

ZED-F9P GNSS Firmware Documentation

Brian Zhou

Purpose

Our team would like to transition our sensorboard away from Arduino's HAL towards STM32's HAL. The biggest roadblock for this transition is recreating complicated peripherals like GNSS, which is currently a black box. This document records our research into rewriting the ZED-F9P's firmware and will allow our team to understand the plausibility of migrating.

Links

Official Documentation:

- [UBLOX Official Documentation & Resources Page](#)
- [ZED-F9P USER GUIDE](#)
 - RTK Setup
- [ZED-F9P Integration Manual](#)
- [Application Considerations](#)
 - Suggestions for RTK setup
- [ZED-F9P Interface Manual](#)
 - **Communication Protocols**

Arduino Libraries for ZED-F9P

- [SparkFun Ublox Arduino Library \(OLD\)](#)
- [SparkFun u-blox GNSS Arduino Library](#)

Original 2023 Arduino Firmware

- <https://github.com/ubcaerodesign/sensorboard-rev2-firmware>
 - [main.cpp](#)

Preliminary Research

Skimming through the UBlox's documentation gave the impression that the [ZED-F9P Interface Manual](#) contains the bulk of information needed to communicate with the GPS module. However, the information might be incomplete as [comments on the Sparkfun Arduino Library](#) suggest partial reverse engineering (this is for RTCM, irrelevant to UBX).

ZED-F9P Interface Manual Summary

There are three supported protocols: NMEA, UBX, and RTCM, which can be used interchangeably to communicate with the F9P?

The “current configuration” for the receiver can be changed by sending UBX-CFG-VALSET of any IO port except UART2. The receiver always uses the “current configuration”. Changes to the current configuration will be reset on power off (p17)

1.4 Message naming

A full message has protocol name (eg “UBX”) class name (“NAV”) and the message name (“PVT”) separated by hyphens (p17).

1.6 Message types

UART1

The ZED-F9P includes two UART serial ports. UART1 can be used as a host interface for configuration, monitoring and control. UART2 is available as an optional interface and cannot be used as a single host interface (Integration manual, p48)

[integration manual](#)

“The default baud rate is 38400 baud. To prevent buffering problems it is recommended not to run at a lower baud rate than the default.” (integration, 48)

The baud rate for safeboot mode is 9600 baud. See more information about safeboot mode in the SAFEBOOT_N section.

