Product Manual - SparkPNT SXT(-D)

Table of contents:

Welcome

- Get Started
- Quick Start
- Product Overview
- Hardware Overview
- Hardware Assembly
- Software Overview
- Disciplined Oscillator
- Fugro AtomiChron®
- Enable the L5 Band
- Support

Product Overview

- Variants
    - SparkPNT SXT (GNSSSDO)
    - SparkPNT SXT-D (GNSSSDO+)
- Parts List
- Specifications
    - Features
    - Connections
    - Schematic
    - Dimensions
- How It Works
    - GNSS Signal Acquisition and Warm Up
    - Frequency Locking
    - Phase Locking (Frequency Ramping)
    - Phase Lock
    - Error States

Hardware Overview

- Power Options
- Septentrio logo mosaic-T
- Espressif logo ESP32-WROVER
- Ethernet PHY Interface
- USB-C Connectors
- µSD Socket
- SMA Connectors

- Input/Output Terminals
- Switches
- Status LEDs
- OLED Display
- Buttons
- Jumpers (SXT)
- Jumpers (SXT-D)

Hardware Assembly

- USB-C Ports
- GNSS Antenna
- Ethernet Jack
- 10MHz Signal
- SD Card Slot
- IO Terminals

Software Overview

- Septentrio logo mosaic-T
  - RxTools Software Suite
    - Installation Instructions
  - Septentrio USB Driver
  - Web Interface
- Espressif logo ESP32
  - CH340 USB Driver
  - Terminal Emulator
- Software Settings
  - Configure Operation
  - TCP Server (IPS1)
  - Debug Software
  - Print Conditions (periodic messages)
- Firmware Upgrade
- Compiling Firmware - using Docker
  - Clone, fork or download the GNSSDO repo
  - Install Docker Desktop
  - Using Docker to create the firmware binary
- Uploading Firmware - using esptool
- Footnotes

Disciplined Oscillator

- SXT: SiTime SiT5358 Disciplined Oscillator

- SXT-D: Rakon STP3593LF Double-Oven Disciplined Oscillator
- Software Control Loop

Fugro AtomiChron®

- Specifications
- Subscribing to Fugro AtomiChron®
- Performance
- Firmware configuration

Enable the L5 Band

- Web Interface
- Command Interface

Support

- Troubleshooting Tips
- 🗃️ Resources

Troubleshooting Tips

- What is in the Box?
- OLED Dead Pixels
- Data Logging
- L5 Band
- Reprogramming the ESP32
- Enclosure Disassembly
- SXT Clock Frequency Reference
- SXT-D Power Supply

Resources

- Product Resources
- Product Comparison

Product Resources

- Product Specific Resources
- SparkPNT SXT (GNSSDO)
- SparkPNT SXT-D (GNSSDO+)
- Additional Resources
    - 🏭 Manufacturer's Resources

Product Comparison

Welcome

This is the online product manual for the SparkPNT SXT (GNSSSDO) and SXT-D (GNSSSDO+); high-performance, precision, 10MHz GNSS disciplined oscillators with dedicated timing features for time and clock synchronisation. If this is your first time using the SXT(-D), follow check out our Quick Start Guide for initial instructions on setting up the SXT(-D). Again, thank you for purchasing our SparkPNT SXT(-D)!

Designed and manufactured in Boulder, Colorado, USA, the SparkPNT SXT (GNSSSDO) and SXT-D (GNSSSDO+) are the perfect solution for your high-precision timing needs. Based around the multi-constellation, multi-frequency, L5-ready mosaic-T from Septentrio, these are our most accurate GNSS timing product to date. The SXT features a SiTime SiT5358 disciplined 10MHz Digitally-Controlled Temperature-Controlled Crystal Oscillator (DCTCXO) providing excellent clock accuracy and stability. The SXT-D gets an upgrade from the Rakon STP3593LF disciplined 10MHz Digitally-Controlled Double-Oven Crystal Oscillator (OCXO) providing superb clock accuracy, stability and holdover. The mosaic-T also has built-in on-module support for the Fugro AtomiChron® L-band timing service.

Under the hood, the SXT(-D) is based on the mosaic-T GNSS module from Septentrio, plus the Espressif ESP32-WROVER processor (16MB flash, 8MB PSRAM). The mosaic-T has USB-C connectivity (with Ethernet-over-USB), multiple UARTs and supports full Ethernet connectivity. You can connect the mosaic-T directly to your Ethernet network - our product supports Power-over-Ethernet too. The ESP32 rides shotgun, disciplining the oscillator and controlling the OLED display. The SXT(-D) has multiple power options including USB-C, PoE and 9V-36V DC, making it easy to connect them to a battery-backed supply. Robust SMA connections and screw cage terminals provide access to the Pulse-Per-Second and 10MHz clock signals.

> **RTK NOT SUPPORTED**
>
> The mosaic-T is a superb GNSS module designed for highly accurate timing applications. **However, RTK is not supported.**

Get Started

Quick Start

3 items

Product Overview

A high-level overview of the kit, design specifications, and its functionality

Hardware Overview

A detailed overview of the device's interfaces and components

Hardware Assembly

A setup guide and instructions for connecting to the device interfaces

Software Overview

A guide on the software used to configure and operate the device

Disciplined Oscillator

A guide on the software control loop for the disciplined oscillator

Fugro AtomiChron®

A guide for the service authentication and clock synchronisation

Enable the L5 Band

A guide on how to enable the use of GPS signals on the L5 band

Support

Resources for technical support and customer service inquiries

Product Overview

Designed and manufactured in Boulder, Colorado, USA, the SparkPNT SXT (GNSSSDO) and SXT-D (GNSSSDO+) are the perfect solution for your high-precision timing needs. Based around the multi-constellation, multi-frequency, L5-ready mosaic-T from Septentrio, these are our most accurate GNSS timing product to date. The SXT features a SiTime SiT5358 disciplined 10MHz Digitally-Controlled Temperature-Controlled Crystal Oscillator (DCTCXO) providing excellent clock accuracy and stability. The SXT-D gets an upgrade from the Rakon STP3593LF disciplined 10MHz Digitally-Controlled Double-Oven Crystal Oscillator (OCXO) providing superb clock accuracy, stability and holdover. The mosaic-T also has built-in on-module support for the Fugro AtomiChron® L-band timing service.

Under the hood, the SXT(-D) is based on the mosaic-T GNSS module from Septentrio, plus the Espressif ESP32-WROVER processor (16MB flash, 8MB PSRAM). The mosaic-T has USB-C connectivity (with Ethernet-over-USB), multiple UARTs and supports full Ethernet connectivity. You can connect the mosaic-T directly to your Ethernet network - our product supports Power-over-Ethernet too. The ESP32 rides shotgun, disciplining the oscillator and controlling the OLED display. The SXT(-D) has multiple power options including USB-C, PoE and 9V-36V DC, making it easy to connect them to a battery-backed supply. Robust SMA connections and screw cage terminals provide access to the Pulse-Per-Second and 10MHz clock signals.

> **RTK NOT SUPPORTED**
>
> The mosaic-T is a superb GNSS module designed for highly accurate timing applications. **However, RTK is not supported.**

Variants

SparkPNT SXT (GNSSSDO)

With excellent clock accuracy and stability, the SXT features a SiTime SiT5358 disciplined 10MHz Digitally-Controlled Temperature-Controlled Crystal Oscillator (DCTCXO) and supports the Fugro AtomiChron L-band timing service.

Purchase from SparkPNT

SparkPNT SXT-D (GNSSSDO+)

With superb clock accuracy, stability and holdover, the SXT-D features an upgraded Rakon STP3593LF disciplined 10MHz Digitally-Controlled Double-Oven Crystal Oscillator (OCXO) and supports the Fugro AtomiChron L-band timing service.

Purchase from SparkPNT

PRODUCT COMPARISON

Check out our product comparison page for a basic table listing the basic features of the SparkFun mosaic-T GNSS breakout board, Septentrio development kit, SXT, and SXT-D kits.

Parts List

The SXT / SXT-D comes with everything you need to get up and running.

Everything that is included in the SparkPNT SXT(-D) kit.

- Quick Start Guide
- Cased GNSS Receiver
  - Aluminum Enclosure
  - 10-Way Terminal Blocks
  - Qwiic 1.3" OLED Display
- L1/L2/L5 GNSS Surveying Antenna

- Reinforced RG58 TNC-SMA Cable (10m)
- 32GB microSD Card (Class 10)
- USB A and C Power Delivery (PD) Wall Adapter - 65W
- USB-C to USB-C Cable (Flexible Silicone, 3m)
- Ethernet Cable (CAT-6, 1m)
- SMA to BNC Cable (RG316, 1m)
- Silicone Bumpers - 5x11mm (4 Pack)

Specifications

Features

- Septentrio mosaic-T multi-constellation, multi-frequency GNSS timing receiver
    - Accurate and resilient with dedicated timing features
    - Highly secure against jamming and spoofing with AIM+ unique interference mitigation technology combined with Fugro AtomiChron services
    - Update rate of 10 Hz
    - 448 hardware channels for simultaneous tracking of all visible supported satellite signals:
        - GPS: L1C/A, L1PY, L2C, L2P, L5
        - GLONASS: L1CA, L2CA, L2P, L3 CDMA
        - Beidou: B1I, B1C, B2a, B2b, B2I, B3
        - Galileo: E1, E5a, E5b, E5 AltBoc, E6
        - QZSS: L1C/A, L1 C/B, L2C, L5
        - NavIC: L5
        - SBAS: Egnos, WAAS, GAGAN, MSAS, SDCM (L1, L5)
        - On-module L-band
    - Time pulse precision: 5ns
    - Time pulse precision with AtomiChron (L-Band or IP): <1ns
    - Event accuracy: <20ns
    - Operating temperature: -40 to 85 °C
    - USB-C interface (UART and Ethernet-over-USB)

- **RTK is not supported**
- ESP32-WROVER processor (16MB flash, 8MB PSRAM)
  - USB-C interface (UART via CH340)
- SXT (GNSSSDO): SiT5358 disciplined 10MHz oscillator
  - ±50ppb stability
  - ±1ppb/°C frequency slope
  - ±58ppb typical 20-year aging
  - Digital frequency pulling via I²C
  - Allan Deviation approaches 1E-14 at 10000 seconds with AtomiChron enabled
  - Operating temperature: -40 to 85 °C (Industrial)
- SXT-D (GNSSSDO+): STP3593LF disciplined 10MHz oscillator
  - Better than 50ppb frequency calibration at 25°C
  - ±0.2ppb per day frequency stability (aging)
  - ±0.03ppb frequency stability with temperature (-32°C to +70°C)
  - Digital frequency pulling via I²C
  - Allan Deviation below 1E-14 at 10000 seconds with AtomiChron® enabled
  - Operating temperature: -32 to 70 °C
- microSD socket
  - Connected directly to the mosaic-T for fast data logging
- OLED display
  - 128x64 pixels
- Status LEDs

Connections

- SMA Connections:
  - GNSS Antenna (L1/L2/L5/L-Band) - provides 3.3V for an active antenna
  - SXT (GNSSSDO):
    - 10MHz Output - disciplined, configurable for 5V / 3.3V / 2.8V / 1.8V and 50 Ohm
    - 10MHz Input - switchable, input impedance 50Ω, detection level -14dBm, max supported level +12dBm
    - Pulse-Per-Second Square Wave - configurable for 5V / 3.3V / 2.8V / 1.8V and 50 Ohm

    - SXT-D (GNSSSDO+):
        - 10MHz Output - disciplined, 3.3V
        - Pulse-Per-Second Square Wave - 3.3V, aligned to the 10MHz clock signal
    - EventA Input - configurable for 5V / 3.3V / 2.8V / 1.8V and 50 Ohm
- 3.5mm Screw Cage Connections:
    - 9V-36V DC input (isolated)
    - GND
    - mosaic-T COM2 (TX/RX/CTS/RTS) - 3.3V / 5V switchable
    - CTS can be configured as a 3.3V / 5V power output via a solder jumper
    - EventB input - 3.3V / 5V switchable
    - SXT (GNSSSDO):
        - I^2C (SCL2 / SDA2) for an external TCXO / OCXO - 3.3V / 5V switchable
    - SXT-D (GNSSSDO+):
        - mosaic-T GP1 and GP2 outputs - 3.3V / 5V switchable
- Ethernet:
    - KSZ8041NLI Ethernet PHY interface
    - 10Base-T / 100Base-TX with auto-negotiate and Auto MDI/MDI-X
- Power Options:
    - USB-C
        - SXT-D (GNSSSDO+): requires a Power Delivery Wall Adapter
    - Power-over-Ethernet (PoE)
    - 9V-36V DC input (fully isolated)

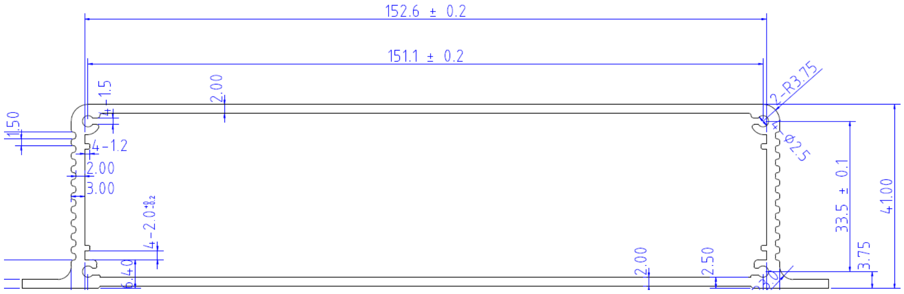
Schematic

Users can download the full schematics of the SXT and SXT-D in `*.pdf` format.

Dimensions

Metal Enclosure **Printed Circuit Board** **Antenna**

Details about the aluminum enclosure can be found on the Metal Enclosure - Custom Aluminum Extrusion (6in. x 4in. PCB) product page.



Dimensions (PDF) of the SXT(-D) aluminum enclosure and the front/rear panels, in millimeters.

How It Works

As the name suggests, the SparkPNT SXT GNSSDO (GNSS Disciplinned Oscillator) and SXT-D GNSSDO+ are very precise 10MHz GNSS disciplined oscillators. Inside, the Septentrio mosaic-T GNSS provides all-in-view satellite tracking: multi-constellation, and multi-frequency. It has dedicated timing features for time and clock synchronisation. Using the clock bias measurements from the mosaic-T, which it calculates by comparing the internal 10MHz clock to those of the GNSS satellites, the SXT(-D) is able to discipline (adjust) the frequency of the digitally-controlled oscillator to bring it as close to 10MHz as possible. Once the frequency has been adjusted, the accuracy of the mosaic-T's Pulse Per Second output is: better than 5ns; and better that 1ns with an optional subscription to Fugro's AtomiChron timing service. Take two GNSS disciplinned oscillators, place them anywhere on the planet and you can be confident that their time pulses are aligned to within 5ns or 1ns of each other...

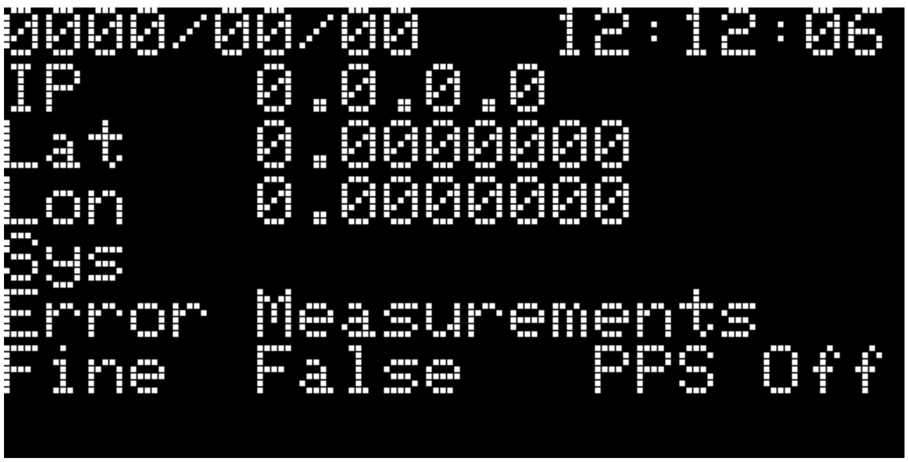
The mosaic-T PVT (Position Velocity Time) messages report the Receiver clock bias (RxClkBias) relative to the GNSS time system. If RxClkBias is positive, the receiver time is ahead of the GNSS system time. By reducing the oscillator frequency (ever so slightly), the clock bias can be reduced towards zero. If RxClkBias is negative, the oscillator frequency needs to be increased to minimise the bias. This is achieved by controlling the oscillator frequency using a classic Phase-Locked Loop, with Proportional and Integral (PI) control terms. On the SXT-D, we also include a Derivative term (PID) in the feedback loop.

GNSS Signal Acquisition and Warm Up

When you turn the SXT(-D) on, it needs to do two things: acquire a GNSS signal (so that the clock bias can be measured), and warm up the oscillator.

If you connect to the SXT(-D)'s `CONFIG ESP32` USB port and open a terminal emulator / serial console, you will see diagnostic data which tells you the status of the SXT(-D). You can find more information on how to do that in the quick start guide and in the software overview. The **Mode** column tells you which mode the SXT(-D) is in.

While the mosaic-T is acquiring a GNSS signal, the OLED display will show an Error such as "Measurements" or "Ephemerides":



OLED showing Error Measurements.

In the serial console CSV messages, the **Mode** will be shown as **CONFIGURED**. This indicates that the GNSS has been configured, but that it does not yet have a time solution.

Once the GNSS has acquired a signal and has a (preliminary) time solution: Error will change to "None"; Fine will change to "True" (indicating that the receiver time-of-week is within bounds).

With the SXT-D (GNSSSDO+), the firmware will now go into **Mode WARMUP**. It can take approximately 10 minutes for the STP3593LF double-oven oscillator temperature to stabilise.

OLED showing TCXO Warming Up.

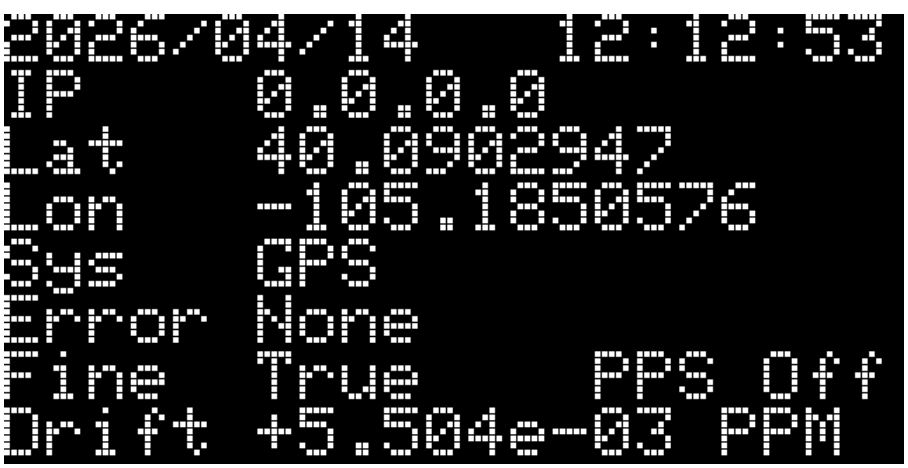
Warm-up is complete when the **Temp_Smooth** reaches the required 0.01°C stability.

Frequency Locking

> **NOTE**
>
> Frequency locking was introduced with GNSSDO firmware v3.0

After initial warm-up, the temperature-controlled oscillator is neither frequency-locked nor phase-locked. The first step is to ensure it is running at the correct frequency. This is acheived using a simple frequency-locked loop. The *change* in the Bias (RxClkBias) is used as the error term for the PI loop. The *change* in the Bias is driven to zero, by altering the oscillator frequency.

During this state, the OLED display will show "Drift" followed by the oscillator drift in PPM:



OLED showing Drift.

The **Mode** will be **FINETIME** during this state. I.e. a fine time solution has been obtained, and the firmware is now in the process of frequency locking.

The firmware will stay in this state until the required bias stability has been achieved. In the CSV messages, **Bias_Smooth** will reach the required 1.0e-10. The reported "Drift" will reach ~1e-05 PPM at about the same time.

Phase Locking (Frequency Ramping)

> **NOTE**
>
> Phase locking (frequency ramping) was introduced with GNSSDO firmware v3.0

Once the *change* in the Bias has been driven close to zero, we can be confident that the oscillator frequency is correct. (If the frequency was not correct, the Bias would change.) The next step is to ensure the oscillator phase is correct.

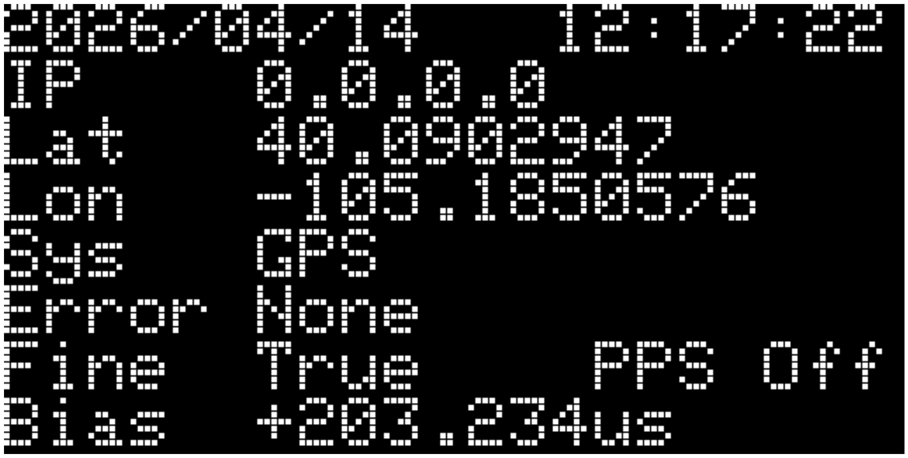
In v3.0 of the GNSSDO firmware, we set the mosaic-T **setClockSyncThreshold** (scst) `StartupSync` parameter to `Off`. In previous versions of the firmware, this was set to `On`. This ensured a small residual Bias when entering Finetime, but it also introduced an uncertainty in the PPS output with respect to the 10MHz clock.The only way to remove that uncertainty is to set the `StartupSync` parameter to `Off`. But, this means the initial clock Bias can be as large 500us.

You may be lucky and there may be only a small Bias left to remove after Frequency Locking. Worst case, there is a Bias approaching 500us to remove.

The large Bias is removed by ramping the oscillator frequency one way and then back the other way. If the initial bias is, say, +400us, the oscillator frequency must be ramped down, held, and then ramped up again. The firmware accelerates the change in frequency at: 1ns/s for the SXT, and 0.5ns/s for the SXT-D. This helps the PI loop to retain control and avoids blowing up the integrator.

Ramping the frequency is a slow process. It can take over 30 minutes, depending on the size of the initial error.

The OLED display will display the clock Bias. You can watch the Bias change, slowly at first, reaching a limit of: 500ns/s for the SXT; and 250ns/s for the SXT-D.



OLED showing Bias in microseconds.

The **Mode** will be **FREQUENCY_LOCK** during this state. I.e. frequency lock has been achieved, and the firmware is now in the process of phase locking.

