ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

The Ublox GPS (SparkFun GPS-16329) is a very accurate, powerful location sensor in a tiny I2C enabled package. Along with the power comes considerable complexity. Please see the end of this document for how to connect your Antenna to board.

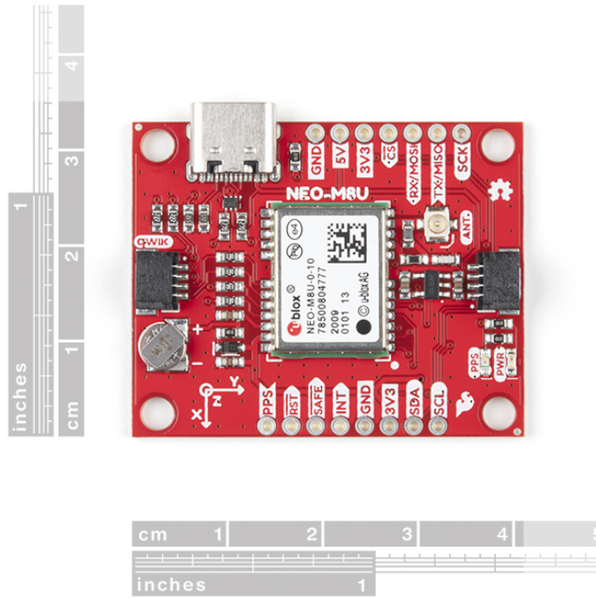
Ublox GPS overview

The SparkFun NEO-M8U GPS Breakout is a high quality, GPS board with equally impressive configuration options. The NEO-M8U takes advantage of u-blox's Untethered Dead Reckoning (UDR) technology. The module provides continuous navigation without needing to make any electrical connection to a vehicle, thus reducing cost of installation for after-market dead reckoning applications.

The NEO-M8U module is a 72-channel u-blox M8 engine GNSS receiver, meaning it can receive signals from the GPS, GLONASS, Galileo, and BeiDou constellations with ~2.5 meter accuracy. The module supports concurrent reception of three GNSS systems. The combination of GNSS and integrated 3D sensor measurements on the NEO-M8U provide accurate, real-time positioning rates of up to 30Hz.

Compared to other GPS modules, this breakout maximizes position accuracy in dense cities or covered areas. Even under poor signal conditions, continuous positioning is provided in urban environments and is also available during complete signal loss (e.g. short tunnels and parking garages). With UDR, position begins as soon as power is applied to the board even before the first GNSS fix is available! Lock time is further reduced with on-board rechargeable battery; you'll have backup power enabling the GPS to get a hot lock within seconds!

Additionally, this u-blox receiver supports I2C (u-blox calls this Display Data Channel) which makes it perfect for the Qwiic compatibility so you don't have to use up your precious UART ports. Utilizing SparkFun's handy Qwiic system, no soldering is required to connect it to the rest of your system. However, it still has broken out 0.1"-spaced pins in case you prefer to use a breadboard.



U-blox based GPS products are configurable using the popular, but dense, windows program called **u-center.** Plenty of different functions can be configured on the NEO-M8U: baud rates, update rates, geofencing, spoofing detection, external interrupts, SBAS/D-GPS, etc.

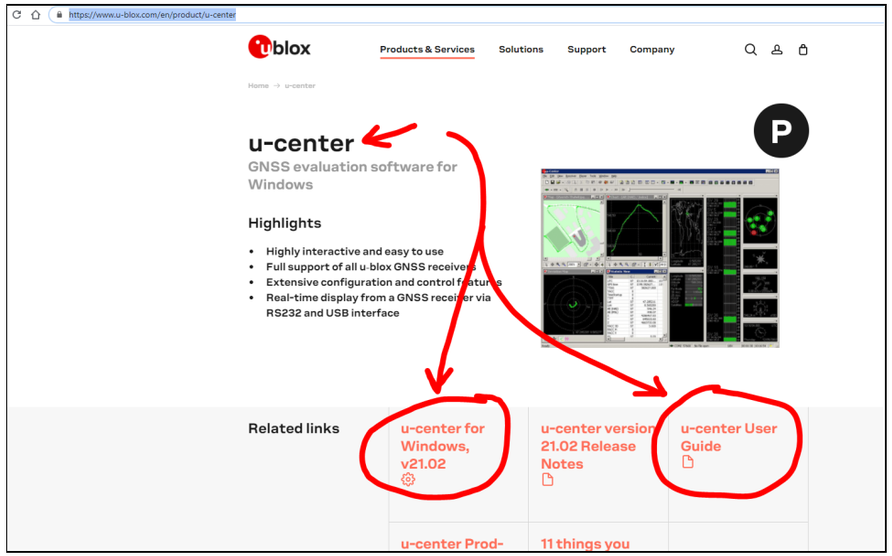
U-Center Overview

Before you can hook up your GPS to MatLab, you will need to pre-configure it over a USB cable from your laptop with a Ublox provided utility called **U-Center**. First go to:

