

BIBLIOTEKA
POLSKIEGO KRÓTKOFALOWCA

68

KRZYSZTOF DĄBROWSKI
OE1KDA

SYSTEM LoRa

WIEDEN 2023



© Krzysztof Dąbrowski OE1KDA
Wiedeń 2023

Opracowanie niniejsze może być rozpowszechniane i kopiowane na zasadach niekomercyjnych w dowolnej postaci (elektronicznej, drukowanej itp.) i na dowolnych nośnikach lub w sieciach komputerowych pod warunkiem nie dokonywania w nim żadnych zmian i nie usuwania nazwiska autora. Na tych samych warunkach dozwolone jest tłumaczenie na języki obce i rozpowszechnianie tych tłumaczeń.

Na rozpowszechnianie na innych zasadach konieczne jest uzyskanie pisemnej zgody autora.

System LoRa

Krzysztof Dąbrowski OE1KDA

Wydanie 1
Wiedeń, lipiec 2023

Spis treści

Wstęp	6
1. System LoRa z rozpraszaniem widma	7
1.1. Znaczenie parametrów transmisji	14
1.2. Rozwiązania stacji	17
1.2.1. Eksperymenty dla początkujących	21
1.2.2. Dydaktyczny program nadajnika APRS w systemie „LoRa”	24
1.3. LoRa APRS	33
1.4. Wyposażenie stacji w systemie „LoRa”	39
2. Sieć systemu „LoRa MeshCom”	41
2.1. Przykładowe zastosowania sieci	43
2.2. Instalacja i uruchomienie oprogramowania	44
2.3. Androidowy program <i>MeshCom</i>	49
2.4. Wymiana komunikatów z innymi systemami	52
2.5. Elementy dodatkowe	52
2.6. Zestaw poleceń	54
2.7. Aktywność	55
3. Odbiór satelitów „LoRa”	56
4. Próbnik połączeń „LoRa”	60
Dodatek A. Instalacja <i>Pythona</i> i <i>ESPTOOL</i>	62
Literatura i adresy internetowe	63
Spis tomów „Biblioteki polskiego krótkofalowca”	65

Sommaire

Système LoRa

Préface	6
1. Système LoRa à spectre étalé	7
1.1. Les paramètres de LoRa et leurs relations	14
1.2. Constructions des postes	17
1.2.1. Expériences pour débutants	21
1.2.2. Logiciel didactique de poste APRS „LoRa”	24
1.3. LoRa APRS	33
1.4. Équipement des postes „LoRa”	39
2. Le réseau „LoRa MeshCom”	41
2.1. Exemples d'applications	43
2.2. Installation et démarrage de logiciel	44
2.3. Logiciel <i>MeshCom</i> pour Android	49
2.4. Échange de messages avec autres systèmes	52
2.5. Accessoires supplémentaires	52
2.6. Jeu d'instructions	54
2.7. Activité	55
3. La réception des satellites „LoRa”	56
4. Testeur des liaisons „LoRa”	60
Annexe A. Installation des logiciels <i>Python</i> et <i>ESPTOOL</i>	62
Bibliographie et les pages web	63
Liste des volumes de la „Bibliothèque de radioamateur polonais”	65

Wstęp

Tematem aktualnym i coraz bardziej zyskującym na znaczeniu nie tylko w krótkofalarstwie jest radiowy system nadawania danych oparty o transmisję z rozpraszaniem widma sygnału – *LoRa*. Jest on wykorzystywany m.in. do transmisji danych telemetrycznych albo do komunikacji w lokalnych lub bardziej rozległych sieciach. Rozpraszanie widma powoduje zwiększenie szerokości pasma sygnału nadawanego w stosunku do sygnału modulującego. Stosunek szerokości pasm sygnału nadawanego i modulującego określa zysk systemowy, dzięki któremu możliwe jest stosowanie niskich lub bardzo niskich mocy nadawania. Leżą one przeważnie nawet poniżej 100 mW, a w sytuacjach w których nie zapewniają one pożądanego zasięgu pomocne stają się sieci o mniejszym lub większym zagęszczeniu stacji (ich oczek). W sieci krótkofalarskiej wykorzystywane jest oprogramowanie *MeshCom* w obecnych wydaniach różniące się od dawniejszych opartych na rozwiązaniu firmy *Meshtastic*. Jego instalacja jest wyraźnie mniej skomplikowana, a dalszy rozwój pozostaje wyłącznie w rękach krótkofalowców. Natomiast dla szerokich rzesz użytkowników powstały sieci Internetu Przedmiotów (IoT) typu *LoraWAN* itp. W odróżnieniu od użytkowników sieci powszechnie dostępnych krótkofalowców nie obowiązują ani procentowe ograniczenia czasu nadawania, ani ograniczenia liczby komunikatów nadawanych w ciągu doby ani ograniczenia mocy na poziomach poniżej warunków licencji.

Urządzenia nadawczo-odbiorcze *LoRa* są nie tylko tanie, ale też na tyle uniwersalne, że mogą być stosowane w innych systemach transmisji. W połączeniu z odpowiednim oprogramowaniem możliwy jest odbiór danych telemetrycznych z satelitów *LoRa* albo z balonów naukowo-badawczych i przekazywanie ich do sieci *TinyGS*. Krótkofalowcy wykorzystują tę technikę m.in. w transmisji danych pozycyjnych APRS. Komunikaty APRS też zresztą mogą być uzupełniane o dane telemetryczne. Do wyboru jest szeroka gama modeli różniących się wyposażeniem: wyświetlaczami, wyposażeniem w odbiorniki GPS, w pojemniki dla akumulatorów, przyciski itp.

Sprawą istotną dla użytkowników systemu *LoRa*, a tym bardziej dla eksperymentatorów jest należyte zrozumienie znaczenia parametrów transmisji, ich wzajemnych zależności i ich wpływu na przepustowość kanału albo na uzyskiwane zasięgi i niezawodność transmisji.

Krótkofalowcy austriaccy pracują od dłuższego czasu nad rozwiązaniami dla sieci radiowych *MeshCom*, dla transmisji danych APRS i do innych zastosowań. W stosunkowo szybkim tempie powstają nowe wersje oprogramowania i rozwiązania dla kolejnych modułów transmisyjnych *LoRa*. Temat transmisji w systemie *LoRa* była przedstawiany w poprzednim tomie *Biblioteki*, ale tempo rozwoju spowodowało konieczność poświęcenia *LoRa* oddzielnego i całkowicie nowego tomu. Z inicjatywy OE1KBC i kilku innych osób został powołany Instytut wiedzy obywatelskiej (*icssw.org*) zajmujący się aktualnie opracowaniami na polu sieci *MeshCom*, a w przyszłości także innymi nowoczesnymi tematami.

Dla programistów i konstruktorów opracowujących nowe rozwiązania przydatny może być odbiornik kontrolny systemu *LoRa* pracujący na modułach Heltec V2 lub T-Lora. Jest on sterowany przez program androidowy korzystający ze złącza *Bluetooth*.

Niskie moce nadawania i stosunkowo mały pobór prądu przez moduły transmisyjne pozwala na zasilanie urządzeń z akumulatorów. Dzięki temu, że wykorzystywane w komunikacji komputery lub telefony androidowe mogą być również przez dłuższy czas zasilane z wewnętrznego akumulatora sieć *MeshCom* jest dobrze predystynowana do wykorzystania w łącznościach awaryjnych i ratunkowych przy przerwaniu dopływu energii elektrycznej w sieci.

Krzysztof Dąbrowski OE1KDA

Wiedeń

9 lipca 2023

1. System „LoRa” z rozpraszaniem widma

Oznaczenie „LoRa” jest skrótem od określenia „Long Range” mającego oznaczać telemetrię daleko-sięzną, przy czym o ile w porównaniu z rozpowszechnionymi systemami transmisji danych pomiarowych na dystansach kilku lub kilkudziesięciu metrów zasięgi dochodzące w korzystnych warunkach do 20 lub więcej km można uznać za dalekie. W oczach krótkofalowców nie są to oczywiście żadne DX-y, ale przecież nie o to najbardziej chodzi.

System „LoRa” korzystający z transmisji z rozpraszaniem widma sygnału (ang. *spread spectrum*) został opracowany w 2013 roku we francuskiej firmie „Cycleo” wykupionej następnie przez firmę „Semtech”. Obecnie na rynkach dostępne są moduły dla pasm 433 i 868 MHz, a w USA także pracujące w paśmie 915 MHz. Do zastosowań krótkofalarskich nadają się oczywiście moduły na zakres 433 MHz (typu RFM98W-433S2 itp.), a do zastosowań nie wymagających licencji amatorskiej – moduły dla pasma 868 MHz (RFM95W-868S2 itp.). W paśmie 868 MHz w zależności od podzakresu obowiązuje ograniczenie mocy do 10 lub 25 mW i nadawania do 0,1% lub 1% czasu pracy – liczonego w skali godzinowej.

Pomimo niskich mocy nadawania – od kilkunastu do maksimum 120 mW – osiągalne są, dzięki zastosowaniu transmisji z rozpraszaniem widma, stosunkowo dalekie zasięgi. W porównaniu z transmisją z kluczkowaniem FSK lub GMSK są one około dziesięciokrotnie dalsze. Zależny od współczynnika rozpraszania widma zysk systemowy wynosi tutaj około 20 dB.

Niskie moce nadawania pozwalają na bateryjne zasilanie urządzeń. Obecnie produkowane moduły pobierają przy nadawaniu od 20 mA przy mocy 5 mW do 120 mA przy mocach rzędu 120 mW.

Krótkofalowcy mogą, w odróżnieniu od użytkowników nielicencjonowanych korzystać z dowolnych anten zewnętrznych i nie obowiązują ich ograniczenia czasu nadawania, a moce są ograniczone jedynie warunkami licencji. Dozwolone jest korzystanie z dodatkowych wzmacniaczy mocy.

Dopuszczalne szybkości transmisji wynoszą wprawdzie od 180 bit/s do 37,5 kbit/s, ale dla uzyskania możliwie największej czułości stosowane są szybkości z dolnego zakresu, co praktycznie jest w pełni wystarczające ponieważ ilość transmitowanych danych jest raczej niewielka. Dla minimalnych szybkości osiągnąć czułości -148 dBm, natomiast przykładowo dla 1200 bit/s już tylko -119 dBm. Możliwa jest też automatyczna zmiana szybkości transmisji w zależności od jakości łącza (ang. *adaptive data rate* – ADR).

Maksymalna długość transmitowanego pakietu wynosi 256 bajtów, włącznie z danymi korekcyjnymi FEC (korekcji wyprzedzającej). Jest ona ograniczona długością rejestru przesuwonego zawartego w obwodzie scalonym modemu.

Współczynnik rozpraszania widma (stosunek szerokości pasma sygnału nadawanego do szerokości widma kanału podstawowego) leży w granicach 6 – 12, a pasmo sygnału transmitowanego zajmuje od 7,8 do 500 kHz. Najczęściej stosowane jest pasmo 125 lub 250 kHz.

Moduły radiowe „LoRa” są stosowane w profesjonalnych systemach telemetrycznych, w zdalnym odczycie liczników elektrycznych i innych przyrządów, w układach zdalnego sterowania, w systemach bezpieczeństwa, alarmowych, w śledzeniu przesyłek, bagażu na lotniskach albo innych obiektów itp.

Do spraw problematycznych należy wprawdzie kwestia zabezpieczenia transmitowanych danych, ale w zastosowaniach amatorskich sprawa ta nie ma większego znaczenia.

Moduły „LoRa” pozwalają dodatkowo na transmisję danych z kluczkowaniem FSK, GFSK, MSK, GMSK i OOK. Możliwe jest więc wykorzystanie ich w innych systemach transmisji i sieciach, śledzenie sond meteorologicznych itp.

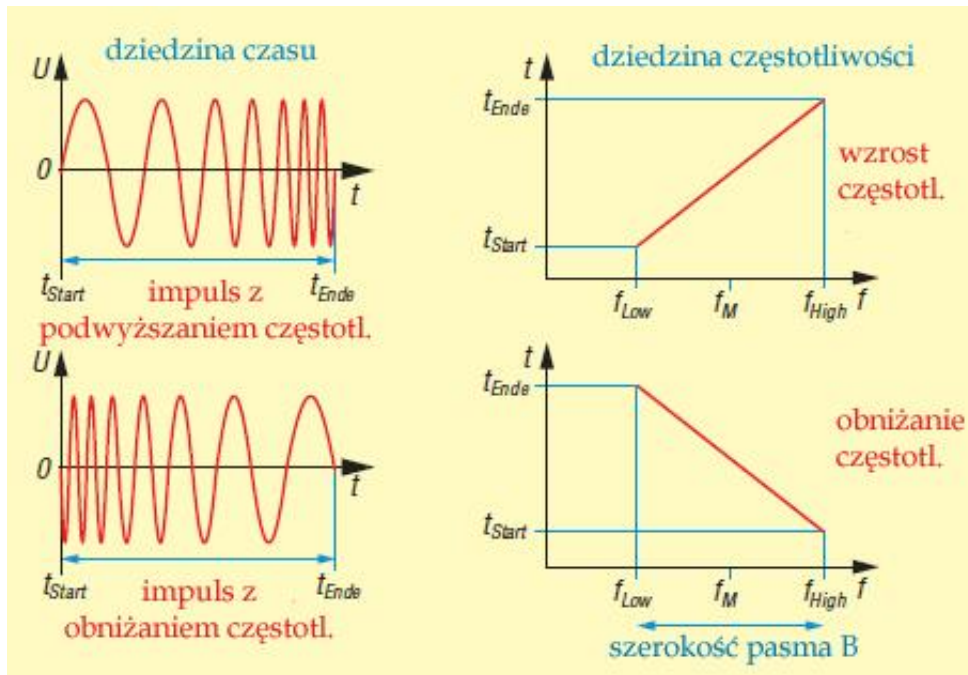
Do najczęściej stosowanych metod rozpraszania widma sygnału należą kluczkowanie częstotliwości (ang. *frequency hopping* – FH), kluczkowanie fazy kodem losowym (ang. *direct sequence spread spectrum* – DSSS) i linowa zmiana częstotliwości (ang. *chirp*). Do rozpraszania widma stosowane są kody pseudoprzypadkowe. W najprostszym przypadku są one generowane za pomocą rejestrów przesuwanych z odpowiednio dobranymi sprzężeniami zwrotnymi, ale mogą być także generowane programowo. Istnieją całe rodziny takich kodów, a nowe kody mogą być też generowane przez kombinację istniejących. Teoria kodów pseudoprzypadkowych jest dość rozbudowana i skomplikowana.

Zysk systemowy jest zależny od stopnia rozproszenia widma czyli stosunku szerokości pasma sygnału rozproszonego do sygnału podstawowego. Sygnał o rozproszonym widmie jest odbieralny przez klasyczne odbiorniki wąskopasmowe jako szum. Dodaje się on do szumów pochodzących z innych źródeł.

Przy wielu zachodzących na siebie sygnałach o rozproszonym w dowolny sposób widmie selektywny wybór możliwy jest dzięki stosowaniu różnych kodów. Dla prawidłowego odbioru konieczna jest oprócz zgodności kodów także zgodność ich faz (w sygnale odbieranym i w kodzie generowanym w odbiorniku), a także zgodność ich częstotliwości zegarowych. Sprawa zapewnienia synchronizacji należy do najważniejszych problemów w komunikacji z rozpraszaniem widma sygnału.