The firmware will stay in this state until the ramps are complete. In the CSV messages, you can watch:

- **Rate** increase then decrease at: 1ns/s on SXT; 0.5ns/s on SXT-D
- **Rate_Held** is the capped rate: 500ns/s on SXT; 250ns/s on SXT-D
- **Setpoint** is the target Bias, set by the ramps
- **Turn** is the turn-around point, from the 'up' ramp to the 'down' ramp (this value is absolute - always positive)
- **Error** is the difference between the **Setpoint** and the RxClkBias
- **Repeat** is the number of times the ramps have been repeated: a minimum of once on SXT; twice on SXT-D

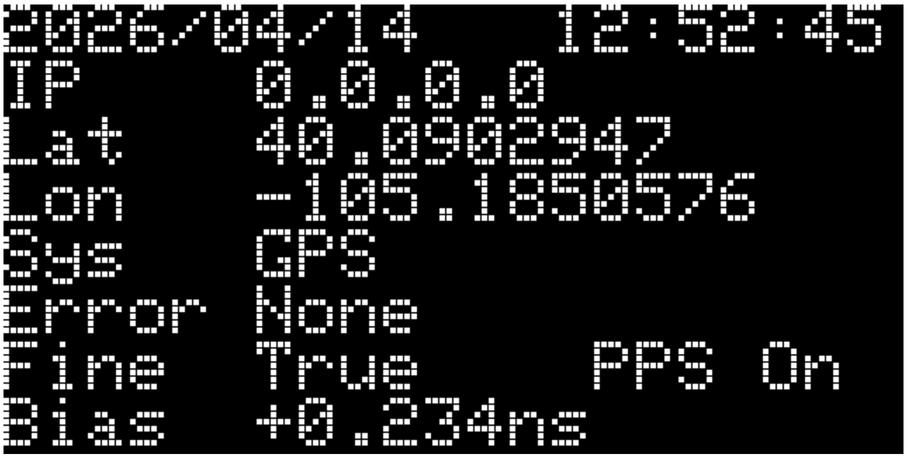
The firmware will exit **Mode FREQUENCY_LOCK** and enter **PHASE_LOCK** if the residual bias is low enough on completion of the 'down' ramp: 100ns on both SXT and SXT-D

Phase Lock

When the oscillator frequency and phase are correct, the firmware continues to drive the Bias towards zero using a PI(D) loop. The residual Bias should be within (approximately) +/-100ps on SXT and +/-10ps SXT-D. Please note that the reported Bias is relative, not absolute. The true absolute accuracy will be higher than this, but within the quoted 5ns or 1ns with Fugro AtomiChron.

The **Mode** will be **PHASE_LOCK** during this state.

PPS output is started when the firmware enters this state for the first time (since power-on).



OLED showing PPS On.

Error States

If the GNSS detects an Error, the firmware will: report the Error on the OLED display and in the CSV **Error** column; maintain the oscillator frequency until the error is cleared. The Error LED will also illuminate.

Once the error is cleared, the firmware will return to **FREQUENCY_LOCK**, nudge the oscillator frequency if required, and return to **PHASE_LOCK** as promptly as possible. The PPS output is not paused during error states; once started in **PHASE_LOCK**, PPS output is not stopped.

For more details, please refer to the Software Control Loop section.

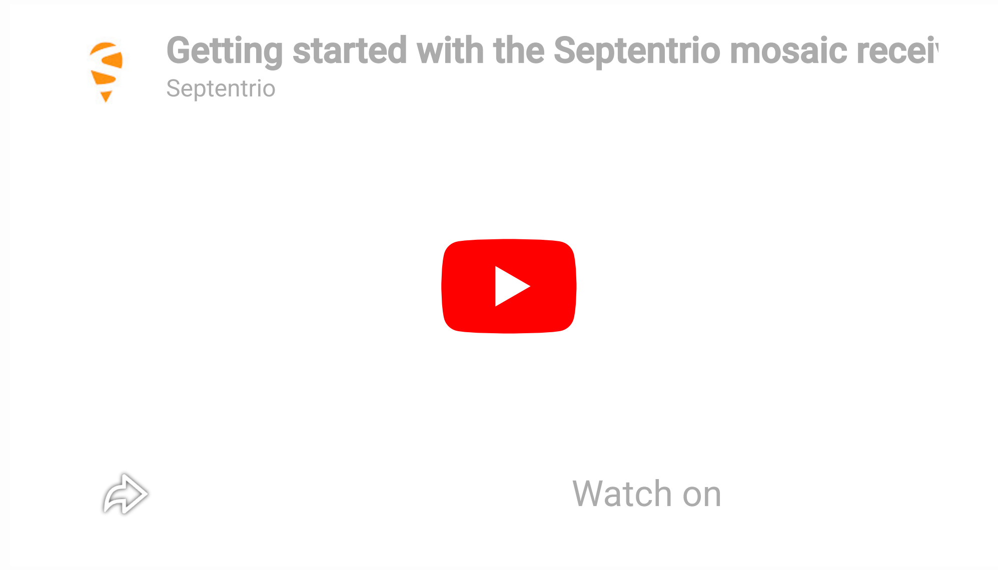
READ BEFORE HANDLING PCB!

ESD SENSITIVITY

The mosaic-T module is sensitive to ESD. Use a proper grounding system to make sure that the working surface and the components are at the same electric potential.

ESD PRECAUTION

As recommended by the manufacturer, we highly encourage users to take the necessary precautions to avoid damaging their module.



ESP32 FIRMWARE

We have intentionally kept the ESP32 firmware as simple as possible - it only disciplines the TCXO oscillator and controls the OLED display. The SparkFun RTK Firmware or SparkFun RTK Everywhere firmware will not run on this product. The intention is that you can easily develop your own firmware for the SXT(-D) if the SparkFun firmware does not meet your needs.

The **/Firmware/Binaries** folder contains the firmware binaries.

You can update or reload the firmware using the SparkFun RTK Firmware Uploader.

You can of course modify the hardware too, should you want to. The design is completely open-source.

Hardware Overview

In this section, we walk you through the hardware design, interfaces, I/O connections, power options and more.

Power Options

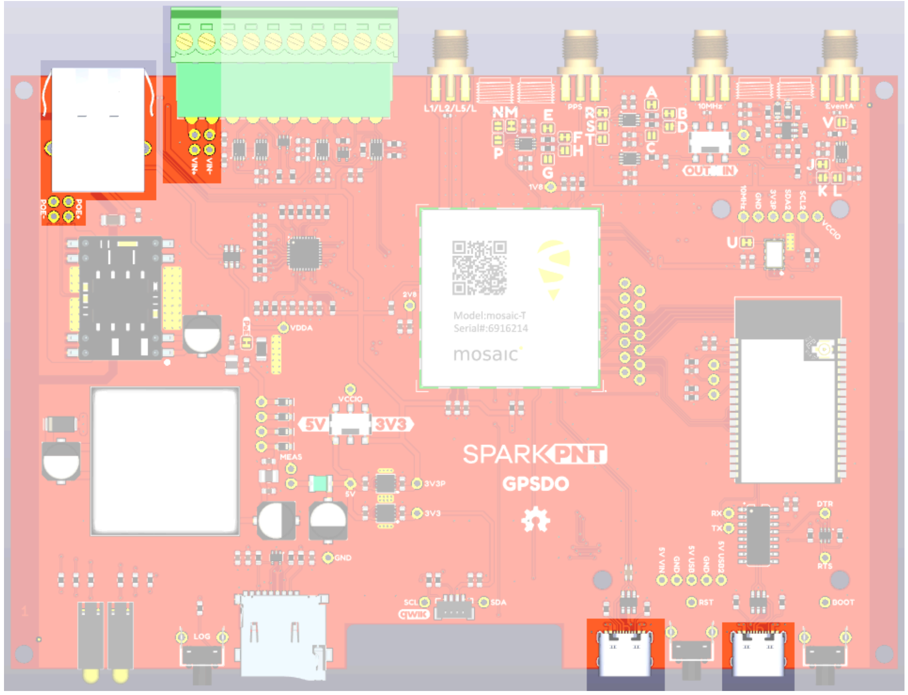
The mosaic-T and the ESP32 both required 3.3V power. To simplify the power circuitry, the four power sources are combined into a common 5V rail which then feeds individual 3.3V regulators for the mosaic-T and the ESP32.

The SXT (GNSSDO) can be powered individually or in combination, with any of the following:

- **`USB Ports`** - **5V**; delivered via the `MOSAIC CONFIG` and/or `ESP32 CONFIG` USB-C connectors.
- **`Power-over-Ethernet`** - **Range: 36 to 57V**; delivered via the `MOSAIC ETHERNET` RJ45 MagJack connector.
- **`External DC Power`** - **Range: 9 to 36V**; delivered via the `VIN+` and `VIN-` screw cage terminals.

PROTECTION COMPONENTS

Diodes are used to combine and protect the power sources from each other. Also, a 2A resettable fuse (green) provides additional protection.



Power connections on the SXT(-D) PCB.

Whereas, powering the SXT-D (GNSSDO+) takes a little more thought and care. The double-oven oscillator in the SXT-D draws more current than a standard USB (2.0) port can provide. You need to ensure that you use the provided 65W Power Delvery Wall Adapter if you are going to power the SXT-D from USB. Also, the SXT-D includes a Power Path priority circuit. From highest to lowest, the Power Path priorities are:

- **`Power-over-Ethernet`** - **Range: 36 to 57V**; delivered via the `MOSAIC ETHERNET` RJ45 MagJack connector.
- **`External DC Power`** - **Range: 9 to 36V**; delivered via the `VIN+` and `VIN-` screw cage terminals.
- **`CONFIG ESP32 USB Port`** - **5V from a Power Delivery Wall Adapter**; delivered via the `CONFIG ESP32` USB-C connector.
- **`CONFIG MOSAIC USB Port`** - **5V from a Power Delivery Wall Adapter**; delivered via the `CONFIG MOSAIC` USB-C connector.

> **NOTE**
>
> - If you have the **`CONFIG ESP32 USB Port`** connected to a standard USB (2.0) port, it will not be able to provide enough current. And due to the Power Path priority, it will not preferentially draw power from **`CONFIG MOSAIC USB Port`**. Your options are:

  - Provide power via `Power-over-Ethernet` or `External DC Power` since these have higher priority
  - Install a power switch in the `CONFIG ESP32 USB Port` cable to disconnect its power, forcing the SXT-D to draw Power Delivery power from the `CONFIG MOSAIC USB Port`.
- If you have `CONFIG MOSAIC USB Port` connected to a standard USB (2.0) port, ensure the Power Delivery Wall Adapter is connected to `CONFIG ESP32 USB Port`. The Power Path priority will ensure power is drawn from the `CONFIG ESP32 USB Port`.

INFO

Consult page 9 of the SXT-D schematic for more details

MEASURE CURRENT DRAW

If you want to measure the board's current draw, you can open the `MEAS` jumper and measure the current via a pair of breakout pads *(see the* ***Jumpers*** *section)*.

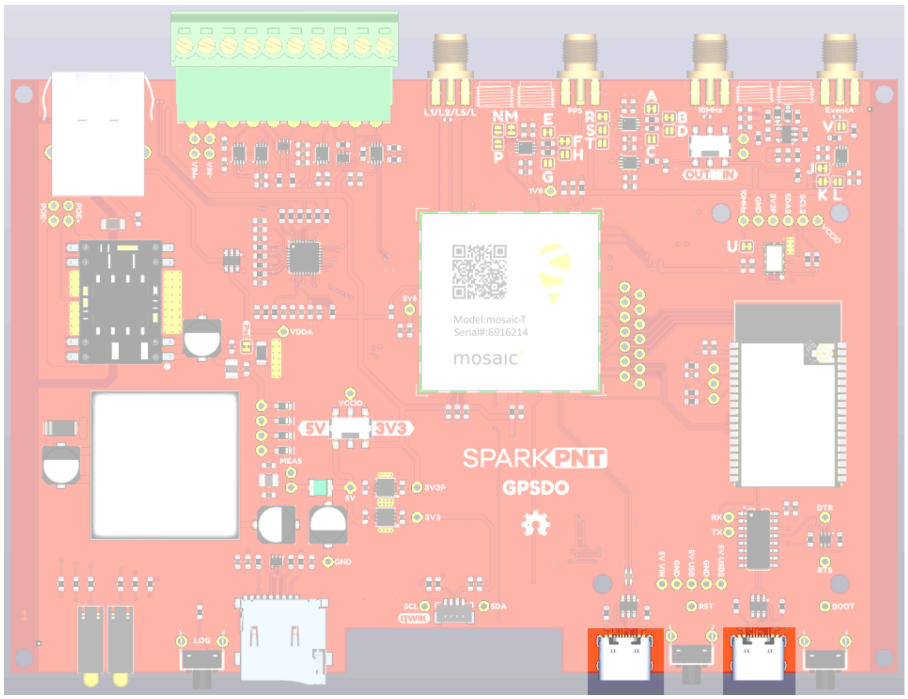
INFO

For more details, users can reference the SXT schematic, SXT-D schematic and the datasheets of the individual components on the board.

USB-C Connectors	**Power-over-Ethernet (PoE)**	**External DC Power (VIN)**

The mosaic-T and ESP32 both have USB-C connections. These USB ports can be used to power the SXT(-D) during the initial configuration when the mosaic-T or ESP32 are connected to a computer. For SXT-D, ensure you are using the provided Power Path Wall Adapter as described above.

USB-C connectors on the SXT(-D).

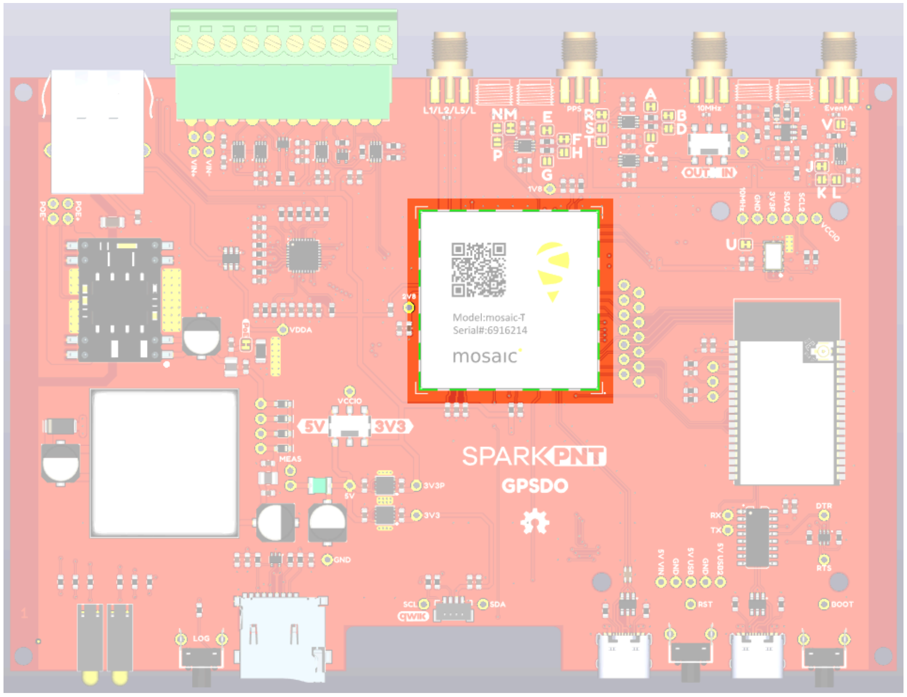


The USB-C device connections on the SXT(-D) PCB.

> **CH340 DRIVER**
>
> The CH340 allows the ESP32-WROVER to communicate with a computer/host device through the USB-C connection. This allows the ESP32 to show up as a device on the serial (or COM) port of the computer. Users will need to install the latest drivers for the computer to recognize the CH340 *(see* ***USB Driver*** *section)*.

mosaic-T

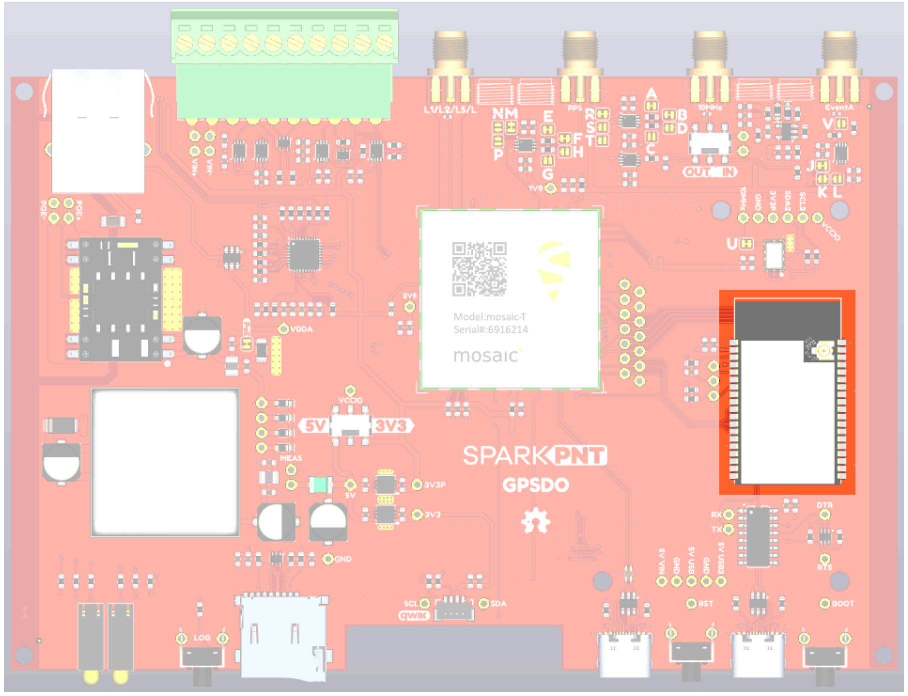
The heart of our product is of course the mosaic-T GNSS module from Septentrio. It is a *very* sophisticated chip with multiple interfaces: UARTS, USB and Ethernet. With the SXT, the `GPIO1` and `GPIO2` pins are available as 0.1" test points should you need access to them. For the SXT-D, `GPIO1` and `GPIO2` are available from the 10-way screw cage connector.



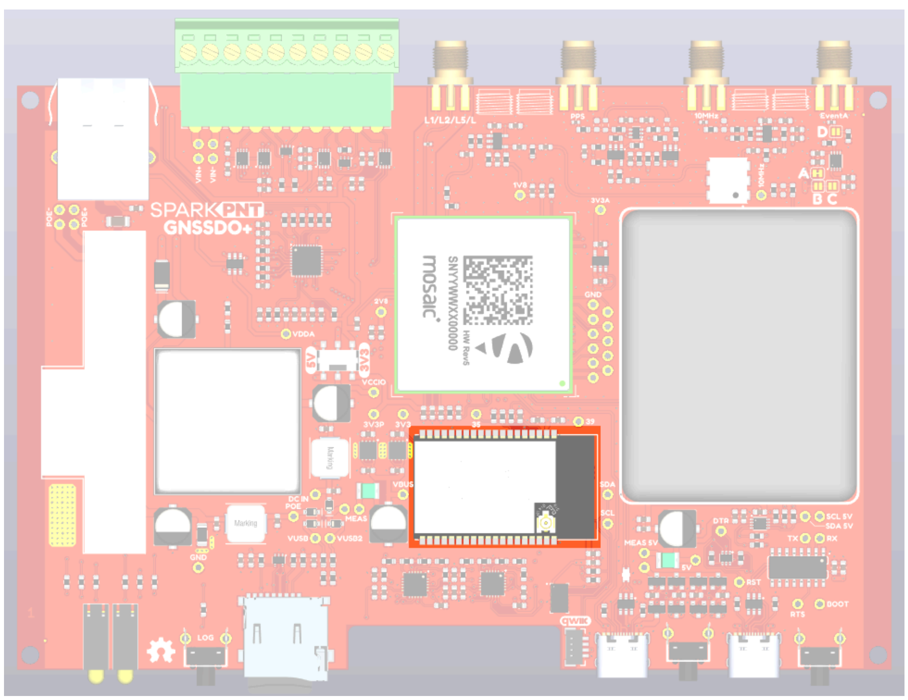
The Septentrio mosaic-T GNSS module.

ESP32-WROVER

The ESP32 processor is there to control (discipline) the 10 MHz TCXO oscillator and the OLED display. We have deliberately kept the ESP32 firmware as simple as possible. The intention is that you can write your own firmware using the Espressif IDF or Arduino IDE if you need to.



The Espressif ESP32-WROVER processor on the SXT.



The Espressif ESP32-WROVER processor on the SXT-D.

Think of the ESP32 as a co-processor, or riding shotgun... The mosaic-T `COM1`, `COM3` and `COM4` UARTs are linked to the ESP32, allowing the two to communicate directly without needing an Ethernet link. In our firmware, the PVTGeodetic and ReceiverTime messages are output on COM1. The ESP32 displays some of their content on the I^2C OLED display, and then uses the content to discipline the TCXO oscillator. See Oscillator for more details.

ESP32 FIRMWARE

We have intentionally kept the ESP32 firmware as simple as possible. The intention is that users can easily develop their, own firmware for the SXT(-D) using the Espressif ESP IDF or the Arduino IDE if the SparkFun firmware does not meet their needs.

The **/Firmware/Binaries** folder contains the firmware binaries.

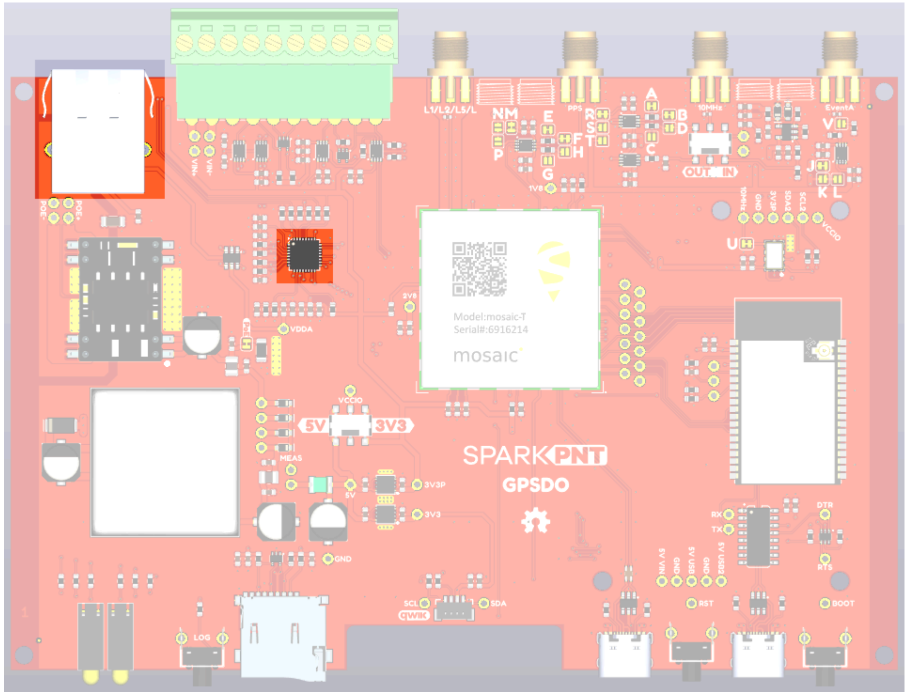
You can update or reload the firmware using the SparkFun RTK Firmware Uploader.

The ESP32-WROVER-IE antenna is not connected. If you write your own firmware and want to use BT/WiFi connectivity, you will need to attach your own antenna to the u.FL connector on the ESP32 module.

Ethernet PHY Interface

The mosaic-T has a KSZ8041NLI Ethernet PHY interface, connected using a Reduced Media-Independent Interface (RMII).

