



GNSS RTK, testy terenowe nawigacji

W poprzednim numerze ZE opisywałem moje zmagania związane z uruchomieniem bazującego na Raspberry Pi odbiornika GNSS RTK, czyli nawigacji o dokładności centymetrowej. Wyniki pierwszych prób okazały się na tyle obiecujące, że postanowiłem zweryfikować działanie tego urządzenia w terenie.

Źródło zasilania

Dostęp do sieci

Interfejs użytkownika

Obudowa

W poszukiwaniu precyzyjnych map

QGIS

Testy z mapami uzbrojenia terenu

Testy z osnową geodezyjną

Podsumowanie

Choć nawigacja poprawnie działała na moim biurku, to przeprowadzenie kolejnych testów poza domem wymagało wprowadzenia pewnych zmian, aby instalacja stała się mobilna: należało zadbać o niezależne zasilanie, łącze internetowe oraz interfejs użytkownika. Poniżej pokrótce opiszę rozwiązania, na jakie się zdecydowałem.

Źródło zasilania

Ponieważ Raspberry Pi zasilane jest poprzez gniazdo USB, oczywistym rozwiązaniem jest użycie powerbanku. Zmierzony pobór prądu mojego Raspberry Pi 3 wraz z podłączonym odbiornikiem

GNSS i wyświetlaczem LCD wynosi około 0,8 A. Pozwala to oczekiwać, że powerbank o pojemności 20 000 mAh zapewni urządzeniu wiele godzin nieprzerwanej pracy.

Dostęp do sieci

W domu Raspberry Pi mogło korzystać z routera Wi-Fi, jednak w terenie należało zapewnić inne łącze internetowe. Zdecydowałem się na „hotspot osobisty”, który można dziś aktywować w niemal każdym smartfonie. W zasięgu sieci 5G szybkość transmisji okazała się w pełni wystarczająca do przesyłania poprawek RTK do modułu GNSS.

X – RÓŻNE

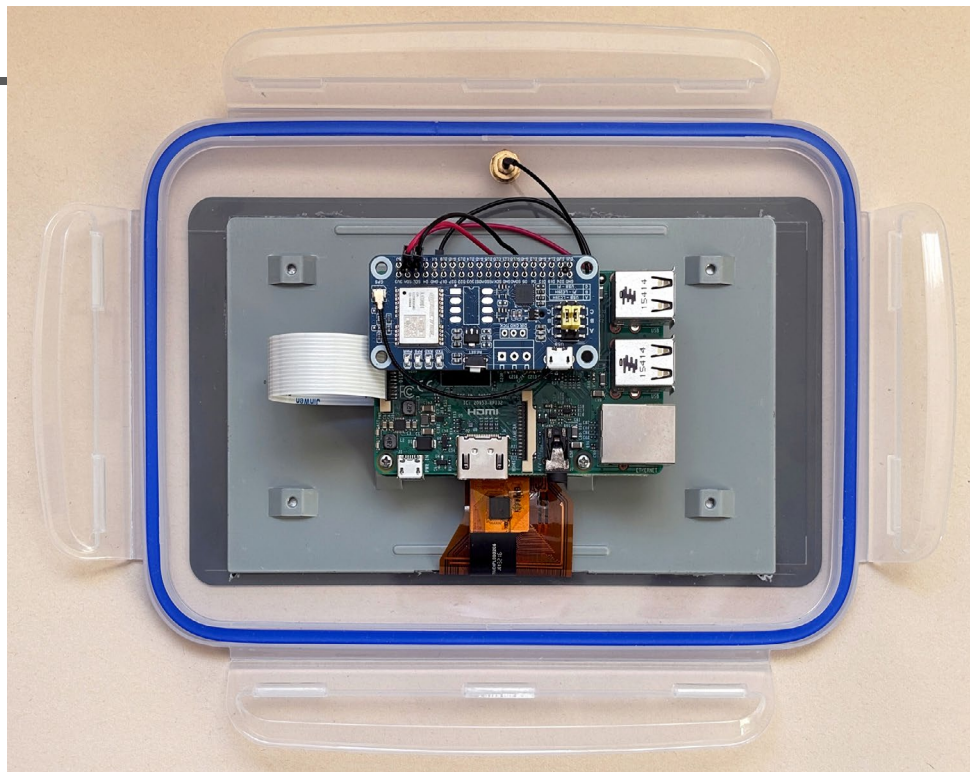
Wydaje się, że ze względu na niskie zapotrzebowanie na przepustowość danych, całość powinna działać również przy połączeniu LTE bądź 3G, jednak nie miałem jeszcze okazji sprawdzić tego w praktyce.

