Riktningsindikering med 2 GNSS mottagare och RTK

Jens Tunare - SM6AFV ver. 0,52

Sammanfattning

Beskrivning av en portabelt Arduinobaserad riktningsindikator med hjälp av två GNSS-mottagare[1]. Indikatorn är tänkt att användas tillsammans med en portabel mikrovågsstation för att rikta antennen med en mätosäkerhet på mindre än 1 grad. Enheten har också sensorer och indikering för temperatur/Fukt/Tryck/Spänning/Tid samt lokatorberäkning med 10 tecken.

Fig. 1. 60 cm antenn med azimuth Indikator

Fig. 2. Indikatorfönster

Bakgrund

Ett stort problem vid portabelkörning på de högre mikrovågsbanden är att peka antennerna i rätt riktning med stor noggrannhet. Öppningsvinkeln för antennerna kan vara mindre än 1 grad och man behöver då motsvarande noggrannhet för riktningsinställningen.
Det vanligaste sättet är att i förväg leta upp lämpliga synbara referenspunkter i närheten av önskad riktning, som t. ex. kyrktorn, fyrar eller andra tydliga landmärken. Med ett internetbaserat kartverktyg som K7FRY[2] kan man i planeringsfasen ta fram riktningen och avstånd till referenspunkten.
Med en större vridbar analog/digital gradskiva monterad på antennstativet kan man sedan ställa in mot en referenspunkt så att man får en absolut kalibrerad riktning över 360 grader.

Med denna metod kan man dock många gånger råka ut för överraskningar när man är på plats. Dis, dimma, flera objekt som liknar varandra skymda objekt kan medföra att man riktar fel.

Oberoende av vilket sätt man använder för att bestämma riktningen så måste radiosignalriktningen vara kalibrerad med den fysiska riktningen. Man kan använda ett kikarsikte och en testsändare på ca 100 m avstånd som kalibreringshjälpmedel. Ett alternativt sätt är att kalibrera den fysiska riktningen med en solbrusmätning dvs maximera solbruset vid mottagning och använda en beräknad azimuth/elevering-riktning för mättillfället.

Det är dock inte alltid möjligt att använda referenspunkter när man är ute i fält. Har man en riktning över havet eller en större sjö så kan det bli problem att hitta någon synbar referenspunkt.

Kan man vrida sitt stativ 360 grader så kan man eventuellt använda en referenspunkt som ligger i motsatt riktning över land. Det kan även inträffa att man behöver flytta position och då har man kanske inga förberedda referenspunkter till hands.

Vanliga kompasser och GPS-appar för smartphones är alldeles för opålitliga i denna tillämpning.
Det finns dock professionella Differentiella/RTK GPS system som skulle kunna användas men kostnaderna för ett sådant system har hittills varit för högt.

GPS-baserad riktningsindikering

Utvecklingen av GNSS/RTK/GPS-enheter för drönare har medfört att kostnaderna för ett noggrant GNSS-GPS-system har kommit ner till en rimlig nivå och även litet fysiskt format [3].
GNSS betyder Global Navigation Satellit System [4]**.**
Det i referens [3] presenterade systemet använder 2 utvecklingssystem, C94-M8P från U-Blox.
Det ena GPS-enheten ”Base” placeras ca 5 m från den andra enheten ”Rover” som är placerad på antennstativet. Korrektionsdata från basenheten skickas till ”Rover”-enheten. Portabelstationens antenn riktas mot Basenhetens GPS-antenn och den absoluta riktningen till Basenheten ställs in på en analog eller digital gradskiva.

Mätosäkerheten för det ovan beskrivna systemet är beroende av avståndet mellan GPS-antennerna, fasegenskaperna hos antennerna samt att antennerna skall vara placerade utomhus och ha fri himmel över 20 graders elevering för att kunna ta emot signaler från så många satelliter som är möjligt. Med de senaste GNSS-kretsarna kan man ta emot signaler på upptill 60 kanaler från de flesta system såsom GPS, GLONASS, GALILEO, COMPASS och BEIDOU . Se fig. 4.

Efter att ha läst en beskrivning i DUBUS [2] började jag att söka efter RTK-GNSS operation (**R**eal **T**ime **K**inetics = relativ bärvågsmätning i realtid) på nätet och fick en mängd träffar för utrustningar och filmer på youTube.

Det fanns nya GPS-enheter med bättre data än de enheter som använts i [3].
Från U-Blox finns t ex ZED-F9P [3]. Detta chipset används hos flera tillverkare av tillämpningskort ”breakout” från SparkFun [4], U-Blox [5], Ardusimple [6] och Drotek[6] med flera. Det var dock inte lätt att få tag i moduler med ZED-F9P kretsarna eftersom de inte funnits tillgängliga så länge. Det gäller även aktiva precisionsantenner lämpliga för denna tillämpning.