Oprócz radiokomunikacji transmisja z rozpraszaniem widma jest stosowana m.in. w zdalnym sterowaniu i w technice pomiarowej. Pomiar przesunięcia kodu odbieranego w stosunku do nadawanego pozwala, zwłaszcza przy użyciu odpowiednio długich kodów, na dokładne pomiary odległości.

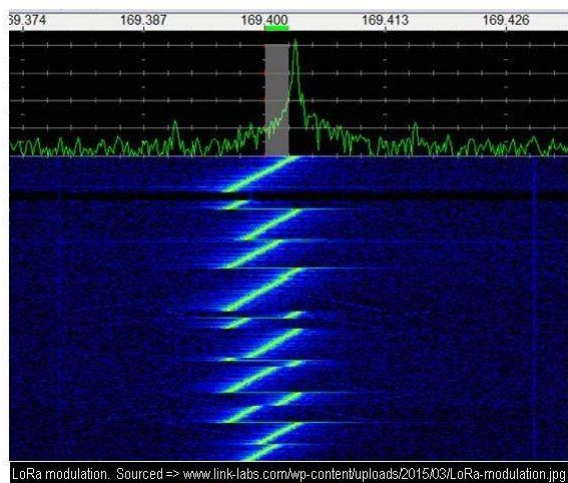
W modułach „LoRa” stosowany jest uproszczony system liniowej zmiany (przemiatania) częstotliwości (ang. *chirp* – CSS) oparty na standardzie IEEE 802.15.4.



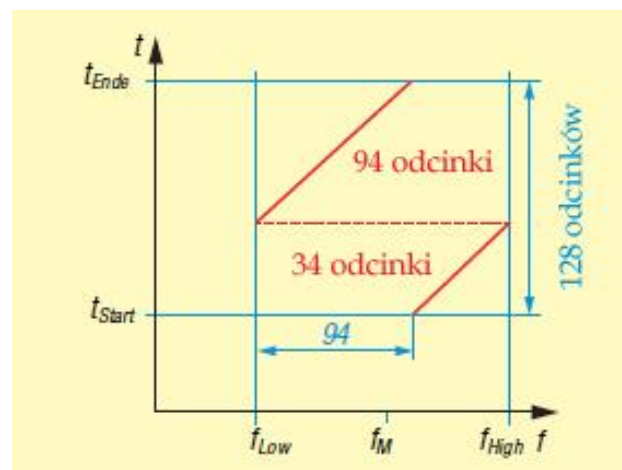
Rys. 1.1. Przebieg sygnałów modulowanych liniowo w dziedzinie czasu i częstotliwości (źródło: *Funkamateurl* 11/2021)



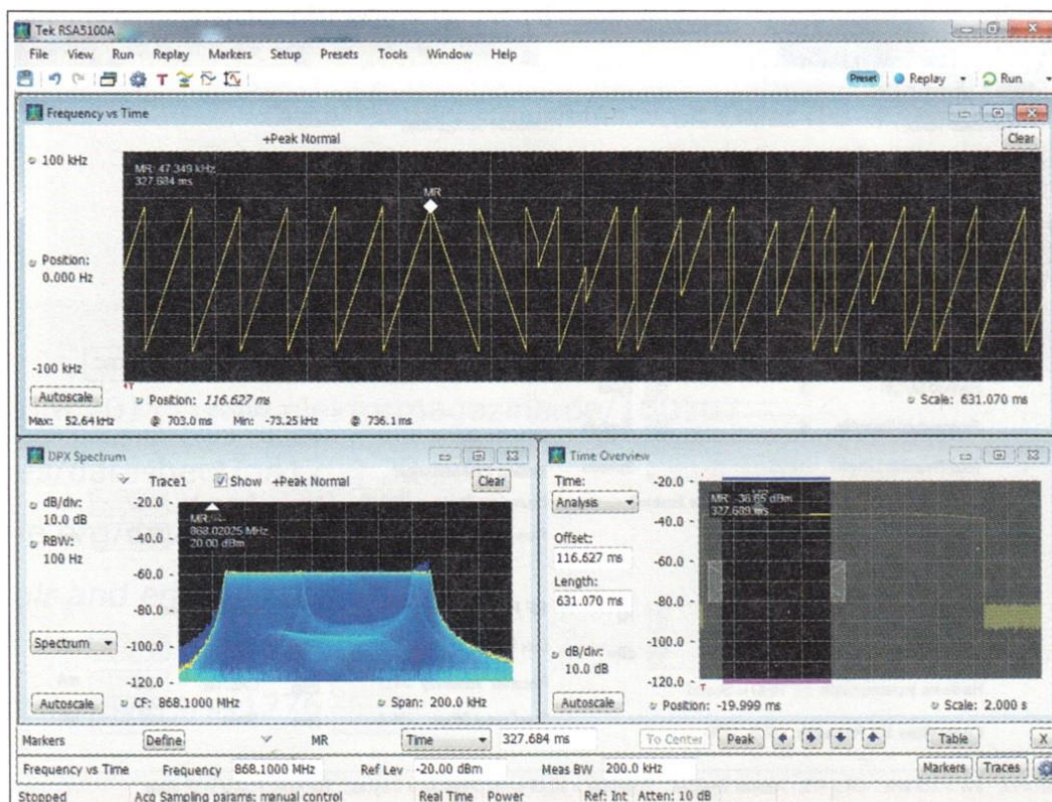
Rys. 1.2. Transmisja sygnału o rozproszonym widmie. Sygnał użytkowy jest po stronie nadawczej rozpraszany, a po stronie odbiorczej skupiany w identyczny sposób i przy użyciu identycznego kodu dla odzyskania danych w paśmie podstawowym. Ewentualnie odebrane sygnały wąskopasmowe znajdujące się w tym samym paśmie ulegają w odbiorniku rozproszeniu i nie powodują zakłóceń



Rys. 1.3a. Przebieg sygnału systemu „LoRa” w funkcji czasu i częstotliwości
(źródło: dokumentacja LoRa)



Rys. 1.3b. Zasada kodowania na przykładzie symbolu o wartości dziesiętnej 94
(źródło: Funkamateurl1/2021)



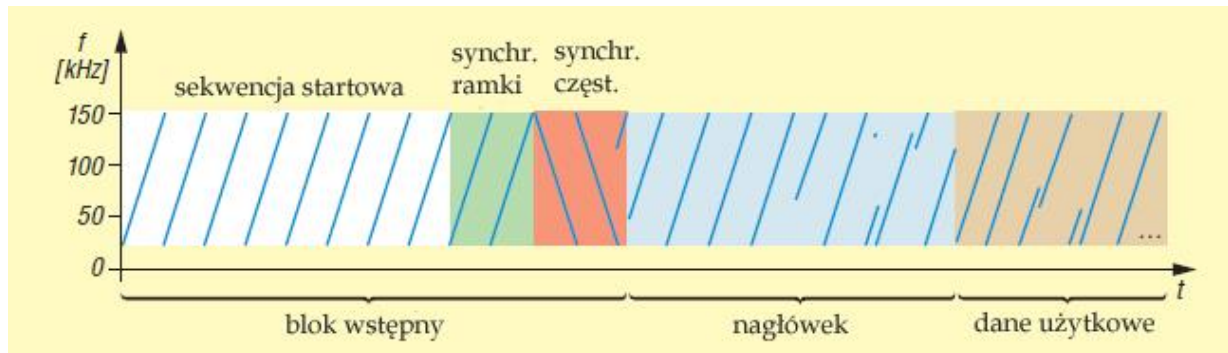
Rys. 1.4. Sygnał systemu LoRa zmierzony analizatorem widma RSA5106 firmy Tektronix
(źródło: Elektor maj-czerwiec 2017)

Na ilustracji 1.4 przedstawiona jest dokonana w laboratorium *Elektora* analiza sygnału systemu „LoRa” dla pasma o szerokości 125 kHz, współczynnika rozpraszania 12, FEC 4/5 i dewiacji również 125 kHz. W bloku synchronizacyjnym przebiegi są odwrócone: częstotliwość maleje, a nie wzrasta w funkcji czasu. Blok synchronizacyjny składa się z 8 symboli, przy czym do synchronizacji odbiornika wystarczają 4 z nich.

Bitów zgrupowane są w symbole 12-bitowe – przyjmują więc wartości od 0 do 4095. Czas trwania symbolu wynosi 32,8 msek. Przepustowość brutto równa jest 366,2 bit/s, co przy FEC 4/5 daje przepustowość netto 292,9 bit/s.

Maksymalna dewiacja 125 kHz jest podzielona na 212 odcinków: $125000/4096 = 30,52$ Hz. Dewiacja chwilowa równa jest wartości symbolu $\times 30,52$ Hz. Transmisja 8 bajtów użytkowych trwa 925 msek.

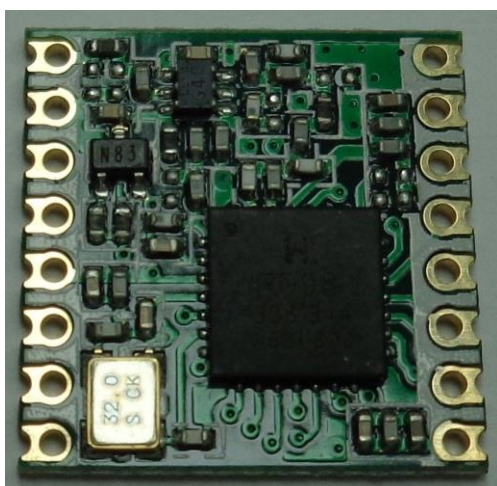
Przeważnie system jest stosowany do niezbyt częstych transmisji niewielkich ilości danych. Transmisja z dużymi szybkościami nie jest nieopłacalna. Oprócz zastosowań naziemnych w praktyce amatorskiej interesująca jest transmisja danych z balonów. Dzięki znacznym wysokościom lotu uzyskuje się duże zasięgi stacji. Inną interesującą dziedziną może być śledzenie zwierząt wyposażonych w nadajniki.



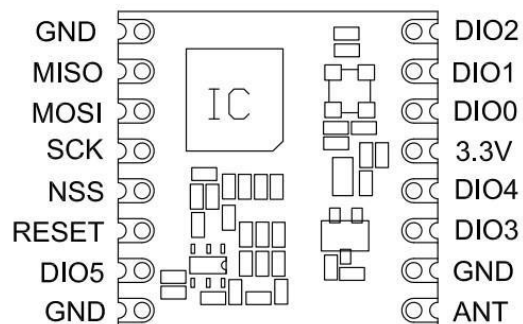
Rys. 1.5. Orientacyjna struktura pakietu „LoRa” (źródło: *Funkamateur* 11/2021)

Moduły RFM98W i podobne komunikują się z mikrokomputerem za pomocą złącza SPI (*Serial Peripheral Interface*), w którego skład wchodzi sygnały MOSI (*Master Output Slave Input*), MISO (*Master Input Slave Output*), SCK (sygnał zegarowy) i SS (*Slave Select*). Sygnał wyboru urządzenia SS musi być połączony z wejściem NSS modułu „LoRa”. Jako wyjście selekcyjne po stronie *Arduino* może służyć dowolne wyjście logiczne.

Kontakty DIO0 – DIO5 modułu są wykorzystywane przez bibliotekę LMIC do różnych celów j.np. informacji o pracy urządzenia. Moduły pracujące w paśmie 433 MHz zawierają przeważnie nadajniki-odbiorniki SX1278 firmy „Semtech”. Ich syntezer pokrywa zakres częstotliwości 137 – 525 MHz. Podobny do niego SX1268 pokrywa zakres 410 – 525 MHz. W zakresie 860 – 1000 MHz pracują układy SX1272/1273, a SX1276 pokrywa zakres 433 MHz oraz pasma przemysłowe 868 i 915 MHz. Moduły SX1280/1281 pracują w paśmie 2,4 GHz.



Fot. 1.6. Moduł RFM dla pasma 433 MHz...



Rys. 1.7. ...i jego wyprowadzenia

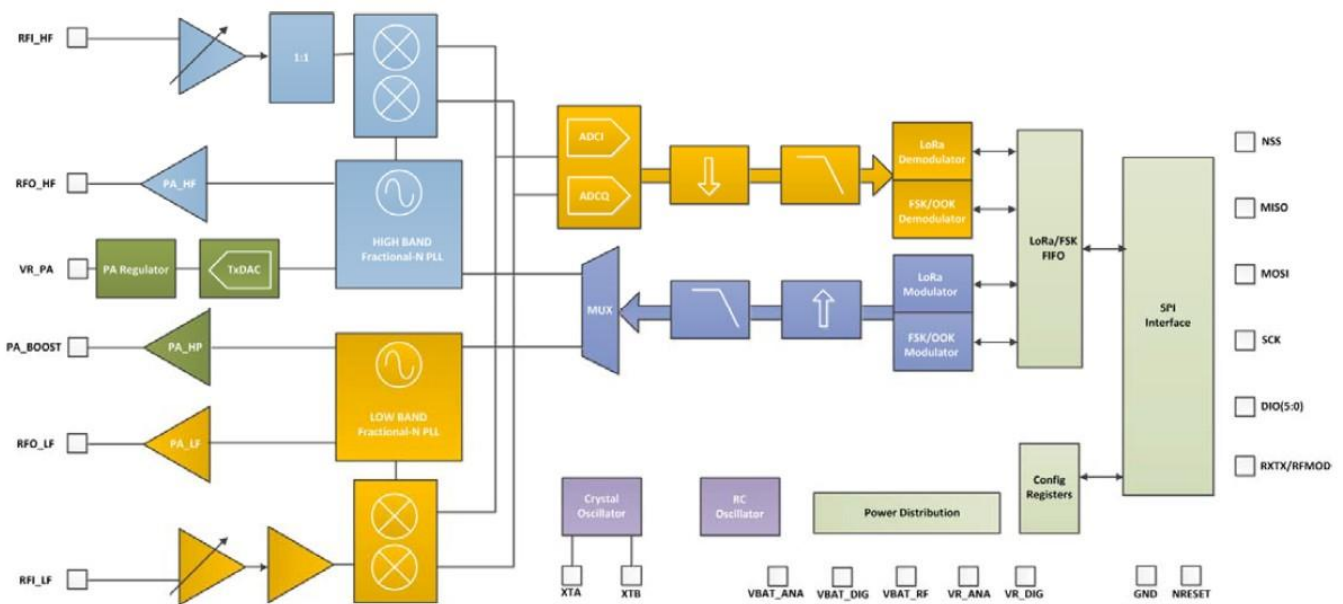
Tabela 1.1

Minimalny stosunek sygnału do szumu w zależności od współczynnika rozpraszania dla modułu SX1278 (GMSK dla porównania)

Współcz. rozpr.	Stos. sygn./szum [dB]	Współcz. rozpr.	Stos. sygn./szum [dB]
6	-5,0	10	-15,0
7	-7,5	11	-17,5
8	-10,0	12	-20,0
9	-12,5	GMSK	9