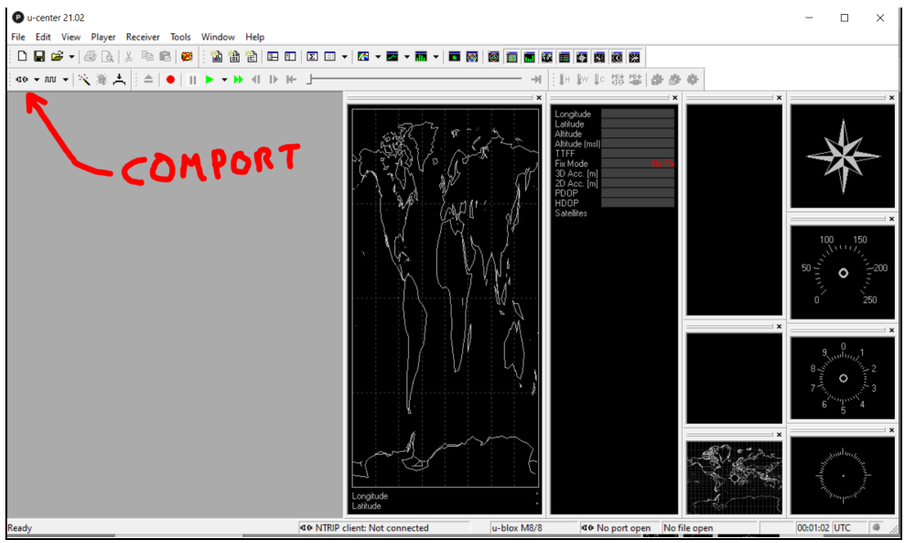
https://www.u-blox.com/en/product/u-center

And download the **U-Center for Windows** and the **U-Center User Guide**

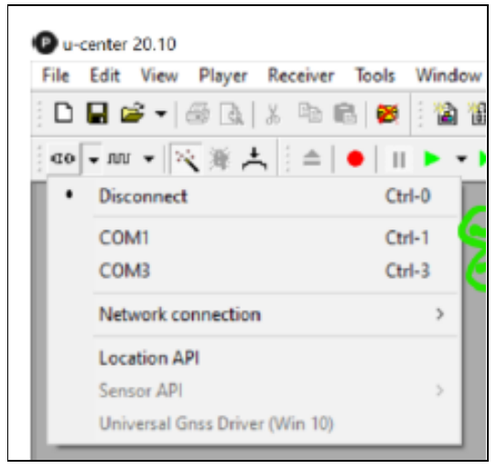
ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)



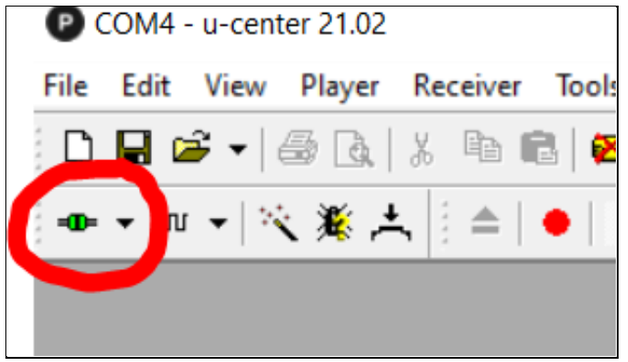
Open up **U-Center** application and it will look something like this:



Plug your Ublox GPS with antenna attached into your Laptop **(see last section of this document for safe ufl antenna plugging in help**), position antenna on flat surface pointing toward sky away from buildings and then click on the Com port button, to get:

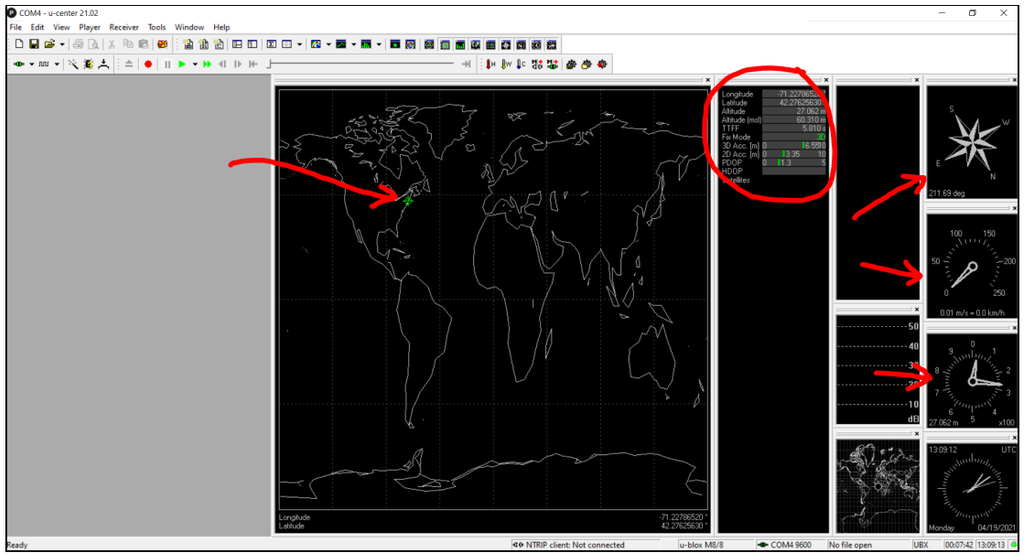


You will probably only have one device hooked up, so click on it and the com port button will go green:

