SparkFun GPS-RTK Dead Reckoning ZED-F9K Hookup Guide

Introduction

The SparkFun GPS ZED-F9K takes advantage of dead reckoning for navigation. The u-blox ZED-F9K is a powerful GPS-RTK unit that uses a fusion of IMU, wheel ticks, a vehicle dynamics model, correction data, and GNSS measurements to provide highly accurate and continuous position for navigation in the difficult conditions. We will quickly get you set up using the Qwiic ecosystem through Arduino. so that you can start reading the output!

SparkFun GNSS-RTK Dead Reckoning Breakout - ZED-F9K (Qwiic)
GPS-18719

Required Materials

To follow along with this tutorial, you will need the following materials at a minimum. You may not need everything though depending on what you have. Depending on your application, you may need additional parts for a correction source or connecting to you a vehicle to obtain wheel tick/direction information. Add it to your cart, read through the guide, and adjust the cart as necessary.

ZED-F9K Wishlist SparkFun Wish List

	GNSS Multi-Band Magnetic Mount Antenna - 5m (SMA) GPS-15192
	Reversible USB A to Reversible Micro-B Cable - 2m CAB-15427
	Qwiic Cable - 100mm PRT-14427
	SparkFun Thing Plus - ESP32 WROOM WRL-15663
	SparkFun GNSS-RTK Dead Reckoning Breakout - ZED-F9K (Qwiic) GPS-18719

Microcontroller

If you are using the breakout board and programming in Arduino, we recommend the SparkFun Thing Plus - ESP32 WROOM with the associated USB cable to start.

SparkFun Thing Plus - ESP32 WROOM
WRL-15663

Qwiic Cable - 100mm
PRT-14427

Reversible USB A to Reversible Micro-B Cable - 2m
CAB-15427

Antenna

We recommend using the magnetic mount antenna for the full RF reception and mounting it on top of a vehicle. The antenna uses an SMA connector so you can direclty connect it to the breakout board. Link for that is below in the antenna accessories. 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, you can try the The following antennas. However, these antennas will not provide the full RF reception for the ZED-F9K.

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

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

GPS Antenna Accessories

Depending on your antenna, you will need an adapter to connect to the GPS-RTK's u.FL connector. If you need more than the metal from the top of a vehicle or are mounting it on a robot that does not have the necessary ground plane, 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

Other Qwiic Cable Accessories

There are different Qwiic cable lengths available. Depending on your application, you can adjust it to your project's specifications.

SparkFun Qwiic Cable Kit

KIT-15081

Qwiic Cable - 100mm

PRT-14427

Qwiic Cable - 50mm
◉ PRT-14426

Qwiic Cable - 200mm
○ PRT-14428

Heads up! If you are using the RedBoard **without** a Qwiic connector, we recommend getting the Qwiic Shield for Arduino.

SparkFun Qwiic Shield for Arduino
◉ DEV-14352

Suggested Reading

If you aren't familiar with the Qwiic system, we recommend reading here for an overview.

Qwiic Connect System

We would also recommend taking a look at the following tutorials if you aren't familiar with them.

GPS Basics

The Global Positioning System (GPS) is an engineering marvel that we all have access to for a relatively low cost and no subscription fee. With the

Serial Peripheral Interface (SPI)

SPI is commonly used to connect microcontrollers to peripherals such as sensors, shift registers, and SD cards.

correct hardware and minimal effort, you can determine your position and time almost anywhere on the globe.

I2C

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

How to Work with Jumper Pads and PCB Traces

Handling PCB jumper pads and traces is an essential skill. Learn how to cut a PCB trace, add a solder jumper between pads to reroute connections, and repair a trace with the green wire method if a trace is damaged.

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.

ESP32 Thing Plus Hookup Guide

Hookup guide for the ESP32 Thing Plus using the ESP32 WROOM's WiFi/Bluetooth system-on-chip in Arduino.

What is Dead Reckoning?

Dead Reckoning is the process of determining current position by combining previously determined positional data with speed and heading. This process can also be applied to determine *future* positions as well! The ZED-F9K uses Dead Reckoning which calculates speed and heading (amongst many other points of data) through the use of an internal **inertial measurement unit** (IMU). The addition of an wheel ticks, RTCM-formatted corrections, and IMU allows the ZED-F9K to produce high precision and more accurate readings in between GNSS data refreshes!

In addition, the module can also give accurate and useful GNSS data in areas where satellite connections are difficult to maintain: areas like the dense urban environments of major cities, long tunnels, parking garages, any large UFO's that may descend from the sky, etc.

Dead Reckoning Overview

As mentioned in the "What is Dead Reckoning?" section, the u-blox F9K module has an internal inertial measurement unit or **IMU** for short. The IMU calculates position based on the last GNSS refresh and its own movement data points. To use the SparkFun GPS-RTK Dead Reckoning Board, there are a few guidelines to orienting and mounting the module to a vehicle that are outlined from u-blox. For more detailed information, check out the integration manual for mounting.