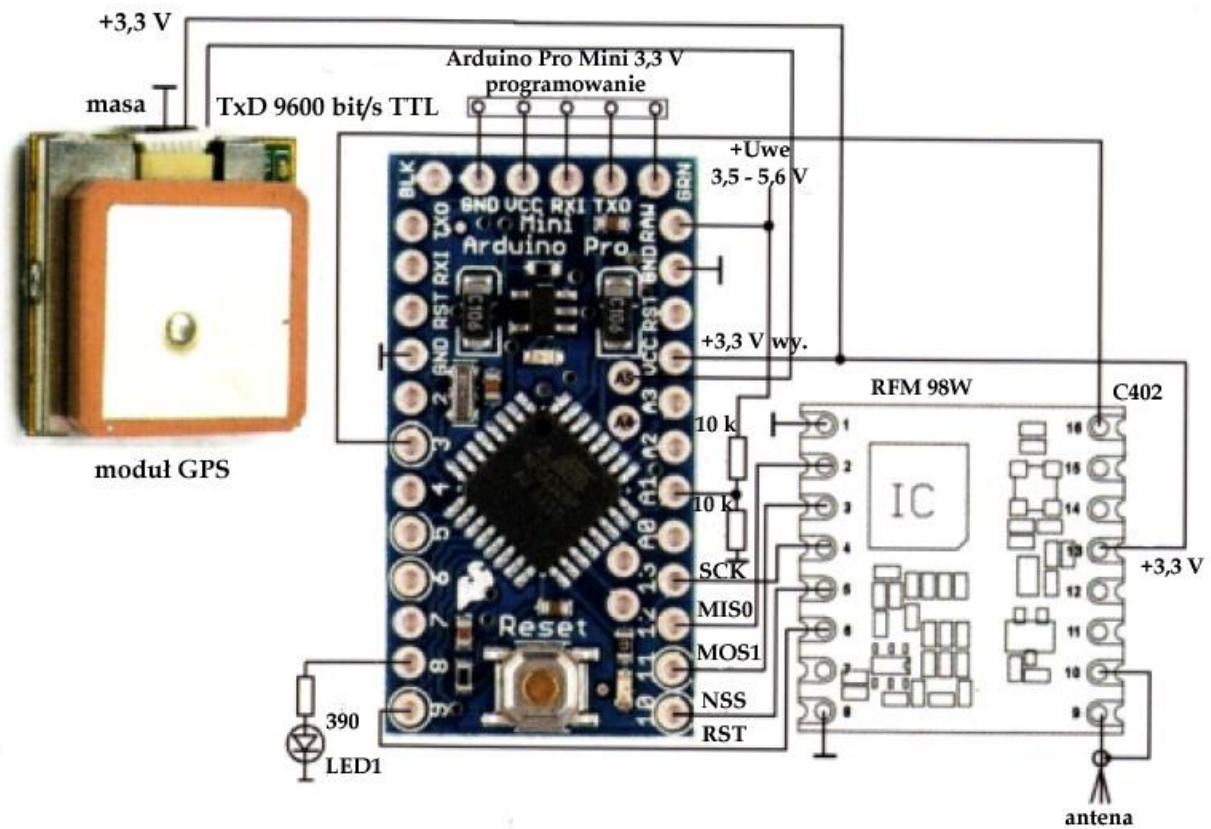
Czułość odbiornika oblicza się ze wzoru:

$ERX = 10 \log(B) + WS + SSS - 174 \text{ dBm}$, gdzie B jest szerokością pasma sygnału w Hz (dla 125 kHz $10 \log(125000) = 51$, WS – współczynnikiem szumów odbiornika (równym 6 w paśmie 70 cm), a SSS – współczynnikiem sygnał/szum według tabeli.

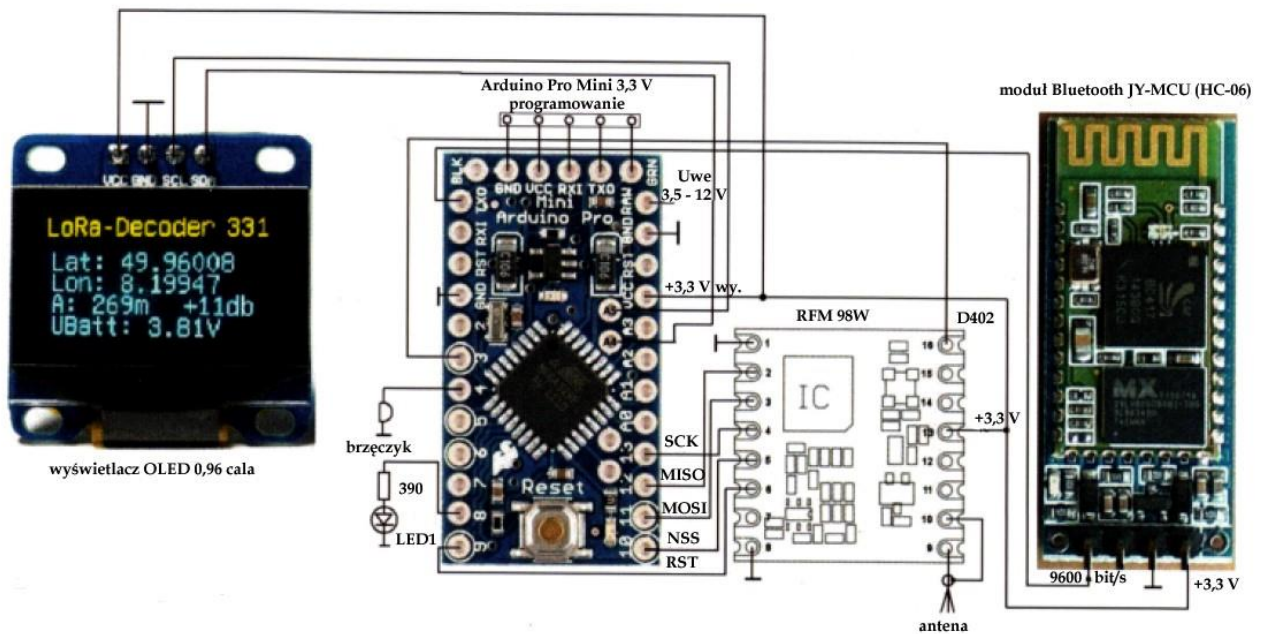


Rys. 1.8. Schemat blokowy modułu z serii SX

W przedstawionym na ilustracji 1.9 nadajniku APRS zastosowano mikrokomputer *Arduino Mini* zasilany napięciem 3,3 V, co ułatwia połączenie obu części, a także modułu odbiornika GPS. Oprócz danych APRS nadawana jest informacja o napięciu zasilania. Napięcie to jest doprowadzone do wejścia A1 *Arduina* przez dzielnik 1:2 złożony z dwóch oporników 10 kΩ. Krótkofalowcy eksperymentowali również z transmisją danych meteorologicznych: ciśnienia atmosferycznego, temperatury i wilgotności. W pasującym do tego odbiorniku APRS zastosowano wyświetlacz OLED połączony z mikrokomputerem za pomocą magistrali I2C (sygnał zegarowy SCL pochodzi z wyjścia A5 *Arduina*, a sygnał danych SDA – z wyjścia A4). Moduł *Bluetooth* służy do połączenia odbiornika z komputerami lub telefonami komórkowymi. Dla telefonów i komputerów androidowych dostępne są m.in. programy *BluTerm*, *W2APRS* i *Lotus Map Pro*. Pierwszy z nich jest programem terminalowym pozwalającym na odczyt danych w postaci tekstowej, a następne umożliwiają przedstawienie pozycji odbieranej stacji na mapie.



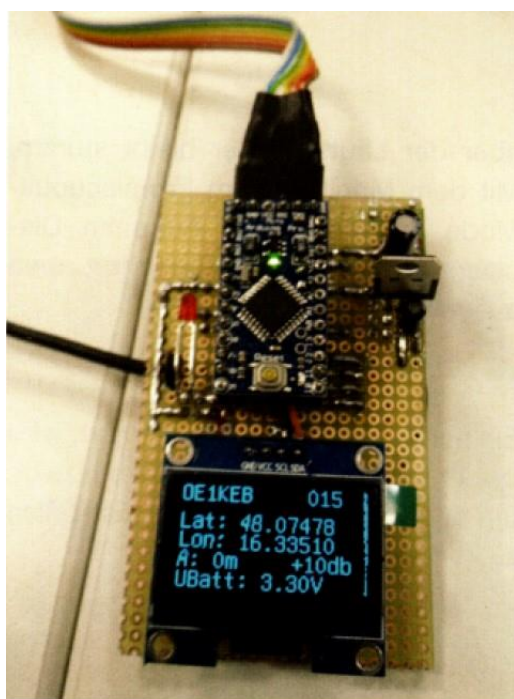
Rys. 1.9. Nadajnik APRS z modulem „LoRa” (źródło: [1.4] i QSP 4/2017)



Rys. 1.10. Odbiornik APRS (źródło: [1.4] i QSP 4/2017)

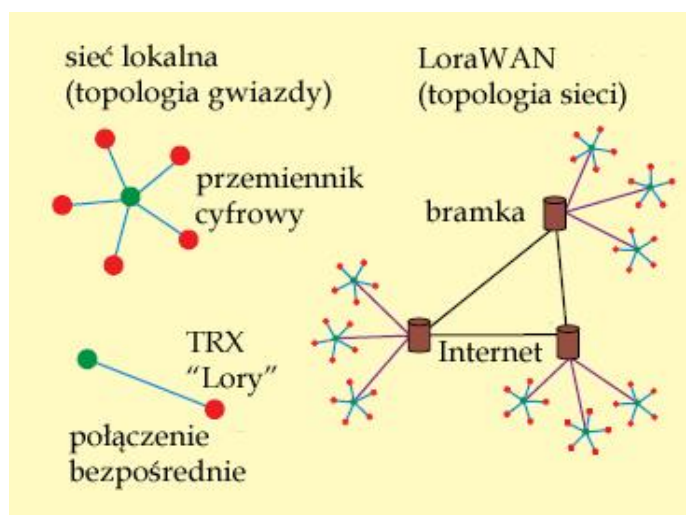


Fot. 1.11. Transmisja danych meteorologicznych: ciśnienia, temperatury, wilgotności względnej i wysokości zmierzonych za pomocą czujnika BME280 firmy „Bosch”. Dodatkowo wyświetlana jest informacja o sile odbieranego sygnału (odstępie poziomu sygnału od szumu)



Fot. 1.12. Odbiornik konstrukcji OE1KEB (źródło: QSP 4/2017)

Rys. 1.13. Możliwe rodzaje połączeń (topologie) w systemie „Lora” (źródło: Funkamateur 11/2021)



Wyposażony w SX1276/1278 moduł z ilustracji 1.14 dysponuje mocą wyjściową 100 mW (pobór prądu wynosi wówczas 120 mA, przy 20 mW – 29 mA), charakteryzuje się wysoką odpornością na blokowanie, czułością -148 dBm i niskim poborem prądu. W trakcie odbioru pobór prądu wynosi około 11 – 12 mA. Moduł jest kompatybilny z płytkami *Arduino UNO*, *MEGA*, *DUE* i *Leonardo* zasilanymi napięciem 3,3 V i 5 V.

Oprócz transmisji w systemie „Lora” do dyspozycji są modulacje FSK, GFSK, MSK, GMSK i kluczowanie amplitudy OOK (od zera do pełnej mocy jak w telegrafii). Dzięki temu możliwy jest odbiór sygnali pracujących w innych sieciach lub systemach, przykładowo sond radiowych wysyłanych na pokładach balonów meteorologicznych. Syntezator częstotliwości pozwala na nastawienie częstotliwości pracy z rozdzielczością 61 Hz. Przy szerokościach pasma sygnału rozproszonego poniżej 125 kHz kwarcowy generator sterujący należy zastąpić stabilizowanym temperaturowo – TCXO.

Płytkę ma wymiary 62 x 43 x 23 mm i masę 22 g.



Fot. 1.14. Moduł nadawczo-odbiorczy z RFM98W-433S2 dla *Arduino* (źródło: *exp-tech.de*)

1.1. Znaczenie parametrów transmisji

Zmiany parametrów transmisji w systemie LoRa nie pozostają bez wpływu na szybkość transmisji i zasięg odbioru stacji. Oczywiście stacja odbiorcza musi nie tylko być nastrojona na częstotliwość pracy nadajnika, ale mieć także nastawione identycznie pozostałe parametry” szerokość pasma (BW), współczynnik rozpraszania (SF) i stopę danych korekcyjnych (CR). Stopa danych korekcyjnych jest parametrem mniej krytycznym i możliwy jest prawidłowy odbiór mimo różnych wartości ustawionym w nadajniku i odbiorniku.

Tabela 1.1.1

Porównanie możliwych parametrów transmisji dla różnych modułów nadawczo-odbiorczych RFM

Typ	Zakres [MHz]	Współczynnik rozpraszania	Pasma [kHz]	Szybkość transmisji netto	Czułość [dBm]
RFM95W	868/915	6 – 12	125 – 500	0,293 – 37,5 kbit/s	-111 – -136
RFM96W/ 98W	433/470	6 – 12	62,5 – 500	0,1465 – 37,5 kbit/s	-112 – -140
RFM98W	169	6 – 12	31,25 – 125	73,24 – 9375 bit/s	-118 – -143

Rozpraszanie widma sygnału powoduje zwiększenie odporności transmisji na wpływ szumów. W zależności od modułu do dyspozycji są następujące szerokości pasma (BW) rozproszonego sygnału (dewiacje): 31,25 kHz, 41,7 kHz, 62,5 kHz, 125 kHz, 250 kHz i 500 kHz. Wybór niższych szerokości pasma powoduje przedłużenie czasu transmisji. Pasma poniżej 125 kHz wymagają wyposażenia stacji w TCXO ze względu na wymaganą większą stabilność częstotliwości pracy. Dla sieci LoRaWAN przyjęto standardowo pasmo 125 kHz (w sieciach krótkofalarskich także 250 kHz). W paśmie 2,4 GHz stosowane są szerokości pasma 500 kHz – 1,6 MHz. Ogólnie rzecz biorąc węższe pasmo sygnału oznacza zmniejszenie szybkości transmisji i zwiększenie zasięgu stacji. Szersze pasmo oznacza zwiększenie szybkości transmisji i jednocześnie zmniejszenie zasięgu.

Również zmiana wartości współczynnika rozpraszania (SF) określająca liczbę symboli (ang. *chirps*) reprezentujących każdy z bitów danych. Przy niskich współczynnikach rozpraszania wzrasta w efekcie liczba transmitowanych bitów danych czyli szybkość transmisji. Jednocześnie powoduje to zmniejszenie czułości strony odbiorczej, co oznacza zwiększenie prawdopodobieństwa utraty danych w kanale transmisyjnym (większą zawodność transmisji). Większe współczynniki oznaczają z kolei zmniejszenie szybkości transmisji danych użytkowych, ale daje to wzrost czułości odbiorników i wzrost niezawod-

ności transmisji. Skutkuje to jednocześnie przedłużeniem czasu transmisji i wzrostem średniego poboru mocy przez nadajnik, co może być istotne w niektórych zastosowaniach (czujnikach zasilanych bateryjnie i umieszczonych w miejscach trudno dostępnych). Tak więc i tutaj wszystko sprowadza się do wyboru między szybkością i niezawodnością transmisji (czułością odbiornika). Dla zakresów częstotliwości poniżej 1 GHz dopuszczalne są współczynniki rozpraszania SF6 – SF12. W praktyce najniższym stosowanym współczynnikiem jest SF7 gdyż SF6 wymaga użycia TCXO jako generatora sterującego. Oznaczają one potęgę dwójki dająca liczbę symboli reprezentujących każdy z bitów. Przykładowo SF7 oznacza 2^7 czyli 128 symboli, a SF11 – 2^{11} czyli 2048 symboli. Przy współczynniku rozpraszania SF7 transmisja 64 bajtów danych trwa około 120 ms, a przy SF11 – ponad sekundę (patrz.tab. 1.1.2).

