give the ZED-F9P a few minutes to get a satellite lock. The GPS coordinates and the accuracy will be output in the Serial Monitor.

More Examples!

Now that you got it up and running, check out the other examples located in the ZED-F9P folder!

SPARKFUN U-BLOX GNSS ARDUINO LIBRARY: ZED-F9P

Of course, to get the most out of the ZED-F9P, you will need an RTCM correction source. Depending on your setup, you will probably need a second ZED-F9P for a correction source. The following project tutorials allow you to set up the ZED-F9P as a reference station or rover.

How to Build a DIY GNSS Reference Station

OCTOBER 15, 2020

Learn how to affix a GNSS antenna, use PPP to get its ECEF coordinates and then broadcast your own RTCM data over the internet and cellular using NTRIP to increase rover reception to 10km!

Setting up a Rover Base RTK System

OCTOBER 14, 2020
Getting GNSS RTCM correction data from a base to a rover is easy with a serial telemetry radio! We'll show you how to get your high precision RTK GNSS system setup and running.

Resources and Going Further

Ready to get hands-on with GPS?

We've got a page just for you! We'll walk you through the basics of how GPS works, the hardware needed, and project tutorials to get you started.

TAKE ME THERE!

Now that you've successfully got your MicroMod GNSS Carrier Board (ZED-F9P) up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- Board Dimensions (PNG)
- Hookup Guide
- Datasheet (PDF) (ZED-F9P)
- UBX and NMEA Protocol Manual (PDF) (ZED-F9P)
- Integration Manual (PDF) (ZED-F9P)

- Product Summary (PDF) (ZED-F9P)
- Release Notes - FW1.00 (PDF) (ZED-F9P)
- Example RTCM output (PDF) from the ZED-F9P
- u-blox ECCN (PDF)
- Arduino Library
- GitHub Hardware Repo

Need some inspiration for your next project? Check out some of these related tutorials:

Three Quick Tips About Using U.FL

Quick tips regarding how to connect, protect, and disconnect U.FL connectors.

Advanced Autonomous Kit for Sphero RVR Assembly Guide

Get your Advanced Autonomous Kit for the Sphero RVR built up with this hookup guide!

How to Build a DIY GNSS Reference Station

Learn how to affix a GNSS antenna, use PPP to get its ECEF coordinates and then broadcast your own RTCM data over the internet and cellular using NTRIP to increase rover reception to 10km!

SparkFun RTK Surveyor Hookup Guide

Learn how to use the enclosed RTK Surveyor product to achieve millimeter level geospatial coordinates.

Or check out this blog post for inspiration.

SparkFun's GPS RTK brings Orienteering Dreams to Life