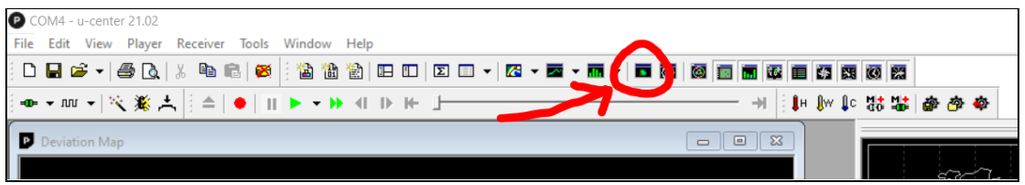


ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

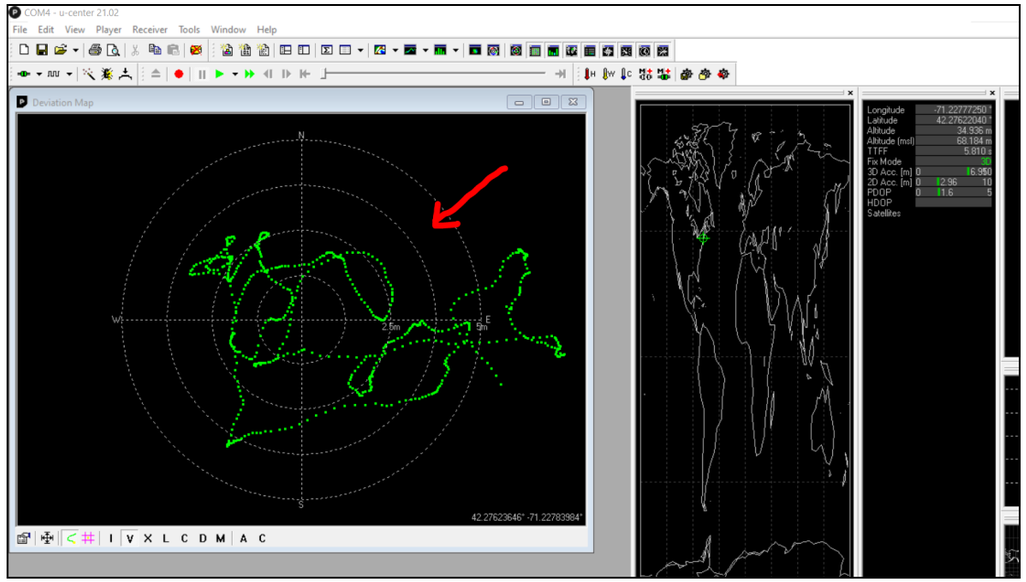
And a full HUD GPS display will come alive and you will see where on earth GPS is, Current Latitude and Longitude as well as a whole host of satellite and orientation data, it can all be a little overwhelming:



There are lots of buttons and tools on the top rail of the HUD display. If you roll the cursor over them, what they do is shown in text in the bottom left rail of the HUD display (bottom left corner). One of the most useful is the **Deviation Map** which shows the current GPS reading with time on a scale you can center and zoom:



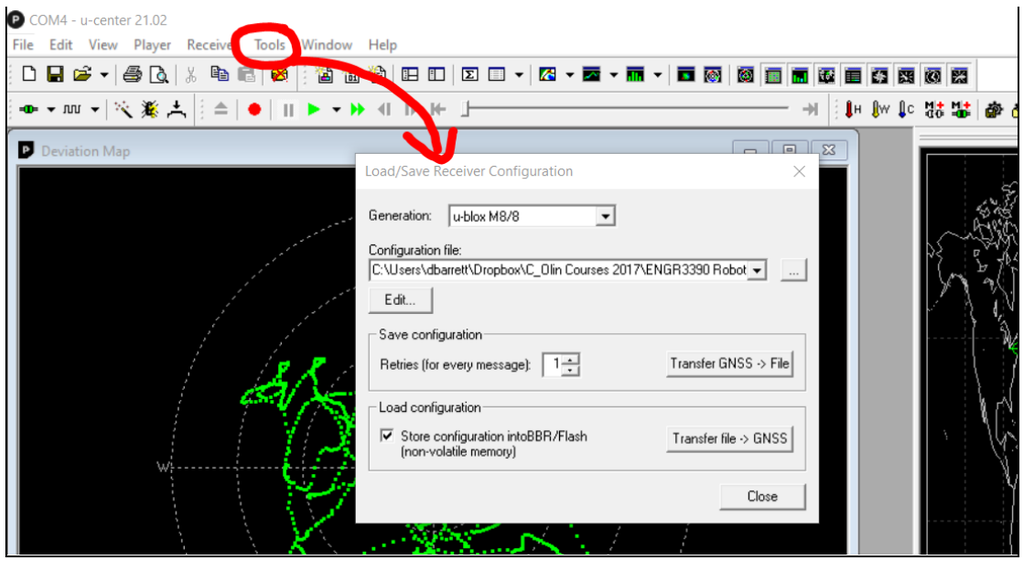
Click on it and then enlarge its display and take a look at your live GPS data coming in. Take your setup for a walk and go over near the Academic buildings and under the overhanging roofs. Take notes on what happens to your GPS fix. It will be quite useful later.



There is a tremendous amount of other tools and views you can play with here, please feel free to do so later. The important next step to do now is to tailor the output messages your Ublox GPS is sending to match those that the Matlab driver we give you is expecting to receive. The Ublox family of GPS modules can output a wide variety of clear ASCII text or binary information to match that required by whatever system they are attached to. In our case we want to output fast binary latitude longitude and a few other data points over the I2c bus back to your Raspberry Pi. To do that we need to configure the output data stream with a pre-made (supplied on Canvas) text file.

ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

Please go to **Tools > Receiver Configuration:**

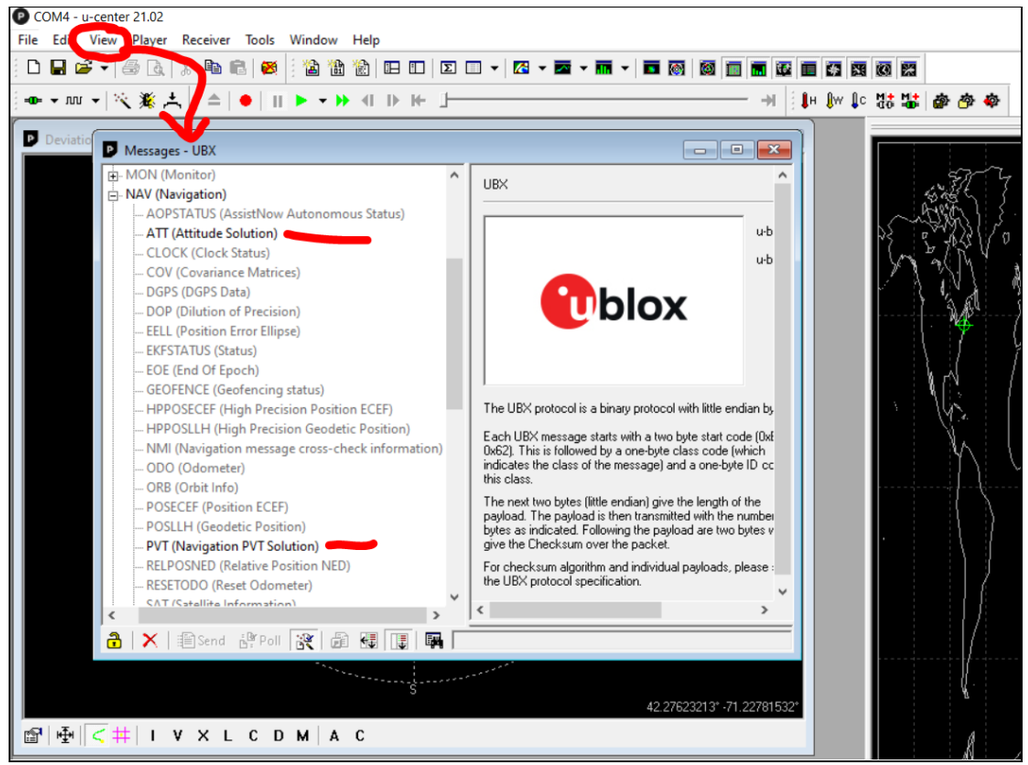


Then:

a. Select **u-blox M8/8** for the **Generation**
b. Select the file **basic_cfg.txt** file by clicking on the **[...]** button next to the dropdown under **Configuration File**, please select the **basic_cfg.txt** file you downloaded from Canvas final project code website.
c. Click the **Check Box** titled **Store Configuration into BBr/Flash (non-volatile memory)**
d. Click **Transfer file->GNSS** button

This should run a bit and download the configuration file to your GPS. You only need to do this once and it is sets for life (or until you change it again).

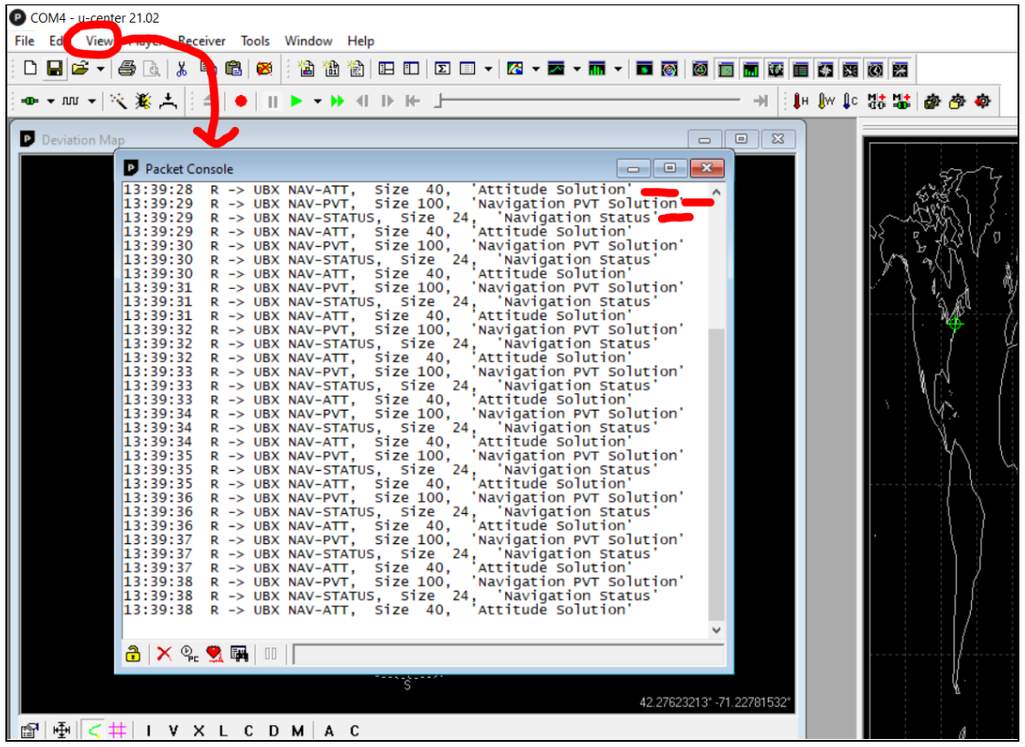
To verify that the device has been successfully configures, go to **View>Messages View**



Scroll down till you see the UBX set of messages (Unified Binary Transmission) and the **ATT, PVT, Status** message types should be colored in black (selected) while all others are gray (non-selected). Be careful not to click on anything else in this list, this is a powerful tool and poking around in it without reading through the full manual is dangerous. In some cases you may need to repeat the Tools>Receiver Configuration step if it didn't get correctly downloaded the first time. Configuring micro electronics can be finicky at times.