Tabela 1.1.2

Orientacyjny czas transmisji bloku 64 bajtów danych w zależności od współczynnika rozpraszania przy paśmie 125 kHz

Współczynnik rozpraszania	Przybliżony czas transmisji [ms]
SF7	120
SF8	220
SF9	400
SF10	700
SF11	1300
SF12	2400

Szerokość pasma i współczynnik rozpraszania definiują fizyczne parametry modulacji. Natomiast stopa danych korekcyjnych CR określa stopień korekcji wyprzedzającej (FEC) pozwalającej na skorygowanie błędnie odebranych danych dzięki danym nadmiarowym (redundantnym) zawartym w nadawnych pakietach. Dane nadmiarowe są obliczane po stronie nadawczej i dodawane do danych użytkowych. Większy udział danych korekcyjnych w całości pozwala na skorygowanie większej liczby przekłamań, zwiększenie niezawodności i zasięgu, ale jednocześnie oznacza, że dane użytkowe zajmują mniejszą część pakietu, czyli zmniejszenie efektywnej szybkości transmisji. Standard *LoRa* dopuszcza cztery możliwe stopy korekcji 4/5, 4/6, 4/7 i 4/8. Przy wartości 4/5 udział danych użytkowych w całości wynosi 4/5 czyli 80%, a przy następnych odpowiednio mniej aż do 50% przy stopie 4/8. Wybierając stopę korekcji warto pamiętać, że w wielu przypadkach korzystniejsze może być dopuszczenie niewielkiej liczby przekłamaných pakietów zamiast stałego znacznego obniżenia skutecznej szybkości transmisji. Zmiany stopnia korekcji nie wpływają na pozostałe właściwości kanału co oznacza, że odbiorniki skonfigurowane na stopę różną od nadawczej mogą go w dalszym ciągu prawidłowo dekodować.

Tabela 1.1.3

Orientacyjne wartości czułości odbiornika i szybkości transmisji w kanale *LoRa* w zależności od parametrów transmisji dla szerokości pasma 125 kHz wg [1.1.1]

Współczynnik rozpraszania	Stopa korekcji	Czułość [dBm]	Szybkość transmisji netto
7	4/5	-124,5	5,47 kbit/s
7	4/6	-124,5	4,56 kbit/s
7	4/7	-124,5	3,91 kbit/s
7	4/8	-124,5	3,42 kbit/s
8	4/5	-127,0	3,13 kbit/s
8	4/6	-127,0	2,60 kbit/s
8	4/7	-127,0	2,23 kbit/s
8	4/8	-127,0	1,95 kbit/s
9	4/5	-129,5	1,76 kbit/s
9	4/6	-129,5	1,46 kbit/s
9	4/7	-129,5	1,26 kbit/s

9	4/8	-129,5	1,10 kbit/s
10	4/5	-132,0	977 bit/s
10	4/6	-132,0	814 bit/s
10	4/7	-132,0	698 bit/s
10	4/8	-132,0	610 bit/s
11	4/5	-134,5	537 bit/s
11	4/6	-134,5	448 bit/s
11	4/7	-134,5	384 bit/s
11	4/8	-134,5	336 bit/s
12	4/5	-137,0	293 bit/s
12	4/6	-137,0	244 bit/s
12	4/7	-137,0	209 bit/s
12	4/8	-137,0	183 bit/s
11 (250 kHz)	4/6	-131,5	895 bit/s

Przy podwojeniu szerokości pasma ze 125 kHz do 250 kHz czułość zmniejsza się o 3 dBm, a szybkość transmisji wzrasta dwukrotnie w stosunku do podanych w tabeli dla każdej z kombinacji parametrów. Wybór pasma 500 kHz powoduje odpowiednio zmianę czterokrotną szybkości transmisji i czułości o 6 dBm.

W tabeli kolorem żółtym zaznaczono kombinacje parametrów stosowane w transmisjach APRS w Austrii i niektórych innych krajach zachodnich (12; 4/5) i w Polsce (9; 4/7). Jak wynika z tabeli kombinacja ze współczynnikiem rozpraszania 12 oznacza większą czułość, zasięgi i niezawodność transmisji kosztem przedłużenia czasu transmisji komunikatu APRS do około 3 sekund (szybkość transmisji jest zbliżona do stosowanej w tradycyjnym wariacie na falach krótkich). Natomiast kombinacja ze współczynnikiem rozpraszania 9 oznacza prawie ośmiokrotne pogorszenie czułości (a co za tym idzie zasięgu i niezawodności transmisji), ale przy szybkości transmisji zbliżonej do stosowanych w tradycyjnym systemie APRS 1200 bit/s daje w przybliżeniu czterokrotne skrócenie jej czasu.

W odróżnieniu od zainstalowanych w miejscach niedostępnych czujników telemetrycznych i innych minimalizacja poboru energii w stacjach amatorskich ma mniejsze znaczenie. Bramki radiowo-internetowe są przeważnie zainstalowane w stałych lokalizacjach i mogą być zasilane z sieci energetycznej a nadajniki danych pozycyjnych przenoszone lub przewożone w samochodzie są łatwo dostępne i dzięki temu nie ma trudności w naładowaniu ich akumulatorów lub wymianie baterii. Dużo większe oszczędności można osiągnąć dopasowując odstęp między transmisjami do szybkości poruszania się stacji – czyli przez ograniczenie liczby transmisji do sensownego minimum (tzw. inteligentna transmisja – ang. *smart beaconing*). Zapobiega to również nadmiernemu zatłoczeniu kanału.

W sieciach *MeshCom* w Austrii stosowane jest obecnie pasmo 250 kHz (poprzednio było to 125 kHz), współczynnik rozpraszania 11 i stopa korekcji 4/6 (ostatnia linia w tabeli z różowym tłem). Jak widać również i w tym przypadku położono nacisk na uzyskanie możliwie dużej czułości i zasięgów. W Niemczech stosowany jest częściowo również współczynnik 10.

Wybór częstotliwości pracy w paśmie 70 cm oznacza, że krótkofalowców nie obowiązują ograniczenia mocy nadawania i procentowego udziału czasu nadawania w całości czasu pracy wymagane od stacji nielicencjonowanych (ISM). Moc nadawania jest ograniczona jedynie warunkami licencji co oznacza, że krótkofalowcy mogą stosować dodatkowe wzmacniacze mocy. Przeważnie jednak moce nadawania nie przekraczają 1 W.

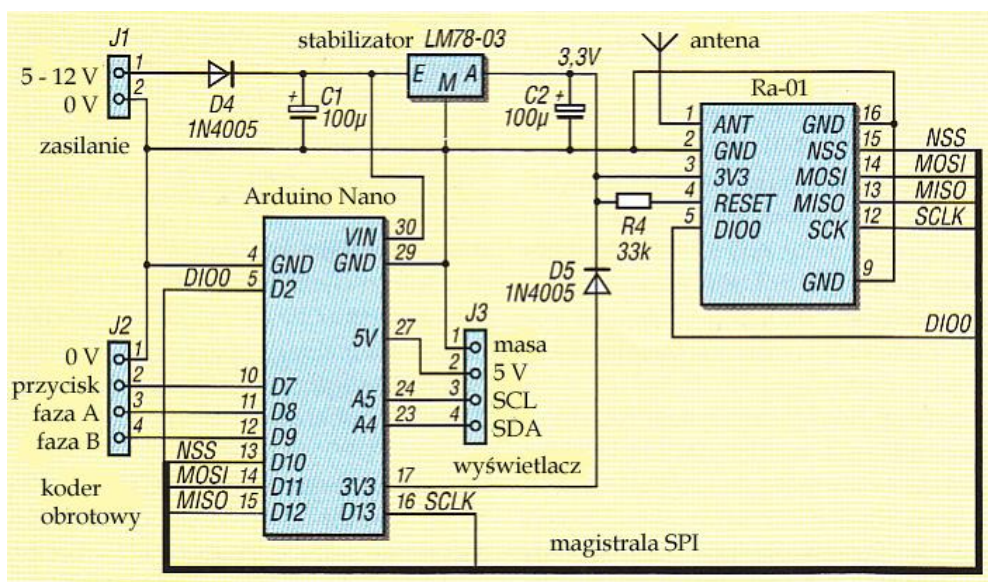
Natomiast w paśmie 868 MHz stacje nadawcze obowiązuje ograniczenie mocy do 25 mW i procentowego czasu transmisji do 0,1% lub 1 % w zależności od podzakresu pasma.

1.2. Rozwiązania stacji



Fot. 1.2.2. TRX na modułach Ra-01 lub Ra-02 (źródło: DL2MEE)

Radiostacja systemu *Lora* opracowana przez DL2MEE została opisana w numerze 3/2022 miesięcznika „Funkamateu”. Zawiera ona moduł nadawczo-odbiorczy Ra-01 firmy *Thinker* i sterujący go mikrokomputer *Arduino Nano* zasilany napięciem 3,3 V. Zamiast modułu Ra-01 można zastosować Ra-02. Pierwszy z nich posiada antenę drukowaną i ma możliwość podłączenia anteny śrubowej (helikalnej) wchodzącej w skład kompletu, a drugi posiada gniazdko μ FL/IPEX do podłączenia anteny zewnętrznej. Są one wyposażone w układ nadawczo-odbiorczy SX1278.



Rys. 1.2.2. Schemat ideowy TRX z modułem *Ra-01* i *Arduino Nano* (źródło: Funkamateu 3/2022)

Syntezer Ra-01 jest przestrajany z krokiem 61 Hz, ale wykazuje też znaczne odchyłki od nominalnej częstotliwości. Na szczęście nie mają one większego znaczenia przy szerokopasmowych transmisjach z rozpraszaniem widma. Szerokość widma sygnału wyjściowego SX1278 leży w zakresie 7,8 – 500

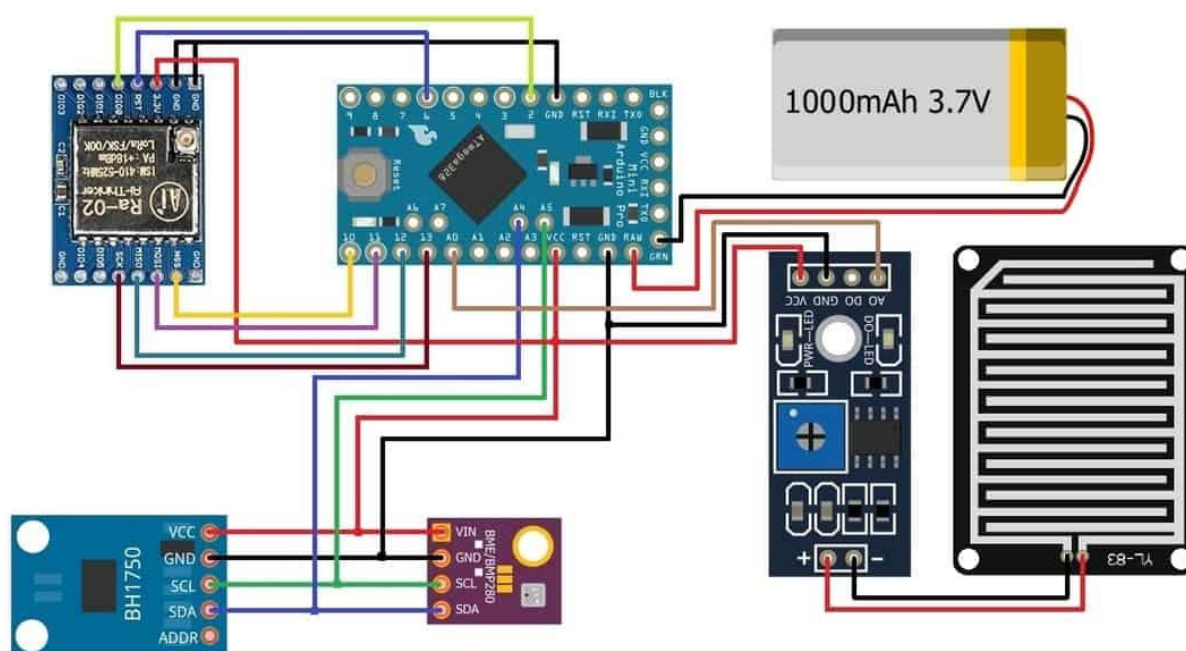
kHz, a dostępne współczynniki rozpraszania leżą pomiędzy 6 – 12. Maksymalna moc wyjściowa nadajnika wynosi 18 dBm (63 mW). Użycie mikrokomputera innego typu, zasilanego napięciem 5 V wymaga dopasowania poziomów z 5 V na 3,3 V dla modułu radiowego. Do obsługi można zastosować koder obrotowy albo przycisk, podłączane do złącza J2.

Układ może być zasilany przez gniazdko mini-USB *Arduino* albo przez jego kontakty 5 V. W przypadku zasilania modułu radiowego przez gniazdko USB *Arduino* należy ograniczyć moc wyjściową do 8 dBm, gdyż pobór prądu przy nadawaniu przekraczający 240 mA może powodować spadek napięcia zasilania mikrokomputera i zakłócenia w jego pracy – z ponownymi startami włącznie. Przy większych mocach wyjściowych konieczne jest oddzielne zasilanie napięciem 3,3 V z akumulatora albo zasilacza sieciowego.

Wyświetlacz o przekątnej 0,96 cala jest sterowany przez magistralę I2C i podłączony do złącza J3. Całość można zmontować na płytce drukowanej albo na uniwersalnej płytce eksperymentalnej. W trakcie montażu należy zwrócić szczególną uwagę na to, aby nie przegrzać modułu radiowego, gdyż spowoduje to późniejsze problemy z pracą pętli synchronizacji fazy w układzie syntezy. Może to skutkować znacznymi odchyłkami częstotliwości wyjściowej od nominalnej, dochodzącymi nawet do kilkunastu kHz.

Oprogramowanie *Arduino* korzysta z bibliotek dla obsługi wyświetlacza i modułu radiowego. Jedną z bibliotek Lory jest dostępna w Internecie pod adresem <https://github.com/sandeepmistry/arduino-LoRa>, a biblioteka dla wyświetlacza – pod adresem www.arduino.cc/reference/en/libraries/adafruit-ssd1306. Biblioteka *Lory* zawiera funkcje nadawcze i odbiorcze dla rodziny układów SX1276/77/78/79. Przykładowe programy są dostępne w strefie pobrań (*download*) witryny www.funkamateur.de. Jednym z przykładów jest program radiolatarni *LoRa* nadający tekst w odstępach 5-sekundowych. Przed uruchomieniem radiolatarni konieczne jest wprowadzenie do programu własnego znaku wywoławczego, częstotliwości nadawania i parametrów transmisji. Innym z przykładów jest program dla odbiornika sygnałów radiolatarni nadający własny tekst po naciśnięciu przycisku. Mini-TRX można także wykorzystać do transmisji własnych współrzędnych geograficznych odczytanych z odbiornika GPS.