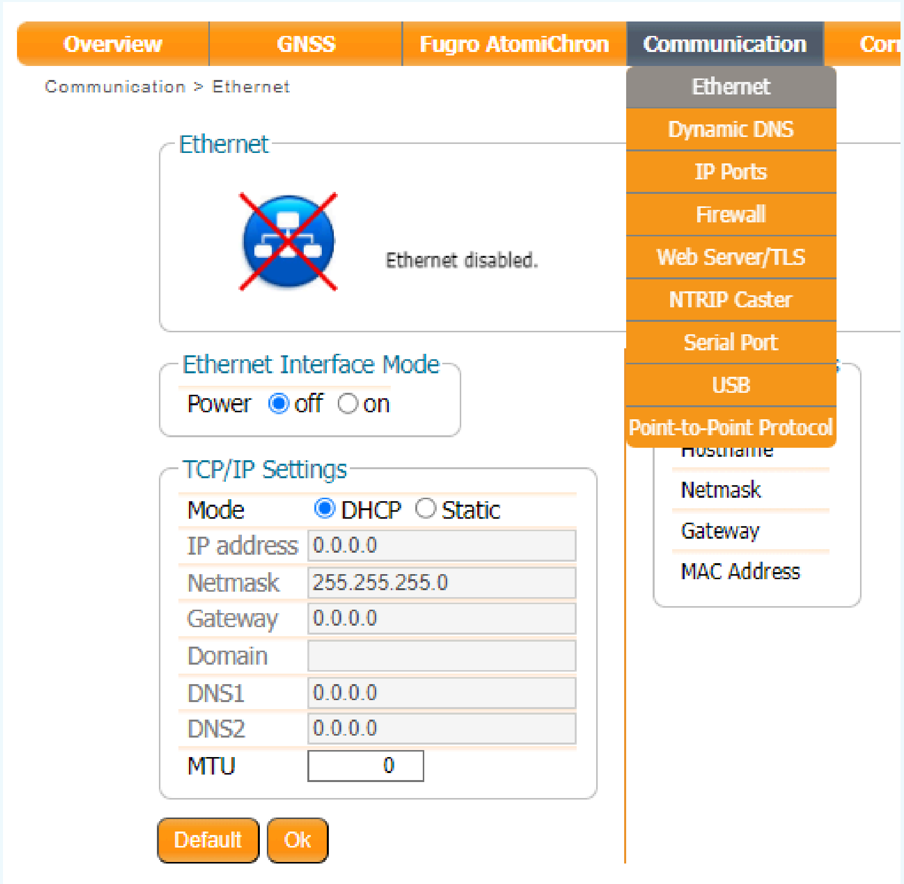
Ethernet (PoE) conection.



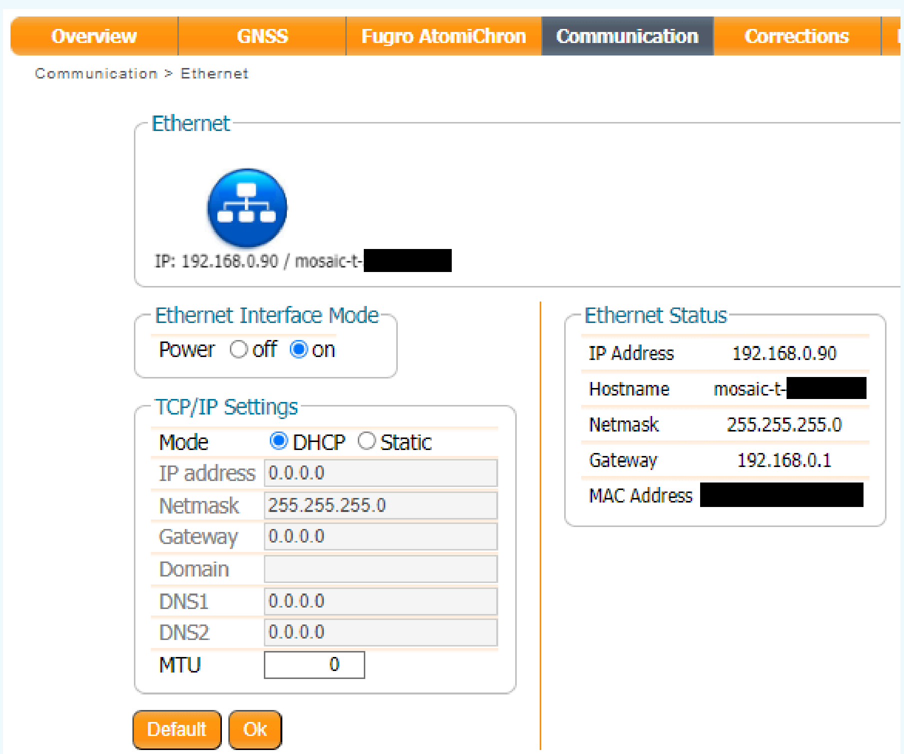
The Ethernet PHY on the SXT(-D) PCB.

NO IP ADDRESS?

Check the Ethernet interface is enabled. It may be disabled. Connect via the `CONFIG MOSAIC` USB-C port and open 192.168.3.1 on a web browser. Check the **Communication \ Ethernet** sub-page.



[mosaic-T web page (PNG)](#) with Ethernet disabled.



mosaic-T web page (PNG) with Ethernet enabled.

By default, the mosaic-T Ethernet port is configured for Dynamic Host Configuration Protocol (DHCP). It expects the router / Ethernet switch to provide it with an IP address. If the IP address is all zeros (0.0.0.0), check that your router has DHCP enabled. Most do.

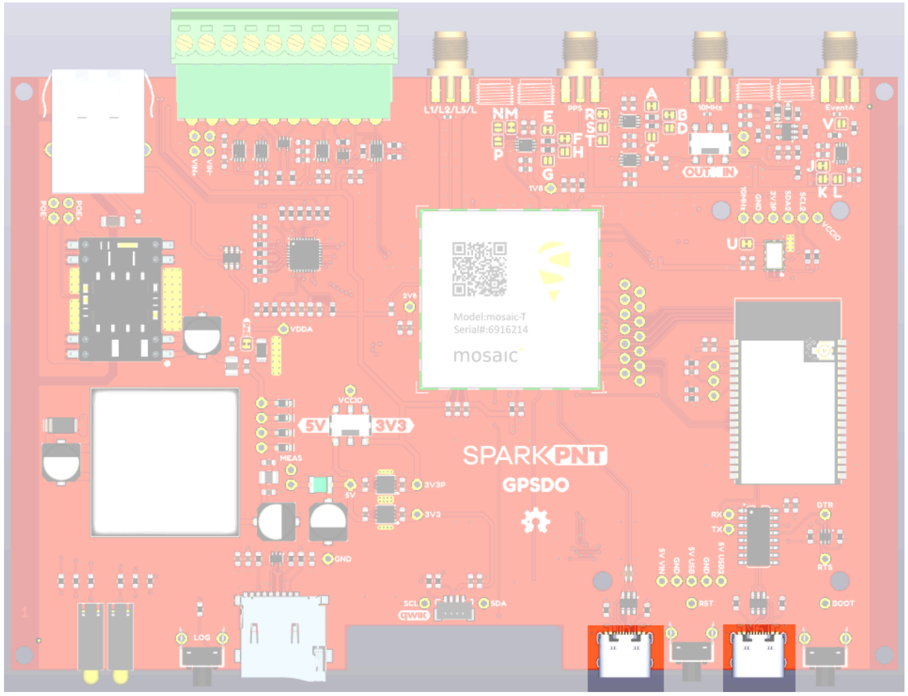
If you need a static IP address, you can configure this through the mosaic-T's **Communication \ Ethernet** sub-page.

Subnet 3 is reserved for the mosaic-T's USB-C connection (Ethernet-over-USB). If your router / switch is allocating addresses using subnet 3 (192.168.3.***), please change its settings so it uses a different subnet.

USB-C Connectors

The mosaic-T and ESP32 both have USB-C connections. The MOSAIC USB port is high-speed and connected to the T through a balancing transformer. The ESP32 USB port is connected through a CH340 USB-UART IC.

USB-C connections: mosaic-T (left) and ESP32 (right).



The USB-C data connections on the SXT(-D) PCB.

INFO

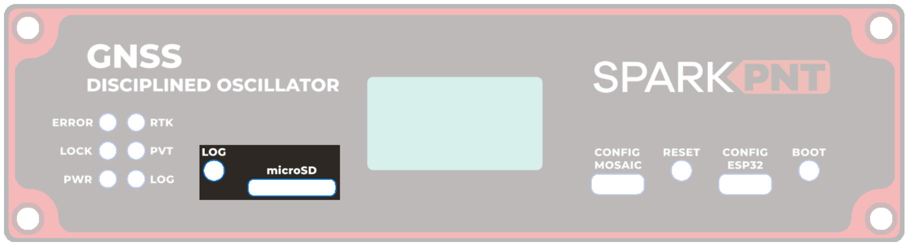
The SXT can draw power from either or both USB ports, in addition to Power-over-Ethernet and the DC-DC external input described above. Powering the SXT-D via USB takes a little more though and care. Please see the notes above.

CH340 DRIVER

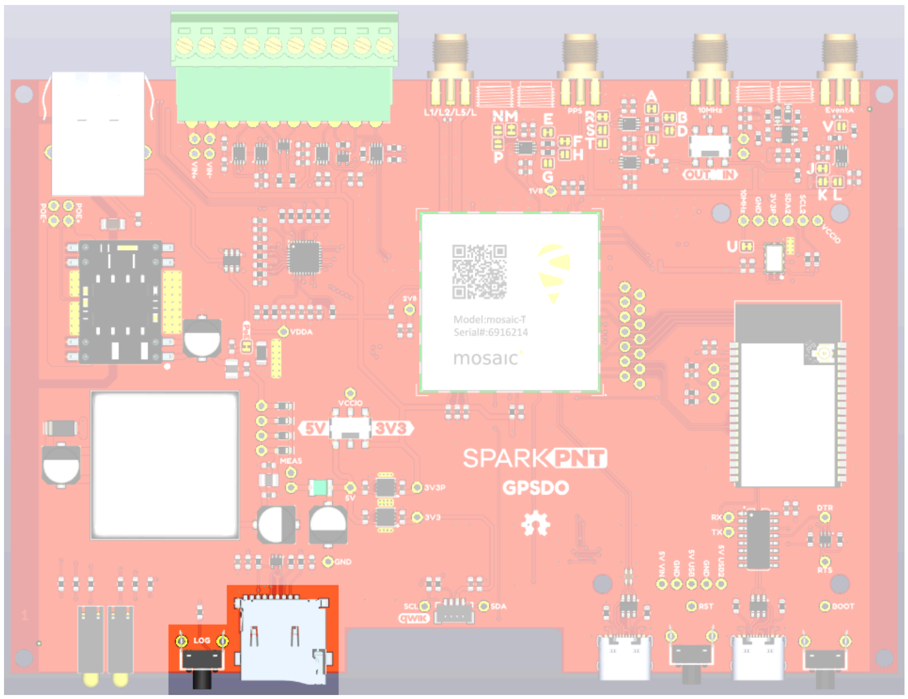
The CH340 allows the ESP32-WROVER to communicate with a computer/host device through the USB-C connection. This allows the ESP32 to show up as a device on the serial (or COM) port of the computer. Users will need to install the latest drivers for the computer to recognize the CH340 *(see **USB Driver** section)*.

µSD Socket

The µSD socket is connected directly to the mosaic-T via a one-bit SDIO interface for fast data logging. The mosaic-T supports µSD cards with a **FAT32** file system *(i.e. only cards **up to 32GB** in size)*.

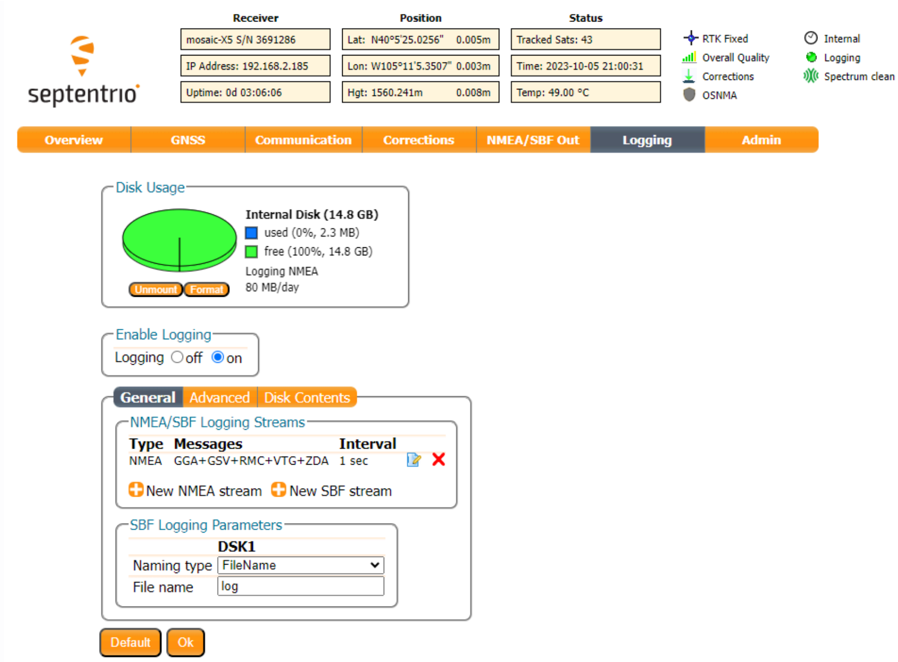


µSD slot and `LOG` button.



µSD socket and `Log` button on the SXT(-D) PCB.

> **OPERATION INSTRUCTIONS**
>
> > **INITIAL CONFIGURATION**
> >
> > Before logging can take place, it is necessary to define a "logging stream" using the **Logging** page or **RxTools**. Streams can contain NMEA or SBF (Septentrio Binary Format) data; SBF can contain RTCM and/or RINEX.



µSD logging stream configuration.

INSTRUCTIONAL VIDEO

[How to log data to the SD card of the Septentrio mosaic receiver module]

Once the stream is defined, users can control the data logging operation through the `LOG` button.

- A short press of the `LOG` button *(<5s)* toggles data logging to the SD card on and off.
  - The red `LOG` LED will flash while logging is taking place.
- A long press, holding the `LOG` button for more than 5 seconds *(>5s)* and then releasing it, will force the board to:
  - Unmount the SD card if it was mounted
  - Mount the SD card if it was unmounted

SMA Connectors

The SXT(-D) has robust SMA connectors for the mosaic-T GNSS antenna, Pulse-Per-Second output, 10 MHz clock signal and the Event A input.

The SMA connections on the SparkPNT SXT(-D).

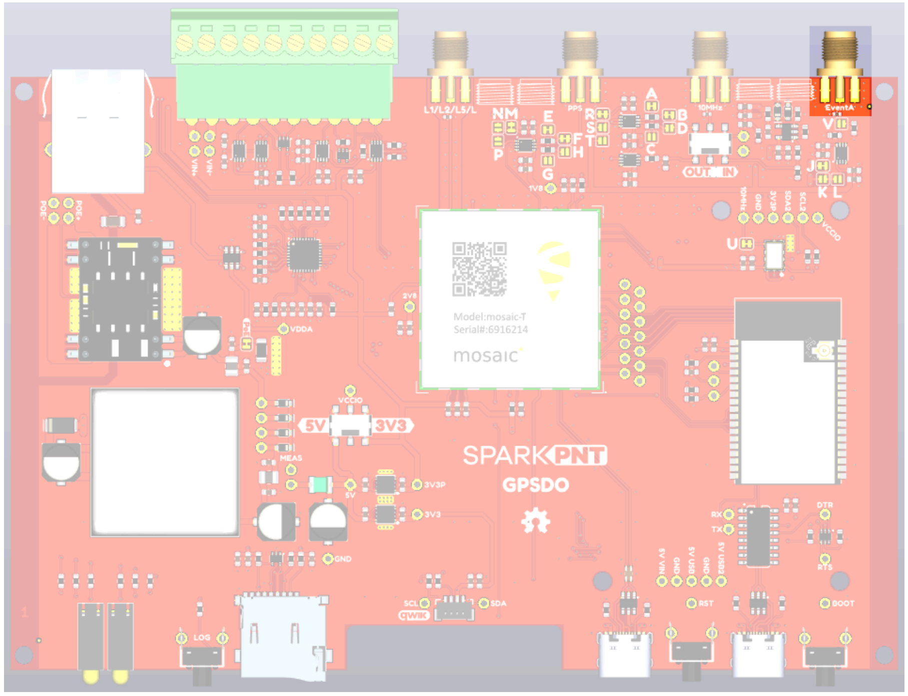
CONNECTOR POLARITY

All these SMA connectors have a standard polarity. When selecting antennas and/or cables for the SXT(-D), double-check the polarity for the connections and cables.

EVENTA **10MHz** **PPS** **GNSS**

The Event A SMA connector is standard polarity. The voltage is adjustable via the VCCIO switch: 3.3V or 5V. 2.8V and 1.8V are also available via the jumper links *(see the **Jumpers** section)*. The output can also be configured for 50 Ohm via the jumper links *(see the **Jumpers** section)*.

The SMA connector for the EventA input.

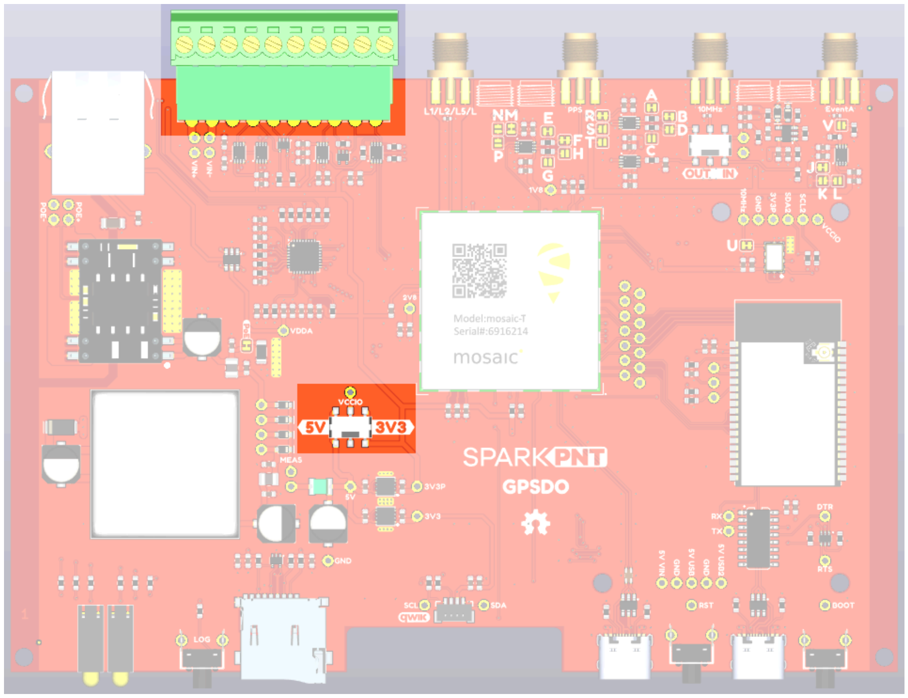


The connection for the EventA input.

Input/Output Terminals

The SXT(-D) is equipped with a 10-way 3.5mm screw cage terminal connector.

I/O Screw Terminal Connections.



I/O Screw Terminal Connections on the SXT PCB.

These terminals are described in the tabs below. For more information on the I/O terminals, you can refer to the SXT schematic and the SXT-D schematic.

External Power | **GND** | **mosaic-T COM2** | **CTS** | **EVENT B** | **SCL2 & SDA2** | **GPIO1 & GPIO2**

The `VIN+` and `VIN-` terminals allow the SXT(-D) to be powered by an external DC power source - typically a 12V / 24V vehicle battery.

Terminal	Function
VIN+	External voltage: **Min: 9V**; **Max: 36V**
VIN-	Ground / Chassis / 0V

INFO

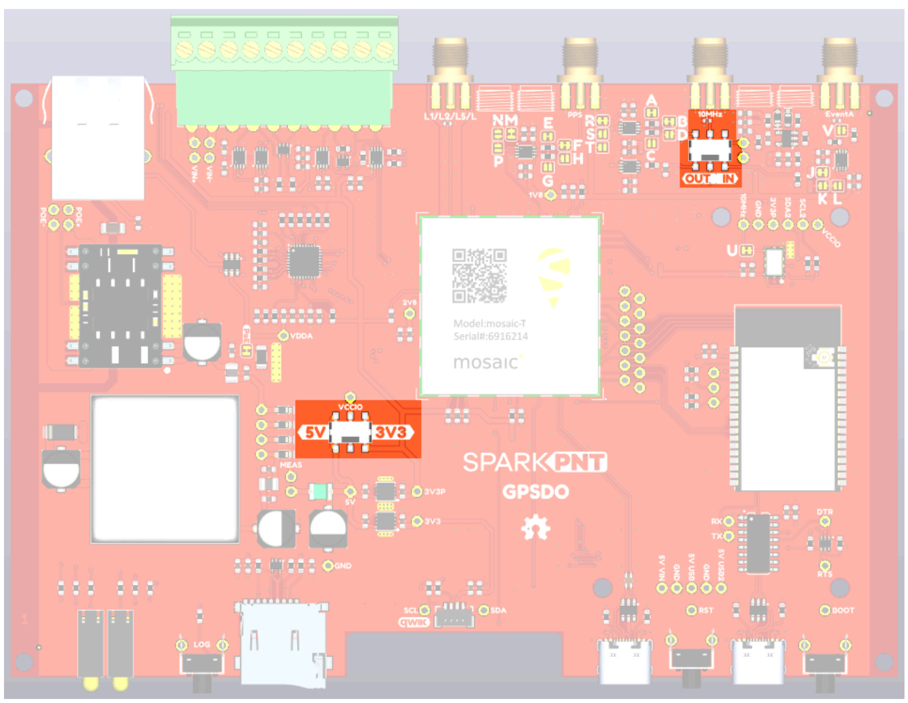
The DC-DC converter in the SXT(-D) provides 1.5kV isolation between `VIN+`/`VIN-` and **5V/GND**. There is no direct electrical connection between `VIN-` and `GND`.

GROUND LOOP

If desired, users can link `VIN-` to the adjacent `GND` screw cage terminal. However, this will bypass the voltage isolation and could introduce an unwanted ground loop, particularly if the GNSS antenna ground (shield, 0V) is also connected to the chassis.

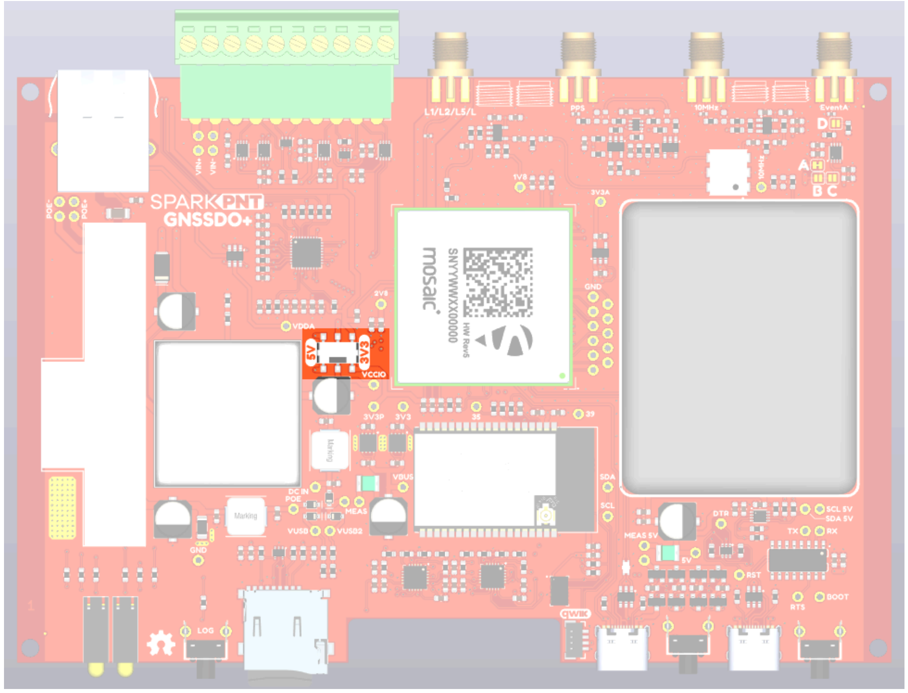
Switches

There are two miniature slide switches on the SXT PCB:



Switches inside the SXT

But only one on the SXT-D PCB:



Switch inside the SXT-D

> ⚠ **WARNING**
>
> The mosaic-T module can either use its internal TCXO or accept an external signal as a frequency reference. However, the module will constantly reboot if the `REF_I` pin isn't provided a 10MHz sinusoidal signal and left floating. Therefore, users should only set the SXT `10MHz` switch to the `IN` position, only if a 10MHz clock signal can be provided; otherwise, the switch should remain in the `OUT` position.
>
> Additionally, switching between an external and internal frequency reference must occur when the mosaic-T is powered off, or the module must be reset after switching states.