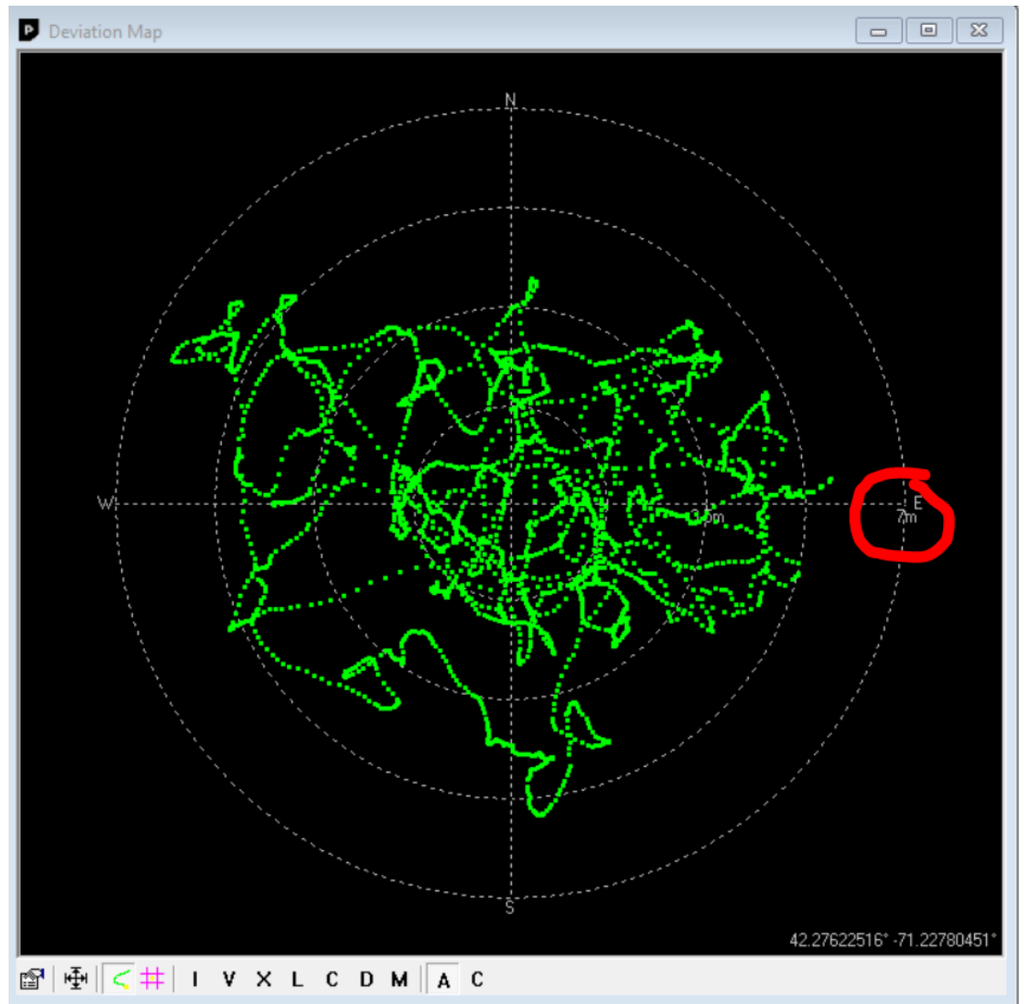
ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

As a final check you can click on **View > Packet Console** and see outgoing three message packets:



Your GPS is now configured and ready to go. You can unplug it from the USB connection to your laptop. Power down your Raspberry Pi (NEVER CONNECT ANYTHING TO PI OR QUIIC BUS HOT AND POWERED!) and connect your GPS to your Pi.

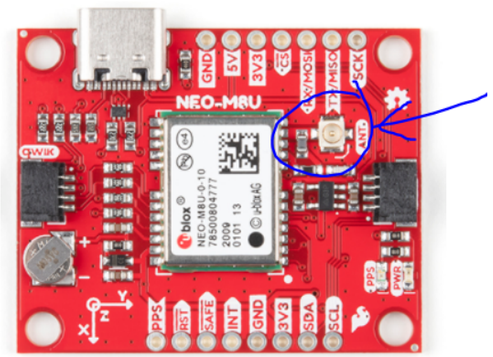
Final take away: On a good day, with good skylock, this GPS can find your Rover's position well within 7m (~20ft). This powerful "Where am I?" sensor can get you within range to use your camera to find visual targets:



ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

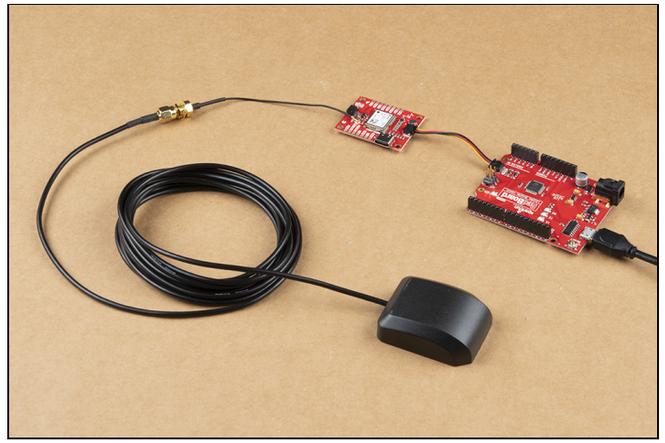
Attaching UFl GPS Antenna

The Sparkfun GPS-16329 GPS board comes with a tiny UFl external antenna connection.