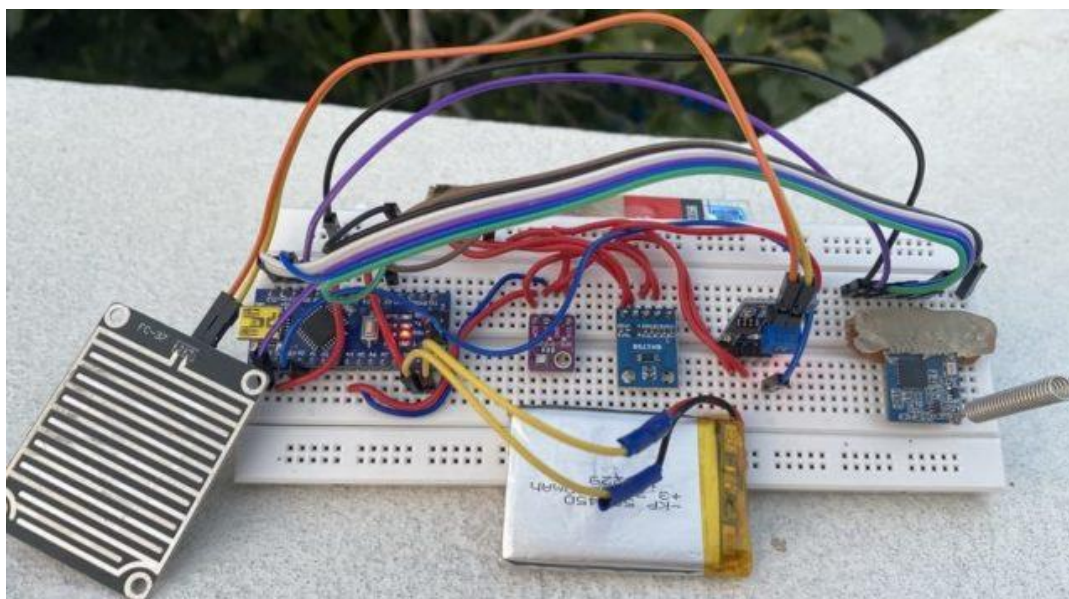
Ostatni z przykładowych programów ilustruje możliwości zdalnego sterowania. Po stronie odbiorczej (sterowanej zdalnie) dioda elektroluminescencyjna (z opornikiem szeregowym 330 Ω) jest włączona między kontaktami 5 V i SCL (A5) na złączu J3. Każdy z przykładów można modyfikować i rozszerzać zgodnie z własnymi potrzebami.



Rys. 1.2.3. Przykładowa realizacja stacji meteorologicznej

W skład stacji z rys. 1.2.3 wchodzi 3,3-voltowy mikrokomputer *Arduino Pro Mini*, czujnik barometryczny BME280, czujnik deszczu FC-37, czujnik oświetlenia BH1750 i moduł radiowy Ra-02

(z SX1278 dla 433 MHz) systemu *LoRa*. Czujniki BH1750 i BME280 są połączone z mikrokomputerem przez magistralę I2C. Do ładowania akumulatora może służyć bateria słoneczna albo moduł ładowania TP4056. Czujnik BH1750 podaje natężenie światła bezpośrednio w luksach – do wartości maksymalnej 65535.

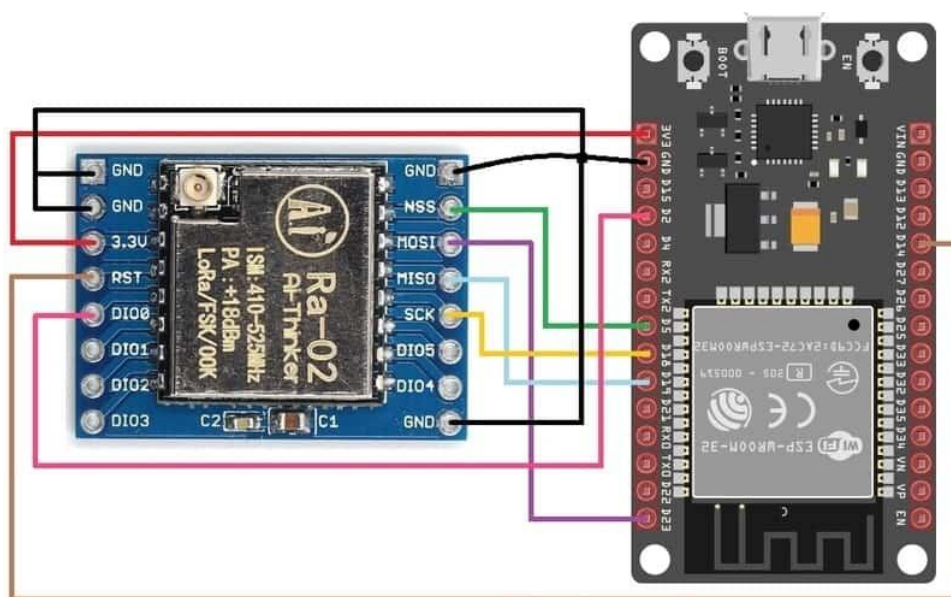


Fot. 1.2.4. Realizacja stacji na płytce próbnej

Przykładowe oprogramowanie stacji meteorologicznej i bramki WiFi wraz niezbędnymi bibliotekami dla *Arduina* znajduje się pod adresem

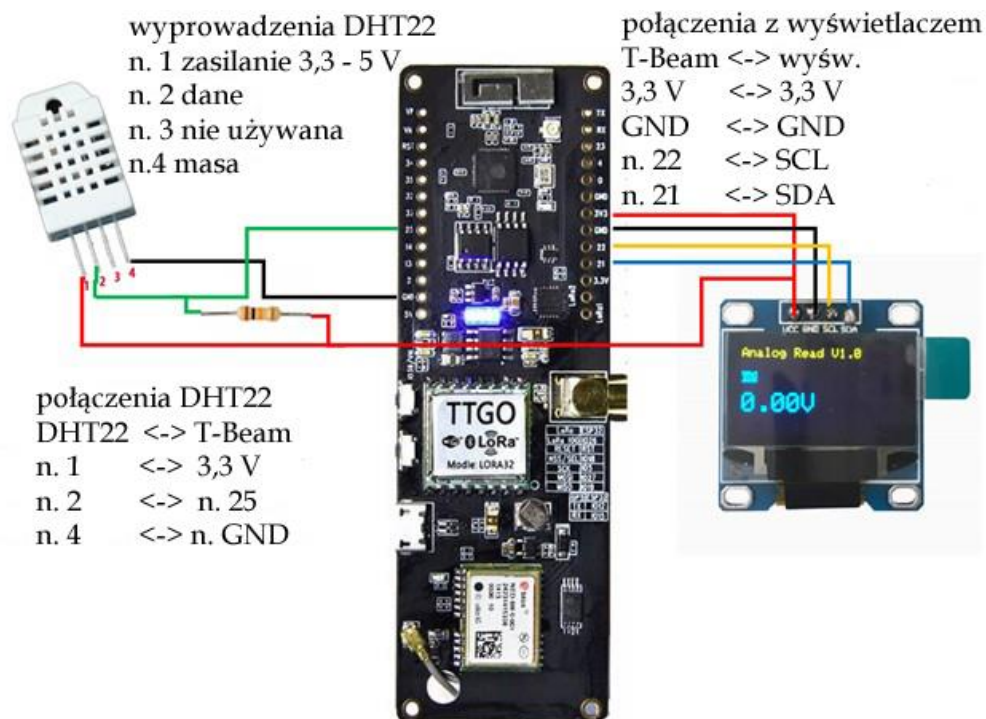
<https://how2electronics.com/lora-based-wireless-weather-station-with-arduino-esp32/>.

Bramka internetowa łączy się w podanym przykładzie z serwerem *Blynk* co pozwala na odczyt danych pomiarowych przez oprogramowanie *Blynk* na telefonie komórkowym. Bramka nie jest potrzebna w przypadku transmisji danych jedynie drogą radiową.



Rys. 1.2.5. Realizacja bramki internetowej na Ra-02 i module WiFi ESP32 (źródło: <https://how2electronics.com/lora-based-wireless-weather-station-with-arduino-esp32/>)

Prosta stacja meteorologiczna konstrukcji OE3CJB składa się z modułu *LoRa* TTGO T-Beam, czujnika temperatury i wilgotności względnej DHT22 i wyświetlacza polimerowego OLED 128 x 64 punkty (<https://oe3.oevsv.at/adl304/technikecke/technik/LoRa-APRS-Tracker/>). Moduł TTGO T-Beam pracuje w paśmie 433 MHz. Zawiera on procesor ESP32, odbiornik GPS NEO-6M i jest zasilany z akumulatora litowo-jonowego typu 18650.

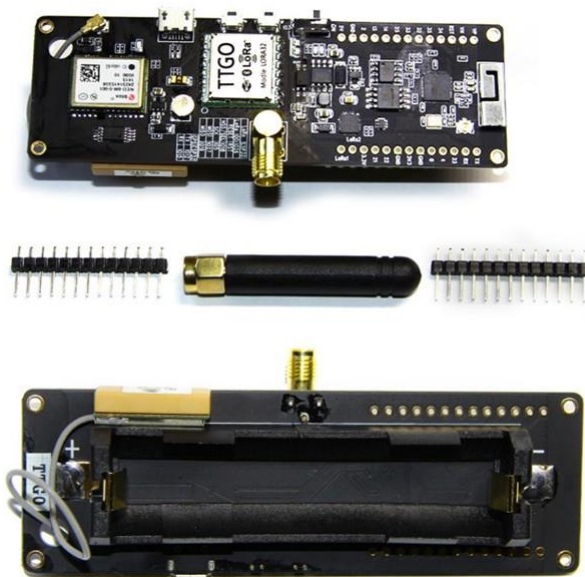


Rys. 1.2.6. Prosta stacja meteorologiczna na *TTGO T-Beam* konstrukcji OE3CJB

W wersji podstawowej bez czujnika DHT22, wyświetlacza i ze standardowym oprogramowaniem może służyć do publikowania współrzędnych geograficznych APRS operatora. Oprogramowanie stacji meteorologicznej z DHT22 jest dostępne w Internecie pod adresem:

<https://github.com/oe3cjb/TTGO-T-Beam-LoRa-APRS>

Jest to wersja eksperymentalna i może służyć za podstawę do dalszych prac i rozbudowy o dodatkowe funkcje.



Fot. 1.2.7. Moduł *TTGO T-Beam*

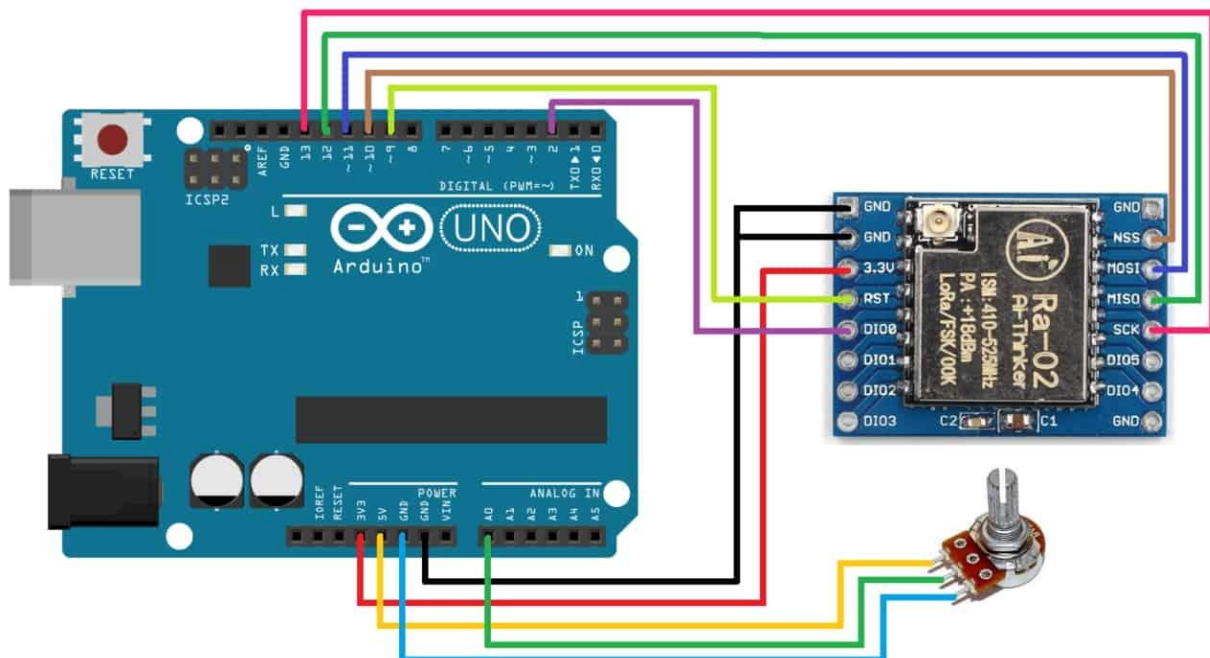
Czujniki są najczęściej połączone z mikrokomputerem za pomocą magistrali I2C lub jej nowszej wersji – magistrali I3C. W odróżnieniu od wyjść nadawczych z otwartym drenem lub kolektorem wymagających stosowania w układach oporników podciągających (ang. *pull-up*) w magistrali I3C stosowane są wyjścia przeciwsołne (ang. *push-pull*). Zapewnia ona większą szybkość transmisji, pozwala na użycie w sieci więcej niż jednego mikrokomputera zarządzającego (nadrzędnego; ang. *master*) ale także kilku nadrzędnych mikrokomputerów pomocniczych oraz na łączenie elementów współpracujących z magistralą I2C i I3C. W obu magistralach do komunikacji służą przewody SCL i SDA.

Bardziej złożone elementy sieci takie jak wyświetlacze, nadajniki systemu *LoRa* itd. są podłączane za pomocą magistrali SPI. Występują w niej przewody MISO, MOSI, SCK i NSS. Do programowej obsługi obydwu magistral służą gotowe biblioteki dla różnych typów mikrokomputerów.

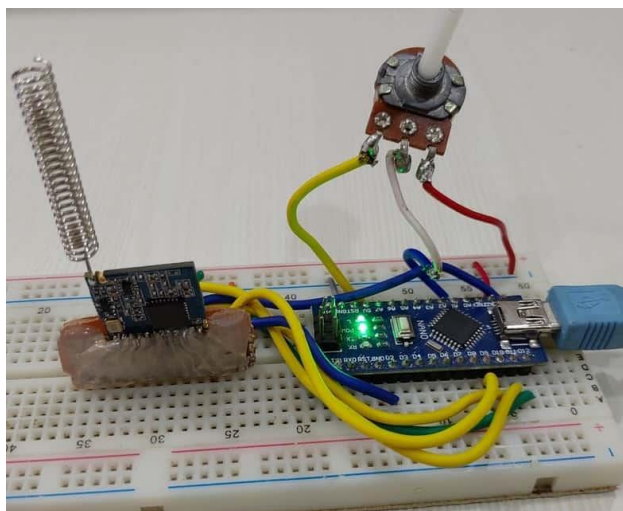
1.2.1. Eksperymenty dla początkujących

W lepszym zrozumieniu działania stacji nadawczo-odbiorczych systemu *LoRa* mogą pomóc zamieszczone poniżej programy dydaktyczne dla *Arduino UNO* połączonego z modułem Ra-02 (SX1278) na pasmo 433 MHz. Zdobyte w ten sposób doświadczenia można wykorzystać też we własnych projektach i zastosowaniach domowych albo we własnej stacji amatorskiej.