ZED-F9K INTEGRATION MANUAL (PDF)

Orientation for the SparkFun Dead Reckoning

The SparkFun Dead Reckoning adheres to two particular frames of reference: one frame of reference for the car and the second a geodetic frame of reference anchoring it to the globe. The latter, known as the **local level frame** uses the following as its' axes:

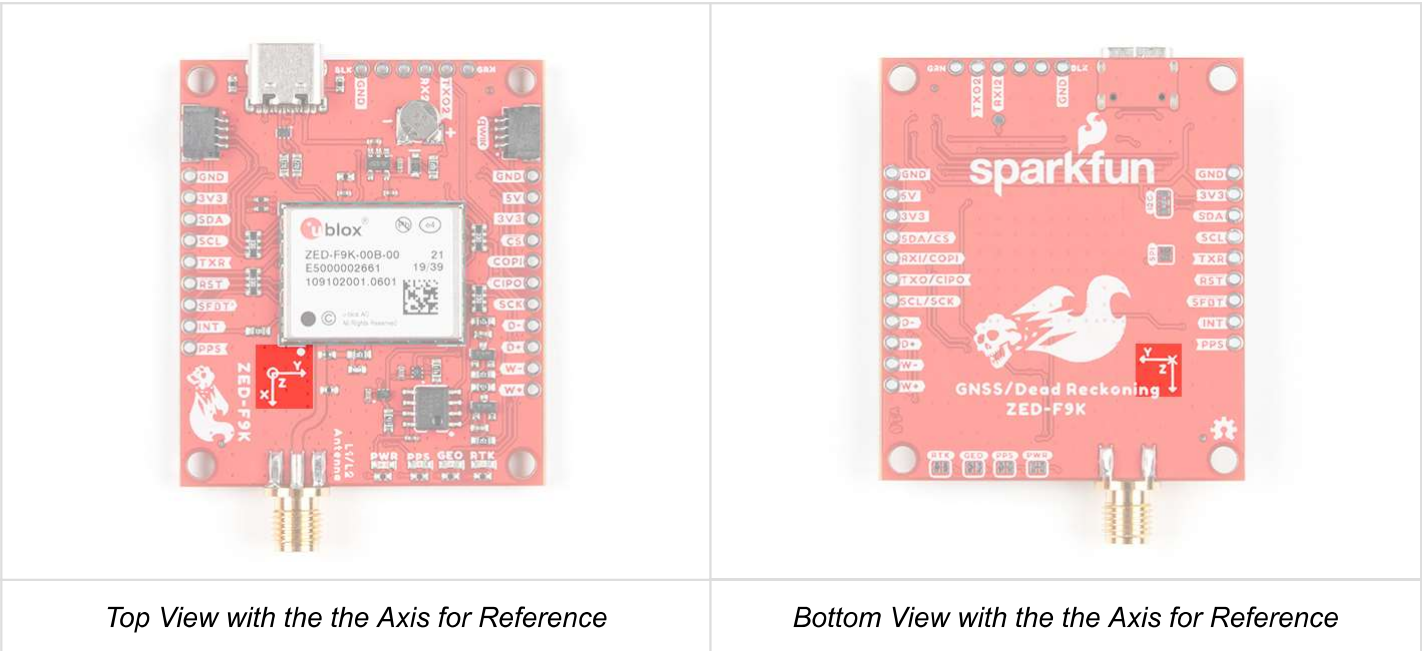
- **X-axis** points to the **North**
- **Y-axis** points to the **East**
- **Z-axis** uses the right hand system by pointing **down**.

This frame will be referred to by its acronym **NED** (North-East-Down) in the image below.

The second frame of references is the **Body-Frame** reference and uses the following as *its'* axes.

- **X-axis** points to the **front** of the vehicle
- **Y-axis** points to the **right** of the vehicle
- **Z-axis** uses the right hand system by pointing **down**.