W tym miejscu warto zauważyć, że mój pracujący pod kontrolą systemu iOS smartfon nie umożliwia podglądu adresu IP klienta korzystającego z udostępnianego łącza. Stanowi to pewne utrudnienie, ponieważ aby połączyć się z Raspberry Pi w ramach lokalnej sieci hotspota, powinienem znać jego adres IP. Ten zaś jest przydzielany dynamicznie, a zatem może ulegać zmianom. Problem ten staje się istotny w sytuacji, gdy klient nie ma podłączonego fizycznego ekranu.

Interfejs użytkownika

Pracując z Raspberry Pi w domu, zazwyczaj korzystam ze zdalnego pulpitu, co pozwala mi na używanie monitora, myszy i klawiatury podłączonych do komputera stacjonarnego. W terenie można oczywiście posiłkować się laptopem lub odpowiednią aplikacją na smartfonie.

Jednak tym razem, ze względu na wspomniane wcześniej wątpliwości związane z dynamicznym adresem IP, zdecydowałem się na podłączenie fizycznego ekranu. Wykorzystałem posiadany od lat dedykowany dla Raspberry Pi wyświetlacz dotykowy 7" Touchscreen Display, którego wcześniej nie miałem jeszcze okazji przetestować. Montaż polegał na przykręceniu minikomputera czterema śrubami do tylnej części obudowy wyświetlacza. Za połączenie elektryczne odpowiada specjalna taśma przesyłająca obraz oraz dane dotyku, a także dwa przewody wpinane w piny GPIO, które służą do zasilania. System automatycznie wykrywa



Fotografia 1

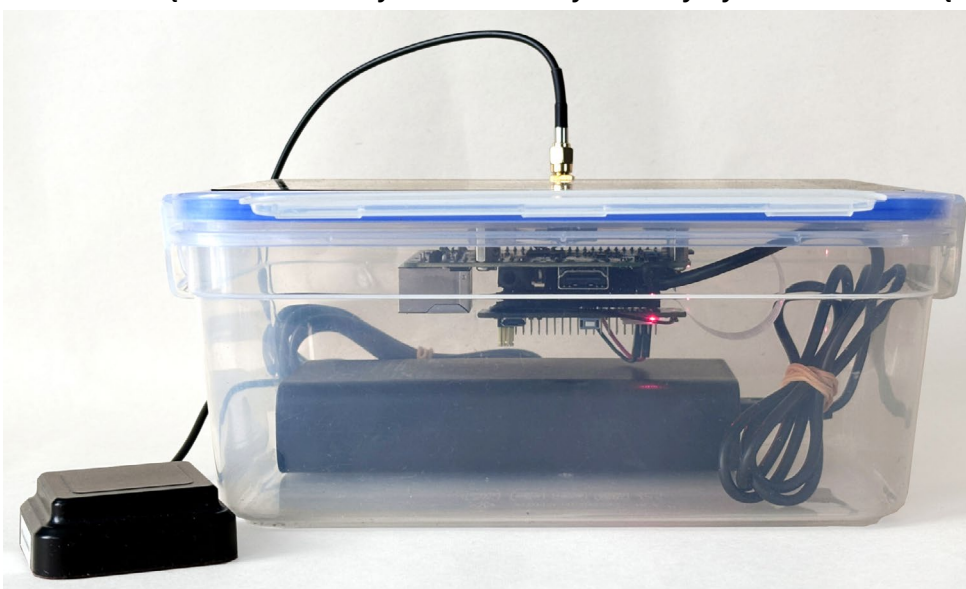
ekran oraz funkcje dotykowe, dzięki czemu konfiguracja nie jest wymagana.

Obudowa

Moja nawigacja była już niemal gotowa do wyprawy w teren, jednak całość należało jeszcze zabezpieczyć przed uszkodzeniem. Zdecydowałem, że na czas testów wszystko umieszczę w plastikowym pudełku, widocznym na **fotografii tytułowej**. Pod względem gabarytów idealny okazał się wykonany z przezroczystego polipropylenu pojemnik spożywczy o pojemności 2,2 litra.

