

# GNSS RTK, czyli nawigacja o dokładności centymetrowej

Współczesna nawigacja to nie tylko GPS, ale cała rodzina systemów GNSS. Jeszcze niedawno centymetrowa dokładność wiązała się z ogromnymi kosztami i dostępna była tylko dla profesjonalistów. Dziś, dzięki modułowi RTK dla Raspberry Pi, niemal każdy może sam spróbować osiągnąć tę niesamowitą precyzję lokalizacji.

**Kilka słów o nawigacji satelitarnej**

**W poszukiwaniu dokładności**

**RTK w życiu codziennym**

**Czy można samemu zrealizować nawigację RTK?**

**Pierwsze eksperymenty pod Windows 10**

**W kierunku Raspberry Pi**

**Uruchomienie trybu RTK**

**Weryfikacja dokładności lokalizacji**

**Łyżka dziegciu w beczce miodu**

**Podsumowanie**

## **Kilka słów o nawigacji satelitarnej**

Nawigacja satelitarna stała się dziś na tyle powszechna, że niemal każdy z nas ma codziennie do czynienia z urządzeniami umożliwiającymi określanie położenia na podstawie precyzyjnych sygnałów wysyłanych przez satelity. Rzadko jednak zastanawiamy się nad zasadami ich działania.

Choć celem niniejszego tekstu nie jest szczegółowe omawianie tego systemu, warto w kilku słowach zwrócić uwagę na kwestię jego dokładności. Pierwszym systemem nawigacji satelitarnej był amerykański NAVSTAR GPS (*Navigation Signal Timing and*

*Ranging – Global Positioning System*), który powstał jako projekt militarny i początkowo był dostępny wyłącznie dla wojska. Dopiero w latach 80. ubiegłego wieku prezydent Stanów Zjednoczonych zezwolił na jego cywilne wykorzystanie. Przez wiele kolejnych lat w systemie tym stosowano jednak tzw. selektywną dostępność (*Selective Availability*), która ograniczała dokładność pozycjonowania dla użytkowników cywilnych do około 100 metrów.

Na początku 2000 roku wyłączono celowe zakłócanie sygnału, dzięki czemu każdy odbiornik GPS mógł określać swoją pozycję z dokładnością do kilku metrów –

był to prawdziwy przełom. To właśnie, w połączeniu ze znacznym spadkiem cen urządzeń, przyczyniło się do masowego upowszechnienia nawigacji satelitarnej.

Choć GPS jest powszechnie wykorzystywany cywilnie, pozostaje wciąż systemem wojskowym. Departament Obrony USA w każdej chwili może wyłączyć jego działanie lub ponownie włączyć celowe zakłócanie sygnału w wybranych rejonach świata. Dlatego inne państwa postanowiły stworzyć własne systemy nawigacji satelitarnej. Obecnie na orbicie Ziemi krążą satelity przesyłające sygnały dla rosyjskiego GLONASS, europejskiego Galileo, chińskiego BeiDou oraz japońskiego QZSS. Wszystkie te systemy razem tworzą Globalny System Nawigacji Satelitarnej, znany pod angielskim skrótem GNSS (*Global Navigation Satellite System*).

## W poszukiwaniu dokładności

Nowsze odbiorniki mogą jednocześnie korzystać ze wszystkich systemów, co pozwala zwiększyć dokładność pomiarów. Jednakże prawa fizyki, ograniczenia sprzętowe oraz wszechobecne zakłócenia sprawiają, że typowe urządzenie nie jest w stanie osiągnąć precyzji lepszej niż 1 metr.

Od wielu lat znane jest jednak rozwiązanie pozwalające na wielokrotne zwiększenie dokładności pomiarów: RTK (*Real Time Kinematic*). W dużym skrócie polega ono na użyciu dwóch odbiorników: jeden znajduje się w dokładnie znanej pozycji i przesyła dane korekcyjne do drugiego, którego lokalizację chcemy określić. Mogłoby się wydawać, że skoro znamy pozycję pierwszego odbiornika, możemy w każdej chwili określić błąd systemu i po prostu odjąć go od wskazań drugiego odbiornika, aby uzyskać właściwą lokalizację. W praktyce jednak takie rozwiązanie jest skuteczne głównie w przypadku usuwania wspomnianych wcześniej celowych zakłóceń; dla niezakłóconych sygnałów poprawa dokładności nie byłaby znacząca.