Fig. 3. Första testuppkoppling

Testuppkoppling

Jag beslutade mig för att testa konceptet och fastnade för SparkFuns moduler som fanns tillgängliga från ELFA. SparkFun har också tagit fram ett Arduinobibliotek med tillämpningsexempel för deras GPS-RTK2-ZED-F9P modul. Detta skulle förmodligen underlätta programutvecklingen.
De billigaste lämpliga antenner som var tillgängliga var ANN-MB från U-Blox. Jag beställde två ANN-MB från Drotek i Frankrike.
Efter en veckas väntan hade jag två GPS-moduler och antenner i brevlådan som jag kunde koppla upp för en första test.
SparkFun-modulerna behöver 5V och drar ca 35 mA. Antennerna är aktiva och ström försörjs via antennkabeln. Internt används 3,3V. Det gäller även övriga portar, UART1/UART2 och I2C
Vid konfigurering av modulerna via USB så strömsätts modulerna även via USB-kontakten (USB-C)

U-blox har ett avancerat test- och konfigureringsprogram, Det är ett Windowsprogram (u-center) som kan laddas ner från U-Blox hemsida [7]. U-center används även för uppdatering av den interna mjukvaran i modulerna.
Jag har använt u-center tidigare för att konfigurera andra GPS-moduler så startsträckan för att konfigurera ZED-F9P-kretsarna var inte så lång. En konfigurering kan sparas i modulens flash-minne eller på fil. Konfigureringen kan också skickas till modulerna vid programstart men när man har två moduler så måste dessa ha olika I2C adresser eller matas via separata I2C portar från en I2C-ansluten ansluten processor.

I min tillämpning så sitter GNSS-mottagarna från SparkFun stackade ovanpå varandra och kommunicerar från ”Base” till ”Rover” sker via UART1 genom några byglingstrådar mellan de två modulerna.
Resulterande data skickas via I2C till en Teensy 4.0 processor försedd med en 2.8” TFT Touch Display.

U-centerprogrammet som jag använder för konfigurering av U-blox-kretsarna i både ”rover” och ”base” enheterna kan även användas för resultatpresentation under utvecklingsfasen och kopplas då till roverenheten via USB-porten.

SparkFuns Arduinobibliotek har funktioner för att hämta positionsdata och absolut riktning (heading) till den andra antennen samt avståndet mellan de två antennerna. Skillnaden mellan det presenterade avståndet och det fysiskt uppmätta avståndet är ett
mått på mätosäkerheten i positionsangivelsen. Även mätosäkerheten i riktningsbestämningen beräknas internt i rovermodulen.

Vid mina första praktiska försök satte jag antennerna med 2,0 m avstånd på en vridbar stång på ett

stativ. Enligt rekommendationer från U-blox satte jag ett jordplan med 110 mm diameter under varje antenn. Antennerna levererades med 5 m antennkabel som medgav en något så när fri placering med GPS-enheterna inomhus. Stången med antennerna riktades mot en uppmätt referenspunkt på 14 km avstånd och riktningen noterades och jämfördes med den beräknade riktningen. Vid försöken monterades GPS-enheterna i en låda med antennkontakter i panelen för att skona de ömtåliga U.FL kontakterna.

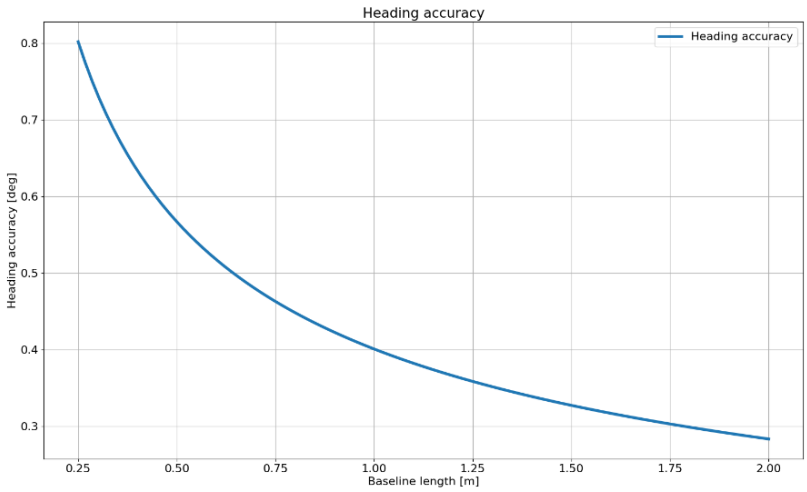


Figure 1: ZED-F9P moving base RTK heading accuracy versus baseline length

Fig. 4. Mätosäkerheten för ZED-F9P enligt datablad

Resultatet med antennerna på 2 m avstånd visade på en avvikelse på < 1 grad. Vilket var en god början.

Fig. 5. Inkopplingspanel

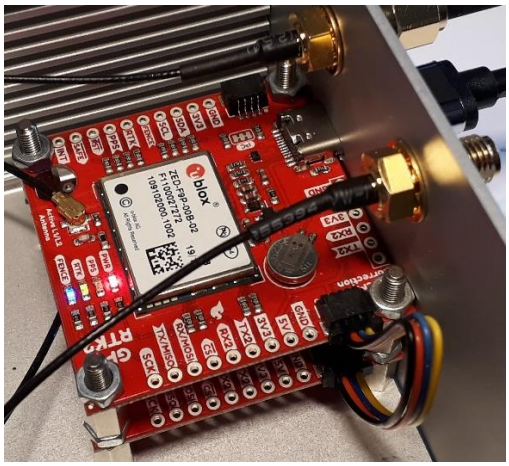


Fig. 6. Stackade SparkFun GPS-RTK2 moduler

Inspirerad av resultatet minskade jag ner avståndet mellan antennerna till 1 m och gjorde om försöket och fick motsvarande resultat. Metoden att flukta längs en stång mot en referenspunkt på 14 km avstånd är dock inte så tillförlitlig så jag väntar på varmare väder för flera tester där jag använder ett kikarsikte för att sikta mot referenspunkten.

Arduinobibliotekets funktioner för ”Heading” och ”Length” visade en mätosäkerhet på 0,6 gr för Azimuth och 10 mm i position!