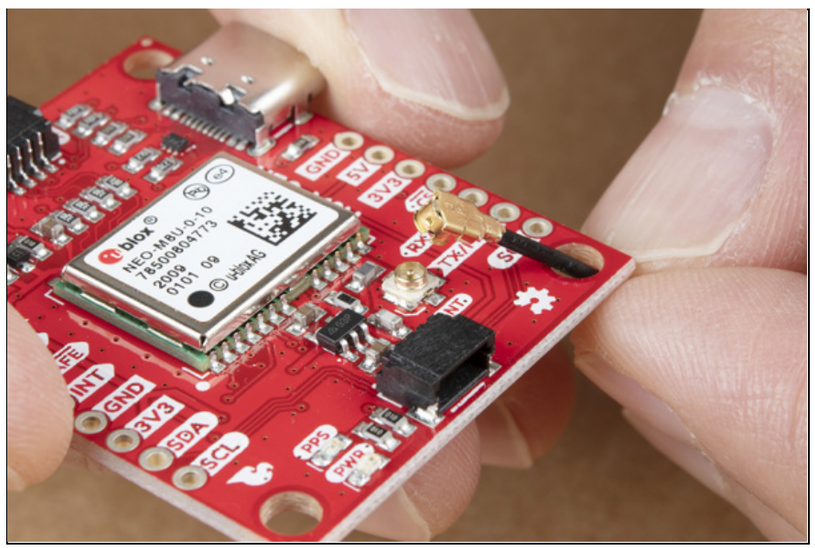


This micro coaxial connector was originally designed for cell-phones and to be connected just once, so its very fragile. In order to connect it to a standard GPS antenna with a standard SMA connector you need a small fragile connecting cable, namely the SparkFun

Example of system connected to an Arduino:



Example of strain relief on UFI cable:



To plug this all in without damaging anything, please read:

https://learn.sparkfun.com/tutorials/three-quick-tips-about-using-ufl

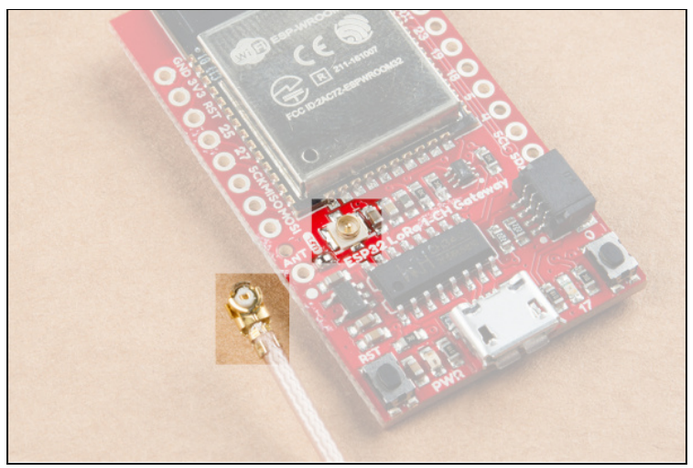
ENGR3390: Fundamentals of Robotics Tutorial (Setup UBLOX GPS)

Main part of the online version repeated here for clarity below:

Since the number of SparkFun products with wireless or radio-frequency (RF) applications continues to increase they thought it would be helpful to make a crash course in one of the most ubiquitous players: the U.FL connector.

U.FL and the host of compatible connectors (like I-PEX, IPX, or UMCC) are all designed to allow attachment of small coaxial cables that are preferable in wireless applications because of the shielding that the outer conductor provides. We use these connectors because they are small and inexpensive and because the smaller coaxial cables used with them are much easier to deal with. They are, however, a little bit harder to use and more fragile than their larger step-cousin (the SMA connector) so it doesn't hurt to get a quick primer on how to use them.