W pokrywie wyciąłem prostokątny otwór o wymiarach nieco mniejszych niż metalowa ramka wyświetlacza. Dzięki temu, po wciśnięciu ekranu w przygotowane miejsce, nie było potrzeby stosowania dodatkowych mocowań. Powyżej ekranu wywierciłem otwór, w którym przykręciłem gniazdo antenowe zamontowane na końcu krótkiego przewodu, wpinanego bezpośrednio do modułu GNSS. Wnętrze pokrywy z zamontowanymi podzespołami szczegółowo prezentuje **fotografia 1**.

Moduł odbiornika został wpięty w dedykowane złącze za pośrednictwem dostarczonej przez producenta przejściówki z długimi goldpinami, które umożliwiają podłączenie kolejnych podzespołów. W moim przypadku wpiąłem w nie jedynie dwa przewody dostarczające napięcie 5 V do wyświetlacza. Całość konstrukcji uzupełnił widoczny na **fotografii 2** powerbank, umieszczony razem z przewodem USB na dnie pudełka.



Fotografia 2

W poszukiwaniu precyzyjnych map

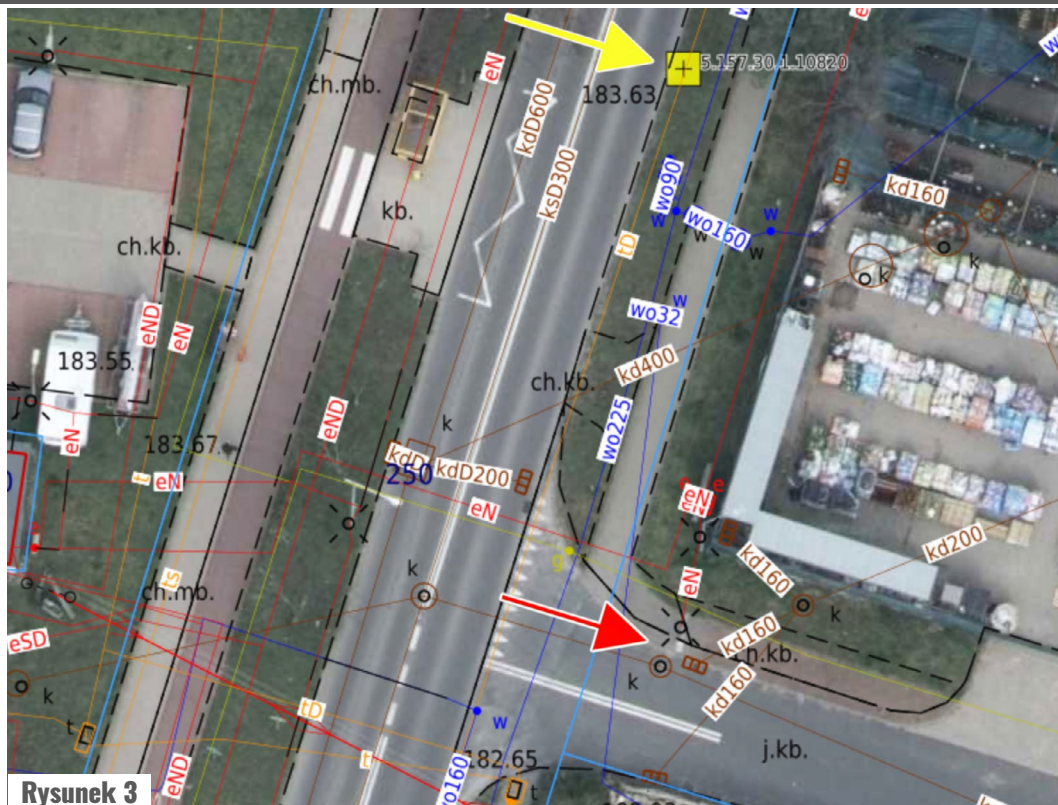
Mając nawigację zamkniętą w prowizorycznej, mobilnej obudowie, mogłem w końcu wyruszyć na testy terenowe. Teraz wyzwaniem stała się precyzja map, z którymi zazwyczaj pracujemy. Przy spodziewanej dokładności odbiornika poziom szczegółowości Map Google czy OpenStreetMap jest po prostu niewystarczająca.