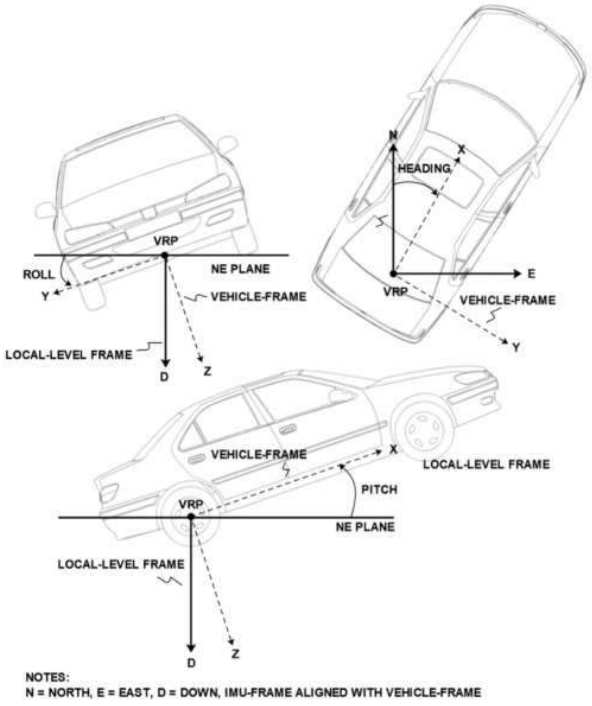
You can reference the **Body-Frame** axes directly on the SparkFun Dead Reckoning ZED-F9K breakout board by looking for the silkscreen with the xyz axis.



Top View with the the Axis for Reference | *Bottom View with the the Axis for Reference*

Vehicle Attitude

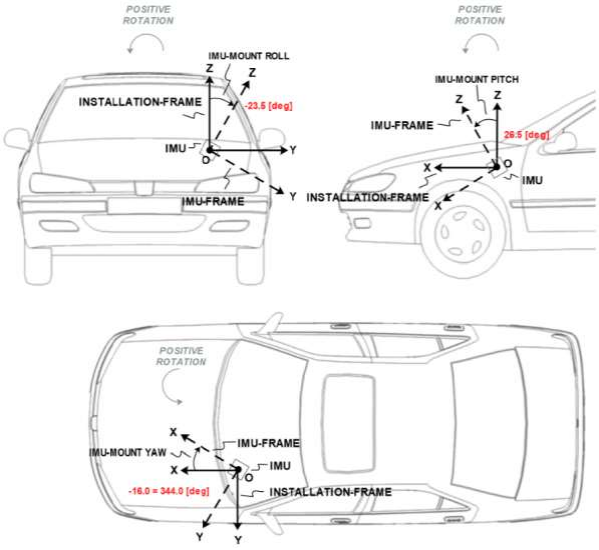
The transformation of the vehicle within these two frames are given as **heading**, **pitch**, and **roll**. In the datasheet these three angles are called the **vehicle's attitude**. Below is an image that illustrates how all of these elements fit together.



Vehicle attitude output as shown on page 30 of the Integration Manual (UBX-20046189-R02)

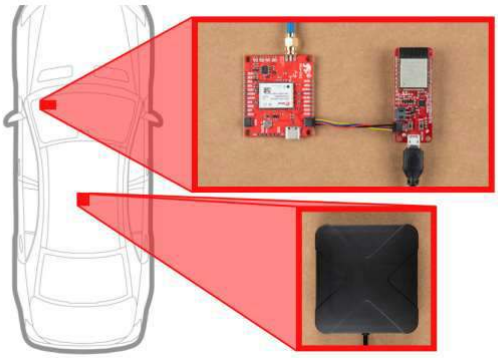
Mounting the SparkFun Dead Reckoning ZED-F9K

The only guideline here is that the SparkFun Dead Reckoning is stable within 5 degrees, and of course that the X-axis points towards the front of the car as mentioned above. Below is an image that illustrates the ZED-F9K with the installation frame as opposed to the vehicle frame.



Installation frame as shown on page 19 of the Integration Manual (UBX-20046189-R02)

With the physical board mounted, it will look similar to the following image.



In the images above, the SparkFun Dead reckoning is seen in the front, driver's side of the car and it may be tempting to think that this is also a necessary requirement. However, it can be mounted anywhere within the vehicle (or RC-car, or boat). Keep in mind that the **pitch** and **roll** is relative to the SparkFun Dead Reckoning's position.

Calibration

After you've mounted the SparkFun Dead Reckoning ZED-F9K, there is still a calibration phase to complete that must satisfy the following movements:

- First, the car needs to be stopped with the engine turned on.
- Secondly, the car must do left and right hand turns.
- Lastly, the car must reach a speed over **30 km/h**.

In SparkFun's u-blox Arduino library, SparkFun has included the calibration example, that prints out the module's calibration status.

Hardware Overview

> **Note:** Overall, the ZED-F9K module's specifications are similar to the ZED-F9R module. The difference is that the ZED-F9K is rated as automotive grade while the ZED-F9R is professional grade. For more information about the different product grades, check out the page on u-blox Product Grades.
>
> Comparing the breakout boards, the ZED-F9K breakout board includes an SMA connector instead of the u.FL connector.

Power

Power for this board is **3.3V** and we have provided multiple power options. This first and most obvious is the **USB-C connector**. Secondly, are the **Qwiic Connectors** on the left and right of the board. Thirdly, there is a **5V pin** on the PTH header along the side of the board that is regulated down to **3.3V**. Make sure that power you provide to this pin does *not* exceed 6 volts. Finally, just below the 5V pin is a **3.3V** pin that should only be provided a clean 3.3V power signal.