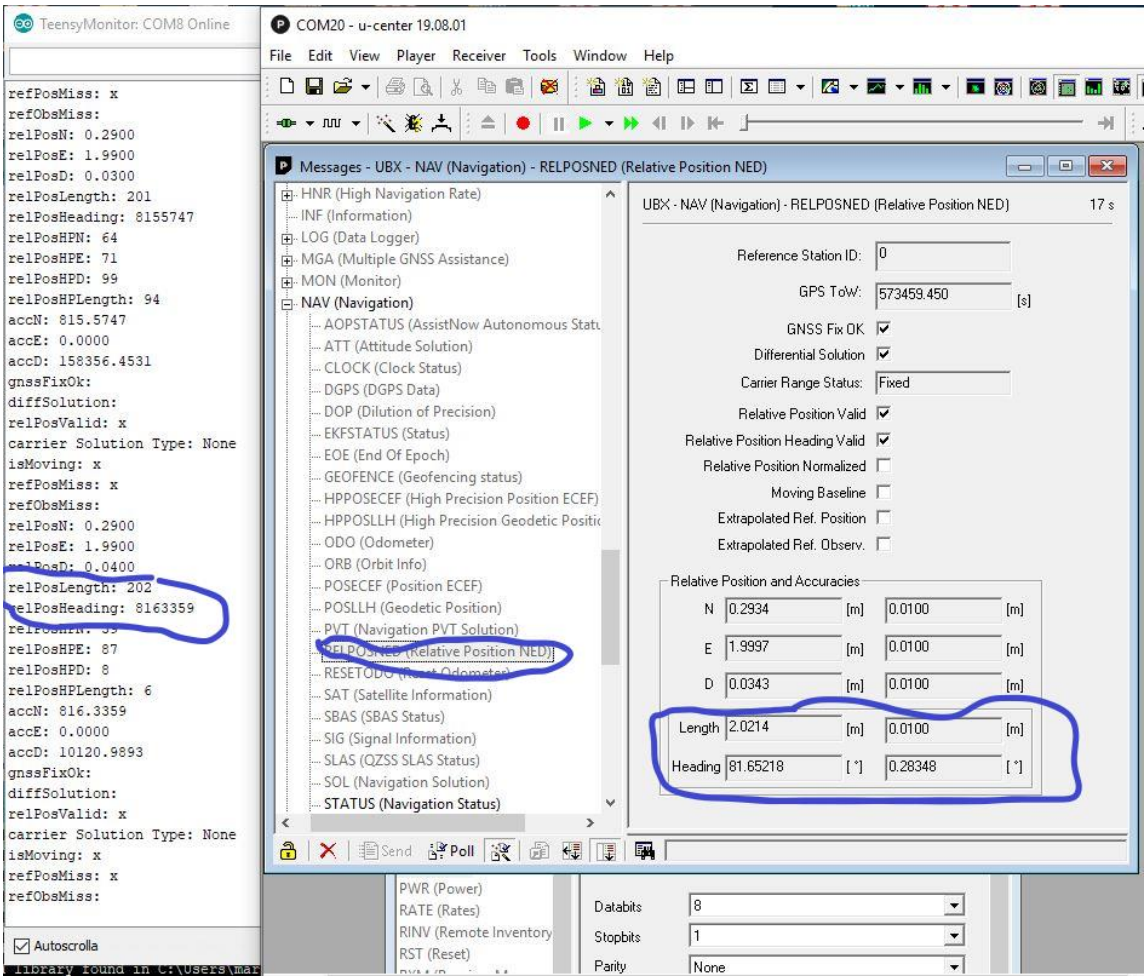


Fig. 7. u-center resultat för längd och riktning (heading)

Efter de lovande första försöken fortsatte jag med att bygga ihop en mera fältmässig utrustning med ett eget displayfönster. Jag fastnade för en Hammond box med ytterformat 160 x 103 x 55 mm.

Portabel Enhet

Tanken är att denna box skall kunna sättas hakas fast på antennstativet och flera funktioner.

1. Distributionsenhet för 12V (PowerPole kontakter)
2. Riktningsindikering (Azimuth)
3. Spänningsövervakning
4. 10 MHz referens för frekvenslåsning (med 1 Hz pulsen från GPS)
5. GPS-tid
6. Lokatorberäkning (baserat på GPS-data)

7. Väderstation (Temperatur/Fukt/Lufttryck/Daggpunkt)

Framtida funktioner

8. Justering av filterfunktioner
9. Loggning av väderdata på minneskort/USB-minne
10. Förbättrad felhantering
11. Touch funktioner
12. Beräkning av sträckdämpning
13. ?.....

Datahantering och beräkningar utförs med en kraftfull och kompakt Teensy-processor som jag använt tidigare för andra applikationer. Detsamma gäller den 2,8” Touch display som jag också använt tidigare och hade liggande.
Teensy-processorerna programmeras i Arduino-miljö via ett Teensyduinotillägg och är ett kompakt, kraftfullt alternativ till Arduino Nano. En Teensy 3.2 med färre pinnar än 3.6 skulle mycket väl räcka till i denna tillämpning. Använder man inte de extra pinnarna i 3.6 så är det bara att byta processor till 3.2 och kompilera om programmet för denna processor. Det har nu kommit en ännu kraftfullare processor, Teensy 4.0 som har samma formfaktor som Teensy 3.2 och det är den som jag använder i denna version av riktningsindikatorn. Teensy 4.0 har 3,3V logik med 600 MHz klockfrekvens.
Kommunikationen mellan GPS-enheterna Base till Rover sker automatiskt efter korrekt Base /Rover konfigurering. Riktningsdata skickas till processorn via I2C kommunikation.
Konfigurering för respektive GPS-enhet kan sparas på fil som kan laddas upp med u-centerprogrammet via USB gränssnittet.