W systemie RTK, jak podaje Wikipedia, stosuje się metody polegające na „szybkim rozwiązaniu problemu nieoznaczoności pomiarów fazowych przez odbiornik ruchomy na podstawie przesłanych ze stacji referencyjnej poprawek do pseudoodległości oraz surowych danych pomiarów fazy sygnałów”. Choć zagadnienie jest skomplikowane, najważniejszą praktyczną informacją jest to, że pomiary RTK wymagają użycia nie jakichkolwiek dwóch odbiorników, lecz dwóch specjalnych urządzeń, które dodatkowo muszą się ze sobą komunikować.

## RTK w życiu codziennym

RTK jest obecnie najnowocześniejszą technologią umożliwiającą najdokładniejsze pomiary. Niewątpliwie jej wadą jest wysoki poziom skomplikowania

oraz koszt niezbędnych urządzeń – z tego powodu początkowo używana była jedynie w geodezji. Jednak zalety tej metody sprawiły, że z czasem zyskiwała na popularności, co przełożyło się również na stopniowy spadek cen odbiorników.

Dodatkowo okazało się, że choć najkorzystniejsze jest, aby stacja referencyjna znajdowała się jak najbliżej miejsca pomiarów, to akceptowalne wyniki można uzyskać także przy odległościach sięgających wielu kilometrów. Wraz z powstaniem zstandaryzowanych formatów wymiany danych korekcyjnych utworzono całe sieci stacji referencyjnych prowadzących ciągłe obserwacje, a rozwój Internetu sprawił, że dostęp do nich stał się możliwy praktycznie w każdym miejscu.

Dzisiaj odbiorniki GNSS RTK wykorzystywane są już nie tylko w geodezji. Znajdziemy je np. w nowoczesnych ciągnikach rolniczych, które są w stanie precyzyjnie realizować takie prace polowe jak siew czy oprysk, albo też w robotach koszących przydomowe trawniki.

## Czy można samemu zrealizować nawigację RTK?

Ceny odbiorników GNSS RTK wykorzystywanych w geodezji to poziom przynajmniej kilku lub kilkunastu tysięcy złotych. Jednak ostatnio na rynku pojawiają się też tańsze konstrukcje, stosowane choćby w autonomicznych kosiarkach. Z kolei w kierunku amatorów chcących spróbować swoich sił pomocną dłoń wyciągają Chińczycy, którzy udostępniają kompletne płytki z już wlutowanymi podzespołami. Eliminuje to konieczność samodzielnego montażu, co w przypadku wielu nowoczesnych układów scalonych jest, bez doświadczenia i bez specjalistycznego wyposażenia, niemalże niemożliwe.

Przykładem takiego produktu jest oferowany przez firmę Waveshare moduł „LC29H(XX) GPS/RTK HAT”. Ten odbiornik GPS/RTK bazuje na układzie Quectel LC29H, a widoczne w nazwie litery HAT (*Hardware Attached on Top*) informują, że jest to przeznaczona dla Raspberry Pi płytka, którą montuje się bezpośrednio na złączu rozszerzeń tego urządzenia.

Moduł dostępny jest w trzech odmianach. Najtańsza z nich, oznaczona literami AA, to jedynie odbiornik GNSS, który nie ma możliwości pracy w trybie RTK. W trybie RTK pracują moduły oznaczone literami DA i BS, przy czym pierwszy z nich umożliwia lokalizację urządzenia, drugi natomiast pozwala stworzyć własną stację referencyjną.

Ja zdecydowałem się na moduł DA, widoczny na **fotografii tytułowej**. Zakupiłem go w polskim sklepie, razem z anteną GPS i wysyłką kosztował mnie niecałe 320 zł.

## Pierwsze eksperymenty pod Windows 10

Moduł ma postać płytki z 40-pinowym złączem, pozwalającym na bezproblemowe połączenie go z Raspberry Pi. Dodatkowo znajdziemy tu też gniazdo microUSB, dwie zworki konfiguracyjne oraz cztery diody pokazujące status. Na szybko działanie płytki możemy sprawdzić korzystając z komputera stacjonarnego. Przystawiamy zworki w ustawienie „USB-LC29H”, a następnie kablem USB podłączamy go do komputera. Windows 10 rozpoznaje go bez potrzeby instalacji jakichkolwiek sterowników i w menedżerze urządzeń pojawia się kolejny port komunikacyjny, u mnie otrzymał on nazwę COM4. Teraz wystarczy połączyć się z tym portem dowolnym terminalem (ja używam bezpłatnego programu PuTTY), ustawiając prędkość transmisji na 115 200 bps, bo z taką domyślnie pracuje opisany moduł.