- `VCCIO`: This switch sets the voltage of the Input Output Terminals (`COM2` UART, `Event B`, `SCL2` & `SDA2` (SXT), `GPIO1` & `GPIO2` (SXT-D))
  - The I/O voltage can be set to 3.3V (default) or 5V.
- `10MHz`: This switch changes the function of the 10MHz SMA connector

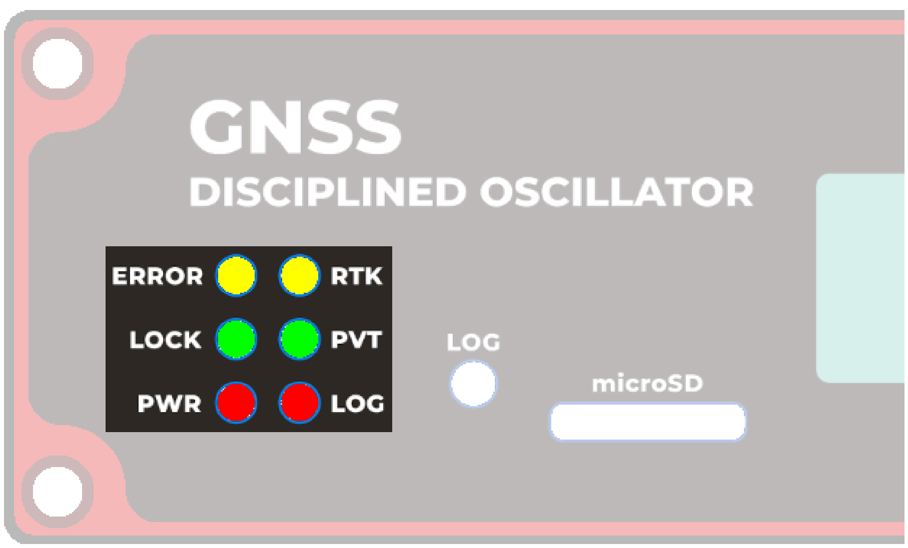
  > 💡 **TIP**

> This switch is only available on SXT. For the SXT-D 10MHz SMA is output-only.

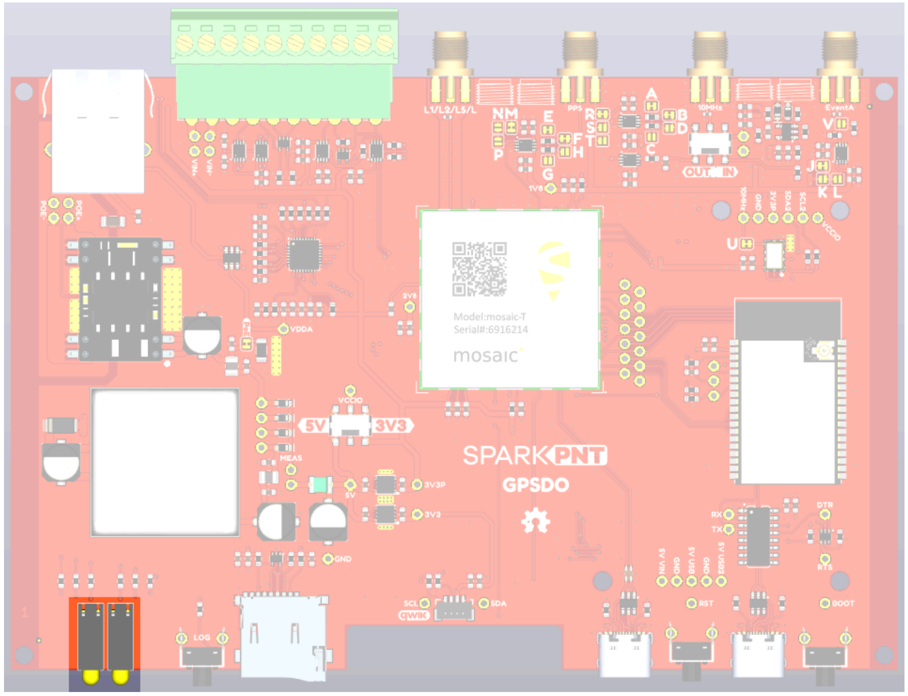
- When set to `OUT` (default):
  - The SMA connector will output a 10MHz "CMOS" disciplined clock signal
  - The signal voltage is set by the VCCIO voltage selection switch
- When set to `IN`:
  - The user **must apply a clock signal from an external 10MHz oscillator**; otherwise, the mosaic-T module will constantly reset
  - The input impedance is 50Ω
  - The detection level is -14dBm
  - The max supported input level is +12dBm

Status LEDs

There are six status LEDs on the SXT(-D):



The status indicator LEDs on the SXT(-D).



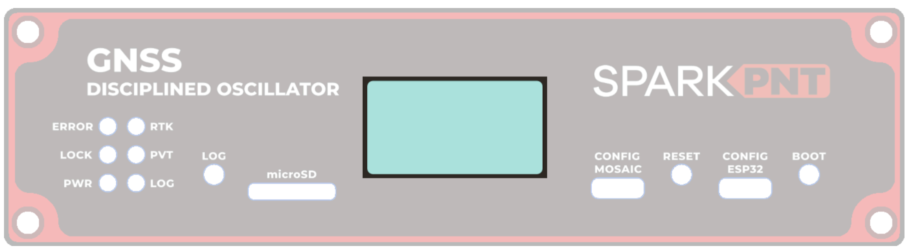
The status indicator LEDs on the SXT(-D) PCB.

- `PWR` - Power *(Red)*
    - Illuminates when power is applied
- `LOG` - µSD Logging *(Red)*
    - Solid Red - µSD card is mounted
    - Blinking Red - Data is being logged
    - Off - µSD is dismounted or not present
- `LOCK` - Oscillator Lock *(Green)*
    - The TCXO is locked to the correct frequency - as reported by **PVTGeodetic RxClkBias**
    - Connected to ESP32 GPIO pin 33
- `PVT` - Position Velocity Time *(Green)*
    - Solid Green - The mosaic-T has valid Position, Velocity and Time
    - Off - Satellite signal not present or acquired
- `ERROR` - GNSS Error *(Yellow)*
    - The GNSS Error status - as reported by **PVTGeodetic Error**
    - Connected to ESP32 GPIO pin 32
- `RTK` - Real-Time Kinematic *(Yellow)*

- This LED has little or no functionality on the SXT(-D) as the mosaic-T uses PPP, not RTK
- Internally, it is connected to the mosaic-T GPLED2 pin
- It can be configured for (e.g.) TRACKLED if desired

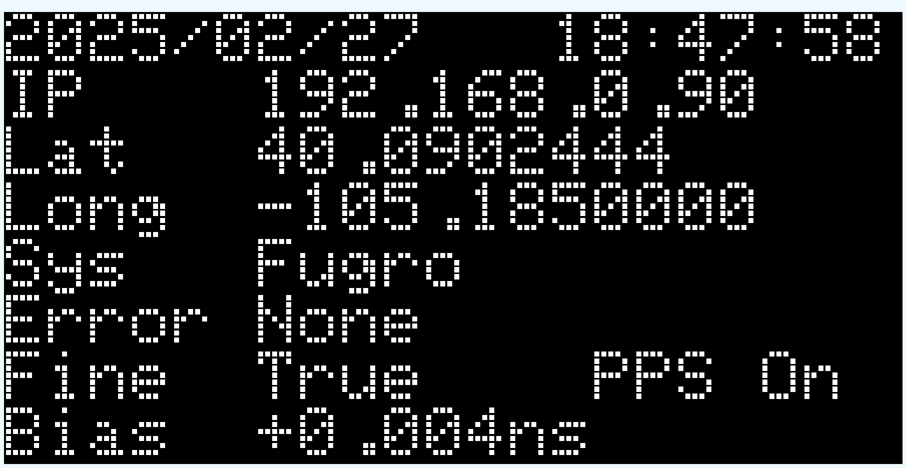
OLED Display

The SXT(-D) has a 128x64 pixel OLED display, controlled by the ESP32 via I^{2}C. After some initial diagnostic messages, the display will show position, time and other data from the mosaic-T **PVTGeodetic**, **ReceiverTime** and **IPStatus** SBF blocks.



The OLED display on the SXT(-D).

ⓘ **INFO**



Information on the OLED display.

- Date & Time : YYYY/MM/DD HH:MM:SS from ReceiverTime
- **IP** : nnn.nnn.nnn.nnn from IPStatus IPAddress
  - When TCP console access is enabled, the TCP port number is also displayed.

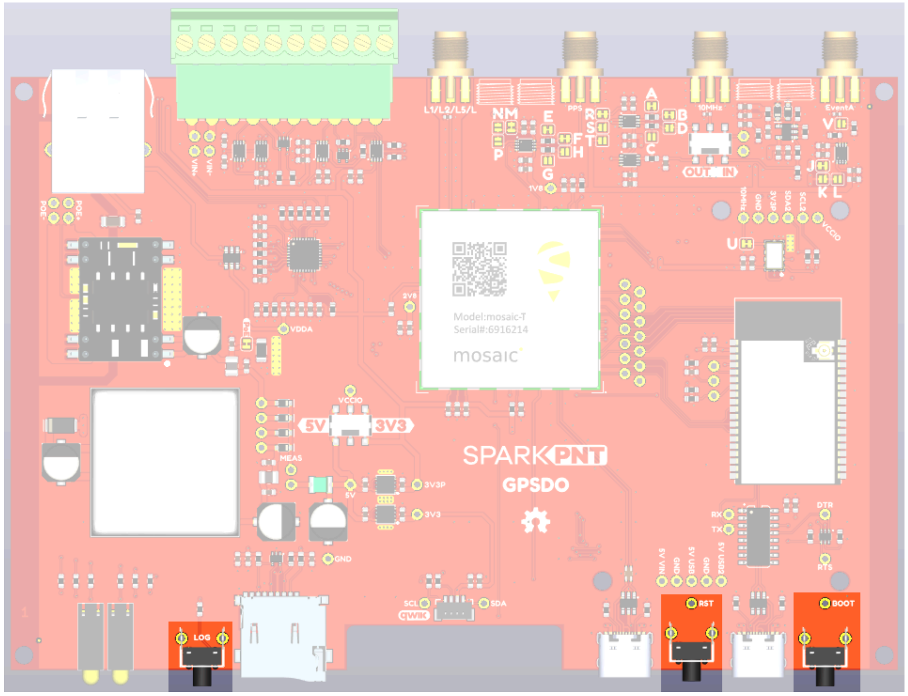
- **Lat** : Latitude from PVTGeodetic (Degrees)
- **Long** : Longitude from PVTGeodetic (Degrees)
- **Sys** : TimeSystem from PVTGeodetic
    - GPS, Galileo, GLONASS, BeiDou, QZSS, Fugro
- **Error** : Error from PVTGeodetic
    - None, Measurements, Ephemerides, DOP, Residuals, Convergence, Outliers, Export, Differential, Base, Ambiguities
- **Fine** : FINETIME from ReceiverTime
    - False, True
- **PPS** : Indicates if the Pulse-Per-Second signal is being generated
    - Off, On
    - PPS is only generated once the RxClkBias has achieved the required accuracy
- **Bias** : RxClkBias from PVTGeodetic (ms/us/ns)

The display is updated on arrival of the **ReceiverTime** message. You may see a small lag between the display and the actual time system time, but it will be minimal.

Buttons

There are three buttons on the SXT(-D): `RESET`, `BOOT`, and `LOG`.

Buttons on the SXT(-D).



Buttons on the SXT(-D) PCB.

- mosaic-T: Data Logging
    - Once a logging stream is defined, users can control the data logging operation through the `LOG` button.
        - A short press of the `LOG` button *(<5s)* toggles data logging to the SD card on and off.
            - The red `LOG` LED will flash while logging is taking place.
        - Holding the `LOG` button for more than 5 seconds *(>5s)* and then releasing it, will force the board to:
            - Unmount the SD card if it was mounted
            - Mount the SD card if it was unmounted
        - The SD card must be mounted to allow data logging by the mosaic-T
        - When the SD card is unmounted, it is accessible as a mass storage device via the `CONFIG MOSAIC` USB-C interface
            - Files can be read and written over USB while the SD card is unmounted
        - You can enter commands using the `Admin \ Expert Console` to:
            - Change what happens when the disk is full
            - Change how the files are named

        - Change the mounting / unmounting of the disk
    - Consult section 3.2.20 of the Firmware Reference Guide for more details

> **INSTRUCTIONAL VIDEO**
>
> How to log data to the SD card of the Septentrio mosaic receiver module

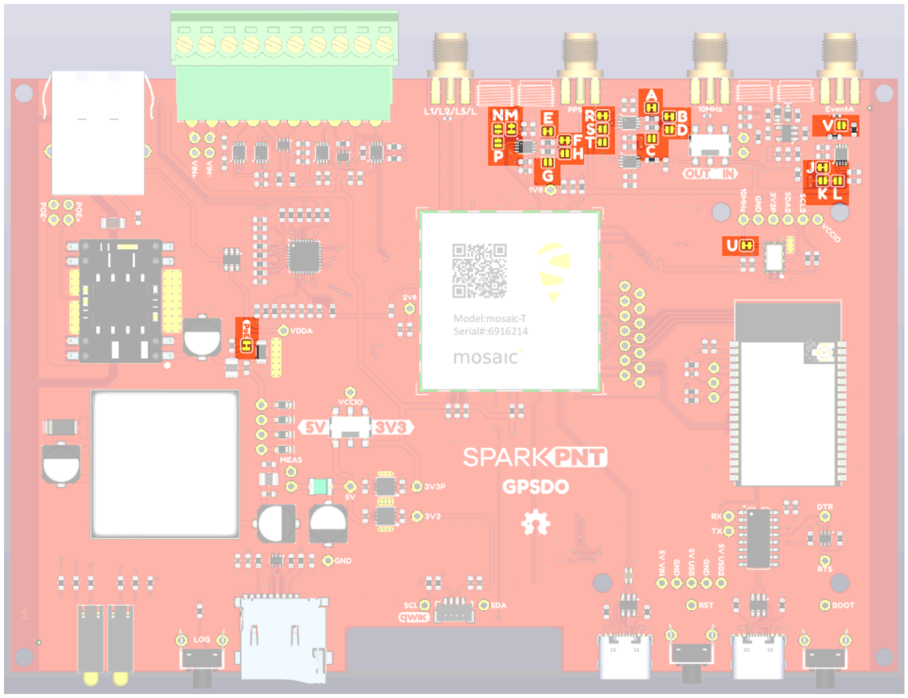
- ESP32: Reset
    - The `RESET` button allows users to reset the firmware running on the ESP32-WROVER module without disconnecting the power.
- ESP32: Boot Control
    - The `BOOT` button can be used to force the ESP32 into the serial bootloader. Holding down the `BOOT` button, while connecting the SXT(-D) to a computer through its USB-C connector or resetting the board will cause it to enter the Firmware Download mode. The ESP32 will remain in this mode until it power cycles (happens automatically after uploading new firmware) or the `RESET` button is pressed.
        a. Hold the `BOOT` button down.
        b. Reset the MCU.
            - While unpowered, connect the board to a computer through the USB-C connection.
            - While powered, press the `RESET` button.
        c. Release the `BOOT` button.
        d. After programming is completed, reboot the MCU.
            - Press the `RESET` button.
            - Power cycle the board.

Jumpers (SXT)

> **NEVER MODIFIED A JUMPER BEFORE?**
>
> Check out our Jumper Pads and PCB Traces tutorial for a quick introduction!

There are several jumpers on the SXT PCB which can be used to (e.g.) disable the LEDs or allow measurement of the board's current draw.

Top Bottom



The jumpers on the top of the SXT PCB.

- **POE** - This jumper can be used to disconnect the Power-over-Ethernet (PoE) module 50Ω load.
    - The PoE module has a minimum load of 200mA. We included the 50Ω load to ensure this is met. If you can ensure this by other means, open this jumper to disconnect the load.
- **Voltage Configuration: A-V**
    - The jumper links A-V can be used to configure the voltage levels and impedance of the SMA connections. Please refer to the schematic for additional information.
    - To configure the 10MHz output for 50 Ohms: open jumper **A** and close jumper **D**.
        - Jumper **A** is closed by default. Open it to select 50 Ohms for the 10MHz output.
        - Jumper **B** is closed by default. It could be used to isolate the gate driving the 10MHz output. Advanced use only.
        - Jumper **C** is open by default. It could be used to select the alternate gate for the 10MHz CMOS output. Advanced use only.

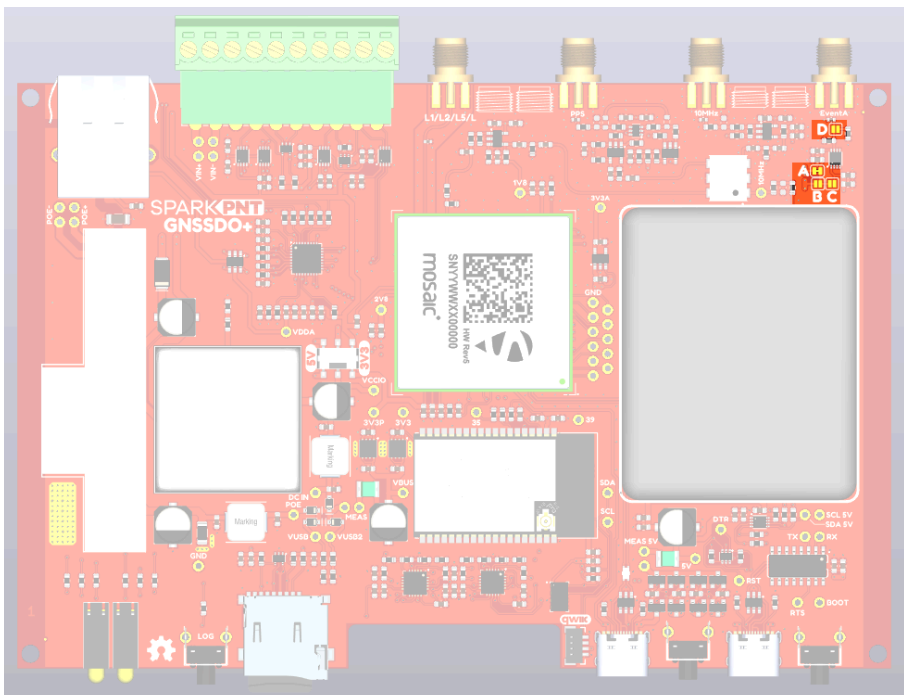
    - Jumper **D** is open by default. Close it to select 50 Ohms for the 10MHz output.
- To configure the PPS output for 50 Ohms: open jumper **E** and close jumper **H**.
    - Jumper **E** is closed by default. Open it to select 50 Ohms for the PPS output.
    - Jumper **F** is closed by default. It could be used to isolate the gate driving the PPS output. Advanced use only.
    - Jumper **G** is open by default. It could be used to select the alternate gate for the PPS CMOS output. Advanced use only.
    - Jumper **H** is open by default. Close it to select 50 Ohms for the PPS output.
- Jumpers J,K,L configure the voltage of the Event A input.
    - Jumper **J** is closed by default. It selects VCCIO as the Event A input voltage.
    - Jumper **K** is open by default. Open jumper J and close jumper K to configure Event A for 2.8V.
    - Jumper **L** is open by default. Open jumper J and close jumper L to configure Event A for 1.8V.
- Jumpers M,N,P configure the voltage of the PPS output.
    - Jumper **M** is closed by default. It selects VCCIO as the PPS output voltage.
    - Jumper **N** is open by default. Open jumper M and close jumper N to configure PPS for 2.8V.
    - Jumper **P** is open by default. Open jumper M and close jumper P to configure PPS for 1.8V.
- Jumpers R,S,T configure the voltage of the 10MHz output.
    - Jumper **R** is closed by default. It selects VCCIO as the 10MHz output voltage.
    - Jumper **S** is open by default. Open jumper R and close jumper S to configure 10MHz for 2.8V output.
    - Jumper **T** is open by default. Open jumper R and close jumper T to configure 10MHz for 1.8V output.
- Jumper **U** can be used to isolate the on-board 10MHz TCXO.
    - Open jumper U when connecting an alternate TCXO via the breakout pads on the PCB. Advanced use only.
- To configure the Event A input for 50 Ohms: close jumper **V**.
    - Jumper **V** is open by default. Close it to select 50 Ohms for the Event A input.

Jumpers (SXT-D)

> **NEVER MODIFIED A JUMPER BEFORE?**
>
> Check out our Jumper Pads and PCB Traces tutorial for a quick introduction!

There are several jumpers on the SXT-D PCB which can be used to (e.g.) disable the LEDs or allow measurement of the board's current draw.

Top Bottom



The jumpers on the top of the SXT-D PCB.

- **Voltage Configuration: A-D**
  - The jumper links can be used to configure the voltage levels and impedance of the SMA connections. Please refer to the schematic for additional information.
  - Jumpers A,B,C configure the voltage of the Event A input.
    - Jumper **A** is closed by default. It selects VCCIO as the Event A input voltage.
    - Jumper **B** is open by default. Open jumper A and close jumper B to configure Event A for 2.8V.
    - Jumper **C** is open by default. Open jumper A and close jumper C to configure Event A for 1.8V.

- To configure the Event A input for 50 Ohms: close jumper **D**.
  - Jumper **D** is open by default. Close it to select 50 Ohms for the Event A input.

Hardware Assembly

WARNING

When assembling the SXT(-D), users should attach any power connections last. While there shouldn't be any issues with hot-swapping peripherals, it is common practice to power electronics as the last step of the assembly process *(and the power should be disconnected before removing components)*.

WHAT IS IN THE BOX?

The SparkPNT SXT(-D) comes packaged as a complete kit, with all the accessories you'd need to set it up.

Everything that is included in the SparkPNT SXT(-D) kit.

Inside the kit, users will find the SparkPNT SXT(-D), GNSS antenna, USB-C cable, CAT-6 Ethernet cable, and another box containing additional accessories. Inside the accessory box, users will find the SMA to TNC cable, SMA to BNC cable, USB PD wall adapter, a pack of silicone bumpers, and a 32GB SD card.

USB-C Ports

The USB ports are utilized to configure the mosaic-T module and ESP32 firmware settings. Additionally, the USB ports can also be used as a power source for the SXT(-D).

The SXT(-D) with USB-C cable being attached.

The USB port to the mosaic-T can be used to configure the module through an IP port, for serial communication to stream the GNSS data, and access the SD card as a mass storage device. To connect to the mosaic-T, users only need to plug a USB-C cable into the `CONFIG MOSAIC` USB port and their computer.

The SXT(-D) with USB-C cable being attached.