Arduinoprogrammet har tre olika skärmvyer med teckenbaserad information.

1. Riktning (Azimuth, Tid, Lokator, Batterispänning
2. Väder Temp, Fukt, Lufttryck, Daggpunkt, Tid, Lokator
3. GPS Data

Programmet är förberett för följande givare

- SHT75 – Sensirion Temp, Fukt (End of life, 2018*)
- BME280 – Bosch Temp, Fukt, Lufttryck
- DS18B20 – 1-wire temperatursensor
- Teensy – Batterispänning (2 x 12 bitars intern A/D konverter)

* Ersättare Sensirion STS31-DIS (I2C bus)

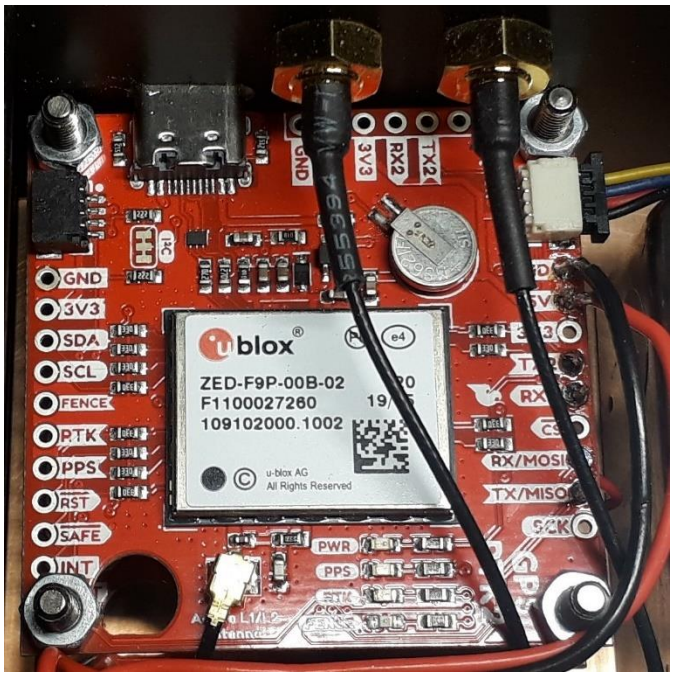
Primära komponenter

Hammond 1455N1601BK	RS Online, 7732993
2 x SparkFun GPS-15136	ELFA, 301-52-826
2 x SMA – U.FL adapterkabel	Electrokit, 41013985
SparkFun Qwiic kabel Kit	Lawicel, KIT-15081
JST SH 4-pin vertical (Qwiic)	Lawicel, ADA-4328
2 x U-blox ANN-MB-00-00 (SMA)	Drotek
Teensy 4.0	ELFA, 301-58-513

2,8” Touch Display, TJCTM24028-SPI	www.pjrc.com
Arduino Nano	
10 MHz OCXO, Oscilloquartz 8663-XS, sinus, 12V	Ebay
DC/DC Converter, Traco Power, THN 20-2412WI	ELFA, 169-11-207
SHT31 Adafruit	ELFA, 300-91-203

USB

USB ANT BASE ANT ROVER



ROVER I2C till Teensy I2C

GND - GND
+5V - +5V

UART1 TX – UART1 RX
UART1 RX – UART1 TX

Fig. 8. 2 stackade SparkFun RTK-GPS2 moduler

Uppbyggnad

Hammond boxen har plats för att skjuta in ett 160 x 100 x 1,6 mm kretskort. De flesta delarna monteras på detta kort med distanser. GPS-enheterna monteras i en stack mot ena gavelpanelen som har uttag för USB-kontakter och antenner. Display med processorkort monteras även på detta kort med distanser så att displayytan ligger i kant med undersidan av locket. Kortets sitter med jordplanssidan uppåt där även 10 MHz OCXO plus DC/DC konverter monteras. På undersidan av Europakortet finns låsningskretsarna med sin egen Arduino Nanoprocessor.

Detta konstruktionssätt innebär en avskärmning uppåt för DC/DC konverter, OCXO och frekvenslåsning. Hela konstruktionen kan dras ut ur Hammondboxen. Det påskjutbara locket till boxen har en rektangulär öppning för en 2,8” display.