Battery

The small metal disk in the upper right corner next to the Qwiic connector is a small lithium battery. This battery does not provide power to the IC like the 3.3V system does, but to relevant systems *inside* the IC that allow for a quick reconnection to satellites. The time to first fix will about **~26 seconds**, but after it has a lock, that battery will allow for a **two second** time to first fix. This is known as a **hot start** and lasts for four hours after the board is powered down. The battery provides over a years worth of power to the backup system and charges slowly when the board is powered. To charge it to full, leave your module plugged in for 48 hours.

LEDs

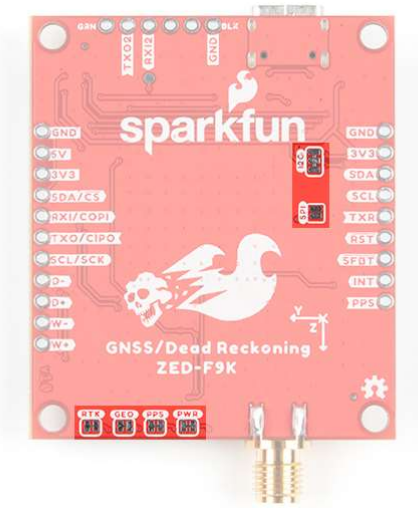
There are four LEDs on the bottom left of the board. Starting from the left:

- **PWR:** The power LED labeled as `PWR` will illuminate when 3.3V is activated.
- **PPS:** The pulse per second LED labelled as `PPS` will illuminate each second once a position lock has been achieved. This generates a pulse that is synchronized with a GPS or UTC time grid. By default, you'll see one pulse a second.
- **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-F9K is getting RTCM from various sources. Once an RTK fix is obtained, the LED will turn off.
- **GEO:** The `GEO` LED can be configured to turn on/off for geofencing applications.

Jumpers

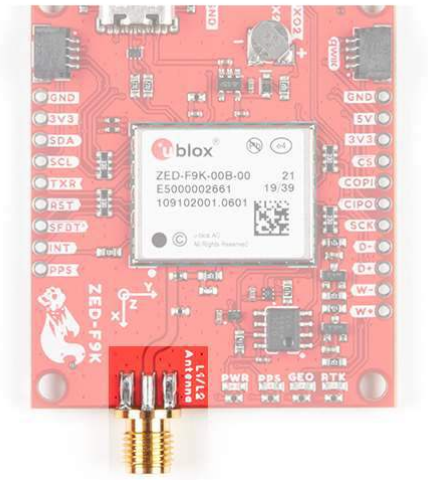
If you flip the board over, you will notice a few jumper pads. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.

- **I²C**: This three way jumper labeled `I²C` connects two pull-up resistors to the I²C data lines. If you have many devices on your I²C data lines, then you may consider cutting these.
- **SPI**: The jumper labeled `SPI` which enables the SPI data bus thus disabling the UART functions on those lines. This also disables I²C interface.
- **PWR**: Starting from the right side is a jumper labeled `PWR` . If you cut this trace, it will disconnect the **Power** LED.
- **PPS**: On the left of the jumper is the `PPS` jumper that when cut disconnects the **PPS** LED.
- **GEO**: Cutting the `GEO` jumper disconnect the LED used to indicate when we reach a certain condition for geofencing applications.
- **RTK**: The `RTK` jumper disconnects the LED used for RTK applications.



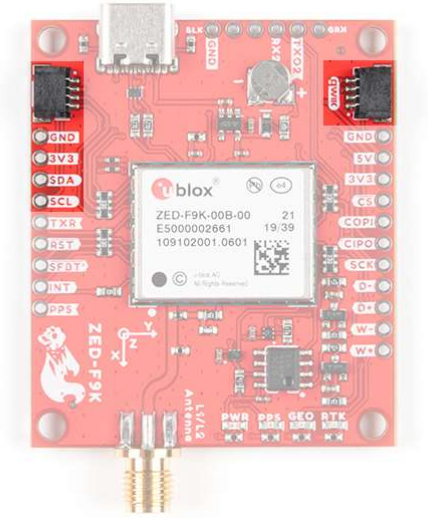
SMA Connector

The ZED-F9K requires a good quality GPS or GNSS (preferred) antenna. A SMA connector is provided for a secure connection. To get the best out of your ZED-F9K, make sure to get a GNSS multi-band antenna.