W terminalu powinien pojawić się tekst, są to komunikaty wysyłane zgodnie ze standardem NMEA (*National Marine Electronics Association*). Przykładowy fragment wygląda następująco:

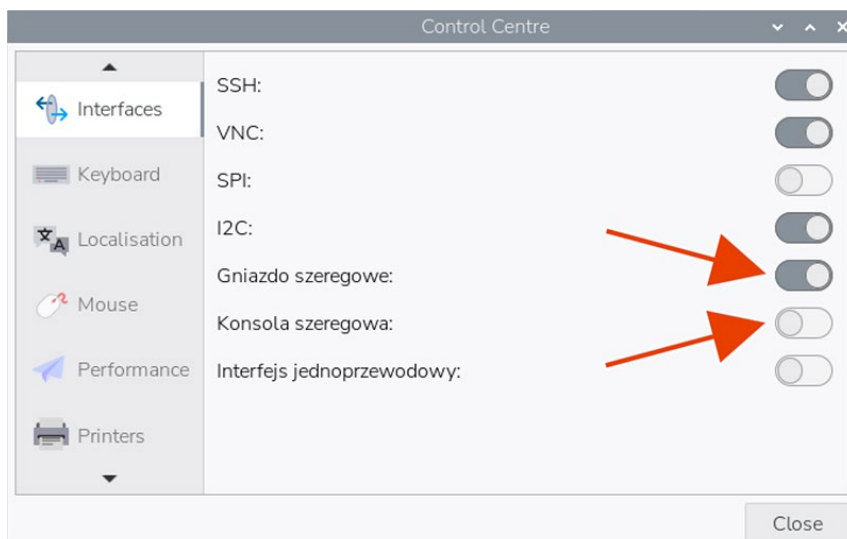
```
$GNGGA,000141,012,,,,,0,00,99,99,M,,M,,*4F
$GNRMC,000141,012,V,,,,,060180,,,N,V*21
$GNGLL,,,,,000141,012,V,N*63
$GNVTG,,T,,M,,N,,K,N*32
$GNGSA,A,1,,,,,,,,,99,99,99,99,99,99,1*33
```

Niczego nie trzeba włączać czy uruchamiać, dane same napływają co sekundę. Na początku ramki nie zawierają współrzędnych, jednak po podłączeniu anteny i jej wystawieniu na zewnątrz urządzenie zacznie wysyłać o wiele więcej informacji, w tym szczególnie interesujące nas położenie geograficzne.

## W kierunku Raspberry Pi

Moduł docelowo chciałem uruchomić na miniaturowym komputerze Raspberry Pi, dlatego też przełożyłem zworki w położenie „Pi-LC29H”, a samą płytkę wpiąłem bezpośrednio do złącza przeznaczonego dla takich rozszerzeń.

Pod adresem internetowym [https://www.wave-share.com/wiki/LC29H\(XX\)\\_GPS/RTK\\_HAT#Raspberry\\_Pi](https://www.wave-share.com/wiki/LC29H(XX)_GPS/RTK_HAT#Raspberry_Pi) znajdziemy udostępniony przez producenta opis uruchomienia modułu. W pierwszej kolejności instrukcja zaleca włączenie obsługi interfejsów UART oraz I2C. Zawarty w niej zrzut ekranu może być jednak mylący, ponieważ przedstawia nieaktualny wygląd menu, którego nie znajdziemy w now-



Rysunek 1

szej wersji systemu Raspberry Pi OS, opartej na Debianie Trixie. Dziś do konfiguracji tej dotrzemy po wybraniu z menu „Preferencje / Control Centre”, a następnie zakładki „Interfaces”, gdzie powinniśmy włączyć obsługę „I2C” oraz „Gniazdo szeregowe”. Według producenta płytka umożliwia kontakt przez I2C, jednak ja tego nie testowałem.

Istotne jest, aby w tym miejscu **wyłączyć** obsługę interfejsu „Konsola szeregowa” (patrz **rysunek 1**), gdyż w przeciwnym razie system operacyjny będzie traktował port szeregowy jako konsolę systemową, co skutecznie utrudni nam komunikację z modułem – a przynajmniej ja zmarnowałem przez to sporo czasu.