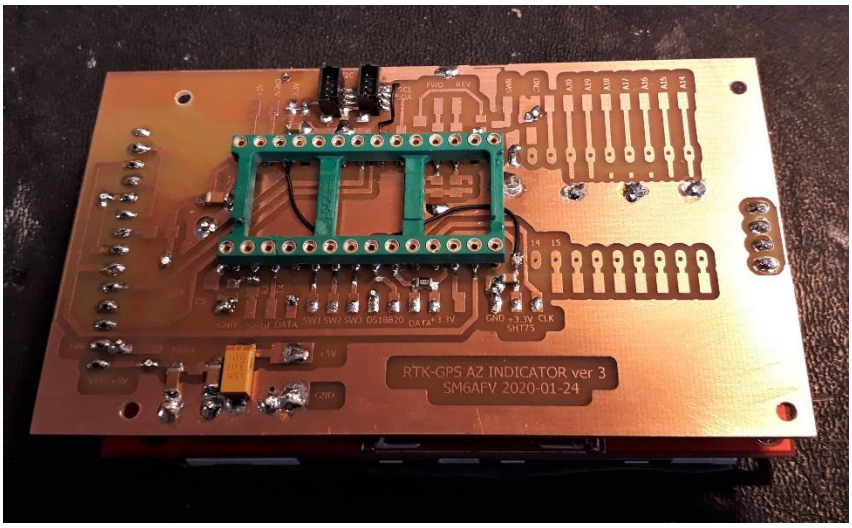


Fig. 9. Kretskort för Teensy med TFT Display

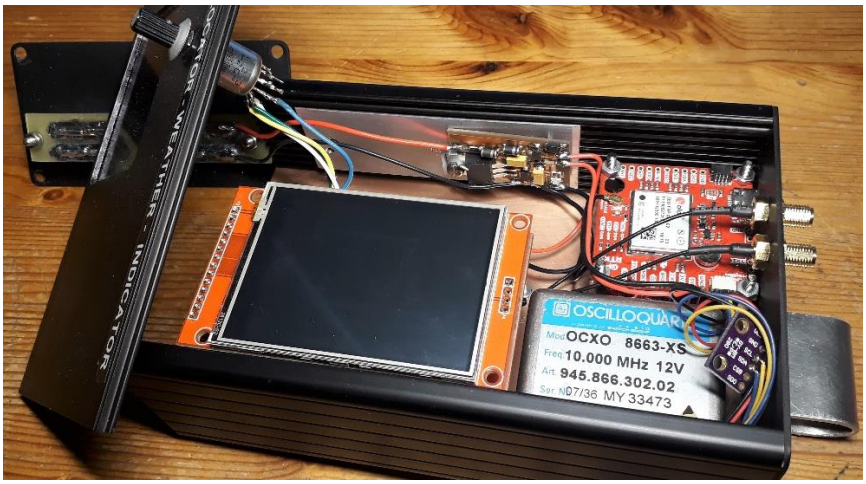


Fig. 10. Insida


Fig. 11. Kretskort med Teensy och TFT

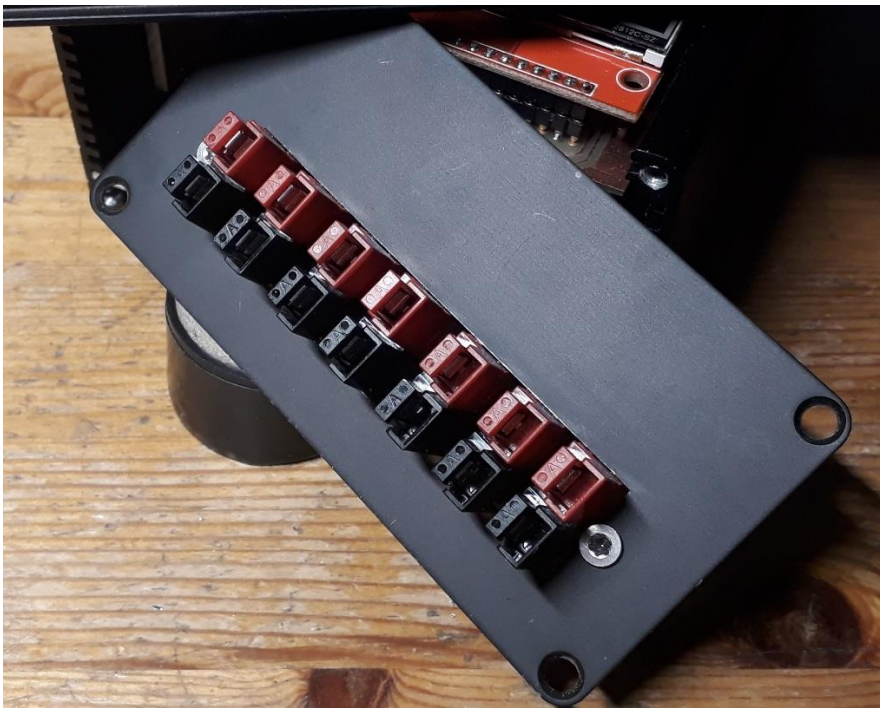


Fig. 12. 12V uttag med Power Pole kontakter

10 MHz OCXO har en separat 12V matning från en 9-18/12V DC/DC konverter för att arbeta stabilt även om batterispänningen sjunker under 12V. DC/DC-konvertern ger också en isolering och från 5 V matningen. GPS-enheter, processor och display får 5V från en LM1085-5.0 regulator.

Antennerna med jordplan monteras på 15 x 15 x 1 mm fyrkantsprofil av aluminium. Den horisontella antennbommen är monterad mot den vertikala profilen (25 x 25 x 2 mm) med en gaffelkoppling som medger en justering av horisontalläget och kalibrering av Azimuth för respektive parabol. Den vertikala fyrkantsprofilen skruvas fast mot parabolens bakstruktur med två M5 vingskruvar.

”Base” och ”Rover” Konfigurering

GPS enheterna kan konfigureras på flera sätt med u-centerprogrammet (windows 10) via USB-porten. Först måste man bestämma vilka portar och porthastigheter som skall användas.

I mitt fall skickar jag data från ”Base” till Rover via UART1/460800 bps på båda enheter. Data mellan ”Rover” och Teensyprocessor går via I2C. Identifiera vilken COM-port som GPS-enheten tar i anspråk i Enhetshanteraren/Portar (COM & LPT) efter att USB-kabeln kopplats in till datorn.