Qwiic and I²C

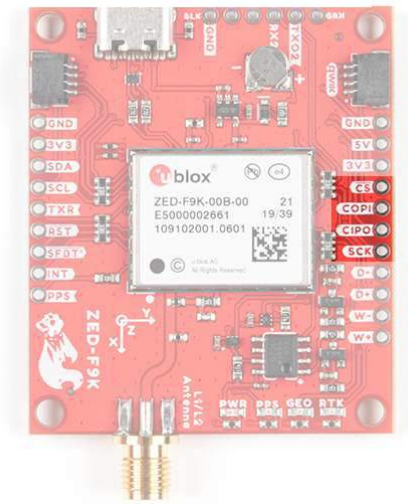
There are two pins labeled `SDA` and `SCL` which indicates the I²C data lines. Similarly, you can use either of the Qwiic connectors to provide power and utilize I²C. The Qwiic ecosystem is made for fast prototyping by removing the need for soldering. All you need to do is plug a Qwiic cable into the Qwiic connector and voila!



> The only I²C address for this and all u-Blox GPS products is **0x42**, though each can have their address changed through software.

SPI

There are four pins on the right most header that are labeled with their corresponding SPI functionality. As mentioned in the jumpers section, you'll need to close the `SPI` jumper on the underside to enable SPI.



UART

There are two pins on the right most header currently labeled as MISO and MOSI. These are shared with the UART pins. By default, the UART interface is enabled. Be sure that the DSEL jumper on the back of the board is open.

- TX/MISO = TX out from ZED-F9K
- RX/MOSI = RX into ZED-F9K

Top View

Bottom View

There is a second serial port available on the ZED-F9K. This 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 like the other RTK breakout boards for the NEO-M8P-2 and ZED-F9P. The RTCM Correction port pins are arranged to match the industry standard serial connection (aka the 'FTDI' pinout). This pinout is compatible with our Serial Basic so you can send RTCM correction data from a computer. Note that RTCM3 data can also be sent over I2C, UART1, SPI, or USB if desired.

Wheel Tick and Direction Pins

For advanced users that are interested in taking advantage of your vehicle's sensor readings, you can connect the following pins. Caution is advised however as this requires you to open up up the hood of your car and hack into the its system.

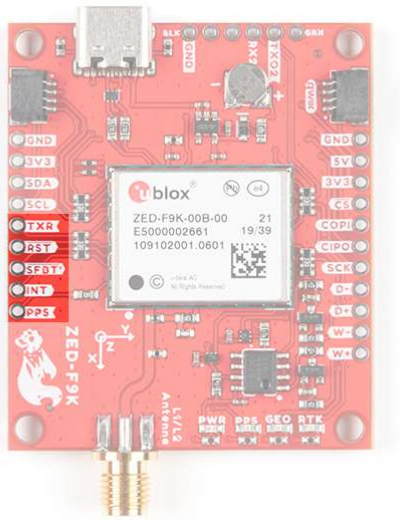
- **D-**: The reference GND pin (`D-`) when connecting the direction pin.
- **D+**: The direction pin is labeled as (`D+`) tells the ZED-F9K what direction the vehicle is moving (forward/reverse).
- **W-**: The reference GND pin (`W-`) when connecting the wheel tick pin.
- **W+**: The wheel tick pin (`W+`) tells the ZED-F9K the distance a vehicle's wheel has traveled. Depending on the odometer type that you connect to, the ZED-F9K can also receive speed data from the vehicle.

Broken Out Pins

There are four other pins broken out:

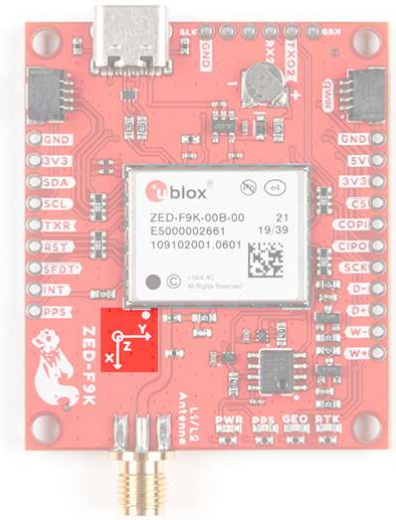
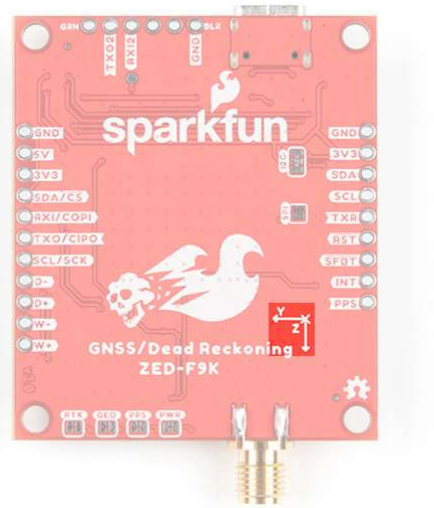
- **TXR**: The transmit ready pin (`TXR`) enables a port to notify a device when bytes are ready to be transmitted.
- **$\overline{\text{RESET}}$**: The reset pin ($\overline{\texttt{RESET}}$) resets the chip.
- **$\overline{\text{SFBT}}$**: The safeboot pin ($\overline{\texttt{SFBT}}$) is used to start up the IC in safe boot mode, this could be useful if you somehow manage to corrupt the module's Flash memory.
- **INT**: The interrupt pin (`INT`) can be used to wake the chip from power save mode.