Teraz moduł powinien być dla nas dostępny jako urządzenie `/dev/ttyS0`, co szybko możemy zweryfikować po wpisaniu w terminalu poniższych poleceń:

```
sudo stty -F /dev/ttyS0 115200
cat /dev/ttyS0
```

Pierwsze z nich ustawia wymaganą prędkość portu szeregowego, drugie odczytuje z niego dane i wypisuje je na ekran. Jeśli wszystko działa tak jak powinno, a antena widzi wystarczającą liczbę satelitów, to na ekranie powinniśmy ujrzeć strumień danych NMEA zawierający między innymi współrzędne geograficzne. W samym module niczego nie trzeba lutować ani konfigurować, całość automatycznie wystartuje po włączeniu.

## Uruchomienie trybu RTK

Niestety nie jest to jeszcze dokładnie to, co chcieliśmy osiągnąć, gdyż urządzenie działa teraz podobnie jak zwykły odbiornik GPS, a jedyna jego перевага to fakt, że korzysta ze wszystkich dostępnych systemów (czyli GPS, GLONASS, Galileo i BeiDou, bo satelity japońskiego QZSS nie są widoczne nad Polską), co pozwala na zwiększenie precyzji.



Aby moduł wszedł w tryb pracy RTK, musimy zacząć dostarczać mu aktualne dane korekcyjne ze stacji referencyjnej. Możemy sami taką zbudować, ale możemy też skorzystać z danych innej pobliskiej stacji. W Polsce od lat funkcjonuje sieć ASG-EUPOS, zarządzana przez Główny Urząd Geodezji i Kartografii. Udostępnia ona poprawki przeznaczone do odbiorników RTK, a dostęp do niej jest całkowicie bezpłatny, wymaga jedynie rejestracji.

Poprawki te musimy pobrać z właściwego serwera i przesłać do naszego urządzenia. Producent modułu zamieszcza na swojej stronie przykładowy program. Znajdziemy go w udostępnionym pliku „Lc29h\_gps\_rtk\_hat\_code.zip”, gdzie w folderze „\python\rtk\_rover” czeka na nas plik o nazwie „main.py”. Analiza tego skryptu pozwala zauważyć, że działa on w następujący sposób: najpierw odczytuje nasze współrzędne z odbiornika i przekazuje je do serwera poprawek. Ten udostępnia mu dane pochodzące ze stacji najbliższej geograficznie, a skrypt sukcesywnie przesyła je do modułu GNSS, który uwzględnia je w obliczeniach pozycji. Wyniki tych obliczeń są na bieżąco przekazywane przez port szeregowy, z którego skrypt czyta dane i wyświetla je na ekranie.

Wystarczy więc jedynie uruchomić skrypt z właściwymi parametrami. Po wielu próbach stwierdziłem, że dla wykorzystywanej przeze mnie sieci ASG-EUPOS poprawne wywołanie jest następujące:

```
python main.py 91.198.76.2 2101 RTN4G_VRS_RTCM32 -u USER -p PASS.
```

Skrypt wyświetli informacje dotyczące połączenia z serwerem poprawek, a następnie cyklicznie pobiera z niego dane i wysyła je do modułu, jednocześnie dane pobrane z modułu wyświetla na ekranie. Brak tu niestety czytelnej informacji na temat aktualnego statusu, gdyż tak naprawdę wyświetlane są jedynie niesformatowane linie zaczynające się od GNGGA. Linie te w szóstym swoim polu (po czasie i współrzędnych geograficznych) zawierają jednocyfrowy status, którego znaczenie przedstawia **rysunek 2**.

Po uruchomieniu skryptu najprawdopodobniej będzie tam cyfra 2, która jest najdokładniejszym trybem możliwym do uzyskania bez uwzględnienia poprawek. Po wysłaniu poprawek status powinien zmienić się na 5, a po jakimś czasie nawigacja powinna pokazywać status 4, który świad-

czy o pracy w trybie RTK. Ciekawostką jest to, że w standardzie NMEA najdokładniejszemu trybowi RTK przypisano cyfrę 4, a nie 5, przez co kolejność statusów wydaje się nieco myląca.