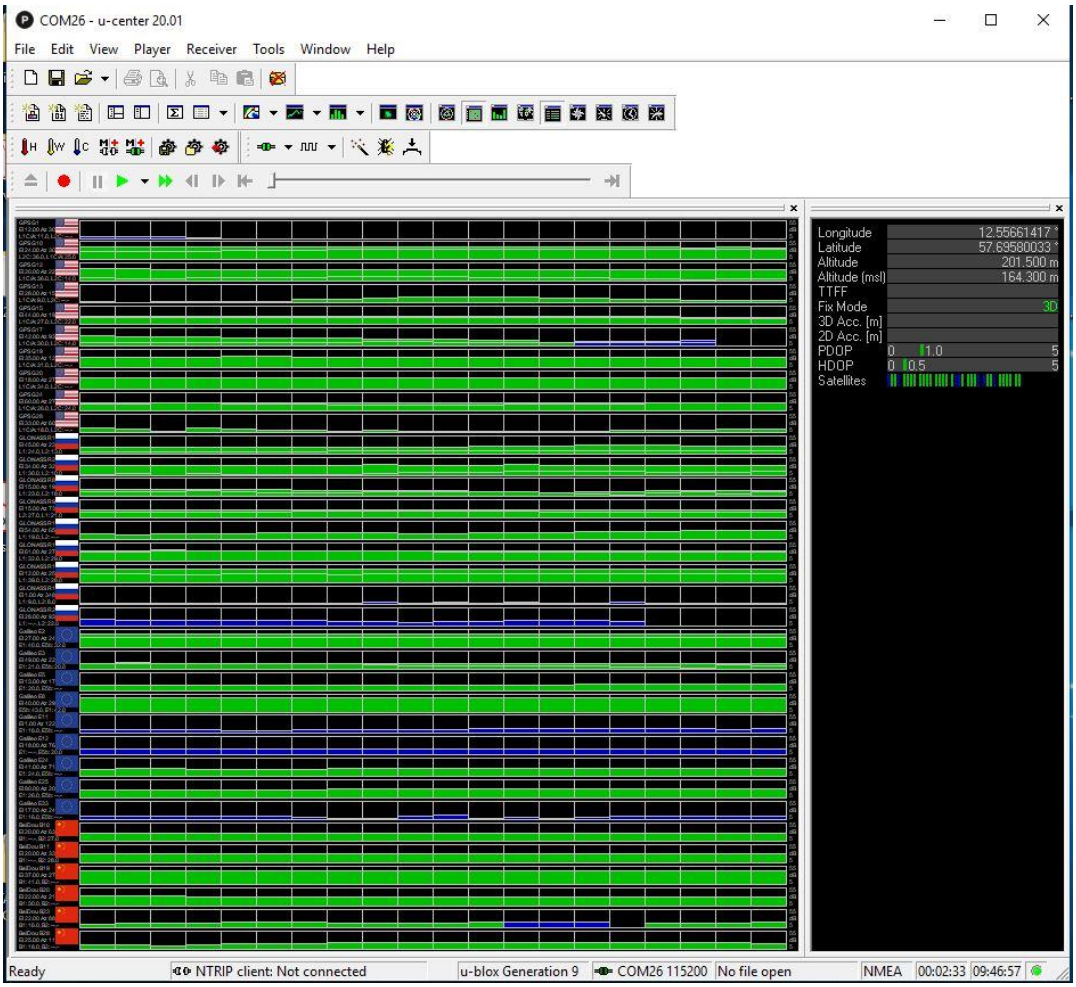


Fig. 12. u-center uppkopplad till en GPS-enhet via USB

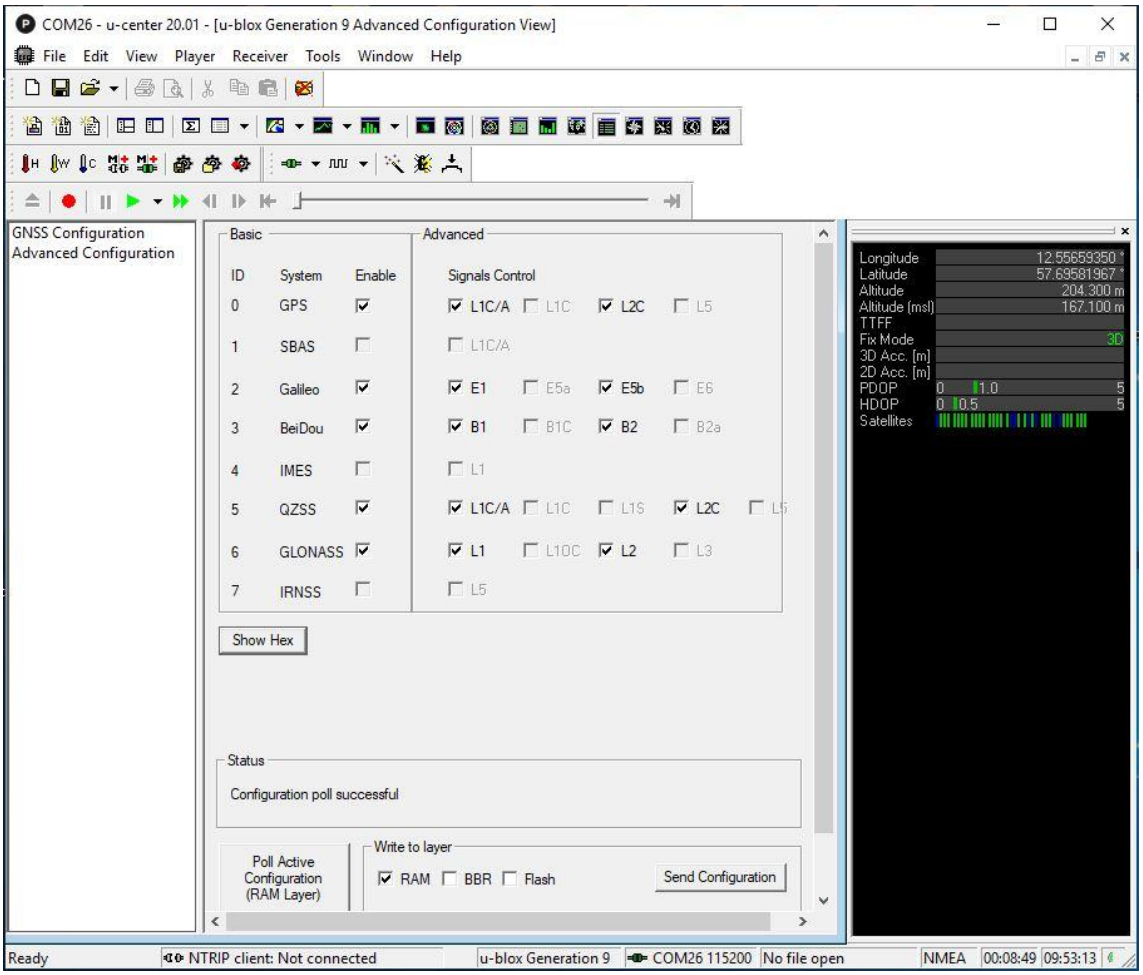


Fig. 13. u-blox GNSS Configuration

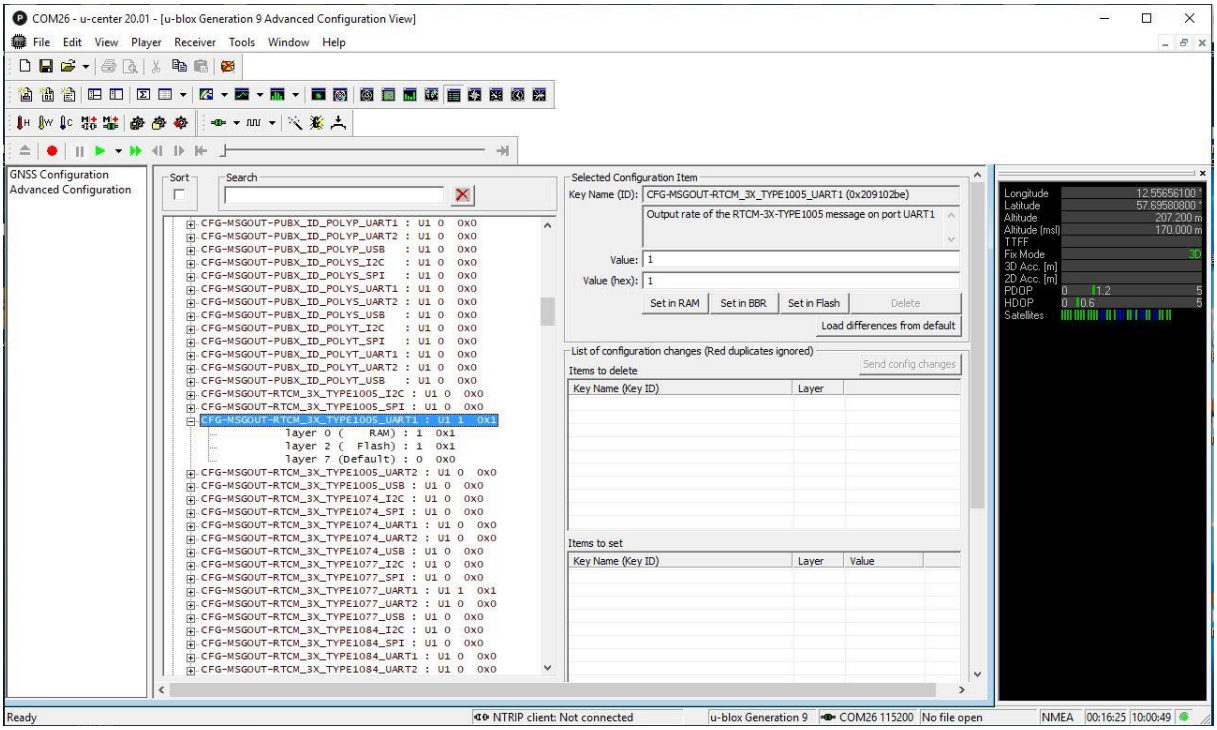


Fig. 14. u-blox Generation 9 Advanced Configuration

”Base” konfigurering

CFG-UART1-BAUDRATE	460800
CFG-RATE-MEAS	1000 (200 = 5 Hz)

CFG-MSGOUT-RTCM_3X_TYPE4072_0_UART1
CFG-MSGOUT-RTCM_3X_TYPE4072_1_UART1
CFG-MSGOUT-RTCM_3X_TYPE4077_1_UART1
CFG-MSGOUT-RTCM_3X_TYPE4087_1_UART1
CFG-MSGOUT-RTCM_3X_TYPE4097_1_UART1
CFG-MSGOUT-RTCM_3X_TYPE1127_1_UART1
CFG-MSGOUT-RTCM_3X_TYPE1230_1_UART1

”Rover” konfigurering

CFG-UART1-BAUDRATE	460800
CFG-RATE-MEAS	1000 (200 = 5 Hz)

CFG-MSGOUT-UBX_NAV_RELPOSNED_UART1

Flera detaljer finns i ZED-F9P Moving base application (UBX-19009093- R01) samt i ZED-F9P_IntegrationManual, clause 5.1.5 RTK configuration.