With the default firmware, the USB port for the ESP32 is used for serial communication to tune the parameters for the TCXO oscillator and the Pulse-Per-Second output. To configure the settings, users only need to plug a USB-C cable into the `CONFIG ESP32` USB port and their computer, and then open a terminal emulator at 115200 baud.

> **SOFTWARE REQUIREMENTS**
>
> Depending on their computer's operating system, users may need to install USB drivers to interface with the mosaic-T and/or the ESP32. Users may also need to install a terminal emulator for serial communication with the mosaic-T and the ESP32.

GNSS Antenna

In order to receive GNSS signals, users will need a compatible antenna. With the parts included in this kit, connect the L1/L2/L5 (tri-band) GNSS antenna to the SXT(-D) using the TNC-to-SMA cable.

Attaching a TNC-SMA cable to the `GNSS` SMA connector on the SXT(-D).

Attaching a tri-band GNSS antenna to the TNC-SMA cable.

MOUNTING LOCATION

Users should mount their GNSS antenna outside, where it will have a clear, unobstructed view of the sky. Avoid areas with nearby buildings, EMF structures (i.e. radio towers or power lines), and vegetation (i.e. trees). These objects can increase errors due to signal muti-path, interference, and elevated noise plane.

The tri-band GPS antenna, mounted outside with an unobstructed view of the sky.

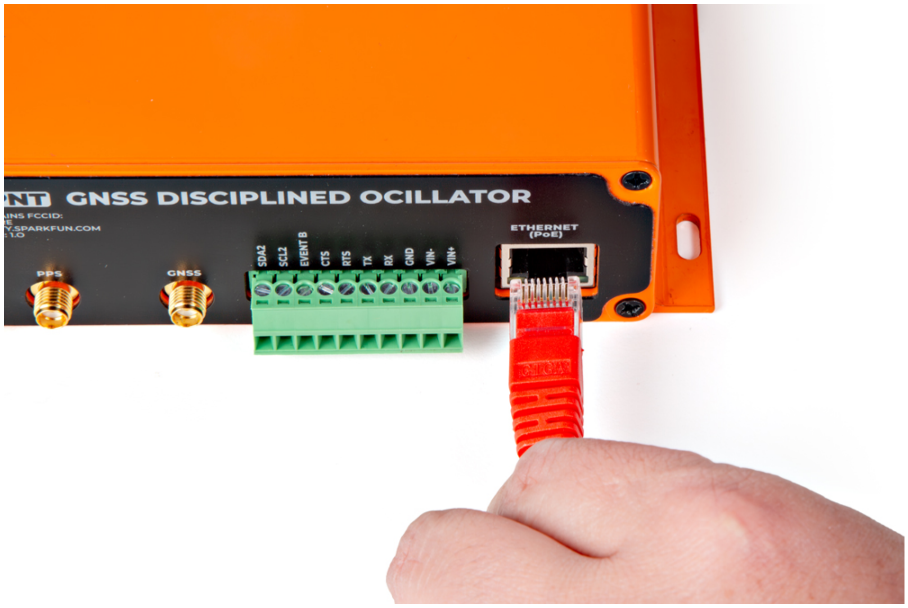
CONNECTOR POLARITY

When selecting antennas and/or cables for the SXT(-D), double-check the polarity of the connection.

Ethernet Jack

There is a single ethernet jack on the SXT(-D), which can be used to provide network access to the mosaic-T module. It supports power over ethernet (PoE) to power the device. To provide network access, users should connect the SXT(-D) from the `ETHERNET (PoE)` jack to their local network with the (CAT-6) ethernet cable provided in the kit.

- To power the device, a PoE network switch or PoE injector should be installed in between the network connection to the SXT(-D).

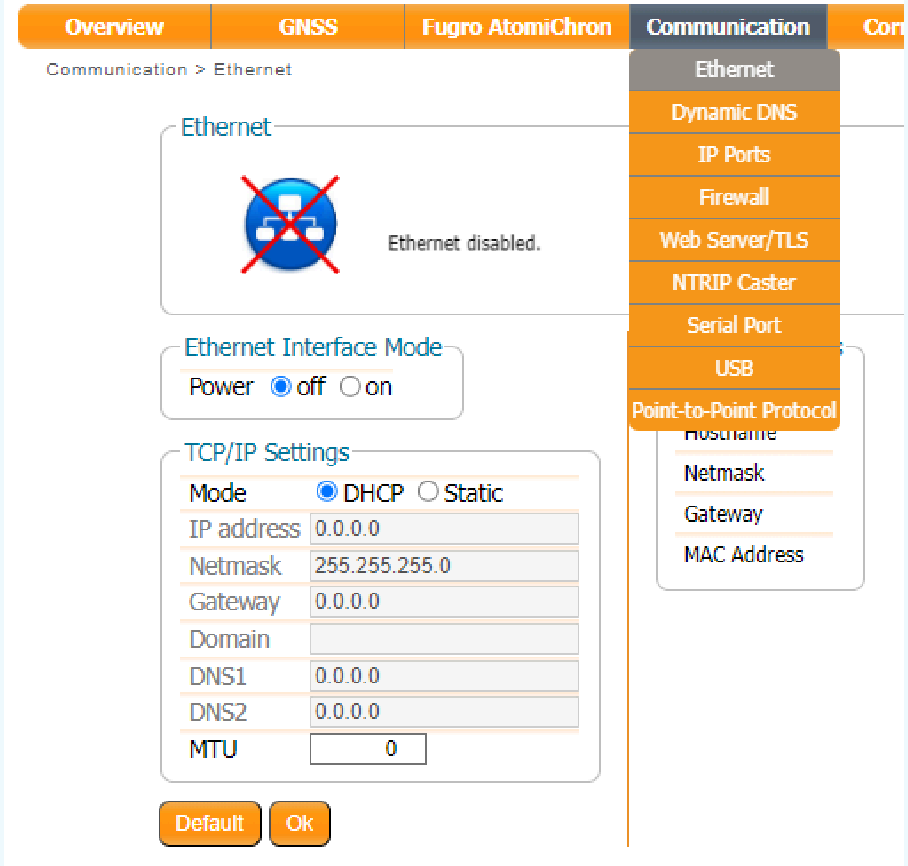


The SXT(-D) with ethernet cable being attached to the `ETHERNET (PoE)` jack.

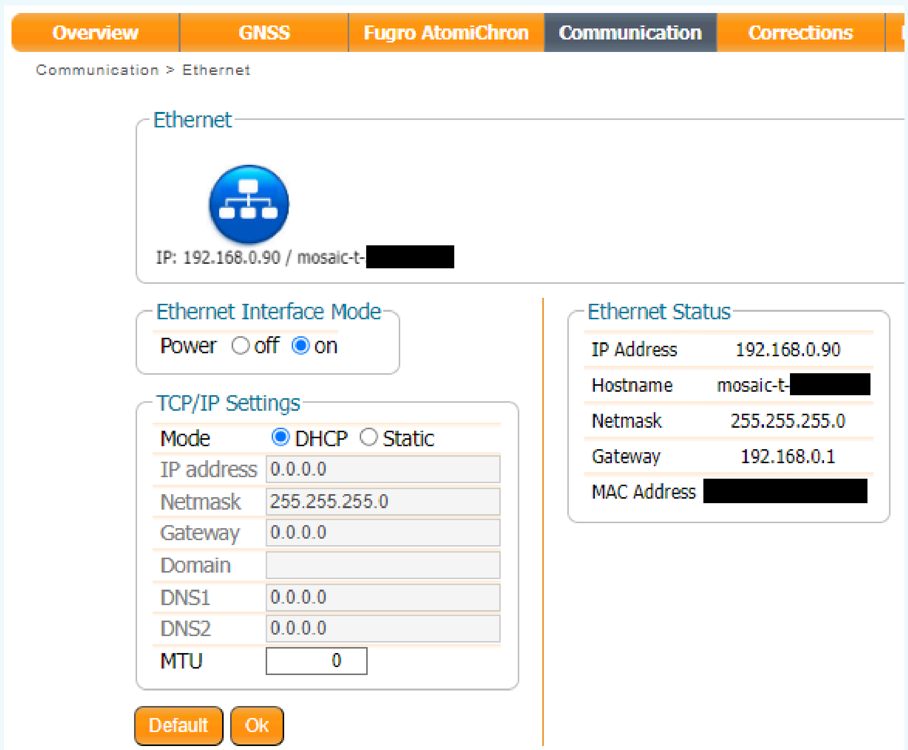
CONFIGURATION: MOSAIC-T SETTINGS

Users can configure the mosaic-T module through the network connection.

> **NO IP ADDRESS?**
>
> Check the Ethernet interface is enabled. It may be disabled. Connect via the `CONFIG MOSAIC` USB-C port and open 192.168.3.1 on a web browser. Check the **Communication \ Ethernet** sub-page.



The mosaic-T web page showint the Ethernet disabled.



The mosaic-T web page showint the Ethernet enabled.

By default, the mosaic-T Ethernet port is configured for Dynamic Host Configuration Protocol (DHCP). It expects the router / Ethernet switch to provide it with an IP address. If the IP address is all zeros (0.0.0.0), check that your router has DHCP enabled. Most do.

If you need a static IP address, you can configure this through the mosaic-T's **Communication \ Ethernet** sub-page.

Subnet 3 is reserved for the mosaic-T's USB-C connection (Ethernet-over-USB). If your router / switch is allocating addresses using subnet 3 (192.168.3.***), please change its settings so it uses a different subnet.

10MHz Signal

For timing applications, we have broken out the TCXO 10MHz signal to an SMA connector. In our kit, users will find an SMA to BNC adapter cable that they can use to hookup this signal to their equipment. If necessary, users can add an SMA extension cable to reach their equipment.

Attaching a cable to the `10MHz` SMA connector on the SXT(-D).

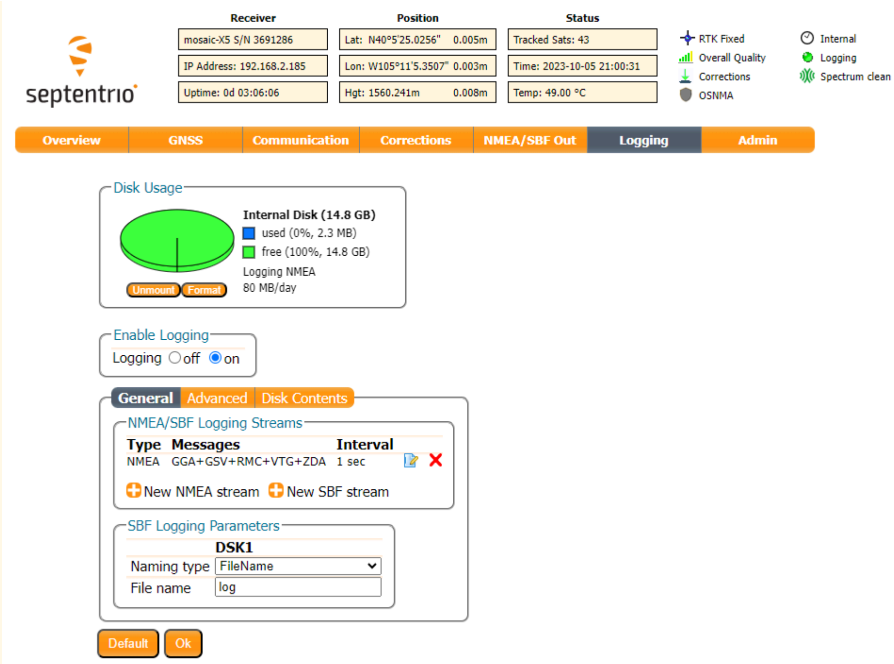
SD Card Slot

A µSD card slot is available for users to log and store data, locally on the board. Users will need to insert a compatible SD card and configure the mosaic-T module for data logging.

Inserting an SD card into the SXT(-D).

> **SD CARD COMPATIBILITY**
>
> The mosaic-T supports µSD cards with a **FAT32** file system *(i.e. only cards **up to 32GB** in size)*.

> **INITIAL CONFIGURATION**
>
> Before logging can take place, it is necessary to define a "logging stream" using the **Logging** page or **RxTools**. Streams can contain NMEA or SBF (Septentrio Binary Format) data; SBF can contain RTCM and/or RINEX.



microSD logging stream configuration.

INSTRUCTIONAL VIDEO

How to log data to the SD card of the Septentrio mosaic receiver module

BUTTON OPERATION

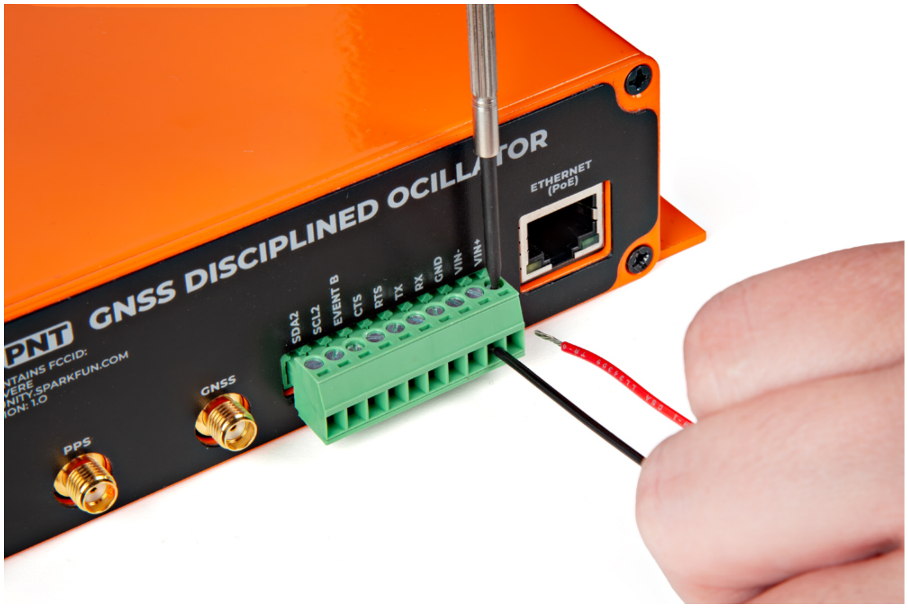
There are multiple ways to configure and enable data logging to an SD card. However, the simplest method is with the LOG button. Once the stream is defined,

- Pressing the LOG button *(<5s)* toggles data logging to the SD card on and off.
- Holding the LOG button for more than 5 seconds *(>5s)* and then releasing it, will force the board to:
  - Unmount the SD card if it was mounted
  - Mount the SD card if it was unmounted

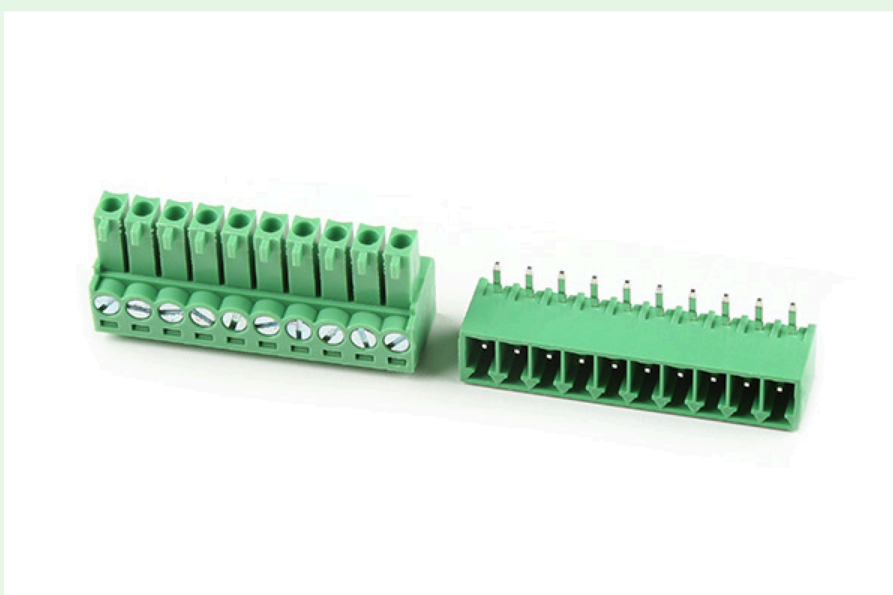
For more information, please reference the SD Card Slot section.

IO Terminals

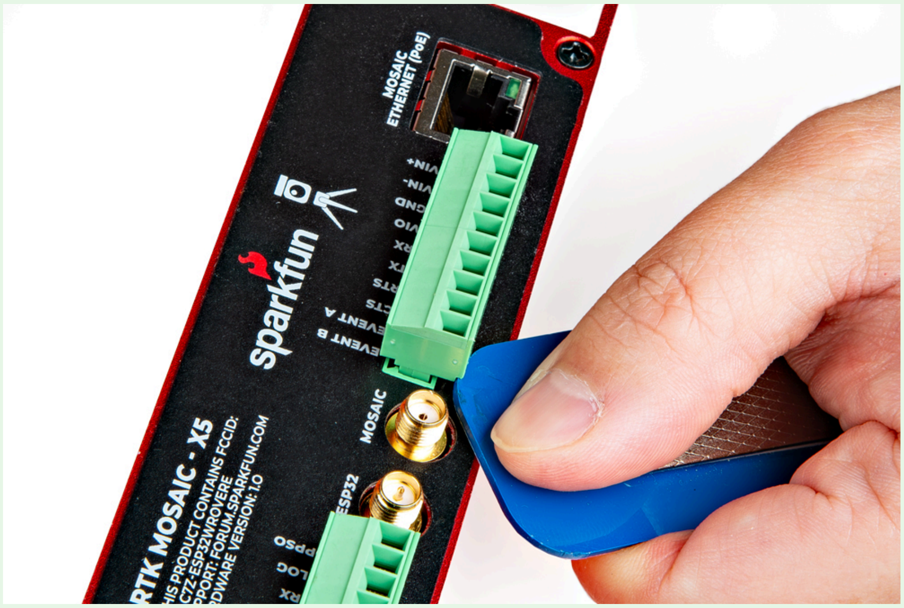
Users can easily attach accessories to the SXT(-D) by wiring them into the terminal blocks on the back of the enclosure.



Connecting a wire to the terminal block.

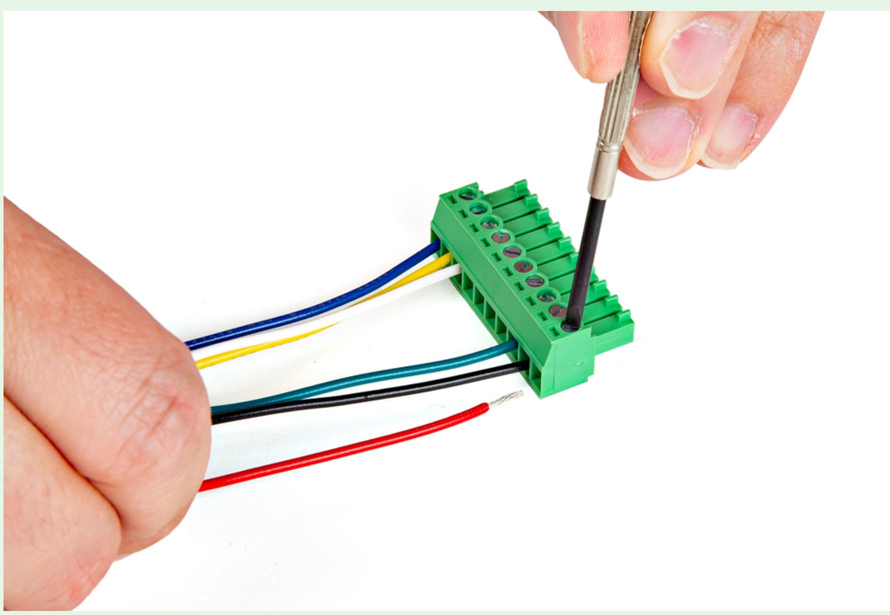
> **MULTIPLE CONNECTIONS**
>
> For multiple connections or wiring harnesses, users can disconnect the terminal block from its socket on the SXT(-D).
>
> 
>
> *Components of the terminal block.*

Users can wiggle or use a soft/rigid object to carefully pry the terminal block off from its connector. In the picture below, a plastic name tag (~1.5mm thick) is used to carefully pry the terminal block up. We have also found the edge of a PCB ruler works great too.



Using a soft/rigid object to carefully pry the terminal block free from its socket.

Once wired up, users can simply push the terminal block back into its socket.



Connecting a wire to the terminal block.

Attaching the terminal block to its socket on the SXT(-D).

> **WARNING**
>
> To avoid shorts or damaging the SXT(-D), verify the wiring with the labels on the back of the enclosure.

Software Overview

ESP32 FIRMWARE

We have intentionally kept the ESP32 firmware as simple as possible - its only tasks are to: discipline the TCXO oscillator; control the OLED display. The intention is that you can easily develop your own firmware for the SXT(-D) if the SparkFun firmware does not meet your needs.

The **/Firmware/Binaries** folder contains the firmware binaries.

You can update or reload the firmware using the SparkFun RTK Firmware Uploader.

You can of course modify the hardware too, should you want to. The design is completely open-source.

NOTE

The mosaic-T module has numerous capabilities and a multitude of ways to configure and interface with them. Without regurgitating all the information that is documented in Septentrio's user manuals and videos, we have tried to highlight a good majority of the module's aspects.

With that said, please feel free to file an issue if you feel we have missed something that may benefit other users. *(Don't forget to provide us with a link to the documentation and what section the information is located.)*

mosaic-T

TIP

Even if you aren't necessarily interested in it, we highly recommend that users install the RXTools software suite before plugging in their board. For Windows PCs, it also includes the USB driver for the module that enables the Ethernet-over-USB support and virtual `COM` ports.

RxTools Software Suite

Users should install the RXTools software suite on their computer to interact with the mosaic-T module through the USB interface. The software package includes the USB-IP driver[1] necessary to recognize the board as an ethernet device on Windows PCs.

Download the RxTools Software from Septentrio

SYSTEM REQUIREMENTS

Operating Systems

- Windows 7
- Windows 8
- Windows 10
- Fedora 23 *(or later)* using Qt technology.
    - The standalone tools (except `bin2asc`) will run on older distributions.

Hardware Requirements The minimal hardware requirements *(1Hz update[2])*:

- CPU: 1 GHz processor
- RAM: 1 GB RAM
- Screen Resolution: 1024×768 or higher resolution

NOTE

The system requirements and installation instructions are from the RxTools *v22.1.0* user manual. This information may change in later iterations of the software suite. Please refer to the user manual *(of the version you are utilizing)* for the most accurate information.

Installation Instructions

Windows Linux

Users can install RxTools software suite by running the installation executable[3], located in the `RxTools\windows` directory of the downloaded `*.zip` file[4]. During the installation process, users will be notified if a previous version of RxTools is already installed then the previous version will be uninstalled. Next, users will need to provide an installation directory for the RxTools software suite. Users will then select which of the following applications[5] are installed:

- RxControl
- SBF Converter
- SBF Analyzer
- RxLogger
- RxUpgrade
- RxDownload
- RxPlanner
- Data Link
- RxAssistant
- RxLauncher

> **INFO**
> For RxTools v22.1.0, the name of the installation executable is `RxTools_22_1_0_Installer.exe` for Windows PCs.

Septentrio USB Driver

Windows Linux

If users haven't already installed the RxTools software suite on their Windows PC, they will need to install the USB driver[1] necessary to recognize and interact with the mosaic-T module through the USB interface.

A Windows USB driver for the mosaic-T can be installed through two methods:

- RxTools Software Suite[10]
- mosaic-T GNSS Receiver Module[11]

Once installed, the driver emulates two virtual serial ports, which can be accessed as standard `COM` ports to the receiver.