Aby w pełni wykorzystać potencjał urządzenia, należało sięgnąć po znacznie dokładniejsze opracowania – mapy geodezyjne. Można je przeglądać np. w serwisie Geoportal, który pozwala nakładać na siebie interesujące nas warstwy danych (przykład widać na **rysunku 3**). Niestety, przeglądarkowa wersja serwisu oferuje maksymalną skalę 1:250, co przy centymetrowej dokładności byłoby istotnym ograniczeniem. Dlatego też zdecydowałem się skorzystać ze specjalnego programu do pracy z mapami.

QGIS

Popularnym narzędziem do pracy z danymi przestrzennymi jest QGIS. To potężny, a zarazem całkowicie darmowy program, z którego korzystają głównie profesjonaliści. Jednak dzięki ogromnej liczbie dostępnych poradników opanowanie podstawowych czynności powinno być w zasięgu także osób, które wcześniej nie miały do czynienia z tą tematyką. Co istotne, najnowszą wersję tego programu zainstalujemy i uruchomimy nawet na leciwym Raspberry Pi 3.

W aplikacji można w prosty sposób dodawać warstwy wektorowe, przedstawiające np. dane dotyczące uzbrojenia terenu czy granice działek pochodzące z ewidencji gruntów i budynków. QGIS oferuje również funkcję wyświetlania aktualnej pozycji, pobieranej na bieżąco z odbiornika nawigacji. Niestety, samodzielnie nie obsłuży on wykorzystywanego tu modułu „LC29H(DA) GPS/RTK HAT”, gdyż ten wymaga stałego dostarczania danych korekcyjnych ze stacji referencyjnej. Jednak łatwo tak zmodyfikować skrypt *main.py* (udostępniony przez producenta modułu), aby dane z odbiornika były wystawiane na porcie lokalnym. Dzięki temu QGIS może je odczytać i wyświetlić na mapie.



Rysunek 3



Fotografia 4

Testy z mapami uzbrojenia terenu

Teraz, mając dostęp do map geodezyjnych oraz precyzyjnej informacji o położeniu, można zweryfikować działanie nawigacji w terenie. Dla przykładu, na **fotografii 4** czerwona strzałka wskazuje antenę odbiornika ustawioną na krawężniku, pomiędzy trzema charakterystycznymi punktami: studzienką kanalizacyjną, kratką kanalizacji deszczowej oraz latarnią elektryczną. To samo miejsce zostało dodatkowo oznaczone czerwoną strzałką na rysunku 3.

Z kolei na Raspberry Pi uruchomiłem wspomniany wcześniej skrypt w języku Python, obsługujący odbiornik RTK, oraz program QGIS z wczytanymi mapami. Na rzucie ekranowym przedstawionym na **rysunku 5** widać aktualną pozycję, odczytywaną z modułu nawigacji, którą na mapie symbolizuje okrąg ze znakiem plus, oznaczony tu czerwoną strzałką. Całość działa więc zgodnie z oczekiwaniami, a wskazania pokrywają się z danymi zawartymi na mapach geodezyjnych.

Testy z osnową geodezyjną

Choć współczesne mapy geodezyjne niekiedy potrafią zaskoczyć swoją dokładnością, to z pewnością studzienki i latarnie nie są najlepsze do weryfikacji poprawności wskazań odbiornika o dokładności rzędu centymetrów.

Można jednak znaleźć obiekty, których położenie znane jest z o wiele większą precyzją. Są to punkty szczegółowej osnowy geodezyjnej poziomej. Ich lokalizację można podejrzeć we wspomnianym już serwisie Geoportal. Na rysunku 3 żółta strzałka wskazuje jeden z takich punktów, podpisany tu numerem 5.157.30.1.10820. Choć same mapy nie pozwalają odczytać jego precyzyjnych współrzędnych, to dane te bez problemu uzyskałem w Wydziale Geodezji i Kartografii Starostwa Powiatowego.