Programvara

Konfigurering av respektive GPS-enhet sker med u-center senaste version (20.01).
U-center används även för uppgradering av firmware till GPS-enheterna

U-center ver. 20.01 eller senare
Firmware 1.12 eller senare

Arduino Utvecklingsmiljö:

Arduino IDE 1.8.5, TeensyDuino 1,49

Arduinobibliotek:

```
#include <SPI.h>                                  // Lib for SPI-bus
#include <ILI9341_t3.h>                           // Lib for LCD Touch Display
#include <Wire.h>                                 // Needed for I2C to GPS
#include "SparkFun_Ublox_Arduino_Library.h"  // Lib for SparkFun ZED-F9P ver. 1.7.2
#include <SHT1X.h>                                // Lib for Sensirion Sensor
#include <Adafruit_Sensor.h>                      // Lib for BME280 Sensor
#include <Adafruit_BME280.h>                      // Lib for BME280 Sensor
#include <OneWire.h>                              // Lib for DS18B20 Sensor
#include <DallasTemperature.h>                    // Lib for DS18B20 Sensor
```

Kodrader för att hämta GPS-data

```
SFE_UBLOX_GPS myGPS;  // Connect to ROVER GPS Unit via I2C

// Navigation Data

latitude = float(myGPS.getHighResLatitude()) / 10000000.;
longitude = float(myGPS.getHighResLongitude()) / 10000000.;
haccuracy = float(myGPS.getHorizontalAccuracy()) / 10;
vaccuracy = float(myGPS.getVerticalAccuracy()) / 10;
altitude = float(myGPS.getMeanSeaLevel()) / 1000.;
heading = float((myGPS.relPosInfo.relPosHeading) / 100000.);
relposlength = float(myGPS.relPosInfo.relPosLength) * 10.;
 fixedtype = myGPS.getFixType();
 siv = myGPS.getSIV();
 gnssfixok = myGPS.relPosInfo.gnssFixOk;
 diffsoln = myGPS.relPosInfo.diffSoln;
 relposvalid = myGPS.relPosInfo.relPosValid;
 carrsoln = myGPS.relPosInfo.carrSoln;
 ismoving = myGPS.relPosInfo.isMoving;
 refposmiss = myGPS.relPosInfo.refPosMiss;
```

Programmet är uppdelat I en rad olika funktioner där inhämtning av data och visning av data är separerade. Detta för att minimera blinkning vid uppdatering av bildskärmsdata. Detta ger också en möjlighet för olika tidsintervall mellan inhämtning/presentation av olika data.

En mera komplett beskrivning och Kod finns att hämta på min hemsida: www.sm6afv.se

Förbättringspotential

Det finns flera delar som kan förbättras inte minst programvaran. En större och mera solvänlig display står på min egen önskelista.
Nätverks-RTK är en alternativ metod för noggrann riktning/positionsbestämning. Lantmäteriet har en sådan tjänst som dock kräver ett abonnemang. 30 dagar kostar SEK 2000. Det finns också ett gratis prova på abonnemang för 10 dagar.

Uppkoppling med en mobiltelefon och en blåtandsmodul till GNSS/GPS mottagaren är vad som krävs för att strömma korrektionsdata i realtid till mottagaren.

Öppna data – DGNSS. Detta är ett kostnadsfritt alternativ där man hämtar korrektionsdata via en Ntrip-klient i en Smartphone. Jag är dock osäker på att dessa korrektionsdata ger bättre slutresultat än det här beskrivna fristående systemet med två GNSS mottagare.

Förfarandet med att hämta in extern RTK/DGNSS data via mobiltelefonnätet komplicerar användandet av RiktningsIndikatorn. Jag har därför inte testat dessa tjänster. Jag vill först testa och utvärdera den nu färdigställda utrustningen.

Referenser

[1] GNSS och RTK. Lantmäteriet.
https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/gps-geodesi-och-swepos/GPS-och-satellitpositionering/Metoder-for-GNSS-matning/RTK/

NätverksRTK

https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/GPS-geodesi-och-Swepos/Swepos/vara-tjanster/natverks-rtk/

[2] http://www.k7fry.com/grid/

[3] DUBUS 3/2019 sid 9 -15

[4] https://sv.wikipedia.org/wiki/Satellitnavigation

[5] GPS Compendium Book, https://www.u-blox.com/sites/default/files/products/documents/GPS-Compendium_Book_%28GPS-X-02007%29.pdf

[6] https://store.drotek.com

[7] https://drotek.gitbook.io/rtk-f9p-positioning-solutions/how-to-get-started/zed-f9p-rtk-configuration

[8] ZED-F9P Summary: https://www.u-blox.com/sites/default/files/ZED-F9P_ProductSummary_%28UBX-17005151%29.pdf

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

[10] SparkFun: https://www.sparkfun.com/products/15136

[11] Arduinobibliotek: https://github.com/sparkfun/SparkFun_Ublox_Arduino_Library

[12] www.pjrc.com

[13] https://www.youtube.com/watch?v=g92GboiOkeQ

[14] https://www.youtube.com/watch?v=qlkN70bBfFQ

[15] https://www.youtube.com/watch?v=XM_hXPfiCws