HAVING TROUBLE?

For users who are having trouble installing the USB driver, we have an archived version *(v3.0.2[12])* of the installation file. Users can download **version 3.0.2** of the driver, by clicking on the button below.

Download USB Driver (v3.0.2)

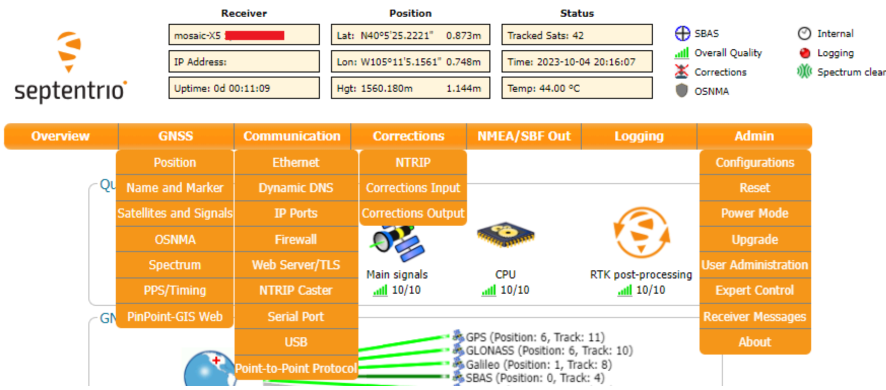
TERMINAL EMULATORS

Most terminal emulation programs will not make a distinction between virtual or native COM ports. However, for virtual serial ports, the port settings *(i.e. baud rate, etc.)* are not relevant and the default configuration is used in the terminal emulation program. However, the physical/native `COM` ports will have the following default setting:

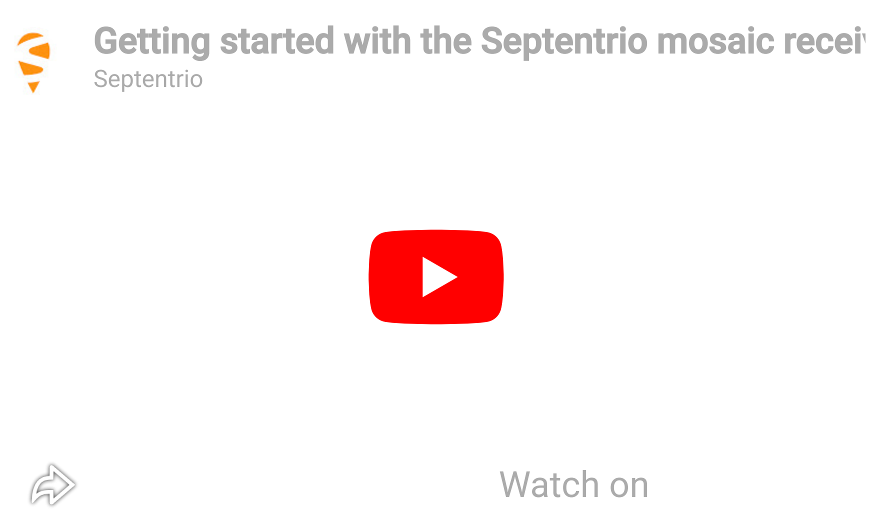
- Baudrate: 115200bps
- Data Bits: 8
- Parity: No
- Stop Bits: 1
- Flow Control: None

Web Interface

With the USB driver installed, the mosaic-T module supports Ethernet-over-USB. The default IP address allocated for the Ethernet-over-USB interface is `192.168.3.1`. This IP can be entered in any browser to open a connection to the receiver's Web Interface as shown below.



All the drop-down navigation tabs in the web interface.



> **INFO**
>
> The default IP address cannot be changed; this feature is only to be used when a single receiver is connected to your computer.

> **INVALID IP ADDRESS *(WIFI ONLY)***
>
> One of the documentation pages on Septentrio's website, specifies a default IP address of `192.168.20.1` for the web interface. However, that address is for a WiFi enabled product and cannot be used with this product.

CH340 USB Driver

Users will need to install a USB driver for the CH340 serial-to-USB chip, in order to communicate with the ESP32 module. The USB drivers for the CH340 USB-to-Serial converter can be downloaded below:

Windows | MacOS | Linux

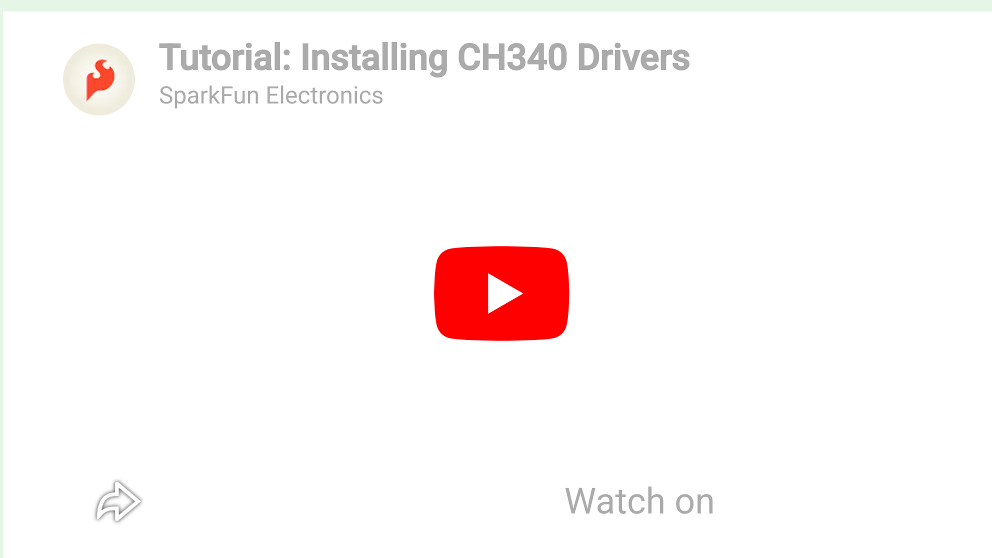
Download the `CH341SER.EXE`

NOTE

If these files don't work, the latest USB drivers for the CH340 are available from the manufacturer on the WCH website.

TIP

For users having trouble installing the CH340 USB driver, check out our video and hookup guide:



Terminal Emulator

In order to configure the firmware settings on the ESP32, users will need to install a serial terminal emulator on their computer.

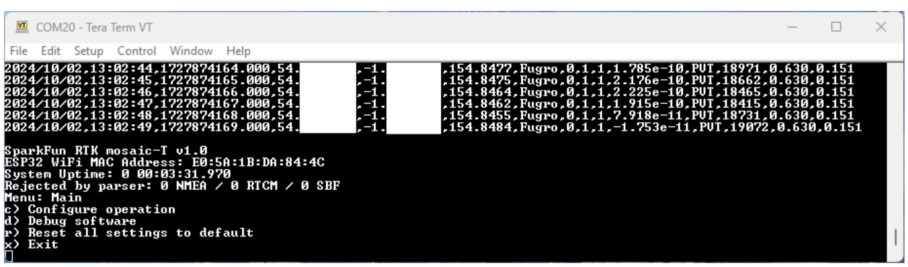
Windows **Linux**

For Windows computers, we highly recommend TeraTerm.

> ⓘ **DEFAULT SETTINGS**
>
> - Baudrate: 115200bps
> - Data Bits: 8
> - Parity: No
> - Stop Bits: 1
> - Flow Control: None

Software Settings

When connected to the ESP32 CH340 COM port at 115200 baud, pressing any key in the terminal emulator will open the firmware Main Menu:



The ESP32 firmware main menu.

Configure Operation

Select option **c** (type `c` followed by `Enter`) to configure the firmware:

```
Menu: Main
c) Configure operation
d) Debug software
r) Reset all settings to default
x) Exit
c

Menu: Operation

1)  Minimum TCXO Warm Up (s):                            120
2)  Required TCXO Temperature Stability (ADU):           0.01
10) PI P term for initial frequency steering:            2.000e-01
11) PI I term for initial frequency steering:            5.000e-01
12) Required bias stability for frequency lock:          1.000e-10
13) PI P term for fast ramping:                          4.000e-01
14) PI I term for fast ramping:                          4.000e-01
15) TCXO steering ramp rate limit (s/s):                 2.500e-07
16) TCXO steering ramp step size (s):                    5.000e-10
17) TCXO steering ramp asymmetry:                        1.000
18) TCXO steering ramp minimum repeats:                  2
19) Required bias for phase lock (s):                    1.000e-07
20) PI P term for final TCXO disciplining:               1.600e-01
21) PI I term for final TCXO disciplining:               1.067e-03
30) Prefer non-composite GPS bias - if available:        Disabled
31) Prefer non-composite Galileo bias - if available:    Disabled
40) Pulse-Per-Second Interval:                           sec1
41) Pulse-Per-Second Polarity:                           Low2High
42) Pulse-Per-Second Delay (ns):                         0
43) Pulse-Per-Second Time Scale:                         GPS
44) Pulse-Per-Second Max Sync Age (s):                   60
45) Pulse-Per-Second Pulse Width (ms):                   5
50) TCP Server (IPS1):                                   Disabled
51) TCP Server Port:                                     28785
60) Save TCXO control word to TCXO memory

x) Exit
```

The ESP32 firmware configuration menu.

- **1) Minimum TCXO Warm Up (s)**
  - This defines the minimum oscillator warmp-up in state STATE_TCXO_WARMUP
  - Default: 120s
  - **SXT-D (GNSSDO+) Only**
- **2) Required TCXO Temperature Stability (ADU)**
  - This defines the required temperature stability to exit STATE_TCXO_WARMUP
  - Default: 0.01 ADU
  - **SXT-D (GNSSDO+) Only**
- **10) PI P term for initial frequency steering**
  - This defines the PI control loop Proportional term used during initial frequency locking
  - The default value depends on the oscillator (SXT vs SXT-D)
- **11) PI I term for initial frequency steering**
  - This defines the PI control loop Integral term used during initial frequency locking
  - The default value depends on the oscillator (SXT vs SXT-D)

- **12) Required bias stability for frequency lock**
  - The firmware will stay in state STATE_GNSS_FINETIME until the *change* in **Bias** reaches this
  - Default: 1.0e-10
- **13) PI P term for fast ramping**
  - This defines the PI control loop Proportional term used during fast ramping (state STATE_GNSS_FREQUENCY_LOCK) to remove the large initial bias
  - The default value depends on the oscillator (SXT vs SXT-D)
- **14) PI I term for fast ramping**
  - This defines the PI control loop Integral term used during fast ramping (state STATE_GNSS_FREQUENCY_LOCK) to remove the large initial bias
  - The default value depends on the oscillator (SXT vs SXT-D)
- **15) TCXO steering ramp rate limit (s/s)**
  - This setting limits the maximum rate (change) in the setpoint during fast ramping
  - The default value depends on the oscillator (SXT vs SXT-D) but is typically 250ns/s
- **16) TCXO steering ramp step size (s)**
  - This setting sets the rate of change of the setpoint during fast ramping
  - The default value depends on the oscillator (SXT vs SXT-D) but is typically 0.5ns
- **17) TCXO steering ramp asymmetry**
  - If needed, this setting can be used to adjust the step size used during the 'down' ramp compared to the 'up'
  - The default value is 1.0 = no asymmetry
  - A value of 0.5 makes the 'down' ramp steps half the size of the 'up' ramp
  - If 16) is set to 1.0ns, and 17) is set to 0.5, the 'down' ramp steps will be 0.5ns. The 'down' ramp will take twice as long as the 'up'
  - If 16) is set to 1.0ns, and 17) is set to 2.0, the 'down' ramp steps will be 2.0ns. The 'down' ramp will take half the time of the 'up'
- **18) TCXO steering ramp minimum repeats**
  - This sets the minimum number of times state STATE_GNSS_FREQUENCY_LOCK is repeated
  - The default value depends on the oscillator (SXT vs SXT-D)
- **19) Required bias for phase lock (s)**

    - The firmware will stay in state STATE_GNSS_FREQUENCY_LOCK if the **Bias** exceeds this after fast ramping
    - Default: 100ns
- **20) PI P term for final TCXO disciplining**
    - This defines the Proportional term for the TCXO frequency PI control loop
    - The default value depends on the oscillator (SXT vs SXT-D)
- **21) PI I term for final TCXO disciplining**
    - This defines the Integral term for the TCXO frequency PI control loop
    - The default value depends on the oscillator (SXT vs SXT-D)
- **30) Prefer non-composite GPS bias - if available**
    - With a subscription to Fugro AtomiChron, this option allows the individual GPS clock bias to be preferred over the composite Fugro bias.
    - You can enable a preference for either GPS or Galileo. Enabling GPS will disable Galileo.
- **31) Prefer non-composite Galileo bias - if available**
    - With a subscription to Fugro AtomiChron, this option allows the individual Galileo clock bias to be preferred over the composite Fugro bias.
    - You can enable a preference for either GPS or Galileo. Enabling Galileo will disable GPS.
- **40) Pulse-Per-Second Interval**
    - This defines the interval of the PPS signal. The intervals are defined by the mosiac-T firmware. Type `40` and hit `Enter` to scroll through the intervals:
        - "off"
        - "msec10"
        - "msec20"
        - "msec50"
        - "msec100"
        - "msec200"
        - "msec250"
        - "msec500"
        - "sec1"
        - "sec2"
        - "sec4"
        - "sec5"

        - "sec10"
        - "sec30"
        - "sec60"
- **41) Pulse-Per-Second Polarity**
    - This defines the PPS signal polarity: Low2High or High2Low
- **42) Pulse-Per-Second Delay (ns)**
    - This allows the timing of the PPS signal to be advanced or retarded. The units are nanoseconds
    - **SXT-D (GNSSDO+) Only:**
        - Advanced users may want to adjust this to ensure a slight mis-alignment between PPS and the 10MHz clock
        - This would ensure their rising edges are slightly separated and that the 74LVC1G175 flip-flop (U15) produces a clean output
        - Consult page 7 of the SXT-D schematic for more details
- **43) Pulse-Per-Second Time Scale**
    - This defines which time scale is used to generate the PPS signal.
        - "GPS"
        - "Galileo"
        - "BeiDou"
        - "GLONASS"
        - "UTC"
        - "RxClock"
- **44) Pulse-Per-Second Max Sync Age (s)**
    - This defines how long PPS pulses will be produced when the GNSS signal is lost or jammed: 0 to 3600 seconds
- **45) Pulse-Per-Second Pulse Width (ms)**
    - This defines the width of the PPS signal. The units are milliseconds: 0.000001 to 1000.000000
- **50) TCP Server (IPS1)**
    - See TCP Server (IPS1) below for more details
- **51) TCP Server Port**
    - The port for the TCP connection. Default is `28785`
- **60) Save TCXO control word to TCXO memory**

- This option allows the current TCXO control word to be saved manually to TCXO internal memory
- **SXT-D (GNSSDO+) Only:**
  - This may be useful when setting up a new SXT-D. E.g. the control word could be saved manually when state `STATE_GNSS_FINETIME` is *almost* complete
  - The control word will be saved automatically when the firmware has been in state `STATE_GNSS_PHASE_LOCK` for one hour

To reset all settings to their default values, select `r`, `Enter`, `y`, `Enter`

The settings are saved to non-volatile memory (NVM, LittleFS) on exiting the menu. Ensure you fully exit the menu (`x`, `Enter`, `x`, `Enter`) to save any modified settings.

The TCXO frequency control word is saved to NVM once per hour, to allow a quicker startup at the next power-on.

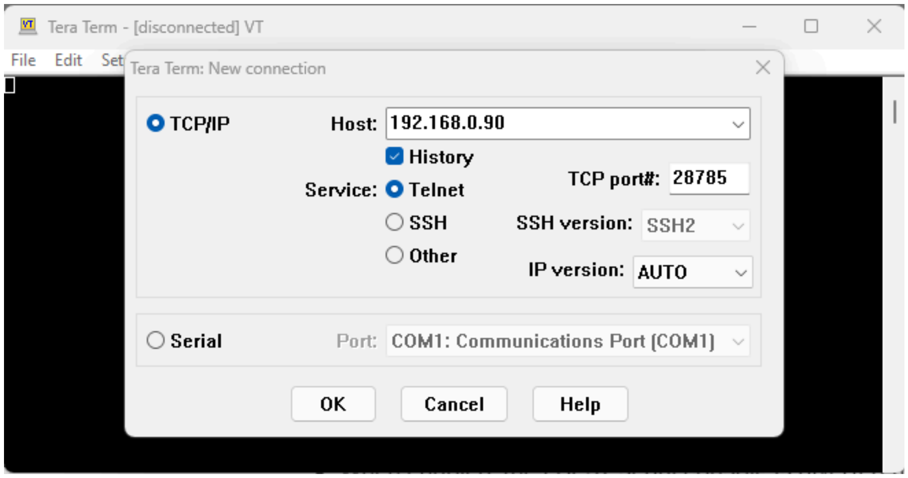
TCP Server (IPS1)

When **TCP Server (IPS1)** is enabled, the ESP32 serial console is diverted from the CH340 USB (CONFIG ESP32) interface to the mosaic-T COM3 UART interface and daisy chained to the IPS1 TCP server. The configuration menu and debug messages can then be accessed over TCP / Telnet on the chosen port. The CH340 USB (CONFIG ESP32) interface is then no longer active.

TCP / Telnet is supported over both Ethernet and the Ethernet-over-USB connection. Only one TCP2Way connection is supported.

If you are using Tera Term:

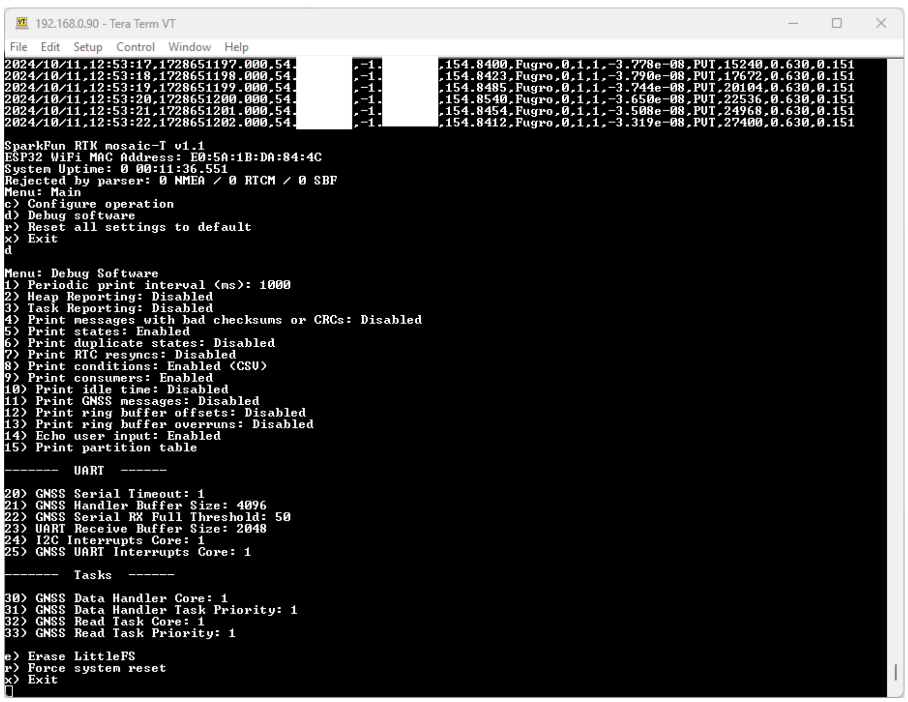
- Select **TCP/IP** and **Telnet**
- Enter the mosaic-T's IP address in **Host**
  - If you are connected via the `CONFIG MOSAIC` Ethernet-over-USB interface, the IP address is 192.168.3.1
- Enter the TCP Server Port number in **TCP port#**
  - The default port is 28785



Tera Term configuration for TCP.

Debug Software

Select option **d** (type `d` followed by `Enter`) to configure the software debug options:



The ESP32 firmware debug menu.

The debug options are what we use at SparkFun to check that the firmware is running correctly. You should not need change any of the options, except perhaps option 8 **Print conditions**. This option controls the periodic CSV messages seen in the console when the menu is closed. The format can be changed to human-readable text, or the messages can be disabled if desired.

Print Conditions (periodic messages)

The format of the **Print conditions** CSV data is:

`YYYY/MM/DD`,`HH:MM:SS`,`Epoch`,`Lat`,`Lon`,`Alt`,`TimeSys`,`Error`,`Fine`,`PPS`,`Bias`,`Drift`,`Source`,`TCXO`,`PkSteer`,`IkSteer`,`PkRamp`,`IkRamp`,`Pk`,`Ik`,`Press`,`Temp`,`Hum`,`TCXO_Temp`,`Mode`,`Temp_Smooth`,`Bias_Smooth`,`Rate`,`Rate_Held`,`Setpoint`,`Turn`,`Error`,`Repeat`

- **YYYY/MM/DD** is the date from the `ReceiverTime` SBF message
- **HH:MM:SS** is the time from the `ReceiverTime` SBF message
- **Epoch** is the date and time in Unix epoch format: `seconds`.`milliseconds` from *midnight UTC January 1st 1970*
  - This is useful when plotting the data against time
- **Lat** is the Latitude in degrees from the `PVTGeodetic` SBF message (7 decimal places)
- **Lon** is the Longitude in degrees from the `PVTGeodetic` SBF message (7 decimal places)
- **Alt** is the Altitude in metres from the `PVTGeodetic` SBF message (0.1mm resolution)
- **TimeSys** is the named `TimeSystem` from the `PVTGeodetic` SBF message
- **Error** is the `Error` byte from the `PVTGeodetic` SBF message
  - `0` indicates no error
- **Fine** is the `FINETIME` bit from the `SyncLevel` byte from the ReceiverTime SBF message
- **PPS** indicates if the Pulse-Per-Second output is enabled
  - PPS is enabled when the RxClkBias reaches the required accuracy, set by **RX Clock Bias Lock Limit**
- **Bias** is the receiver clock bias in seconds
- **Drift** is the receiver clock drift in PPM
- **Source** is the source of the receiver clock bias reported in **Bias**
  - By default, this is **PVT** indicating the source is the composite RxClkBias from the `PVTGeodetic` SBF message
  - If AtomiChron is enabled and if **Prefer non-composite GPS bias** or **Prefer non-composite Galileo bias** has been selected, this will change to **GPS** or **Galileo** indicating that the individual non-composite bias from the FugroTimeOffset SBF message is available and is being used
- **TCXO** is the frequency control word written to the temperature-controlled oscillator

- **PkSteer** is the PI control loop Proportional term used during initial frequency locking
  - Set in the configuration menu
- **IkSteer** is the PI control loop Integral term used during initial frequency locking
  - Set in the configuration menu
- **PkRamp** is the PI control loop Proportional term during fast ramping to remove the large initial bias
  - Set in the configuration menu
- **IkRamp** is the PI control loop Integral term during fast ramping to remove the large initial bias
  - Set in the configuration menu
- **Pk** is the PI control loop Proportional term for the main Phase-Locked Loop
  - Set in the configuration menu
- **Ik** is the PI control loop Integral term for the main Phase-Locked Loop
  - Set in the configuration menu
- **Press** is the atmospheric pressure in hPa (mbar)
  - **SXT-D (GNSSDO+) Only**
- **Temp** is the internal temperature in °C
  - **SXT-D (GNSSDO+) Only**
- **Hum** is the relative humidity in %RH
  - **SXT-D (GNSSDO+) Only**
- **TCXO_Temp** is the temperature-controlled oscillator internal temperature in ADU
  - **SXT-D (GNSSDO+) Only**
- **Mode** is the firmware mode / state. Possible values are:
  - `NOT_CONFIGURED`
  - `CONFIGURED`
  - `ERROR_BEFORE_FINETIME`
  - `WARMUP`
  - `ERROR_DURING_WARMUP`
  - `FINETIME`
  - `ERROR_AFTER_FINETIME`
  - `FREQUENCY_LOCK`
  - `ERROR_AFTER_FREQUENCY_LOCK`
  - `PHASE_LOCK`
  - `ERROR_AFTER_PHASE_LOCK`

- **Temp_Smooth** is the exponentially smoothed **TCXO_Temp**
  - State: `STATE_TCXO_WARMUP` only
  - **SXT-D (GNSSDO+) Only**
- **Bias_Smooth** is the exponentially smoothed *change* in **Bias**
  - State: `STATE_GNSS_FINETIME` only
- **Rate** is the ramp rate (s/s) during state `STATE_GNSS_FREQUENCY_LOCK`
- **Rate_Held** is the capped ramp rate (s/s) during state `STATE_GNSS_FREQUENCY_LOCK`
- **Setpoint** is the Bias target value which during ramping in state `STATE_GNSS_FREQUENCY_LOCK`
- **Turn** is the turn-around point in state `STATE_GNSS_FREQUENCY_LOCK`, from the 'up' ramp to the 'down' ramp
- **Error** is the error term (difference between the **Setpoint** and the **Bias**) in state `STATE_GNSS_FREQUENCY_LOCK`
- **Repeat** is the count of the number of times state `STATE_GNSS_FREQUENCY_LOCK` has been repeated

> **NOTE**
>
> The SXT-D (GNSSDO+) contains a MS8607 pressure, temperature and humidity sensor. The readings are reported in the periodic messages as **Press**, **Temp** and **Hum**.
>
> The temperature reading will be higher than expected due to the sensor's proximity to the OCXO. The OCXO runs at elevated temperature; heat will be coupled to the sensor by convection, radiation and conduction (through the PCB).