- **PPS**: The pulse per second pin (`PPS`) outputs pulse trains synchronized with the GPS or UTC time grid. The signal defaults to once per second but is configurable over a wide range. Read the **u-blox Receiver Protocol Specification** in the Resources and Going Further tab for more information.



3D IMU Orientation and Reference

For easy reference, we've documented the IMU's vectors with 3D Cartesian coordinate axes on the top and bottom side of the board. Make sure to orient and mount the board correctly so that the ZED-F9K can accurately calculate navigation information. Remember, it's all relative.

	
9DoF Reference (Top View)	*9DoF Reference (Bottom View)*

GPS Capabilities

The ZED-F9K is able to connect to up to four different GNSS constellations simultaneously with the 3D gyro and 3D accelerometer making it very accurate for its size. Below are the listed capabilities of the GPS unit.

Parameter	Specification	
Max navigation update rate (RTK)	Priority navigation mode	30 Hz

	Non-Priority navigation mode	2 Hz
Velocity Accuracy		0.05m/s
Dynamic Attitude Accuracy	Heading	0.2 degrees
	Pitch	0.3 degrees
	Roll	0.5 degrees
Navigation Latency	Priority Navigation Mode	0.5 degrees
Max Sensor Output Rate		100Hz

	GNSS	**GPS+GLO+GAL +BDS**	**GPS+GLO+GAL**	**GPS+GAL**	**GPS+GLO**	**BDS+GLO**
Time-To-First-Fix	Cold Start	26s	25s	30s	25s	28s
	Hot Start	2s	2s	2s	2s	2s
	Aided Start	3s	3s	3s	3s	3s
Re-convergence time	RTK	≤ 10s	≤ 10s	≤ 10s	≤ 10s	≤ 30s
Sensitivity	Tracking and Navigation	-160dBm	-160dBm	-160dBm	-160dBm	-160dBm
	Reacquisition	-157dBm	-157dBm	-157dBm	-157dBm	-157dBm
	Cold Start	-147dBm	-147dBm	-147dBm	-147dBm	-145dBm
	Hot Start	-158dBm	-158dBm	-158dBm	-158dBm	-158dBm
Position Accuracy RTK	Along Track	0.20m	0.20m	0.25m	0.25m	0.60m
	Cross Track	0.20m	0.20m	0.25m	0.25m	0.60m
	2D CEP	0.30m	0.30m	0.40m	0.40m	0.85m
	Vertical	0.30m	0.30m	0.40m	0.40m	1.00m

Performance in Different GNSS Modes from the ZED-F9K Datasheet

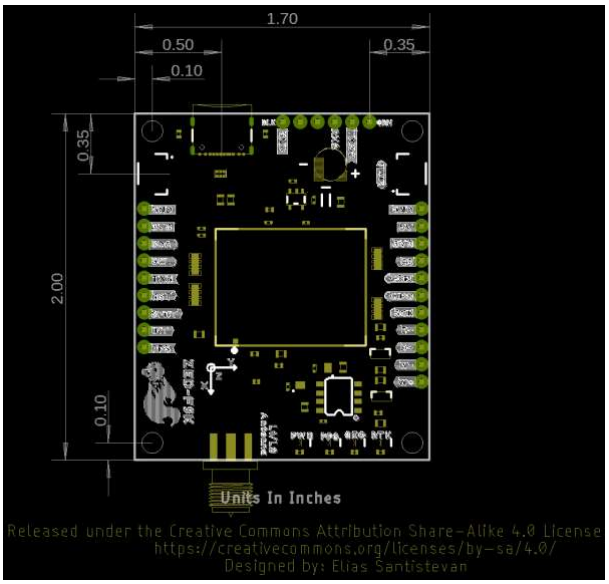
	GNSS	**GPS**	**GLONASS**	**BeiDou**	**Galileo**
Time-To-First-Fix	Cold Start	30s	28s	40s	-

	Hot Start	2s	2s	2s	-
	Aided Start	3s	3s	3s	-
Sensitivity	Tracking and Navigation	-158dBm	-158dBm	-158dBm	-156dBm
	Reacquisition	-157dBm	-155dBm	-157dBm	-153dBm
	Cold Start	-147dBm	-147dBm	-141dBm	-137dBm
	Hot Start	-158dBm	-157dBm	-158dBm	-155dBm
Position Accuracy RTK	2D CEP	0.80m	1.00m	-	1.50m
	Vertical	1.00m	1.50m	-	2.00m