### Weryfikacja dokładności lokalizacji

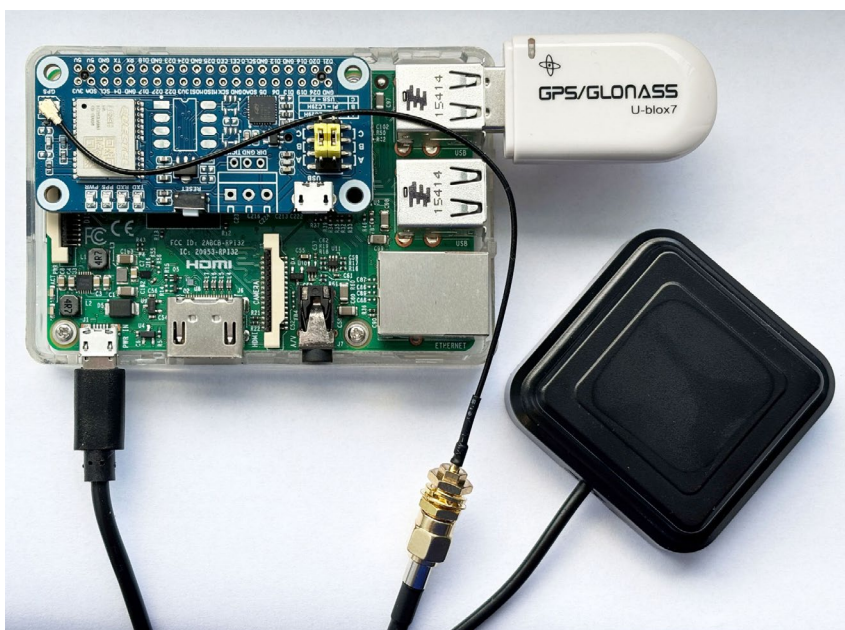
Dokładność centymetrowa oczywiście cieszy, ale należałoby ją w jakiś sposób zweryfikować. Ja w pierwszej kolejności postanowiłem sprawdzić, na ile wskazania te są stabilne, pomijając na razie ich porównywanie z punktami geodezyjnymi. Jest to więc bardziej test powtarzalności, a nie dokładności bezwzględnej.

W tym miejscu chciałbym też zauważyć, że opisane dalej rezultaty osiągnąłem z odbiornikami umieszczonymi na balkonie, gdzie mogły one zobaczyć co najwyżej połowę nieba. Z pewnością nie były to warunki optymalne do tego typu zadań, jednak na początku stycznia, przy mrozie i padającym śniegu, nie zdecydowałem się na wyprawę w otwarty teren.

Dla porównania osiąganych wyników do mojego Raspberry Pi podłączyłem też tani lokalizator, wpinany do gniazda USB (**fotografia 3**).

Status:	Nazwa:	Opis:
0	Fix not available	Brak ustalonej pozycji.
1	GPS fix (SPS)	Standardowa dokładność, tryb autonomiczny bez poprawek.
2	Differential GPS fix (DGPS)	Fix z poprawkami różnicowymi, dokładniejszy niż zwykły.
5	RTK float integers	Tryb RTK z „pływającą” fazą, dokładność decymetrowa.
4	RTK fixed integers	Fix RTK, czyli pełen sukces i dokładność centymetrowa.

**Rysunek 2**



**Fotografia 3**

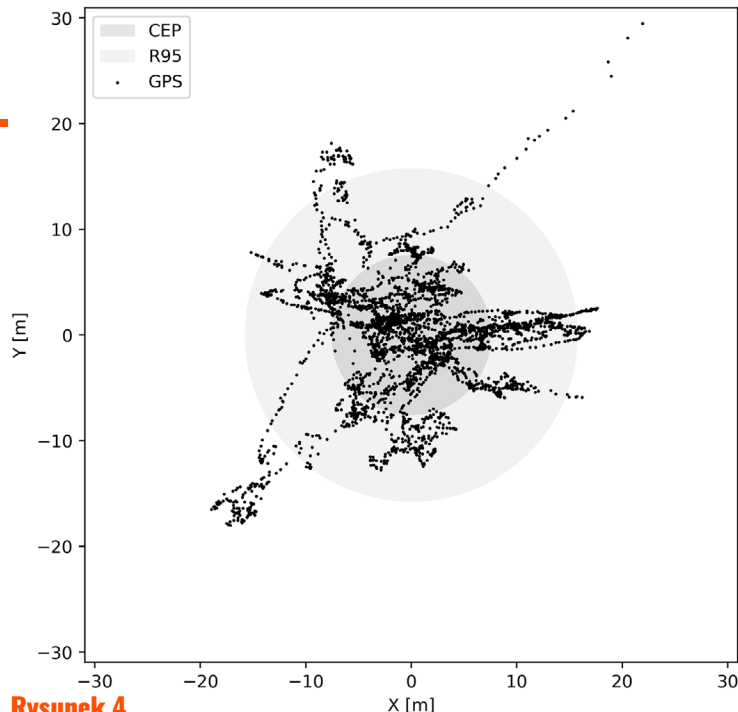
Wprawdzie z umieszczonego na nim opisu wynika, że odbiera on sygnały GPS/GLO-NASS, jednak mój egzemplarz widzi jedynie satelity GPS.