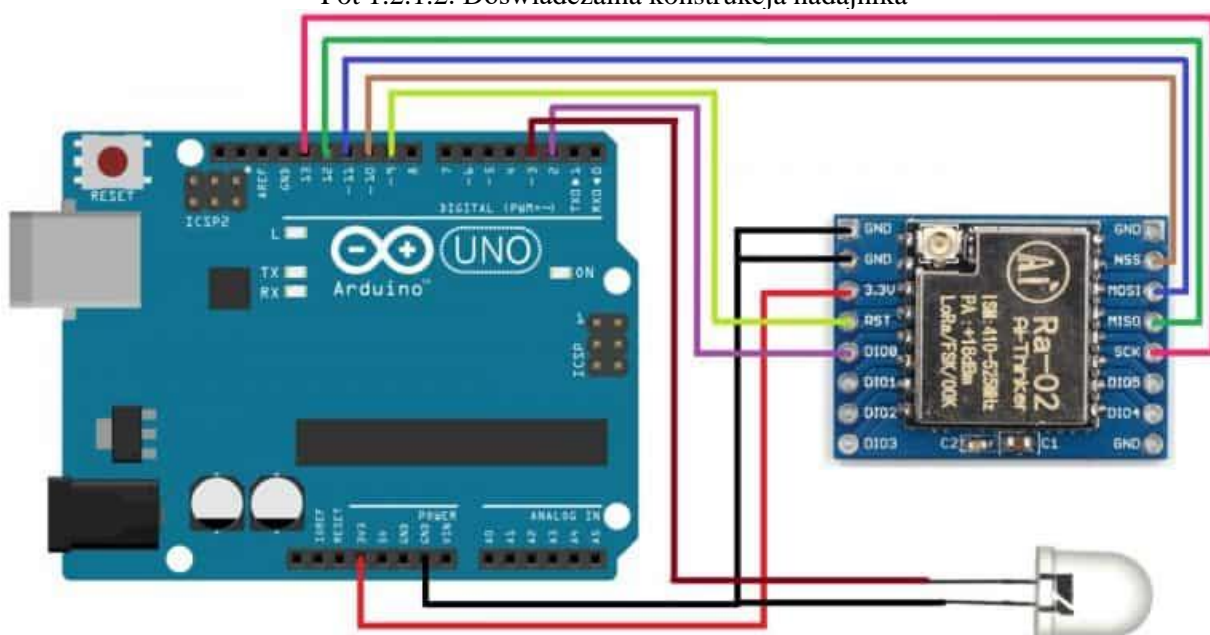
W poniższym przykładzie jasność diody świecącej po stronie odbiorczej jest regulowana za pomocą potencjometru 10 kΩ podłączonego do nadawczego mikrokomputera *Arduino*. Użyta w programie biblioteka obsługuje wszystkie moduły *LoRa* oparte na modemach SX1272, 1273, 1276, 1277, 1278, 1279 albo RFM95, 96, 97 i 98. Mogą one być zamontowane na płytkach rozszerzeń *Arduina* np. *Dragino* itp. Czulości odbiorników modemów SX dochodzą nawet do -148 dBm, a moce wyjściowe do 20 dBm (100 mW). Połączenie elementów nadajnika przedstawia schemat z rysunku 1.2.1.1. Moduł nadawczo-odbiorczy Ra-02 wymaga podłączenia anteny do widocznego na płytce w pobliżu górnej krawędzi gniazdka koncentrycznego. Brak anteny powoduje uszkodzenie nadajnika. Moduł SX1278 wymaga zasilania napięciem 3,3 V. Zasilanie napięciem 5 V spowoduje jego zniszczenie. Do komunikacji między *Arduinem* i *Ra-02* służy złącze SPI.



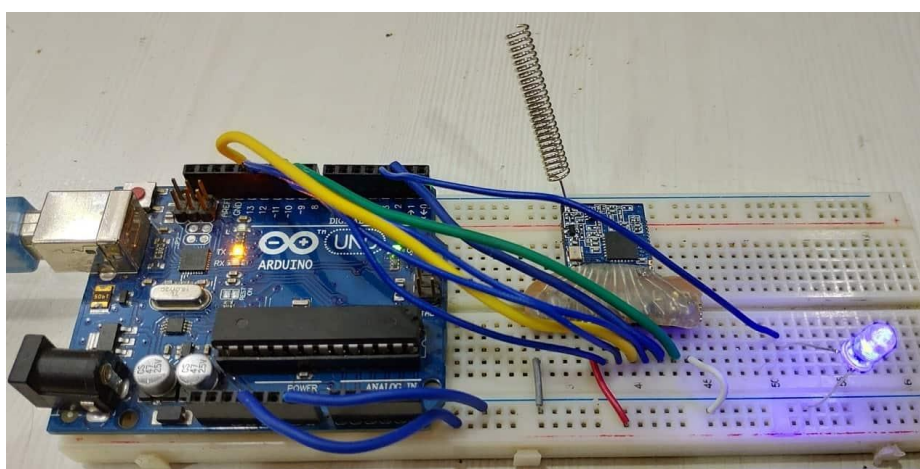
Rys. 1.2.1.1. Połączenie elementów stacji nadawczej



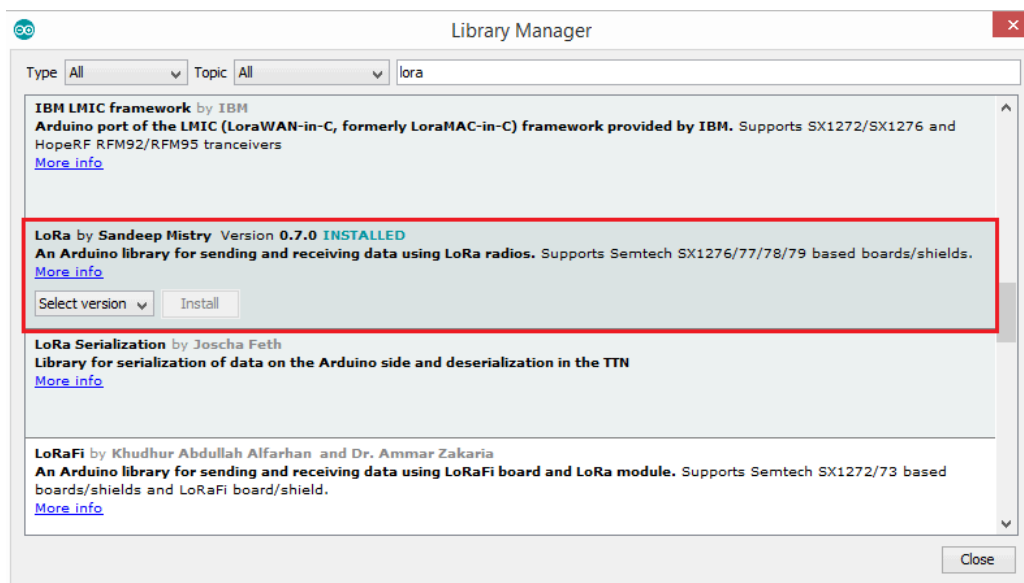
Fot 1.2.1.2. Doświadczalna konstrukcja nadajnika



Rys. 1.2.1.3. Połączenia elementów stacji odbiorczej. Dioda świecąca dowolnego typu jest połączona z wyjściem D3 Arduino



Fot. 1.2.1.4. Doświadczalna konstrukcja odbiornika

Rys. 1.2.1.5. Instalacja biblioteki za pomocą menadżera bibliotek w środowisku *Arduino IDE*

Program nadawczy

```
#include <SPI.h>
#include <LoRa.h>
int pot = A0;

void setup() {
  Serial.begin(9600);
  pinMode(pot,INPUT);

  while (!Serial);
  Serial.println("LoRa Sender");
  if (!LoRa.begin(433E6)) { // lub 915E6, pasmo pracy modułu w MHz
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  int val = map(analogRead(pot),0,1024,0,255);
  LoRa.beginPacket();
  LoRa.print(val);
  LoRa.endPacket();
  delay(50);
}
```

Program odbiorczy

```
#include <SPI.h>
#include <LoRa.h>
int LED = 3;
String inString = ""; // zmienna na dane wejściowe
int val = 0;

void setup() {
```

```

Serial.begin(9600);
pinMode(LED,OUTPUT);

while (!Serial);
Serial.println("LoRa Receiver");
if (!LoRa.begin(433E6)) { // lub 915E6
  Serial.println("Starting LoRa failed!");
  while (1);
}
}

void loop() {

  // składanie pakietu
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // odczyt pakietu
    while (LoRa.available())
    {
      int inChar = LoRa.read();
      inString += (char)inChar;
      val = inString.toInt();
    }
    inString = "";
    LoRa.packetRssi();
  }

  Serial.println(val);
  analogWrite(LED, val);
}

```

Po załadowaniu programów obracanie ośki potencjometru powinno powodować zmianę jasności diody. Eksperymenty są przeznaczone dla osób mających już pewną wiedzę na temat programowania *Arduina*.

1.2.2. Dydaktyczny program nadajnika APRS w systemie „Lora”

Program poniższy został opracowany przez OE1KEB, a dodatki zapewniające odczyt i transmisję ciśnienia atmosferycznego pochodzą od OE1KDA. Program jest przeznaczony dla Arduina z przedstawioną powyżej płytką rozszerzeń *LoRa* Dragino. Jest on przytoczony w całości jako program dydaktyczny mający ułatwić poznanie od kuchni działania nadajników danych pozycyjnych i telemetrycznych w systemie *LoRa*. Dane pozycyjne są na stałe wpisane do programu dzięki czemu niepotrzebny jest odbiornik GPS. Zamierzeniem autora nie jest opracowywanie rozwiązań konkurencyjnych w stosunku do rozwiązań powszechnie stosowanych, ale nadajnik z tym oprogramowaniem dobrze się spisuje od lat, jest odbierany przez austriackie przekaźniki APRS, a dane są przekazywane do *aprs.fi*. W programie zademonstrowano możliwość dowolnego dopasowania parametrów transmisji na poziomie podstawowym i wybór czujników (w poniższej konfiguracji jest to BMP180. Przykładowe parametry transmisji odpowiadają stosowanym w Austrii, ale ich zmiana nie powinna przysporzyć trudności trochę bardziej doświadczonym programistom. W zależności od stawianych sobie celów można całkowicie opuścić odczyt danych z czujnika i poprzestać na nadawaniu wyłącznie współrzędnych geograficznych albo podłączyć do Arduina czujnik innego typu.

```

// LoRa 9x_TX
#include <SPI.h>
#include <RH_RF95.h>
// do termometru analogowego nr 10

```



```
#include <math.h>
// do termometru nr 32 - DS18x20
#include <OneWire.h>
// do płytki LoRa
#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2
#define LED_BUILTIN 13

#define URUCHAMIANIE 1
//#undef URUCHAMIANIE

//wybór czujnika, wybrać jeden
#undef TEMP32
#undef TEMP10
#define BMP180

// formatowanie komunikatu
#define POCZT 36
#define POCZP 45
//-----
#ifdef TEMP32
// DS18S20 termometr
OneWire ds(6); // połączony z wyprowadzeniem 6

// zmienne do temperatury 32
byte i;
byte present = 0;
byte data[12];
byte addr[8];
// do konwersji
int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;
#endif
//-----
#ifdef BMP180
#include "Barometer.h"
#include <Wire.h>
float temperature;
float temperatura_srednia;
float pressure;
float cisnienie_srednie;
float atm;
float altitude;
Barometer myBarometer;
#endif
//-----
// Parametry łącza radiowego
#define BW125 112
#define BW250 128
#define BW500 144
#define SF12 12
#define CR45 2
#define CR48 8
#define ARWT 7
#define LDOPTON 8
```

```

// Zmiana na 433.775 lub inną częstotliwość, musi być zgodzić się z odbiorcą!
#define RF95_FREQ 433.775

// sterownik radiowy
RH_RF95 rf95(RFM95_CS, RFM95_INT);

void setup() {
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  pinMode(LED_BUILTIN, OUTPUT);
  while (!Serial);
  Serial.begin(9600);
  delay(100);
#ifdef URUCHAMIANIE
  Serial.println("Arduino LoRa TX Test!");
#endif
  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);

  while (!rf95.init()) {
    Serial.println("LoRa radio init failed");
    while (1);
  }

  // Ustawienia domyślne 434.000 MHz, 13 dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128 odcinków/symbol,
  CRC włączona
  // Konieczna zmiana na 433,775 MHz, 23 dBm, Bw = 125 kHz, Cr = 4/5, Sf = 4096 bit/symbol, CRC
  włączona
  // Domyślna moc wyjściowa 13dBm, korzystanie z PA_BOOST.
  // przy użyciu modułów RFM95/96/97/98 posiadających wyprowadzenie PA_BOOST
  // moc nadajnika w granicach 5 - 23 dBm:
#ifdef URUCHAMIANIE
  // Rejestry konfiguracyjne po inicjalizacji (pierwsza konfiguracja z tabeli, indeks 0)
  Serial.print ("Rejestr 1d = ");
  Serial.println (rf95.spiRead(RH_RF95_REG_1D_MODEM_CONFIG1));
  Serial.print ("Rejestr 1e = ");
  Serial.println (rf95.spiRead(RH_RF95_REG_1E_MODEM_CONFIG2));
  Serial.print ("Rejestr 26 = ");
  Serial.println (rf95.spiRead(RH_RF95_REG_26_MODEM_CONFIG3));
  Serial.print ("Rejestr 01 (OP_MODE) = ");
  Serial.println (rf95.spiRead(RH_RF95_REG_01_OP_MODE));
#endif
  // Pasma i współczynnik korekcji,
  // Header_Explicit = 0, CRC_off = 0
  rf95.spiWrite(RH_RF95_REG_1D_MODEM_CONFIG1, BW125 + CR45 + 0);
  // Zmiana współczynnika rozpraszania na SF12 - 0xc4
  // ARW + Timeout = 7
  rf95.spiWrite(RH_RF95_REG_1E_MODEM_CONFIG2, SF12 * 16 + ARWT);
  // LowDoptON = 8
  rf95.spiWrite(RH_RF95_REG_26_MODEM_CONFIG3, LDOPTON);
#ifdef URUCHAMIANIE