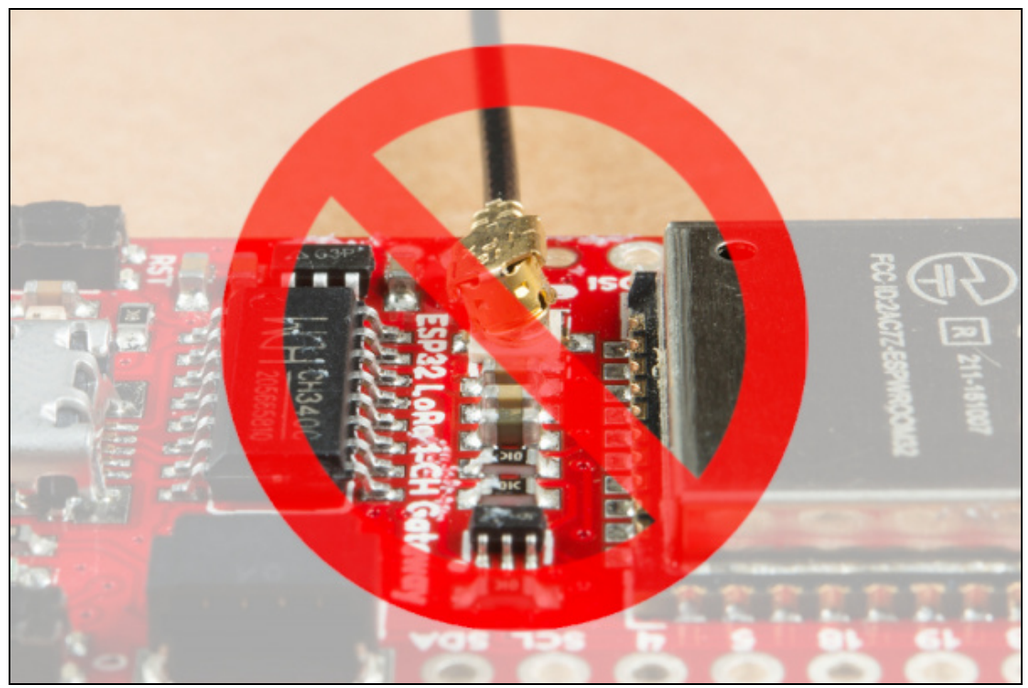
This photo shows off both the male and female ends of a U.FL connector. On the left and attached to the cable is the female side while the male side is soldered to the board. (By the way, a connector's gender is always determined by the gender of the electrical contacts rather than the gender of plastic shrouds or other extraneous materials).



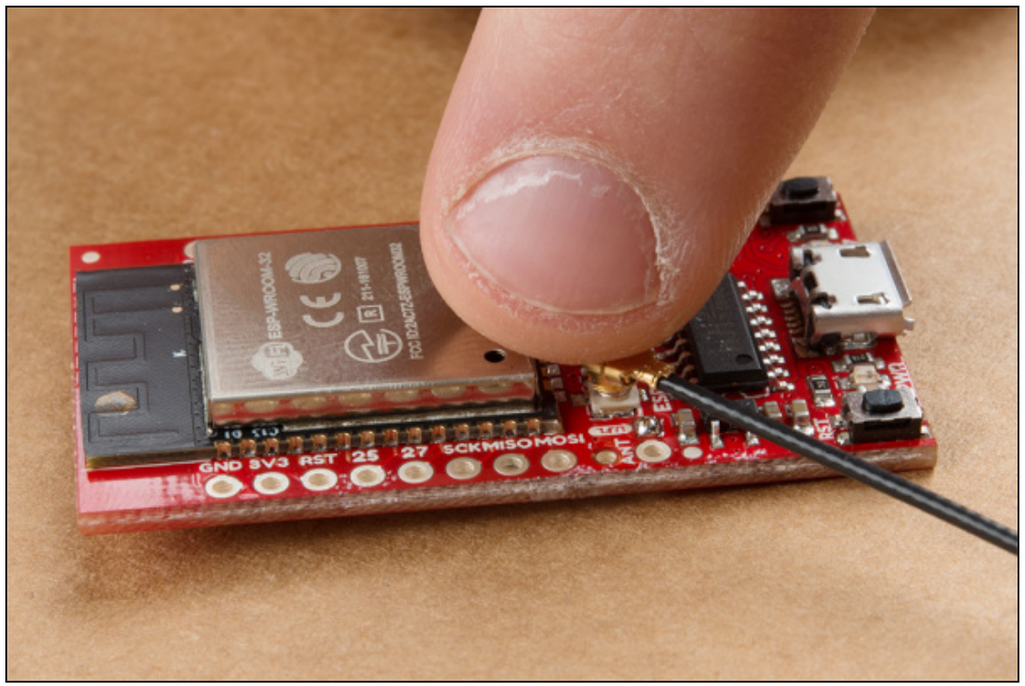
In the hands of an eager inventor, exposed U.FLs get plugged/unplugged much more often than those that live deep inside a consumer's cell phone. A Hirose datasheet shows that they only bothered to test repetitive use to 30 cycles, so follow these tips to keep your connectors working well for as long as possible!

Connect

Caring for your U.FL connector is all about keeping that shiny electrical contact pristine and contacting! Grime, oxidation, and mountain dew will increase the impedance to/from the antenna and will reduce range. Bent contacts won't let any of the electric pixies through at all and those "it works if I hold it just right" problems are definitely the worst to debug!