Na **rysunku 4** widać rozrzut rejestrowanych co sekundę współrzędnych w trakcie godzinnej obserwacji, obejmującej łącznie 3600 pozycji wskazywanych przez ten bardzo tani odbiornik GPS, pracujący dodatkowo w nieoptymalnych warunkach. Zdecydowałem się na prezentację wyników w układzie lokalnym XY, którego początek znajduje się we współrzędnych wskazywanych przez pomiary RTK, o których wspomnę dalej. Jak widać, w trakcie godziny współrzędne „pływają”, jednak wszystkie mieszczą się w kwadracie o boku 30 metrów. Dodatkowo w tle narysowane są dwa okręgi: mniejszy z nich to CEP (*Circular Error Probability*), obejmujący 50% wszystkich odczytów, natomiast większy to R95, wewnątrz którego znajduje się 95% pomiarów.

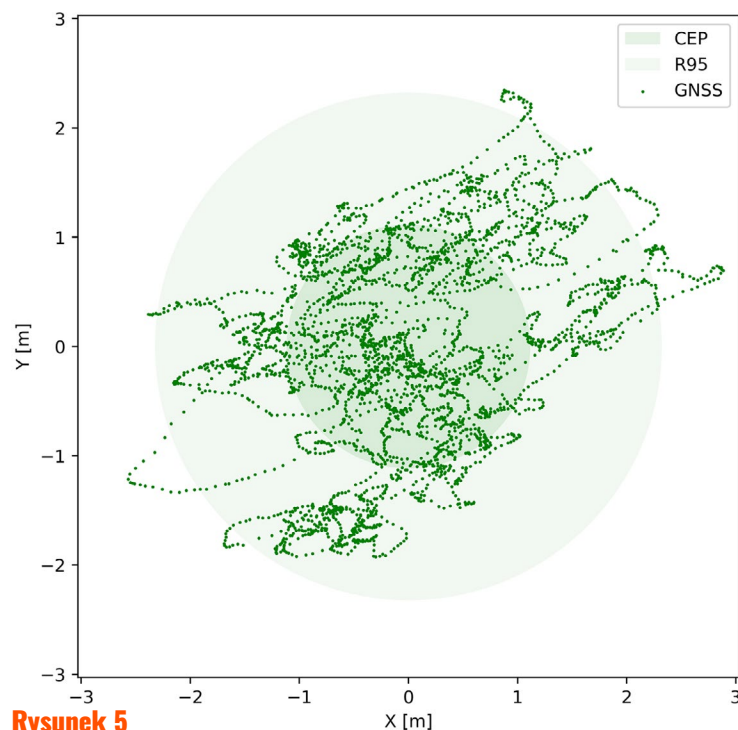
**Rysunek 5** przedstawia analogiczny wykres, tym razem dla 3600 współrzędnych podawanych przez moduł z układem LC29H, obserwującym satelity kilku systemów, dlatego opisałem go na wykresie skrótem GNSS. Widzimy tu wyraźną poprawę, gdyż w trakcie godzinnej obserwacji błąd pozycji nie przekroczył 3 m.

Z kolei **rysunek 6** pokazuje taki sam wykres dla 3600 współrzędnych podanych przez ten sam odbiornik, ale pracujący w trybie RTK. Jak widać, teraz udało się osiągnąć rozrzut wskazań mierzony w pojedynczych centymetrach. Warto zauważyć, że wyniki pomiarów przestały zlewać się w jedną płaszczyznę – wynika to z ograniczeń formatu, w jakim przesyłane są obliczone przez odbiornik współrzędne. Urządzenie przesyła je jako stopnie i minuty kątowe, z dokładnością do sześciu miejsc po przecinku. Jedna milionowa minuty kątowej odpowiada w Polsce około 1,8 mm dla szerokości geograficznej oraz 1,2 mm dla długości geograficznej. Pomiędzy kolejnymi współrzędnymi pozostają więc niewielkie przerwy, które stają się widoczne przy tej skali.