Performance in Single-GNSS Modes from the ZED-F9K Datasheet

Board Dimensions

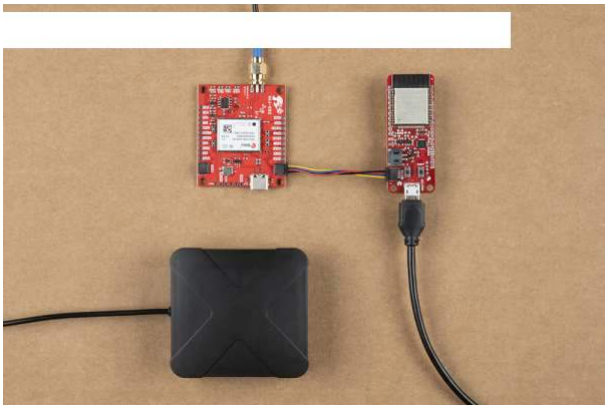
Overall, the board is about 2.00" x 1.70" (50.8mm x 43.2mm). With the USB and SMA connector, it's about 2.40" 1.70" (61.0mm x 43.2mm). There are 4x mounting holes by each corner of the board.



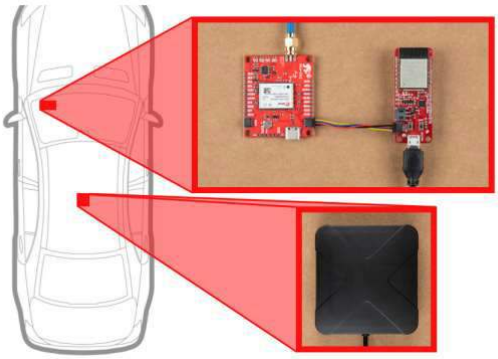
Hardware Assembly

For this example, I used a SparkFun Thing Plus - ESP32 WROOM and associated USB cable. Connecting the boards with Qwiic cable, the assembly is very simple. Plug a Qwiic cable between the Thing Plus - ESP32 WROOM and SparkFun GPS RTK Dead Reckoning ZED-F9K. Then plug in one of our patch antennas to the SMA connector If you're going to be soldering to the through hole pins for I^2C functionality, then just attach lines to power, ground, and the I^2C data lines to a microcontroller of your choice.

When using the ZED-F9K, you will want to orient the board according to the guidelines explained earlier. Below is a top-down view with the board pointing up. Your setup should look similar to the image below.



Make sure to secure the board above a vehicle's dashboard using some tape or sticky tack when prototyping and testing. For best signal reception, it is suggested to guide the antenna from the inside of the car and through a window before attaching the GPS on top of a car. We recommend the magnetic mount GPS/GNSS antenna to easily mount.



Adding a RTCM Correction Source

To get the most out of the ZED-F9K you will need an RTCM correction source. There are a few methods of adding a RTCM correction source. If you have been following along with our GPS-RTK and GPS-RTK2, you can pipe correction data from a wireless network, LoRa, or cellular network. Depending on your setup, you will probably need a ZED-F9P for a correction source. For more information, check out these tutorials.

What is GPS RTK?
SEPTEMBER 14, 2018

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

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.

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.

All of our u-blox based GPS boards share the same library: the breakout board, their predeccesors and the higher precision u-blox cousins. 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 to manually install. Once calibrated, you can take advantage of the examples for the ZED-F9K.

SPARKFUN U-BLOX ARDUINO LIBRARY (ZIP)

There are 13 example sketches provided to get you up and receiving messages from space. The examples listed below highlight the *additional* capabilities of the SparkFun Dead Reckoning ZED-F9K. For the scope of this tutorial, we will not focus on the basic GPS polling sketches as shown in the other u-blox hookup guides.

Note: Example 2 uses the '**MicroNMEA**' library by **Steve Marple**. Make sure to install the library as well by searching for it in the Arduino library manager. You could also grab the zip here from the GitHub repository to manually install.

MICRONMEA ARDUINO LIBRARY (ZIP)

Arduino Examples

Heads up! Make sure stay focused when driving and obey all traffic laws when driving. Ensure that you are not distracted while operating a vehicle and are aware of your surroundings. For each example, make sure to plan your route accordingly to achieve the maneuvers. If you need, grab a friend and adult as you ride in the passenger seat. After you attempt these movements, park your car in a safe location and turn your engine off before checking the status or monitoring the readings!

Example 1 - Calibrate Sensor

Now that the GPS-RTK SparkFun Dead Reckoning is mounted and oriented correctly with regards to the vehicle, it's time to calibrate the sensor. To do this, a few movements with the vehicle must be done all while maintaining good GNSS reception.