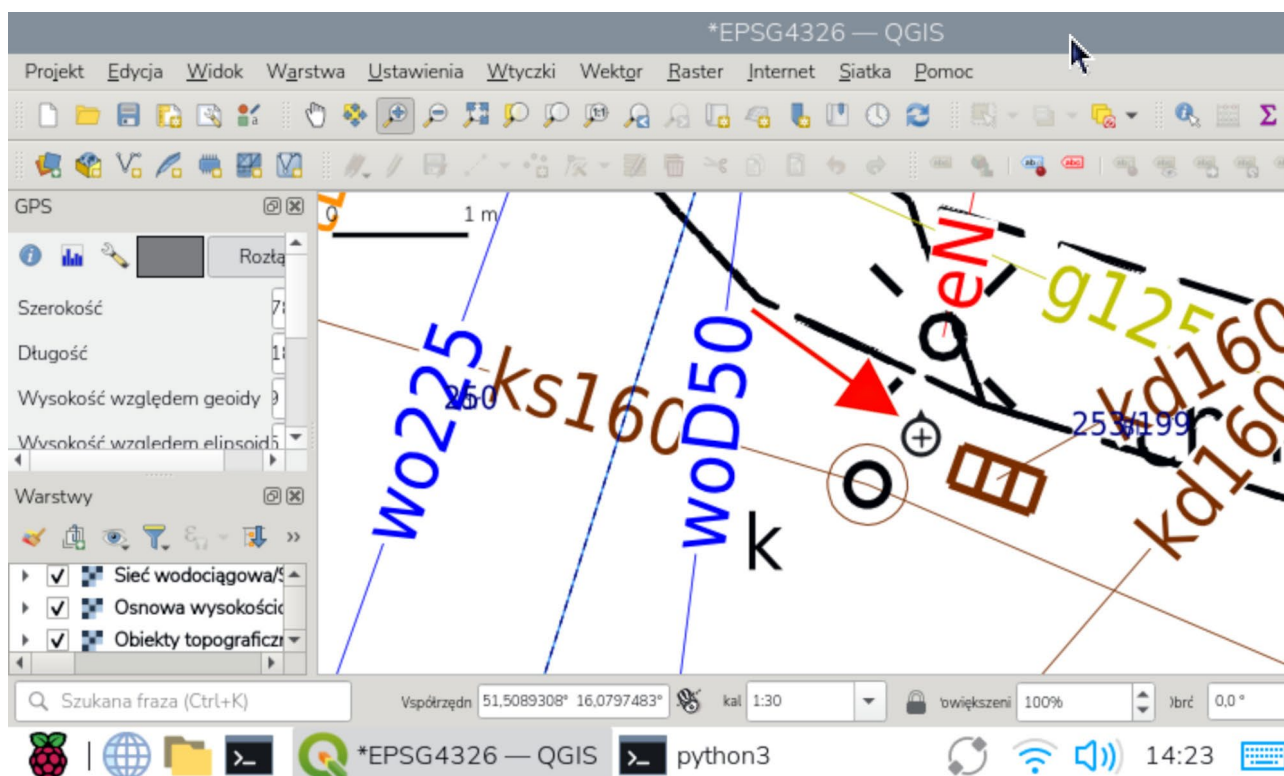
W prowadzonej przez urząd bazie wspomniany punkt ma następujące współrzędne: $x=5708843,84$

oraz $y=5574959,68$. Miałem też takie szczęście, że całkiem przypadkowo natrafiłem na geodetę pracującego w tym miejscu ze swoim profesjonalnym odbiornikiem GNSS RTK. Poprosiłem o udostępnienie wyników jego pomiaru, które były następujące: $x=5708843,821$ oraz $y=5574959,757$.

Wartości te zapisane są w używanym w polskiej geodezji układzie współrzędnych PL-2000, opartym na elipsoidzie GRS-80 i odwzorowaniu Gaussa-Krügera. Obszar kraju podzielono tu na cztery pasy południkowe o szerokości 3° , oznaczone numerami od 5 do 8. Mój punkt znajduje się w strefie 5; w programie QGIS strefa ta ma oznaczenie EPSG:2176.