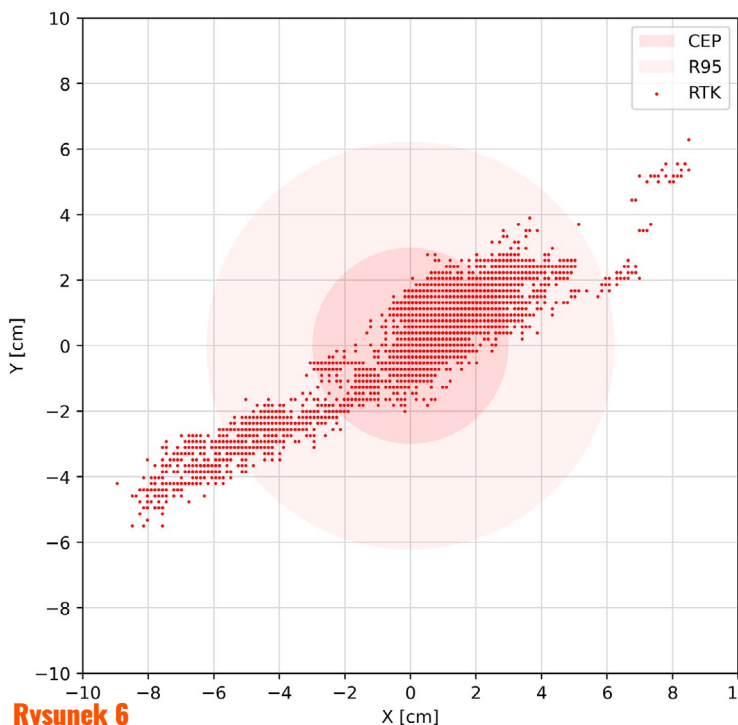
Ostatni już wykres, widoczny na **rysunku 7** (na następnej stronie), powstał poprzez nałożenie na siebie wszystkich trzech poprzednich serii pomiarowych. Widać wyraźnie, że w każdej z nich następowało znaczące zmniejszenie rozrzutu pomiarów.



**Rysunek 4**



**Rysunek 5**



**Rysunek 6**

### Łyżka dziegciu w beczce miodu

Niestety nigdy nie jest tak dobrze, aby nie mogło być lepiej. W przypadku opisywanego tu odbiornika dużym problemem okazało się dla mnie skonfigurowanie oprogramowania, które standardowo używane jest w takich zastosowaniach. Trzeba tu bowiem zauważyć, że prosty skrypt nie zapewni takiej niezawodności i funkcjonalności, jak narzędzia udostępniane przez system operacyjny.

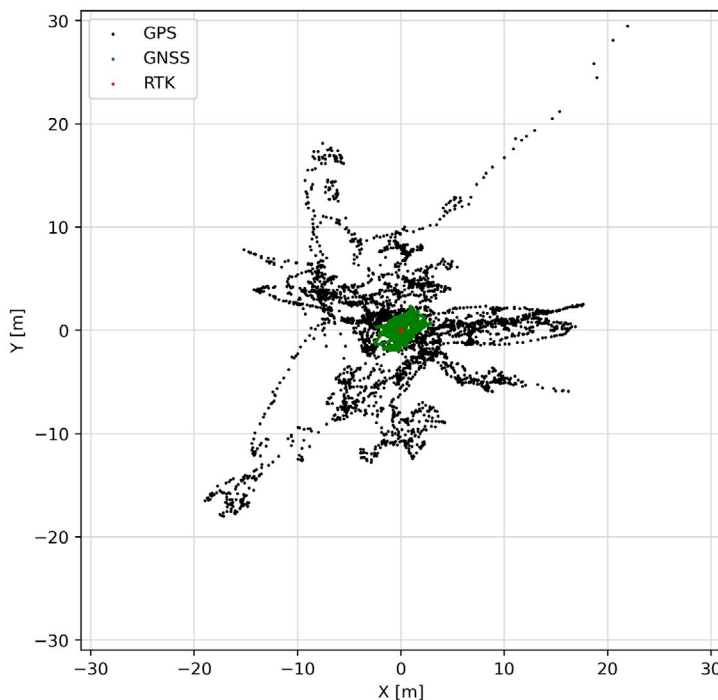
W Linuksie do obsługi nawigacji satelitarnej zastosowanie ma najczęściej program systemowy (usługa/daemon) o nazwie `gpsd`. Ten działający w tle proces zbiera, przetwarza i udostępnia dane z odbiornika. Jest to standard, który akceptują niemal wszystkie aplikacje korzystające z lokalizacji. Przykładem może być program `xgps`, pokazujący na żywo lokalizację widocznych satelitów (**rysunek 8**).