```

```

// sprawdzenie
Serial.print ("Rejestr 1d = ");
Serial.println (rf95.spiRead(RH_RF95_REG_1D_MODEM_CONFIG1));
Serial.print ("Rejestr 1e = ");
Serial.println (rf95.spiRead(RH_RF95_REG_1E_MODEM_CONFIG2));
Serial.print ("Rejestr 26 = ");
Serial.println (rf95.spiRead(RH_RF95_REG_26_MODEM_CONFIG3));

Serial.println ("Czestotliwosc po starcie ");
Serial.println (rf95.spiRead(RH_RF95_REG_06_FRF_MSB));
Serial.println (rf95.spiRead(RH_RF95_REG_07_FRF_MID));
Serial.println (rf95.spiRead(RH_RF95_REG_08_FRF_LSB));
delay (5000);
#endif
if (!rf95.setFrequency(433.775)) {
#ifdef URUCHAMIANIE
  Serial.println("Set Frequency failed");
#endif
  delay(10);
}
// Częstotliwość: 433775 / 61,03515625 -> 7106969,6 ->
// 7106969 -> 6C 71 47 -> 108 113 47
rf95.spiWrite(RH_RF95_REG_06_FRF_MSB, 108);
rf95.spiWrite(RH_RF95_REG_07_FRF_MID, 113);
rf95.spiWrite(RH_RF95_REG_08_FRF_LSB, 71);
#ifdef URUCHAMIANIE
  // sprawdzenie
  Serial.println ("Czestotliwosc ustawiona ");
  Serial.println (rf95.spiRead(RH_RF95_REG_06_FRF_MSB));
  Serial.println (rf95.spiRead(RH_RF95_REG_07_FRF_MID));
  Serial.println (rf95.spiRead(RH_RF95_REG_08_FRF_LSB));
  Serial.println ("Moc po starcie ");
  Serial.println (rf95.spiRead(RH_RF95_REG_09_PA_CONFIG));
  Serial.println (rf95.spiRead(RH_RF95_REG_4D_PA_DAC));
#endif
rf95.setTxPower(23, false);
#ifdef URUCHAMIANIE
  //Sprawdzenie
  Serial.println ("Moc ustawiona ");
  Serial.println (rf95.spiRead(RH_RF95_REG_09_PA_CONFIG));
  Serial.println (rf95.spiRead(RH_RF95_REG_4D_PA_DAC));
#endif
#ifdef BMP180
  myBarometer.init();
  temperatura_srednia = 0;
  cisnienie_srednie = 0;
#endif
}
// podobno maksymalna długość nadawanych pakietów wynosi 40-45 bajtów
//char meteopacket[] = "OE1KDA-11>APRS:!4808.22N/01620.86E_c...s...g...t-20b10250h..";
char radiopacket[] = "OE1KDA-11>APRS:!4808.22N/01620.86E-";
String temp_string = "";
String cisn_string = "";
String wilg_string = "";

```

```

String nap_string = "";
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);

void loop() {
  Serial.println("Sending to rf95_server");
  // Send a message to rf95_server
#ifdef TEMP10
  #ifdef URUCHAMIANIE
    Serial.print(temperatura10(analogRead(0)));
    Serial.println("C");
  #endif
  temp_string = String(temperatura10(analogRead(0)));
  radiopacket[POCZT] = 'T';
  radiopacket[POCZT + 1] = '=';
  for (int i = 0; i <= 3; i++)
    { radiopacket[POCZT + 2 + i] = temp_string[i];
    }
  radiopacket[POCZT + 6] = ' ';
  radiopacket[POCZT + 7] = 'C';
#endif
#ifdef TEMP32
  temperatura32();
  radiopacket[POCZT] = 'T';
  radiopacket[POCZT + 1] = '=';
  for (int i = 0; i <= 3; i++)
    { radiopacket[POCZT + 2 + i] = temp_string[i];
    }
  radiopacket[POCZT + 6] = ' ';
  radiopacket[POCZT + 7] = 'C';
#endif
#ifdef BMP180
  barometrBMP180();
  if (temperatura_srednia == 0)
    { temperatura_srednia = temperature;
    }
  else
    { temperatura_srednia = temperatura_srednia - temperatura_srednia / 8;
      temperatura_srednia = temperatura_srednia + temperature/8;
    }
  temp_string = String(temperatura_srednia);
  if (cisnienie_srednie == 0)
    { cisnienie_srednie = pressure;
    }
  else
    { cisnienie_srednie = cisnienie_srednie - cisnienie_srednie / 4;
      cisnienie_srednie = cisnienie_srednie + pressure / 4;
    }
  cisl_string = String((cisnienie_srednie * 1.021)/100, 1);
  radiopacket[POCZT] = 'T';
  radiopacket[POCZT + 1] = '=';
  for (int i = 0; i <= 3; i++)
    { radiopacket[POCZT + 2 + i] = temp_string[i];
    }
  radiopacket[POCZT + 6] = ' ';

```

```

radiopacket[POCZT + 7] = 'C';
radiopacket[POCZP] = 'P';
radiopacket[POCZP + 1] = '=';
for (int i = 0; i <= 6; i++)
  { radiopacket[POCZP + 2 + i] = (cisl_string[i] != 0) ? cisl_string[i]: ' ';
  }
radiopacket[POCZP + 9] = ' ';
radiopacket[POCZP + 10] = 'h';
radiopacket[POCZP + 11] = 'P';
radiopacket[POCZP + 12] = 'a';
#endif
#ifdef URUCHAMIANIE
  Serial.print("Sending "); Serial.println(radiopacket);
#endif
radiopacket[POCZP + 13] = 0;
#ifdef URUCHAMIANIE
  Serial.println("Sending...");delay(10);
#endif
rf95.send((uint8_t *)radiopacket, 59);
#ifdef URUCHAMIANIE
  Serial.println("Waiting for packet to complete...");delay(10);
#endif
rf95.waitPacketSent();
puls();
// Oczekiwanie na odpowiedź
#ifdef URUCHAMIANIE
  Serial.println("Waiting for reply..."); delay(10);
  if (rf95.waitAvailableTimeout(1000))
  {
    // Po otrzymaniu odpowiedzi
    if (rf95.recv(buf, &len))
    {
      Serial.print("Got reply: ");
      Serial.println((char*)buf);
      Serial.print("RSSI: ");
      Serial.println(rf95.lastRssi(), DEC);
    }
    else
    {
      Serial.println("Receive failed");
    }
  }
  else
  {
    Serial.println("No reply, is there a listener?");
  }
#endif
delay(1000);
}

#ifdef TEMP10
double temperatura10(int RawADC)
{
// odczyt temperatury z czujnika nr 10 podłączonego do A0
double Temp;

```

```

Temp = log(((10240000/RawADC) - 10000));
Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))*
Temp );
Temp = Temp - 273.15;
return Temp;
}
#endif

#ifdef TEMP32
void temperatura32 (void)
{
  ds.reset_search();
  if ( !ds.search(addr) ) {
    Serial.print("No more addresses.\n");
    ds.reset_search();
    return;
  }

  Serial.print("R=");
  for( i = 0; i < 8; i++) {
    Serial.print(addr[i], HEX);
    Serial.print(" ");
  }
  if ( OneWire::crc8( addr, 7) != addr[7]) {
    Serial.print("CRC is not valid!\n");
    return;
  }

  if ( addr[0] == 0x10) {
    Serial.print("Device is a DS18S20 family device.\n");
  }
  else if ( addr[0] == 0x28) {
    Serial.print("Device is a DS18B20 family device.\n");
  }
  else {
    Serial.print("Device family is not recognized: 0x");
    Serial.println(addr[0],HEX);
    return;
  }

  ds.reset();
  ds.select(addr);
  ds.write(0x44,1);    // start konwersji, na końcu włączenie mocy

  delay(1000);    // 750 ms może wystarczyć lub nie
  // można tutaj wywołać ds.depower(), ale wykona to funkcja ds.reset().

  present = ds.reset();
  ds.select(addr);
  ds.write(0xBE);

  Serial.print("P=");
  Serial.print(present,HEX);
  Serial.print(" ");
  for ( i = 0; i < 9; i++) {    // 9 bajtów

```

```

    data[i] = ds.read();
    Serial.print(data[i], HEX);
    Serial.print(" ");
}
Serial.print(" CRC=");
Serial.print( OneWire::crc8( data, 8), HEX);
Serial.println();

// Konwersja
LowByte = data[0];
HighByte = data[1];
TReading = (HighByte << 8) + LowByte;
SignBit = TReading & 0x8000; // sprawdzenie najstarszego bitu
if (SignBit) // wartość ujemna
{
    TReading = (TReading ^ 0xffff) + 1; // uzupełnienie dwójkowe
}
Tc_100 = (6 * TReading) + TReading / 4; // mnożenie przez (100 * 0.0625) czyli 6.25

Whole = Tc_100 / 100; // rozdzielanie części całkowitej i części ułamkowej
Fract = Tc_100 % 100;
if (SignBit) // gdy wartość ujemna
{
    Serial.print("-");
}
Serial.print(Whole);
Serial.print(".");
if (Fract < 10)
{
    Serial.print("0");
}
Serial.print(Fract);

Serial.print("\n");
temp_string = (SignBit)? "-" + String(Whole) : String(Whole);
}
#endif
#ifdef BMP180
void barometrBMP180(void)
{
    temperature = myBarometer.bmp085GetTemperature(myBarometer.bmp085ReadUT()); // odczyt
    temperature, bmp085ReadUT należy wywołać najpierw
    pressure = myBarometer.bmp085GetPressure(myBarometer.bmp085ReadUP()); // odczyt ciśnienia
    altitude = myBarometer.calcAltitude(pressure); // wysokość na podstawie wartości nie
    skompensowanej
    atm = pressure / 101325;
    Serial.print("Temperature: ");
    Serial.print(temperature, 2); // wyświetlenie dwóch miejsc po przecinku
    Serial.println(" deg C");

    Serial.print("Pressure: ");
    Serial.print(pressure, 0); // tylko liczba całkowita.
    Serial.println(" Pa");

    Serial.print("Ralated Atmosphere: ");

```

```
Serial.println(atm, 4); //wyświetlenie 4 miejsc po przecinku

Serial.print("Altitude: ");
Serial.print(altitude, 2); //wyświetlenie dwóch miejsc po przecinku
Serial.println(" m");

Serial.println();

delay(1000); //odczekanie sekundy

}
#endif
void puls(void) {
  char i;
#ifdef URUCHAMIANIE
  digitalWrite (LED_BUILTIN, HIGH);
  delay (5000);
  digitalWrite(LED_BUILTIN, LOW);
  delay (5000);
#else
  for(i = 0;i<=60; i++)
  {
    digitalWrite (LED_BUILTIN, HIGH);
    delay (5000);
    digitalWrite(LED_BUILTIN, LOW);
    delay (5000);
  }
#endif
}
```

W przypadku korzystania z czujnika temperatury i wilgotności względnej DHT11 konieczne jest dodanie pliku nagłówkowego i podanych niżej deklaracji

```
#include "DHT.h"
#define DHTPIN A0 // wejście analogowe dla
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
```

Oraz w void setup()

inicjalizacji

```
dht.begin();
```

Do odczytu danych z DHT11 służą polecenia gdzie temp jest zmienną typu float, a hum – typu int.

```
temp = dht.readTemperature();
```

```
hum = dht.readHumidity();
```

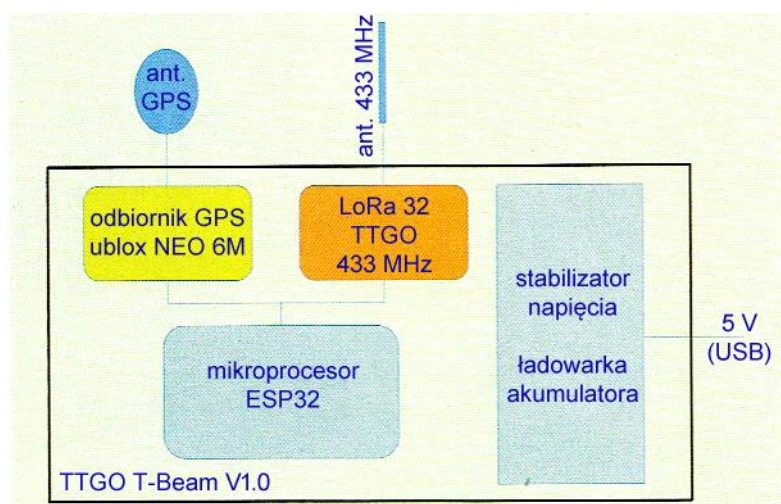

1.3. LoRa APRS

Od początku istnienia systemu APRS komunikaty są transmitowane na kilku ogólnie znanych częstotliwościach w pasmach 2 m i 70 cm emisją AX.25 AFSK – a więc przy użyciu dowolnych radiostacji FM. Szybkość transmisji wynosi standardowo 1200 bit/s. Na częstotliwościach krótkofalowych stosowana jest szybkość transmisji 300 bit/s. W transmisji LoRa APRS używane są częstotliwości pracy w podzakresach 433 i 434 MHz.

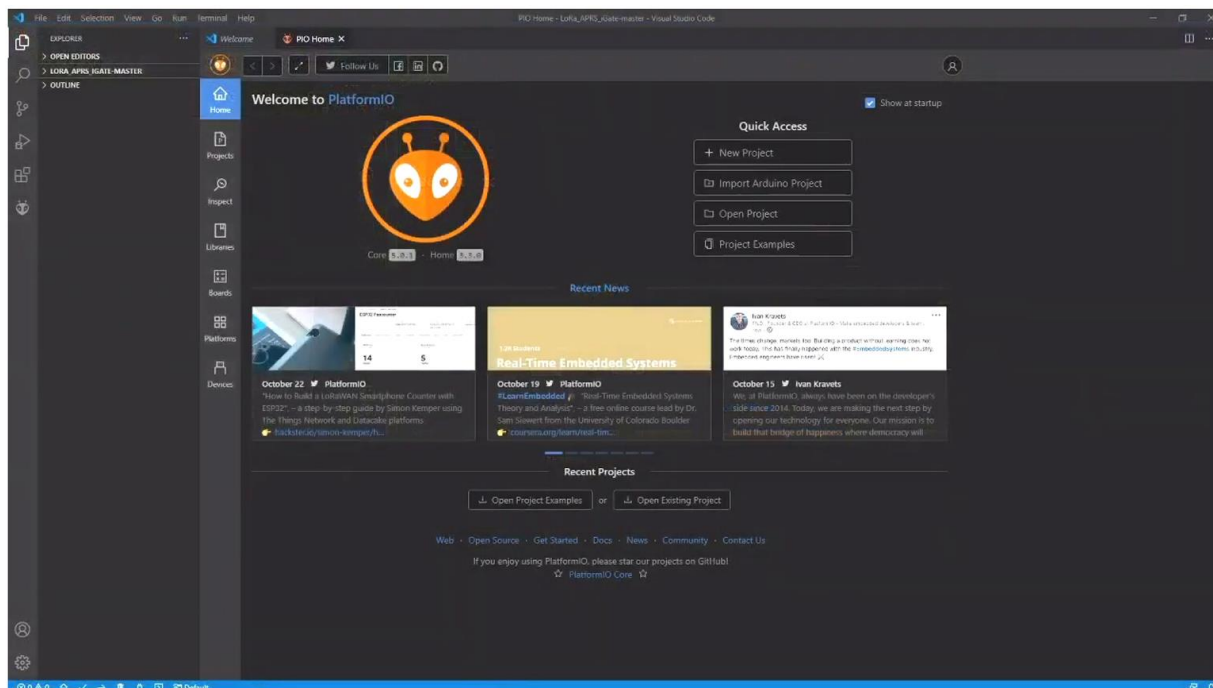
W klasycznym systemie komunikaty APRS nadawane przez stacje indywidualne są retransmitowane przez cyfrowe stacje przekaźnikowe (ang. *digipeater*) i dzięki temu są odbieralne na pewnym obszarze. W szczęśliwej sytuacji docierają one do bramek radiowo-internetowych skąd trafiają do serwerów APRS-IS i są widoczne na tle map w Internecie. Najpopularniejszą z takich witryn jest *aprs.fi*. Każda ze stacji indywidualnych może też służyć jako (uzupełniająca) stacja przekaźnikowa retransmitująca komunikaty do publicznych stacji przekaźnikowych i bramek internetowych.