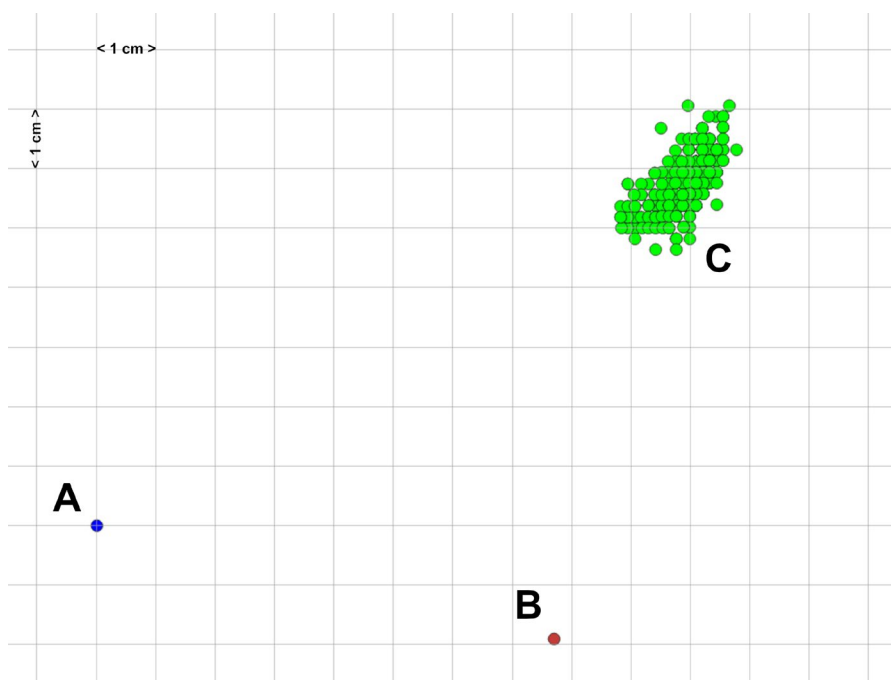
Ponieważ współrzędne podawane są tu w metrach, łatwo obliczyć, że udostępniony mi wynik pomiaru jest przesunięty o 1,9 cm na południe oraz o 7,7 cm na wschód względem wartości zapisanych w bazie. Łączna różnica, czyli odległość między tymi punktami, to 7,9 cm.

Jak wygląda sytuacja w przypadku podłączonego do Raspberry Pi modułu z układem LC29H(DA)? Zarejestrowałem kilkuminutową serię pomiarów, obejmującą 302 odczyty. Po naniesieniu ich wszystkich w programie QGIS punkty zlewają się w jedną „plamę”, oznaczoną na **rysunku 6** (na następnej stronie) literą C, mieszczącą się w kwadracie o boku 3 cm. Plama ta znajduje się około 11 cm od oficjalnych współrzędnych punktu osnowy (oznaczonego tu literą A) oraz 8 cm od pojedynczego pomiaru geodety (oznaczonego literą B).



Rysunek 5

Oczywiście opisany test nie jest całkowicie miarodajny, gdyż należałoby powtórzyć pomiary dla większej liczby punktów, a same odczyty rejestrować kilkakrotnie. Niemniej jednak w tym konkretnym przypadku współrzędne podawane przez pracujący w trybie RTK moduł LC29H(DA) różniły się od współrzędnych punktu osnowy o zaledwie 11 cm. Warto przy tym pamiętać, że sam punkt osnowy też ma ograniczoną dokładność. Zgodnie z obowiązującym rozporządzeniem nowo zakładane punkty osnowy poziomej powinny charakteryzować się średnim błędem położenia nie większym niż 0,07 m. Uwzględniając tę niedokładność punktu odniesienia, rzeczywista odchyłka mojego pomiaru może wynosić od około 4 cm do 18 cm.



Rysunek 6

Podsumowanie

Jeśli ktoś chce poeksperymentować z nawigacją RTK, gorąco polecam opisywany przeze mnie moduł. Trzeba jednak pamiętać, że wersja przedstawiona w artykule powstała głównie do celów testowych, więc nie zawsze warto wzorować się na niej zbyt dokładnie. Do zastosowań praktycznych lepiej sprawdzi się mocniejsze Raspberry Pi, gdyż

wiekowy już model 3 wprawdzie radzi sobie z QGIS, ale wydajność pozostawia wiele do życzenia. Również rozdzielczość użytego wyświetlacza, zaledwie 800×480 pikseli, jest zbyt niska do pracy z takim oprogramowaniem.

W przyszłości zamierzam przetestować wersję z Raspberry Pi 5, gdzie zamiast fizycznego ekranu prawdopodobnie użyję zdalnego dostępu do pulpitu za pomocą programu RealVNC Viewer.

Marian Gabrowski

marian kropka gabrowski @ gmail kropka com