UBX Protocol

Every frame starts with 0xB562 in as first two bytes

3.3.5 UBX data types (p54, Interface Description)

3.5.2 UBX polling mechanism (to poll message, send data but without a payload)

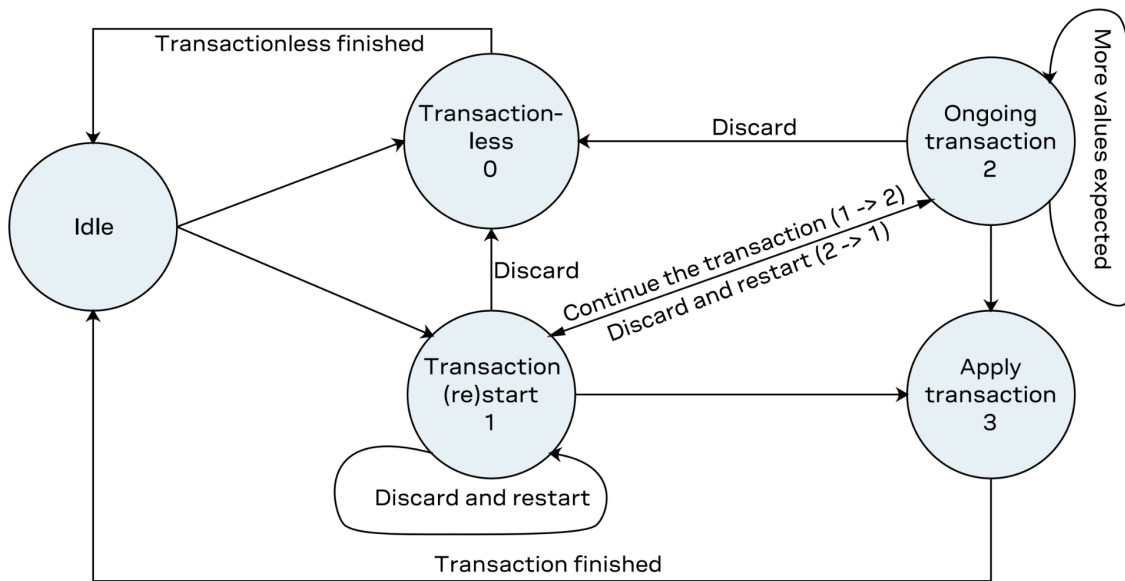
Firmware Implementation Plan

The easiest way to implement the GNSS firmware is to port Sparkfun’s existing [u-blox_GNSS_Arduino_Library](#) over to the STM32C HAL. The library is well documented with comments which should make writing our library easy.

Writing the firmware by referencing only the official documentation is difficult (maybe impossible) due to the difficult to read and potentially incomplete documentation.

6.6 Configuration transactions

State machine for configuration transactions



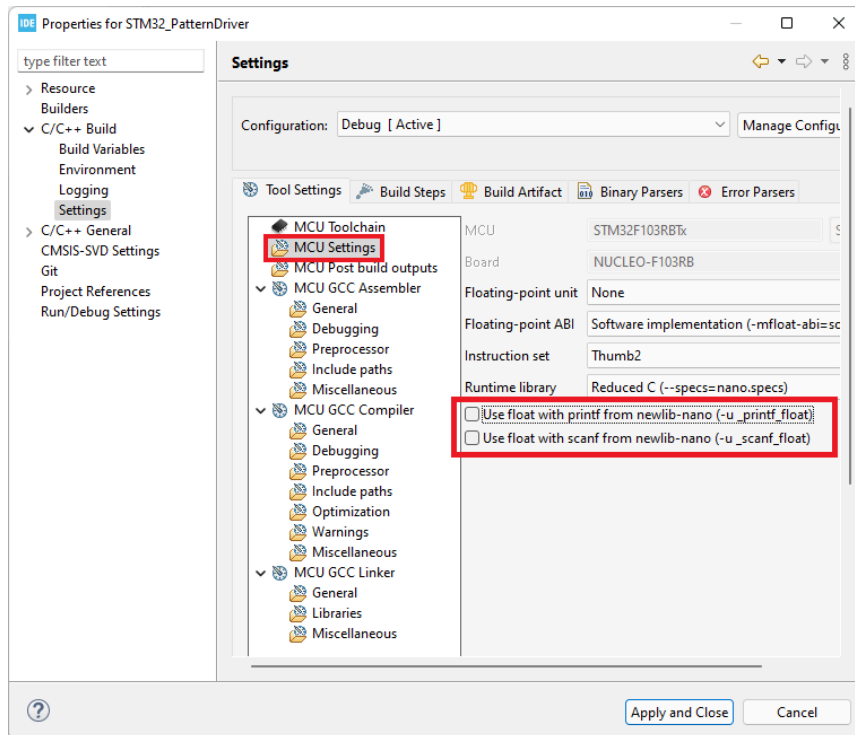
“Note that a transaction can only come from a single source, a UBX-CFG-VALSET message or a UBXCFG-VALDEL message.” (p. 226, Interface Description)

Troubleshooting

Used cubeide ver. 1.16.0

No debug print, we can either print to SWV or Uart2

Printing floats does not work by default, had to do this:



<https://www.eevblog.com/forum/microcontrollers/stm32cubeprintf-does-not-work-with-float/>

Coding Style

Inspired parts by 1.11.1. Coding Standard [Intel UEFI Boot Loader User Guide](#)

CFG-MSGOUT-UBX_NAV_PVT_UART1 0x20910007 U1 - - Output rate of the UBX-NAV-PVT message on port UART1

[Resettings GPS Configuration](#)

COM11 @ 38400 - u-center 24.05