- First, the car needs to be stopped with the engine turned on.
- Secondly, the car must do left and right hand turns.
- Lastly, the car must reach a speed over **30 km/h**.

For the first example (located in **File Examples > SparkFun u-blox GNSS Arduino Library** > **Dead Reckoning** > **Example1_calibrateSensor**), the **calibration** status will be printed to the Arduino's serial monitor to indicate when calibration is ongoing and when it has completed.

If you have not already, select your Board (in this case the **SparkFun ESP32 Thing Plus**), and associated COM port. Upload the code to the board and and set the serial monitor to **115200 baud.** Perform those fancy maneuvers (while obeying the traffic laws) before parking your car in a safe location. Then turn your engine off before

checking the status! You should see a message indicating that the ZED-F9P is calibrated. If you do not, try driving around with the board once again!

Example 2 - IMU Data

After you have your sensor calibrated (see example1), you can now poll the internal IMU to see what data is being fed to the GNSS calculations. Open the second example (located in **File Examples > SparkFun u-blox GNSS Arduino Library** > **Dead Reckoning** > **Example2_getIMUData**) to follow along! First, the sketch checks to see that the board is calibrated before it attempts to read IMU data.

If you have not already, select your Board (in this case the **SparkFun ESP32 Thing Plus**), and associated COM port. Upload the code to the board and set the serial monitor to **115200 baud**. This may be a good time to bring a friend along to drive if you decide to actively monitor the output. Otherwise, check out the data after taking the board for a stroll. Try driving around as the board senses the car's movement. Then park in a safe location with the engine turned off before inspecting the data.

Example 4 - Vehicle Dynamics

What happened to Example 3? It's been skipped over because its used primarily as a diagnostic sketch. What sensors are currently being used, are they functioning correctly, are the measurements being listed as bad or non-existent? Example 3 helps diagnose these various issues. Lets move ahead to the fourth example in the library (located in **File Examples > SparkFun u-blox GNSS Arduino Library** > **Dead Reckoning** > **Example4_vehicleDynamics**)

The **vehicle attitude** is a termed coined by u-blox that encompasses three measurements: **vehicle pitch**, **vehicle roll**, and **vehicle heading**. Much like the other example sketches, this one checks to make sure that the SparkFun Dead Reckoning ZED-F9K has been calibrated before pulling data. If the SparkFun Dead Reckoning ZED-F9K has indeed been calibrated, then it gets the relevant information by calling `myGNSS.getVehAtt()`. As in Example 2, the data is stored within a *struct* called **vehAtt**.

If you have not already, select your Board (in this case the **SparkFun ESP32 Thing Plus**), and associated COM port. Upload the code to the board and set the serial monitor to **115200 baud**. This may be a good time to bring a friend along to drive if you decide to actively monitor the output. Otherwise, check out the data after taking the board for a stroll. Try driving around as the board senses the car's movement. Then park in a safe location with the engine turned off before inspecting the data.

Connecting the GPS-RTK to a Correction Source

To get the most out of the ZED-F9K you will need an RTCM correction source. There are a few methods of adding a RTCM correction source. If you have been following along with our GPS-RTK and GPS-RTK2, you can pipe correction data from a wireless network, LoRa, or cellular network. Depending on your setup, you will probably need a ZED-F9P for a correction source. For more information, check out these tutorials.

What is GPS RTK?

SEPTEMBER 14, 2018

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

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 GPS receiver up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic
- Eagle Files
- Board Dimensions
- ZED-F9K Datasheet
- Product Summary
- Integration Manual
- u-blox Interface Description
- u-blox ECCN

- u-center Software
- Arduino Library
- GitHub Hardware Repo

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

Getting Started with the GeoFence

How to get started using the GeoFence GPS Boundary Widget and GeoFence Software.

Getting Started with the Autonomous Kit for the Sphero RVR

Want to get started in robotics? Look no further than the SparkFun autonomous kit for the Sphero RVR! Whether you purchased the Basic or Advanced kit, this tutorial will get you rolling...

SparkFun GPS Dead Reckoning NEO-M8U Hookup Guide

The u-blox NEO-M8U is a powerful GPS units that takes advantage of untethered dead reckoning (UDR) technology for navigation. The module provides continuous positioning for vehicles in urban environments and during complete signal loss (e.g. short tunnels and parking garages). We will quickly get you set up using the Qwiic ecosystem and Arduino so that you can start reading the output!

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!

Or check out this blog post for ideas.

GPS-RTK, Down the River
AUGUST 18, 2020

Positional Monitoring without GNSS?
MARCH 17, 2021

Ped Dead Reckoning II: This Time it's Inertial
APRIL 28, 2021