Firmware Upgrade

The **/Firmware/Binaries** folder contains the firmware binaries. v3.0 is the latest stable release - as at 4-13-26.

You can update or reload the firmware using the SparkFun RTK Firmware Uploader.

See below for details on how to upload with esptool from the command line.

🛠 Compiling Firmware - using Docker

To compile the GNSSDO Firmware, you need to use the correct version of the ESP32 Arduino core and of each required Arduino library. It is tedious and error-prone. Especially on Windows. We've lost count of the number of times code compilation fails on our local machines, because we had the wrong ESP32 core installed... It is much easier to sandbox the firmware compilation using an environment like Docker.

Docker is open-source. It is our new favourite thing!

Here is a step-by-step guide for how to install Docker and compile the firmware from scratch:

Clone, fork or download the GNSSDO repo

To build the GNSSDO Firmware, you obviously need a copy of the source code.

Click on the icon below to download a copy of the main (released) branch:

Download ZIP

For the real Wild West experience, you can also download a copy of the `release_candidate` code branch. This is where the team is actively changing and testing the code, before it becomes a full release. The code there will *usually* compile and will *usually* work, but we don't guarantee it! We may be part way through implementing some breaking changes at the time of your download...

Download ZIP - release candidate

Install Docker Desktop

- **(Optional)** Head to Docker and create an account. A free "Personal" account will cover occasional compilations of the firmware
- Download and install Docker Desktop - there are versions for Mac, Windows and Linux. You may need to restart to complete the installation.
- Run the Desktop
  - You don't *need* to have an account and you don't *need* to be signed in

- You only need to be signed in if you want to store or share your Container on Docker Hub
  - If you don't sign in, Docker Desktop will run in Personal mode - which will cover local compilations of the firmware
- On Windows, you may see an error saying "**WSL needs updating** Your version of Windows Subsystem for Linux (WSL) is too old". If you do:
  - Open a command prompt
  - Type `wsl --update` to update WSL. At the time of writing, this installs Windows Subsystem for Linux 2.6.1
  - Restart the Docker Desktop
- If you are using Docker for the first time, the "What is a container?" and "How do I run a container?" demos are useful - *but not essential*
  - On Windows, you may want to give Docker Desktop permission to access to your Network, so it can access (e.g.) HTML ports
  - You can Stop the container and Delete it when you are done
- You may want to prevent Docker from running when your machine starts up
  - Uncheck "Start Docker Desktop when you sign in to your computer" in the Desktop settings

Using Docker to create the firmware binary

- **Make sure you have Docker Desktop running.** You don't need to be signed in, but it needs to be running.
- Open a Command Prompt and `cd` into the `SparkFun_GNSSDO` folder
- Check you are in the right place. Type `dir` and hit enter. You should see the following files and folders:

```
.gitattributes
.github
.gitignore
CONTRIBUTING.md
docs
Firmware
```

- `cd Firmware` and then `dir` again. You should see:

```
        app3M_fat9M_16MB.csv
        Binaries
        compile_with_docker.bat
        Dockerfile
        GNSSDO_Firmware
        Utils
```

- The file that does most of the work is the `Dockerfile`
- Run `compile_with_docker.bat` to compile the firmware. It does everything for you
- Type `dir` again. Hey presto! You have your newly compiled firmware binary!

You can then use (e.g.) the SparkFun RTK Firmware Uploader to upload the binary onto the ESP32.

You may find the `.elf` file useful if you are trying to debug the code with me-no-dev's Exception Decoder.

🛠 Uploading Firmware - using esptool

The SparkFun RTK Firmware Uploader contains a copy of Espressif's esptool. It is esptool that actually performs the code upload.

If you want to upload new firmware manually from the command line, here are the commands:

```
esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 460800 --before
default_reset --after no_reset write_flash -z --flash_mode dio --flash_freq 80m
--flash_size detect 0x1000 RTK_Surveyor.ino.bootloader.bin 0x8000
RTK_Surveyor_Partitions_16MB.bin 0xe000 boot_app0.bin 0x10000
GNSSDO_Firmware.ino.bin
```

followed by:

```
esptool.py --chip esp32 --port /dev/ttyUSB0 --before default_reset run
```

On Windows, you can replace `esptool.py` with `esptool.exe`. You can find a copy of `esptool.exe` in the Utils folder.

On Windows, replace `/dev/ttyUSB0` with the name of your CH340 COM port: `COM1` or similar.

You can find copies of `RTK_Surveyor.ino.bootloader.bin`, `RTK_Surveyor_Partitions_16MB.bin` and `boot_app0.bin` in the Firmware Uploader resource folder.

Enjoy!

- Your friends at SparkFun

Footnotes

1. On Linux, the standard Linux CDC-ACM driver is suitable. ↩ ↩[2]
2. Higher data rates will require higher CPU speed and more memory capacity. ↩
3. Users will need administrative privileges to install the RxTools software. ↩ ↩[2]
4. Users may need to extract the RxTools installation files from the downloaded, compressed file. ↩ ↩[2]
5. Please see the release notes for the issues and limitations of the RxTools applications. ↩
6. On most Linux operating systems, the `/dev/ttyS*` devices are owned by `root` and belong to the `uucp` group with read/write (`rw`) access. Additionally, the devices are normally locked by writing a file in the `/var/lock/` directory, with the same permissions. ↩
7. Requires `c` privileges. ↩
8. By default, users will normally have read/write (`rw`) access to the `/dev/ttyS*` serial ports. ↩
9. Changing these permissions also requires `root` privileges. ↩

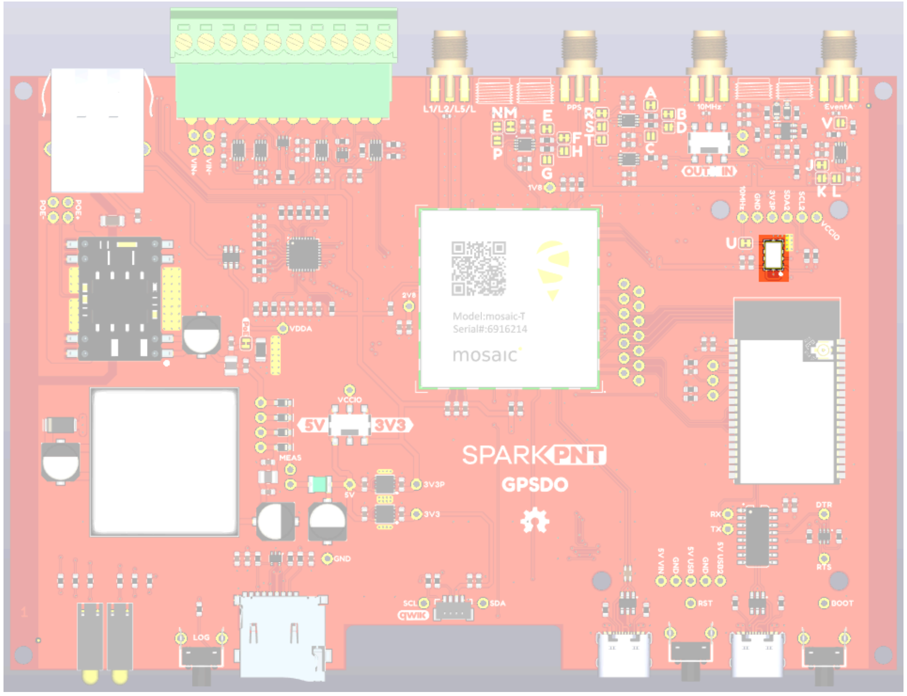
10. The driver is installed during the installation process. ↩

11. The installation file for the Windows USB driver will be available from the mass-storage device when the board is initially connected to the computer. ↩

12. For the latest USB driver from Septentrio, please install their driver through the RxTools software suite.
This driver version was archived at the time that this product manual was written. Please do not request for the file to be updated. ↩

Disciplined Oscillator

SXT: SiTime SiT5358 Disciplined Oscillator



SiT5358 TCXO Oscillator.

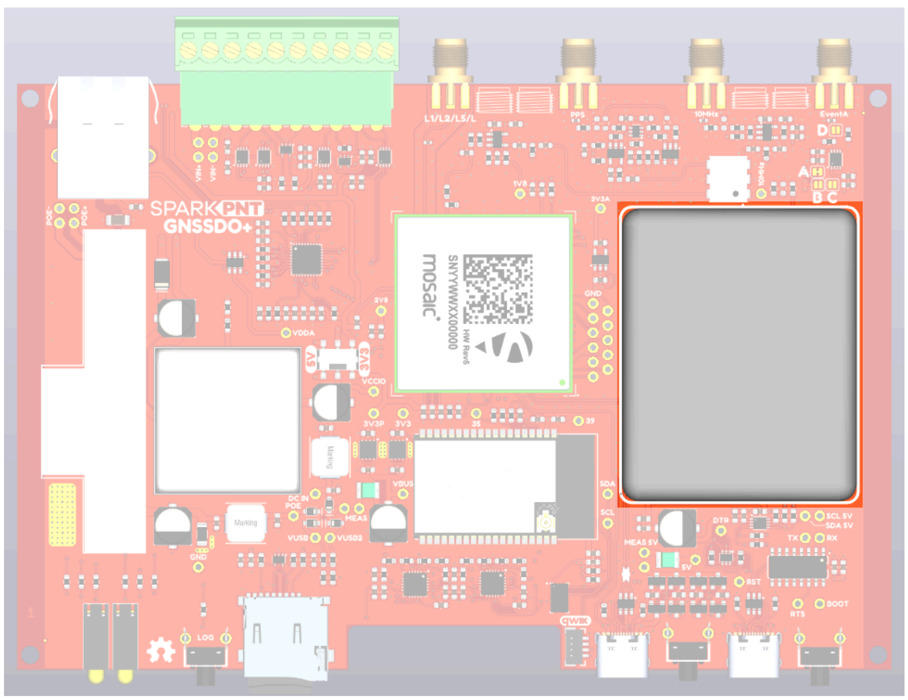
For excellent frequency accuracy and stability, the SparkPNT SXT (GNSSDO) utilizes a SiTime SiT5358 Digitally-Controlled Temperature-Controlled Crystal Oscillator (DCTCXO). The SiT5358 is a precision MEMS Super-TCXO optimized for ±50 ppb stability from -40°C to 105°C. Engineered for best dynamic performance, it is ideal for high reliability telecom, wireless and networking, industrial, precision GNSS and audio/video applications.

- SiT5358AI-FS033IT-10.000000 10MHz DCTCXO
- ±50ppb stability
- ±1ppb/°C frequency slope
- ±58ppb typical 20-year aging
- Allan Deviation approaches 1E-14 at 10000 seconds with AtomiChron enabled

- Digital frequency pulling via I²C
- Operating temperature: -40 to 85 °C (Industrial)

The SiT5358 is interfaced to the mosaic-T through a level-shifting buffer, replacing the mosaic's internal oscillator and allowing the oscillator frequency to be tuned (disciplined) under software control. The 10 MHz SMA output is generated by a duplicate level-shifter to ensure an equal delay and identical thermal phase changes.

SXT-D: Rakon STP3593LF Double-Oven Disciplined Oscillator



STP3593LF OCXO Oscillator.

For the best frequency accuracy, stability and holdover, the SparkPNT SXT-D (GNSSDO+) utilizes a Rakon STP3593LF Double-Oven Crystal Oscillator (OCXO). The STP3593LF is a High End Telecom OCXO with embedded module, typically found in 5G Grand Master or Stratum-2 clocks.

- STP3593LF Double-Oven OCXO - ROX5242T1N Family
- Frequency Stability better than 50ppt peak to peak over operating range from -32°C to +70°C
- Frequency calibration +/- 50ppb at +25°C ± 2°C

- Frequency Stability Vs. Time (Aging): slope per day ± 0.2 ppb
- SSB Phase Noise (typical): -100 at 1Hz; -155 at 1kHz; -170 at 1MHz
- Allan Deviation below 1E-14 at 10000 seconds with AtomiChron enabled
- Digital frequency pulling via I²C
- Operating temperature: -32 to 70 °C

The STP3593LF is interfaced to the mosaic-T through a level-shifting buffer, replacing the mosaic's internal oscillator and allowing the oscillator frequency to be tuned (disciplined) under software control. The 10 MHz SMA output is generated by a duplicate level-shifter to ensure an equal delay and identical thermal phase changes.

The PPS output is aligned to the 10MHz clock by a 74LVC1G175 flip-flop (U15). Consult page 7 of the SXT-D schematic for more details

Software Control Loop

The oscillator is interfaced to the mosaic-T according to Appendix D of the mosaic Hardware Manual: "mosaic-Based Disciplined Clock".

The ESP32 is interfaced to three of the mosaic-T's COM (UART) ports: `COM1`, `COM3`, and `COM4`. The ESP32 configures the mosaic-T via `COM4`. `COM1` is dedicated as an output port for the SBF blocks used to tune the oscillator frequency. `COM3` is used to divert the ESP32 serial console to TCP.

When the firmware boots, the mosaic-T is configured as follows:

```
setPPSParameters, off
setClockSyncThreshold, usec500, off
setSBFGroups, Group1, PVTGeodetic+ReceiverTime
setSBFOutput, Stream1, COM1, Group1, sec1
setSBFOutput, Stream2, COM1, IPStatus+FugroTimeOffset, OnChange
setEthernetMode, on
setPVTMode, Static, , auto
exeCopyConfigFile, Current, Boot
```

These commands configure the mosaic so that it: starts-up with the PPS pulses disabled; performs an initial synchronization to GNSS time; and outputs the **PVTGeodetic** and **ReceiverTime** blocks on COM1 at 1Hz. **IPStatus** is output each time one or more IP parameters change. With a Fugro Atomichron subscription, the **FugroTimeOffset** message will be generated each time one of the clock biases changes.

> **NOTE**
>
> As of GNSSDO firmware version 3.0, the firmware no longer uses **setClockSyncThreshold** (scst) `StartupSync`. `StartupSync` is off.
>
> This prevents a shift in PPS with respect to the 10MHz clock, but it does mean that the initial bias error can be up to 500us and it can take a long time (~30 minutes) to get the bias down to zero by ramping the frequency.
>
> The firmware no longer performs a soft reset if the bias is excessive on startup.

The firmware monitors the message blocks on `COM1`. The `ERROR` LED follows the **PVTGeodetic Error** code. The LED is extinguished when **Error** is zero.

In STATE_GNSS_CONFIGURED:

- The firmware waits for the **ReceiverTime** `FINETIME` bit to be set, indicating the **setClockSyncThreshold** `Threshold` (usec500) has been achieved

In STATE_TCXO_WARMUP (SXT-D GNSSDO+ only):

- The firmware waits for `tcxoMinWarmup_s` and for the OCXO temperature to become stable

In STATE_GNSS_FINETIME:

- The firmware disciplines the TCXO / OCXO frequency in a simple frequency locked loop
- The firmware exits this state when the change in the `tcxoClockBias_ms` (`RxClkBias`) is better than the specified `rxStabilityForFrequencyLock`
- The PI loop uses `PkSteer` and `IkSteer` when frequency locking

In STATE_GNSS_FREQUENCY_LOCK:

- The TCXO / OCXO bias is driven towards zero

- The initial bias error can be up to 500us, so we need to ramp the TCXO frequency one way ('up') and then back again ('down')
- We slowly accelerate the change in frequency, maintain it at `tcxoRampRateLimit_sps`, then decelerate
- The acceleration is set by the `tcxoRampStepSize_s`: 0.5ns/s for the STP3593; the SiT5358 can accelerate faster
- The ramps help to avoid blowing up the integrator
- When the ramps are complete, the `tcxoClockBias_ms` (`RxClkBias`) is re-checked
- This state is repeated if the bias is still excessive (> `rxPhaseErrorLimit_s`)
- The PI loop uses `PkRamp` and `IkRamp` when following the ramps

In STATE_GNSS_PHASE_LOCK:

- The LOCK LED will be illuminated when the firmware is in STATE_GNSS_PHASE_LOCK
- PPS output will be started when entering STATE_GNSS_PHASE_LOCK for the first time
- This state uses a conventional PLL to drive the `tcxoClockBias_ms` (`RxClkBias`) to zero
- The P and I terms are `Pk` and `Ik`
- When entering STATE_GNSS_PHASE_LOCK from STATE_GNSS_FREQUENCY_LOCK, the P and I terms are ramped from `PkRamp` and `IkRamp` to `Pk` and `Ik` over `phaseLockPIRampTime_s` (120s)
  - This helps to avoid shocking the PI loop and sending the bias off for a walk...

The firmware monitors the **PVTGeodetic RxClkBias** (as 64-bit float in milliseconds). When **RxClkBias** is positive, receiver time is ahead of system time and the oscillator frequency should be reduced. When **RxClkBias** is negative, receiver time is behind system time and the oscillator frequency should be increased. The firmware will adjust the oscillator frequency via its Digital Frequency Control register accordingly. As per Appendix D, the frequency will be changed by no more than 3ppb per second.

The Proportional (P) and Integral (I) terms can be adjusted via the ESP32 firmware settings.

The Pulse-Per-Second parameters can be configured by the user through the ESP32 firmware settings: interval, polarity, delay, time scale, max sync age, pulse width.

```
setPPSParameters, Interval, Polarity, Delay, TimeScale, MaxSyncAge, PulseWidth
```

Should **PVTGeodetic Error** indicate an error (i.e. become non-zero), the loop will continue but no changes will be made to the oscillator frequency until the error is cleared.

Fugro AtomiChron®

Fugro AtomiChron® is a global, real-time, precise clock synchronisation and authentication service based on GNSS, for applications that require resilient, accurate and reliable time and frequency. This innovative technology eliminates time drift caused by clocks counting time at slightly different rates and gives users access to extreme frequency stability as well as extreme accurate time references, with levels that surpass high-calibre industrial standard caesium clocks and approaching hydrogen maser performance.

Fugro AtomiChron overview.

Specifications

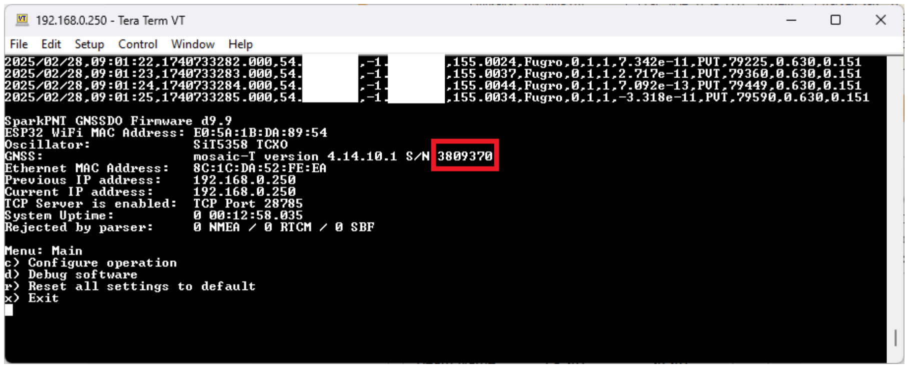
- Timing accuracy: <5 ns UTC (95% of the time)
- Accuracy: <1 ns to Fugro AtomiChron® timescale (independent of world-wide location)
- Performance improvement: >10x better than existing high-end GNSS receivers
- Navigation Message Authentication (NMA) status: per satellite status for all 4 GNSS constellations concurrently

Subscribing to Fugro AtomiChron®

AtomiChron® subscriptions are managed by Fugro. We ask that you complete the SparkPNT Fugro AtomiChron® Service Registration Form and the team at Fugro will be in touch. To subscribe, Fugro will need to know the serial number of the mosaic-T inside your SXT(-D). You can find the mosaic-T serial number in several ways:

- You can view it on the Fugro AtomiChron tab of mosaic-T's internal web page. The web page can be viewed over Ethernet or Ethernet-over-USB. Please see the Quick Start Guide for more details.
- The serial number is also displayed when you open the Software Settings menu in a terminal emulator or serial console.

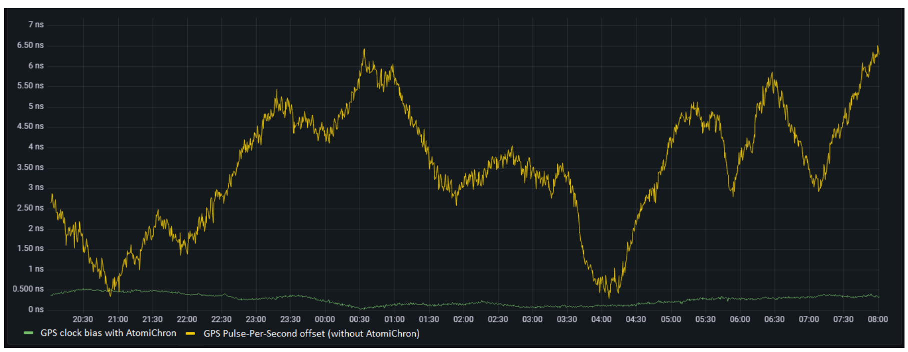
Fugro AtomiChron ID on the mosaic-T web page.



Fugro AtomiChron ID in the settings of the GNSSDO firmware.