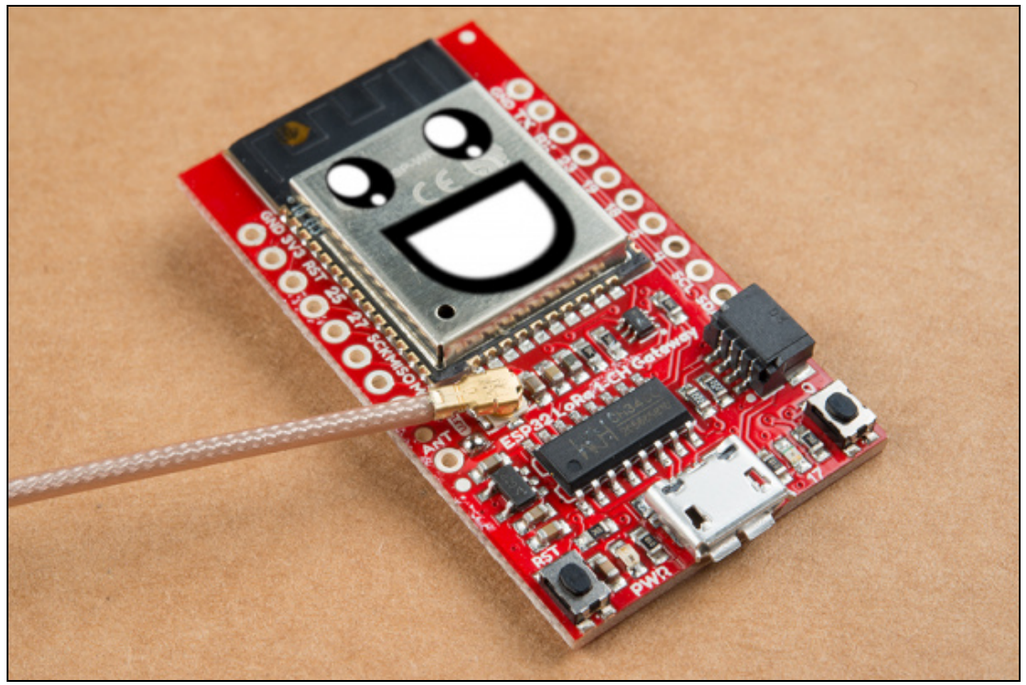


When connecting the parts of the U.FL, first make sure your finger is relatively clean. To avoid putting pressure on the connection, pinch the cable one or two cm from the end and try to align it so that the female connector rests evenly across the surface of the male connector. You can then lightly put your finger on top of the stacked connectors -- if it feels like your elementary school's old see-saw then try again, you want this to be stable. Also be sure that the two connectors are centered on one another left-to-right and up-to-down when you look at the connector from above the board.



If everything is good to go, use the very center of your finger to press firmly down - you'll feel a satisfying "lock sensation" as the datasheet puts it! (P.S. I had to move my finger off to the side so that you could see the connector in there, but you should really make an effort to push down right in the center of the connector)

The end result should look like this very happy ESP32 LoRa 1-Channel Gateway:



Protect

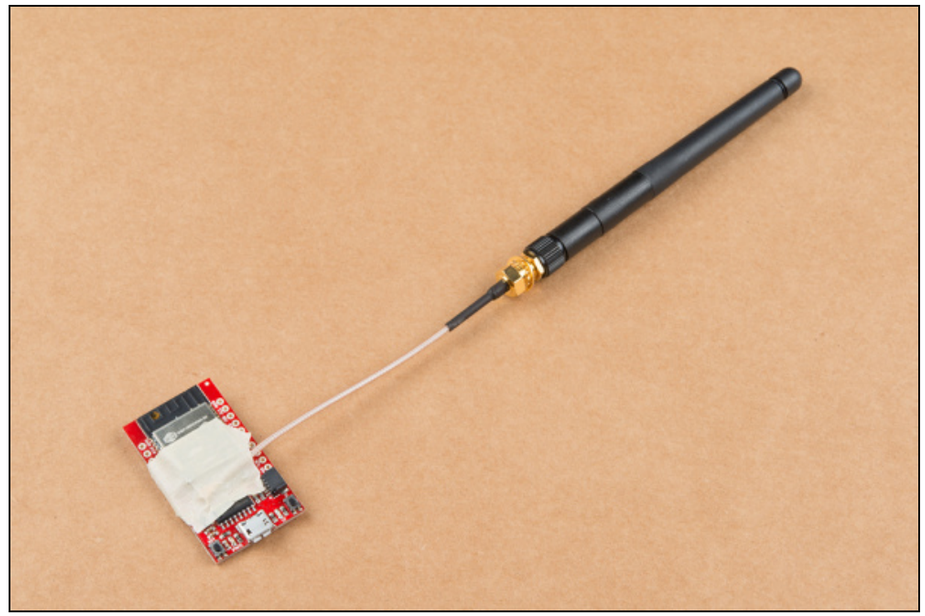
You can kind of think of the outside gripping portion of the female U.FL connector as being made out of dragonfly wings - very pretty to look at but also fragile. Putting unnecessary forces on them, for example a torque from pulling the cable in a funky way, could cause them to bend away from the male side of the connector and no

longer make a reliable connection. Keeping this from happening should also be a priority when moving a project or creating a final installation.

Preventing this weird torque is called strain relief and there are a couple ways to do it:

- **Tape down the cable**
- **Put the cable through a strain-relief hole on the board (if applicable)**
- **Glue down the cable**
- **Or think of something else as long as it relieves any forces that might be accidentally applied to the cable**

With any of the strain relief methods you should allow the cable to bend naturally to the securing location so as not to transmit any forces or torques to the connector. The ESP32 Gateway has a really convenient channel right below the ESP32 module that you can use to make a tape tent. This keeps the connection nice and secure without having to press down on the cable at all!



If there is a through hole nearby that is not being used, you can also thread the u.FL connector before connecting. Below is an example of threading the u.FL cable through a mounting hole before being attached to NEO-M9U GPS board. Feel free to add additional strain relief using hot glue or tape against the board to reduce the amount of forces on the small connector.

Now you know the basics of how to connect, protect, and disconnect a U.FL connector. Go rule the airwaves!