Program `gpsd` można uruchomić z odbiornikiem opisanym w tym artykule, wpisując np. `sudo gpsd /dev/ttyS0 -F /var/run/gpsd.sock`.

Problem polega jednak na tym, że `gpsd` zajmuje wtedy port, przez co nie można już wysłać do niego poprawek i uzyskać trybu RTK. Z kolei dane korekcyjne najlepiej przesyłać narzędziem `str2str`, które zostało do tego celu stworzone. Oba programy należy skonfigurować tak, aby ze sobą współdziałały, lecz mnie nie udało się tego osiągnąć. Nie pomogła nawet próba przesyłania danych przez port wirtualny (np. przy użyciu `socat`). Wydaje się jednak, że konfiguracja taka powinna być możliwa. A może ktoś z Czytelników pisma „Zrozumieć Elektronikę” ma w tym zakresie większe doświadczenie?

### Podsumowanie

Opisane w tym artykule próby pokazują, że technologia RTK jest dostępna również dla amatorów, a tanie moduły GNSS/RTK mogą dostarczać danych o precyzji wcześniej zarezerwowanej jedynie

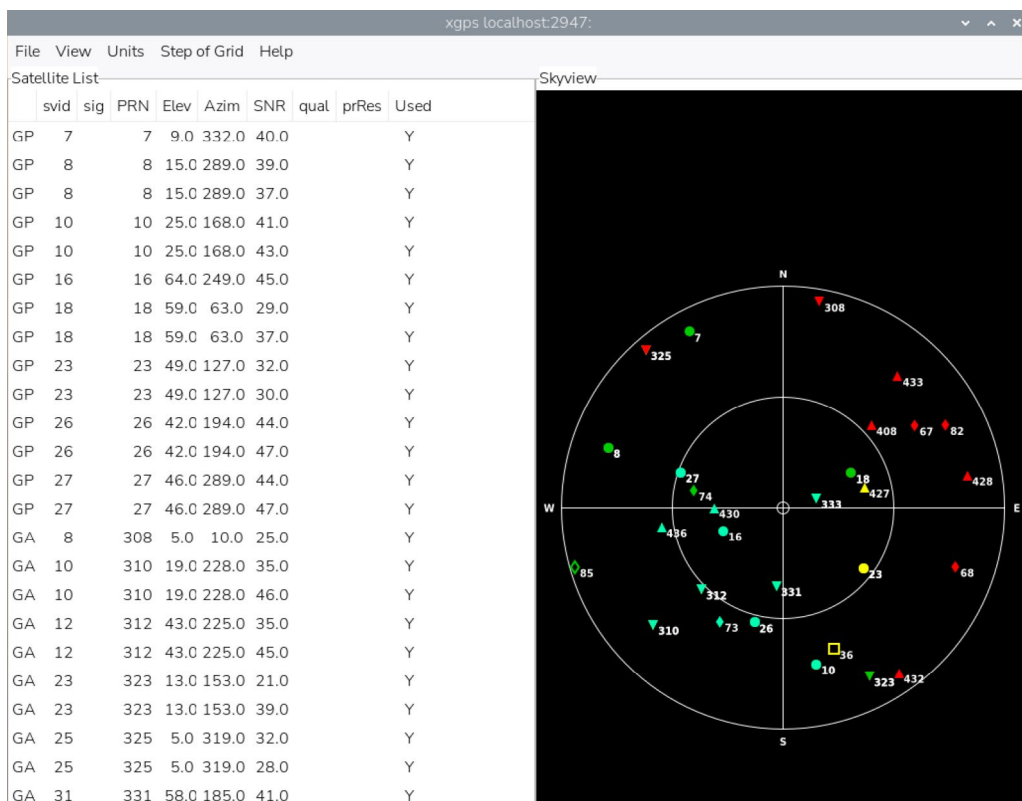


Rysunek 7

dla profesjonalnych odbiorników geodezyjnych. Kolejnym krokiem powinna być weryfikacja poprawności tych wskazań względem punktu o znanych współrzędnych, co pozwoliłoby ocenić dokładność bezwzględną.

**Marian Gabrowski**

marian kropka gabrowski @ gmail kropka com



Rysunek 8