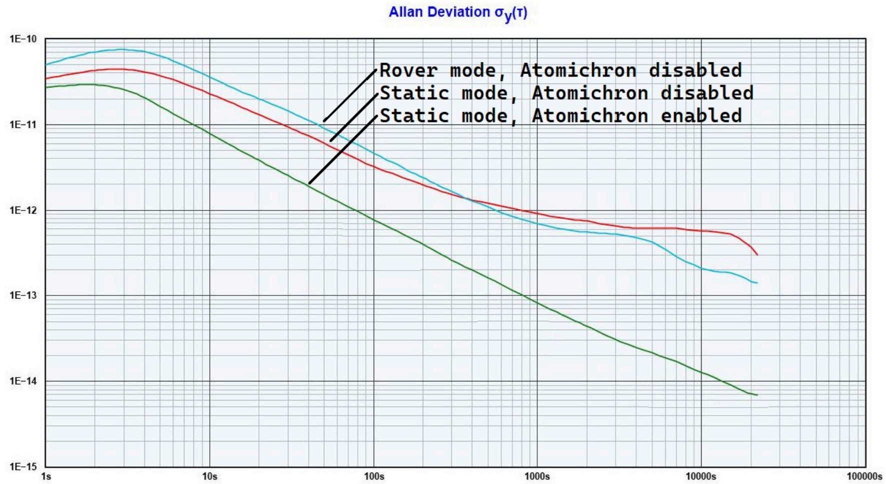
Performance

With AtomiChron enabled, the SXT(-D) achieves a GPS clock bias of better than 1 ns:



GPS clock bias with AtomiChron vs GPS Pulse-Per-Second offset (without AtomiChron).

With AtomiChron enabled, the SXT Pulse-Per-Second Allan Deviation approaches 1E-14 at 10000 seconds:



SXT: *Pulse-Per-Second Allan Deviation with and without AtomiChron.*

On the SXT-D, the Pulse-Per-Second Allan Deviation is below 1E-14 at 10000 seconds:

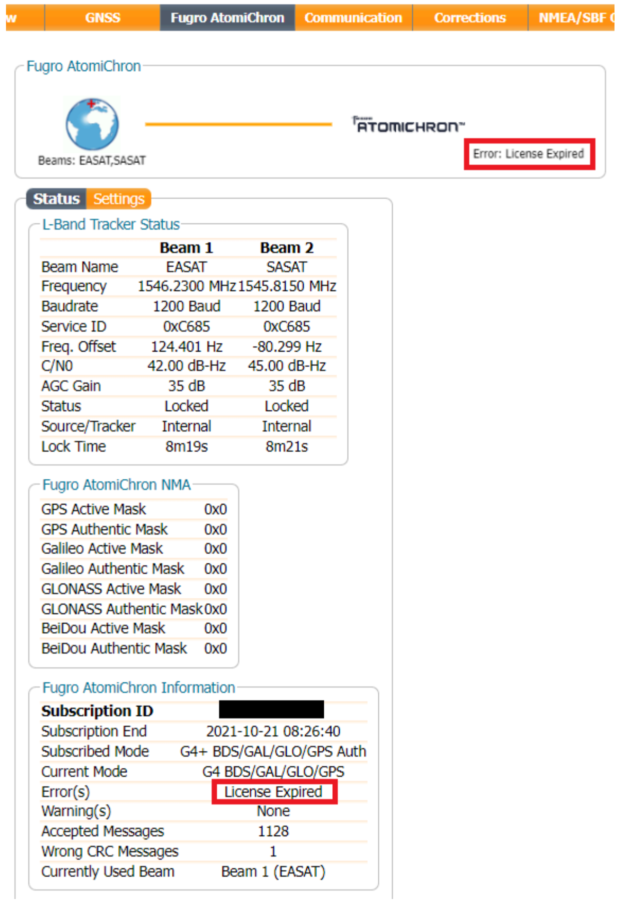
Tau	Sigma(Tau)
1s	9.72E−13
2s	1.12E−12
4s	1.24E−12
8s	1.07E−12
10s	9.66E−13
20s	6.32E−13
40s	3.79E−13
80s	2.19E−13
100s	1.84E−13
200s	1.05E−13
400s	5.70E−14
800s	3.32E−14
1000s	2.75E−14
2000s	1.62E−14
4000s	1.14E−14
8000s	7.84E−15
10000s	7.02E−15
20000s	4.09E−15
40000s	3.02E−15

Trace	Notes	Filename	Input Freq	Sample Interval	ADEV at 1s	Duration	Elapsed	Acquired
GNSSDO+	6db pad	2026.04.24 GNSSDO plus 2d.tim	10 MHz	1 s	9.72E−13	2d 0h 0m 0s	2d 0h 0m 0s	172800 pts

SXT-D: *Pulse-Per-Second Allan Deviation with AtomiChron.*

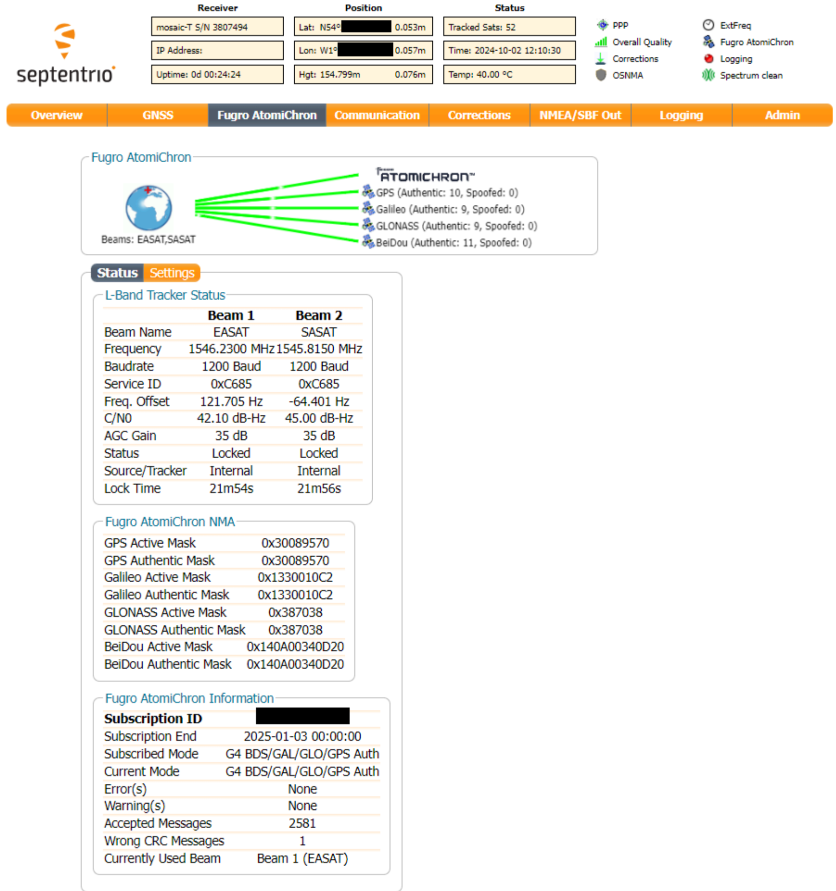
Firmware configuration

Without an AtomiChron subscription, the firmware will use the composite GNSS **RxClkBias** from **PVTGeodetic** to discipline the TCXO frequency.



By default, the Fugro AtomiChron subscription shows as "expired".

With AtomiChron, the firmware will use the Fugro composite bias reported in **FugroTimeOffset** to discipline the TCXO frequency. You also have the option to prefer the individual GPS or Galileo clock bias over the Fugro composite bias. Please refer to the ESP32 firmware settings for more details.



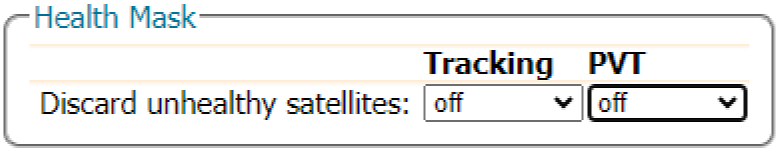
The mosaic-T with an active Fugro AtomiChron subscription.

Enable the L5 Band

The mosaic-T is L5-capable but, because the GPS L5 service is currently pre-operational and marked as "unhealthy", it takes some extra configuration steps to enable L5. The GNSSDO firmware enables L5 reception by default. Here are the configuration steps in case you need to configure L5 manually.

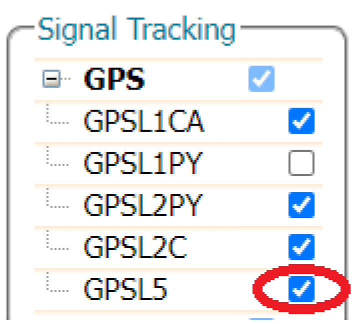
Web Interface

Admin > Expert Control > Control Panel > Navigation > Receiver Operation > Masks - Set "Discard unhealthy satellites" to off for both "Tracking" and "PVT"



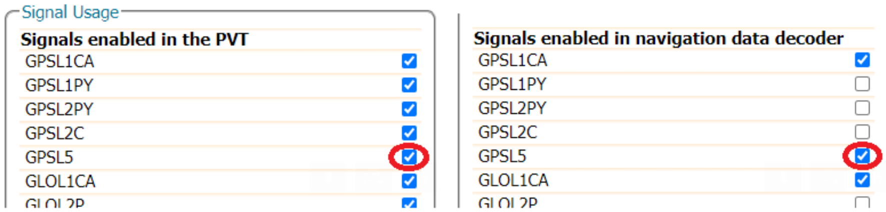
L5 Configuration - Step 1 (PNG).

Navigation > Advance User Settings > Tracking > Signal Tracking : Enable GPSL5 signal tracking



L5 Configuration - Step 2 (PNG).

Navigation > Advance User Settings > PVT > Signal Usage - Enable GPSL5 in both “PVT” and “navigation data decoder”



L5 Configuration - Step 3 (PNG).

> **INFO**
>
> Make sure to click "OK" at the bottom of each page where you are making changes to update the current configuration, and don't forget to save it to boot if you want the configuration to persist when you cycle power.

Command Interface

- `setHealthMask, Tracking, off`
- `setHealthMask, PVT, off`
- `setSignalTracking, +GPSL5`
- `setSignalUsage, +GPSL5, +GPSL5`

Support

🧑 Troubleshooting Tips

A troubleshooting guide for users

🗃️ Resources

Resources for the product, including datasheets, application notes, and manuals

Troubleshooting Tips

If you need technical assistance or more information on a product that is not working as you expected, please head over to our SparkPNT Forum. By posting on the forum you'll assist other users with similar questions; it also gives us feedback as a whole about where our products and documentation need improvement.

> **ACCOUNT REGISTRATION REQUIRED**
>
> On your first visit to our forum, you'll need to create a Forum Account to post questions.

What is in the Box?

The SXT and SXt-D come packaged as a complete kit, with all the accessories you'd need to set them up.

Everything that is included in our kits.

Inside the box, users will find the GNSS antenna, SXT(-D) in its aluminum enclosure, CAT-6 Ethernet cable, USB-C cable, and another box containing additional accessories. Inside the accessory box, users will find the SMA to TNC cable, SMA to BNC cable, USB PD wall adapter, WiFi antenna, silicone bumpers, and 32GB SD card.

If you are missing any of these items, please reach out to us in our forum.

SparkPNT Forum

OLED Dead Pixels

Got a few dead pixels on your OLED display? Please reach out to us in our forum.

SparkPNT Forum

Data Logging

For data logging issues, here are some simple troubleshooting tips:

- Make sure that your SD card is formatted to a `FAT32` file system.
  - The `FAT32` file system also limits the maximum capacity of the card to less than **32GB** *(i.e. a 256GB SD card will not work)*.
- Make sure that the mosaic-T module has a configured data stream output.
- Use the mosaic-T web page to verify that the SD card is mounted as a storage drive.

L5 Band

By default, the L5 band is enabled on the mosaic-T. To disable or re-enable reception of the L5 band on the mosaic-T, we have provided instructions on the Enable the L5 Band page.

Reprogramming the ESP32

In case users accidentally reprogram or corrupt the flash memory on the ESP32, the **/Firmware/Binaries** folder contains the firmware binaries.

You can update or reload the firmware using the SparkFun RTK Firmware Uploader.

The full firmware source code is available in our GitHub repository

Enclosure Disassembly

Due to the ESD sensitivity of the mosaic-T module, we don't recommend disassembling the SXT(-D). However, if users must access the PCB to troubleshoot an issue, make a modification, or repair a component, we highly recommend that they take the necessary ESD precautions to avoid damaging the mosaic-T module.

READ BEFORE DISASSEMBLY!

ESD SENSITIVITY

The mosaic-T module is sensitive to ESD. Use a proper grounding system to make sure that the working surface and the components are at the same electric potential.

ESD PRECAUTION

As recommended by the manufacturer, we highly encourage users to take the necessary precautions to avoid damaging their module.

- The SXT(-D) features ESD protection on the USB-C connectors, ethernet jack, I/O terminals and SMA antenna connections.

Getting started with the Septentrio mosaic recei

Septentrio

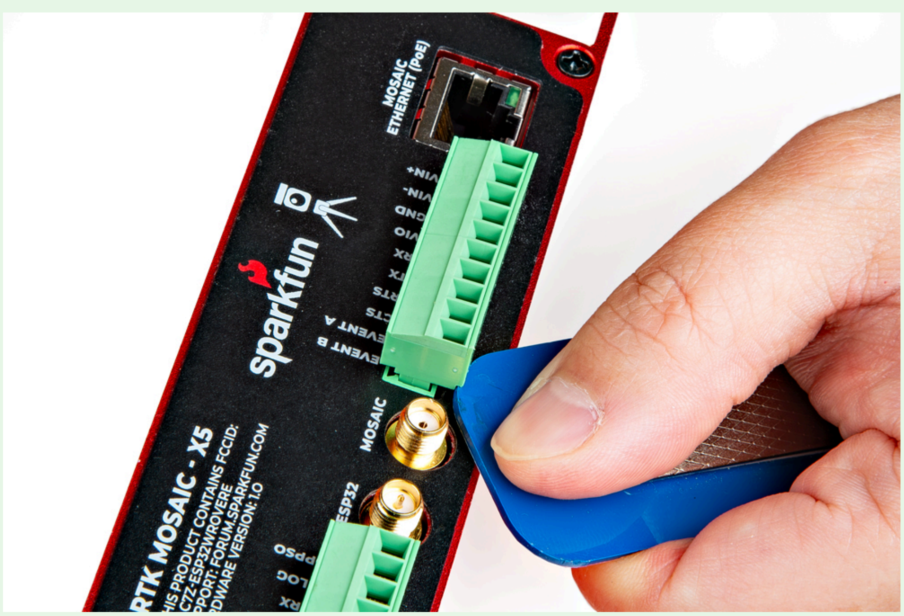
Watch on

Before disassembling the enclosure, users should disconnect the power and all cables from the SXT(-D). This will prevent users from creating any shorts and will facilitate the teardown process further on.

The SXT(-D) PCB is held in place by the front and rear panels of the enclosure. While users can remove the panels first, we recommend that users disconnect the screw terminal blocks first. Users may find it more difficult to separate the terminal block from the connector later in the disassembly process.

REMOVE THE SCREW TERMINAL BLOCK FIRST

Users may find it easiest to remove the screw terminal blocks first; as opposed to later in the teardown process.



Using a soft/rigid object to carefully pry the terminal block off.

Users can wiggle or use a soft/rigid object to carefully pry the terminal block off. In the picture above, a plastic name tag (~1.5mm thick) is used to carefully pry the terminal block up. We have also found the edge of a PCB ruler works great too.

Once the terminal blocks have been removed, users can remove the front and rear panels of the enclosure. They are held in place with four, M3, Phillips head screws on the corners of each panel.

> **REMOVE FRONT PANEL FIRST**
>
> We recommend removing the front panel first to prevent the Qwiic cable from being yanked off the OLED display or main PCB. Once the front panel is free, carefully lift the panel and disconnect the Qwiic cable from the top connector on the OLED display.

At this point, if users have previously disconnected all the cables and the terminal blocks from the back, the SXT(-D) PCB should slide out of the enclosure. Users can then, remove the rear panel from the enclosure to complete the teardown process.

SXT Clock Frequency Reference

The mosaic-T module can either use its internal TCXO or accept an external signal as a frequency reference. However, the module will constantly reboot if the `REF_I` pin isn't provided a 10MHz sinusoidal signal and left floating. Therefore, SXT users should only set the `10MHz` switch to the `IN` position, only if a 10MHz clock signal can be provided; otherwise, the switch should remain in the `OUT` position.

> **NOTE**
>
> Switching between an external and internal frequency reference must occur when the mosaic-T is powered off, or the module must be reset after switching.

SXT-D Power Supply

Powering the SXT-D (GNSSDO+) takes a little more thought and care. The double-oven oscillator in the SXT-D draws more current than a standard USB (2.0) port can provide. You need to ensure that you use the provided 65W Power Delvery Wall Adapter if you are going to power the SXT-D from USB. The priorities of the Power Path priority circuit, from highest to lowest are:

- **`Power-over-Ethernet`** - **Range: 36 to 57V**; delivered via the `MOSAIC ETHERNET` RJ45 MagJack connector.
- **`External DC Power`** - **Range: 9 to 36V**; delivered via the `VIN+` and `VIN-` screw cage terminals.
- **`CONFIG ESP32 USB Port`** - **5V from a Power Delivery Wall Adapter**; delivered via the `CONFIG ESP32` USB-C connector.
- **`CONFIG MOSAIC USB Port`** - **5V from a Power Delivery Wall Adapter**; delivered via the `CONFIG MOSAIC` USB-C connector.

> **NOTE**
>
> - If you have the **`CONFIG ESP32 USB Port`** connected to a standard USB (2.0) port, it will not be able to provide enough current. And due to the Power Path priority, it will not preferentially draw power from **`CONFIG MOSAIC USB Port`**. Your options are:
>   - Provide power via **`Power-over-Ethernet`** or **`External DC Power`** since these have higher priority
>   - Install a power switch in the **`CONFIG ESP32 USB Port`** cable to disconnect its power, forcing the SXT-D to draw Power Delivery power from the **`CONFIG MOSAIC USB Port`**.
> - If you have **`CONFIG MOSAIC USB Port`** connected to a standard USB (2.0) port, ensure the Power Delivery Wall Adapter is connected to **`CONFIG ESP32 USB Port`**. The Power Path priority will ensure power is drawn from the **`CONFIG ESP32 USB Port`**.

> **INFO**
>
> Consult page 9 of the SXT-D schematic for more details

Resources

Product Resources

A list of resources for the SXT(-D) and its components

Product Comparison

A comparison of similar products utilizing the Septentrio mosiac-T GNSS module

Product Resources

Product Specific Resources

SparkPNT SXT (GNSSDO)

- Product Page
- PCB Design Files:
  - Schematic
  - Board Dimensions
  - Board Dimensions
  - KiCad Files
- SiT5358 Oscillator Datasheet
  - Allan Deviation

SparkPNT SXT-D (GNSSDO+)

- Product Page
- PCB Design Files:
  - Schematic
  - Board Dimensions
  - Board Dimensions
  - KiCad Files
- STP3593LF Oscillator Datasheet

- Allan Deviation

Additional Resources

- Quick Start Guide
- CH340 USB-to-Serial Converter
    - Datasheet
    - USB Drivers
        - Windows: `CH341SER.EXE`
        - MacOS: `CH341SER_MAC.ZIP`
        - Linux: `CH341SER_LINUX.ZIP`
- ESP32 Microcontroller
    - Module Datasheet
    - Reference Manual
    - SoC Datasheet
    - ESP32 Arduino Core

      *.json file needed for Epressif's ESP32 Arduino Core

      ```
      https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
      ```

    - Firmware
        - Source Files
        - Pre-Compiled Binaries
- mosaic-T GNSS Receiver
    - Datasheet
    - Reference Manual
    - RxTools Software Suite
    - Firmware
        - `v4.14.10.1`
        - `v4.15.0`
        - `v4.15.3`

- Antenna Info
  - `v25.1.0`
  - `v26.1.0`
- Part Dimensions:
  - Enclosure Dimensions
  - Antenna Dimensions
- Hardware Repo
- Precision Timing Products
- Fugro AtomiChron® Service
- SparkPNT Forum

Manufacturer's Resources

Septentrio **Fugro** **SiTime** **Rakon** **Espressif**

Septentrio also provides great resources for the mosaic-T:

- mosaic-T Product Page
  - Resources: Datasheet, firmware, manual, software tools, USB driver, etc.
- Support Portal:
  - Septentrio Knowledge Base
- YouTube Channel
- Learn More - Insights
  - How GPS brings time to the world
  - How GPS helps keep time, heartbeat of a connected society
  - The Chirp Jammer: a GPS hit and run
  - JammerTest 2023 - Anti-Jamming and Anti-Spoofing Performance

Recommended Articles

General Knowledge

- mosaic - Getting started
- Resetting a receiver
- Resetting and clearing the non-volatile memory
- Generating a diagnostic report of a Septentrio receiver
- RTCMv3 messages supported by the Septentrio receivers
- Interference Detection, Mitigation, and Flagging

Timing Applications

- How to set up the PolaRx5TR for timing applications?
- Activating the mosaic-T for Fugro AtomiChron
- How to log CGGTTS files on a PolaRx5TR?
- PPS IN monitoring with a PolaRx5TR
- How to get scintillation Indices from PolaRx5S receiver?

Providing RTK Corrections

- How to set up a base station
- RTK positioning with a moving baseline
- Set up an (internal) NTRIP caster on a Septentrio receiver

Applying RTK Corrections

- Receive corrections over IP
- Receive corrections via NTRIP
- Using uBlox's correction services for precise positioning

Antenna Corrections

- Antenna calibration
- Applying ARP/APC offsets

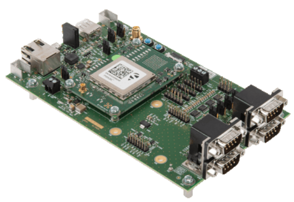
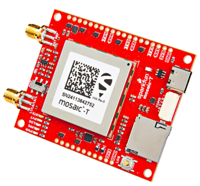
Detecting and Mitigating Interference

- How to Activate Wideband Interference Mitigation
- How to mitigate interference with notch filters
- Pulse-Blanking
- Front-end Saturation Caused by Interference
- Visualizing interference using the spectrum plot
- SBF Analyzer: Carrier to Noise

Recommended Videos

- Getting started with the mosaic receiver module
- Output NMEA data on the mosaic receiver module
- Receiving NTRIP corrections on the mosaic module
- Logging data to the Septentrio mosaic receiver module
- Receiving corrections over an IP connection
- Share the USB internet connection on your receiver
- Septentrio's Agnostic GNSS Corrections program
- Test-run for Septentrio's anti-jamming feature
- How to Activate Wideband Interference Mitigation
- How to set a notch filters
- How to use a spectrum plot
- mosaic - GNSS module receiver range from Septentrio

Product Comparison

	mosaic-T Development Kit	mosaic-T GNSS Breakout	SXT	SXT-D
				
RTK Rover	No	No	**No**	**No**
RTK Base	No	No	**No**	**No**
GNSS Antenna	Dual	Single	Single	Single
USB Connector	micro-B	Type-C	Type-C	Type-C
Ethernet	Yes *10/100 Base-T*	No	Yes *10/100 Base-T*	Yes *10/100 Base-T*
WiFi	No	No	Yes - Network Bridge *10 Base-T*	Yes - Network Bridge *10 Base-T*

COM Ports	4	4	1 - mosaic-T 1 - ESP32	1 - mosaic-T 1 - ESP32
µSD Card Slot	Yes	Yes	Yes	Yes
Reset/Log Buttons	Yes	Yes	Yes	Yes
Logic-Level	1.8V 3.3V	3.3V	3.3V 5V	3.3V 5V
PPS Signal	Header Pin	PTH Pad	SMA Connector	SMA Connector
10MHz Signal	SMA Connector *In Only*	SMA Connector *In/Out*	SMA Connector *In/Out*	SMA Connector *Out only*
Enclosure Material	N/A	N/A	Aluminum	Aluminum
Dimensions	N/A	635 x 508 x 8mm	180.6 x 101.8 x 41mm *Enclosure Only*	180.6 x 101.8 x 41mm *Enclosure Only*
Weight	N/A	23g	415.15g *Enclosure Only*	415.15g *Enclosure Only*