Overview

The ZED-F9P is a multi bandwidth GNSS receiver. It uses the L1 and L2 bands.

Objective

Create a gps data logger utilizing the multiple bands of the ZED-F9P.

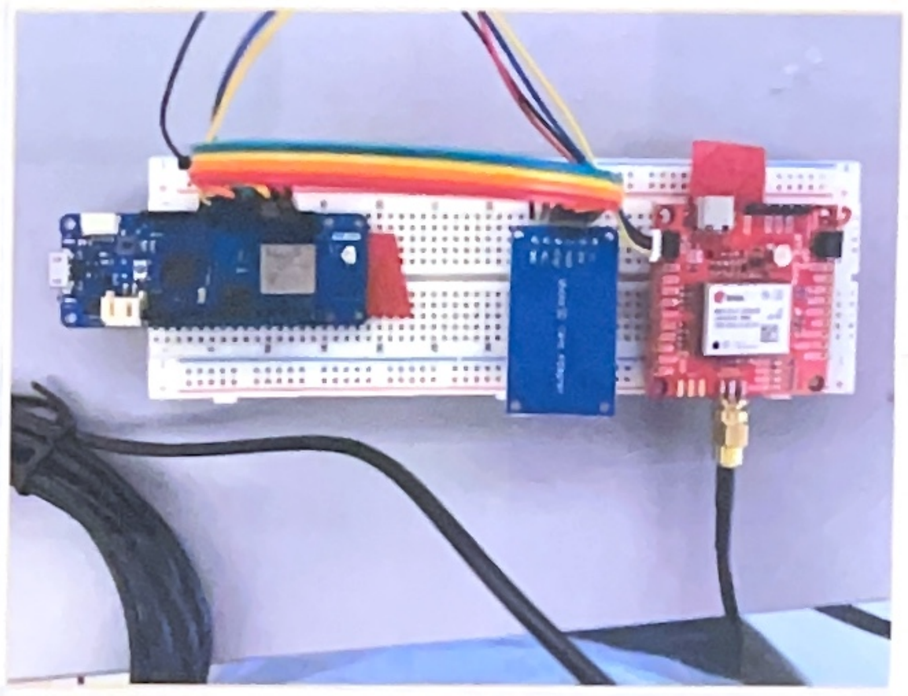

Figure 1

Circuit. The SD card reader is connected via SPI. The Sparkfun board is connected via I2C.

The Pinout for the Qwicc connector is:

GND - Top
3.3V
SDA
SCL - Bottom

If using the 3.3V pin to power the board, it needs to be a very clean source. I chose to power the board using the USB-C cable and an Anker battery pack.

Install V3 of the SparkFun u-blox GNss Library from the Library manager.

```ino
#include <Wire.h> //Needed for I2C to GNSS
#include <SPI.h>
#include <SD.h>

#include <SparkFun_u-blox_GNSS_v3.h> //http://librarymanager/All#SparkFun_u-blox_GNSS_v3
SFE_UBLOX_GNSS myGNSS;

long lastTime = 0; //Simple local timer. Limits amount if I2C traffic to u-blox module.

const int chipSelect = 7;
const int sdLED = 5;

void setup()
{
  Serial.begin(115200);
  Serial.println("SparkFun u-blox Example");

  pinMode(chipSelect, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  Wire.begin();

  if (!SD.begin(chipSelect))
  {
    Serial.println("Card failed, or not present");
    digitalWrite(LED_BUILTIN, HIGH);
    while(1);
  }
  else
  {
    digitalWrite(LED_BUILTIN, LOW);
  }

  File outputfile = SD.open("data.txt", FILE_WRITE);
  outputfile.println("Latitude, Longitude, SIV");
  outputfile.close();

  //myGNSS.enableDebugging(); // Uncomment this line to enable helpful debug messages on Serial

  if (myGNSS.begin() == false) //Connect to the u-blox module using Wire port
  {
    Serial.println(F("u-blox GNSS not detected at default I2C address. Please check wiring. Freezing."));
    digitalWrite(LED_BUILTIN, HIGH);
    while (1);
  }

  myGNSS.setI2COutput(COM_TYPE_UBX); //Set the I2C port to output UBX only (turn off NMEA noise)
  myGNSS.saveConfigSelective(VAL_CFG_SUBSEC_IOPORT); //Save (only) the communications port settings to flash and BBR
}
```

```
51 void loop()
52 {
53   //Query module only every second. Doing it more often will just cause I2C traffic.
54   //The module only responds when a new position is available
55   if (millis() - lastTime > 1000)
56   {
57     lastTime = millis(); //Update the timer
58
59     long latitude = myGNSS.getLatitude();
60     long longitude = myGNSS.getLongitude();
61     byte SIV = myGNSS.getSIV();
62
63     File outputfile = SD.open("data.txt", FILE_WRITE);
64     if (outputfile)
65     {
66       outputfile.print(latitude);
67       outputfile.print(", ");
68       outputfile.print(longitude);
69       outputfile.print(", ");
70       outputfile.println(SIV);
71     }
72     outputfile.close();
73   }
74 }
```

Figures 2 + 3

Data from the SparkFun board plotted on Google Maps.