Stopniowo konkurentem tego rozwiązania staje się transmisja komunikatów w systemie *LoRa* – pracującym z rozpraszaniem widma sygnału. W zastosowaniach krótkofalarskich system pracuje w paśmie 70 cm z mocami nadajników nie przekraczającymi przeważnie 100 mW. Dzięki rozpraszaniu widma sygnału uzyskiwane są zasięgi od kilku do kilkunastu i więcej kilometrów. Szybkości transmisji są naogół niskie i takie są też ilości przekazywanych danych. Łączność przebiega więc według zasady „powoli ale skutecznie”. Poprzez bramki radiowo-internetowe komunikaty docierają, tak jak w pierwszym przypadku, do serwerów APRS-IS i są wyświetlane na mapach. W odróżnieniu od rozwiązania klasycznego radiowa retransmisja komunikatów przez bramki internetowe i stacje indywidualne nie jest zalecana gdyż zwiększa tylko niebezpieczeństwo kolizji ograniczając w ten sposób skuteczną przepustowość kanału.

Do najważniejszych elementów sieci APRS-LoRa należą nadajniki komunikatów pozycyjnych (ang. *tracker*) i bramki radiowo-internetowe (ang. *iGate*). Stosunkowo niewielkim kosztem i przy małym zapotrzebowaniu na energię – pozwalającym na zasilanie z akumulatorów – uzyskuje się sieci przydatne także w ramach akcji ratunkowych i łączności kryzysowych. Bramki internetowe pracują wyłącznie odbiorczo i mogą być uruchamiane również przez nasłuchowców i inne osoby zainteresowane, ale nie posiadające (jeszcze) licencji. Stacje meldujące pozycję pracują nadawczo i do ich uruchomienia konieczna jest licencja amatorska. W ich konfiguracji konieczne jest podanie znaku wywoławczego operatora.



Rys. 1.3.1. Schemat blokowy płytki TTGO T-Beam (źródło: [1.3.7])

Rys. 1.3.2. Okno główne środowiska *PlatformIO*

W obu rodzajach stacji stosowane są moduły mikrokomputerów z procesorami ESP32 i nadawczo-odbiorcze moduły LoRa (modemy) SX1278. Mogą być to przykładowo modele *TTGO T-Beam V1.1* ESP32, *TTGO LoRa32*, *Helltec WiFi LoRa 32* itd. W bramkach internetowych znajdują zastosowanie moduły bez odbiorników GPS – wystarczy wprowadzenie stałych współrzędnych w konfiguracji, natomiast nadajniki komunikatów pozycyjnych powinny być wyposażone w odbiorniki satelitów GPS. Do połączenia z Internetem bramki korzystają ze złącza WiFi. Pierwszy z wymienionych typów posiada odbiornik GPS i znajduje zastosowanie w stacjach meldujących położenie, a dwa dalsze jako nie wyposażone w GPS nadają się głównie do pracy w bramkach. Nic nie stoi jednak na przeszkodzie wykorzystaniu pierwszego modelu do pracy w bramkach, o ile zakup większej liczby identycznych urządzeń okaże się korzystniejszy finansowo. Modele *T-Beam* posiadają pojemniki dla akumulatorów typu 18650 i mogą dzięki temu być zainstalowane w miejscach, w których niemożliwe jest zasilanie z sieci. Jako obudowę można wykorzystać wprawdzie dowolne pudełko o pasujących wymiarach, ale w Internecie oferowane są obudowy dopasowane do wymiarów różnych typów płytek. Możliwe jest również ich samodzielne wydrukowanie na drukarce trójwymiarowej (3-D) w oparciu o dostępne w Internecie pliki konstrukcyjne.

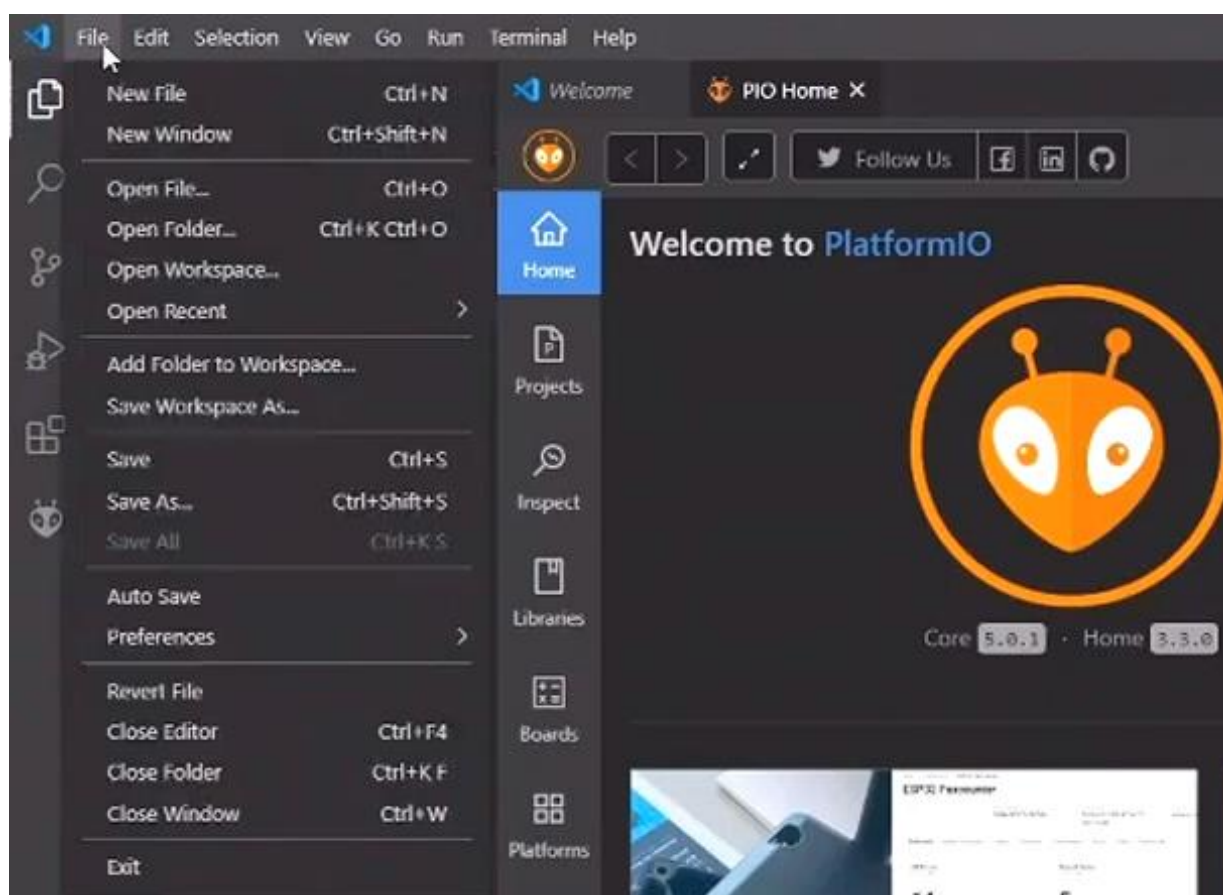
Praktyczne jest, gdy posiadają one wyświetlacz – nawet tylko o rozdzielczości 128 x 64 punkty. Część modułów jest fabrycznie wyposażona w wyświetlacze, ale ewentualne podłączenie go we własnym zakresie nie jest sprawą skomplikowaną. Konieczne jest połączenie masy, zasilania oraz sygnałów SDA i SCL złącza I2C – czyli w sumie czterech przewodów.

Oferowane w zestawach anteny bywają niedostrojone dokładnie do częstotliwości pracy, korzystne może okazać się zastąpienie ich przez skuteczniejsze. Dla upewnienia się czy warto je zamienić można zmierzyć rezonans i przebieg WFS w zakresie pasma 70 cm korzystając z analizatora antenowego.

Komunikaty APRS można uzupełnić o dane telemetryczne takie jak ciśnienie atmosferyczne, temperaturę i wilgotność względną po dodaniu czujnika BME280 lub podobnego, o inne wielkości pomiarowe zależne od typu czujnika (np. również napięcie zasilania) albo o dowolne teksty. Niektóre wersje oprogramowania oferują standardowo takie możliwości. Dla części z nich dostępne są także kody źródłowe co pozwala bardziej doświadczonym programistom na dodanie dowolnej pożądanej funkcjonalności.

Oprogramowanie stacji meldującej pozycję i bramki autorstwa OE5BPA jest dostępne odpowiednio pod adresami [1.3.2] i [1.3.3]. Spakowane archiwum programu jest pobierane po naciśnięciu w oknie *githuba* zielonego przycisku „Code” i wybraniu w menu punktu „Download ZIP”. Archiwum należy następnie rozpakować do dowolnego katalogu na PC. Oprogramowanie nie posiada serwera HTTP i dlatego konieczne jest wprowadzenie parametrów konfiguracyjnych w kodzie źródłowym.

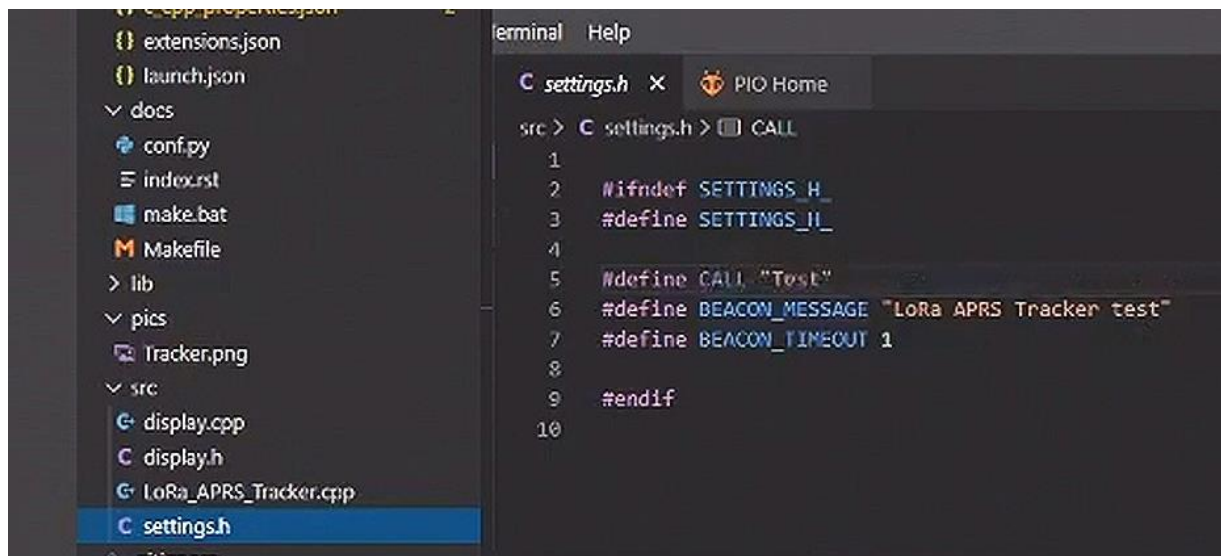
Wpisanie oprogramowania do mikrokomputera wymaga uprzedniego zainstalowania na PC najpierw środowiska programistycznego *Visual Studio Code* (VSCode), a następnie – *PlatformIO* IDE [1.3.4]. Konieczne może być też zainstalowanie sterowników dla złącza USB: TTGO POE Board CH340, Heltec CP210X itp. Podane w objaśnieniach na stronach [1.3.2] lub [1.3.3] dodatkowe biblioteki (j.np. APRS-Decoder-Lib) są pobierane automatycznie przez *PlatformIO* w trakcie kompilacji. Przeważnie znajdują się one na Githubie. Oprócz wersji dla systemu Windows istnieje również wersja dla Linuksa. Po otwarciu *PlatformIO* (rys. 1.3.2) należy przez menu „File | Open folder” („Plik | Otwórz katalog”) wybrać katalog z rozpakowanym programem (nadajnika lub bramki w zależności od realizowanej konstrukcji) – patrz rys. 1.3.3. Program w nim zawarty jest wówczas ładowany do środowiska programistycznego. Po załadowaniu w lewym oknie eksploratora wyświetlany jest spis plików programu (struktura projektu). Z dzungli plików należy wybrać jedynie plik nagłówkowy *settings.h*. Jego treść zostaje wyświetlona w prawym oknie. Plik zawiera kilka poleceń *#define*, w których należy podać własny znak wywoławczy, część tekstową komunikatu APRS i odstęp czasu między transmisjami (rys. 1.3.4).



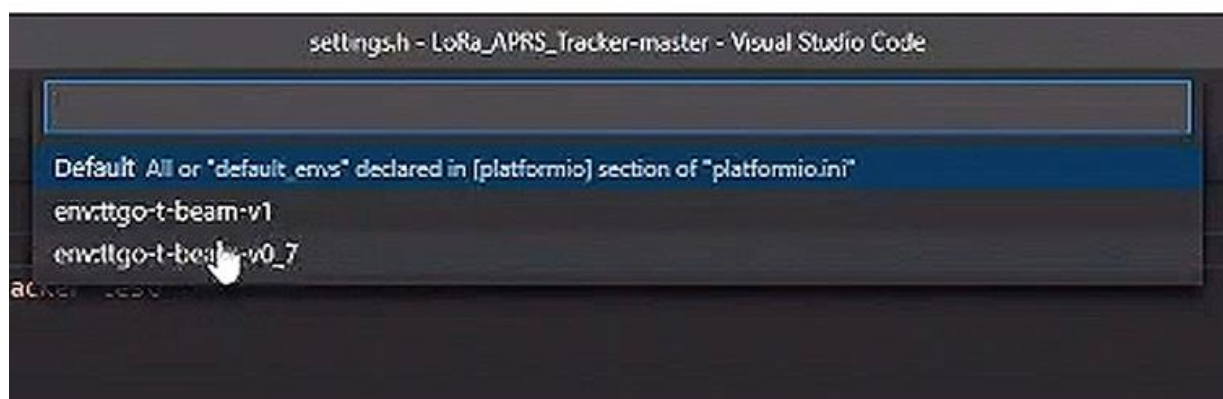
Rys. 1.3.3. Otwieranie katalogu programu

Kolejnym krokiem jest wybór typu mikrokomputera. Po naciśnięciu napisu „Default” w niebieskiej linii informacyjnej u dołu okna otwierane jest u góry okienko wyboru wersji (rys. 1.3.5). Program nadawczy może pracować zarówno na TTGO V0.7 jak i V1. Należy wybrać wersję zgodną z posiadaną płytką. W celu skompilowania i załadowania programu do pamięci mikrokomputera należy nacisnąć w dolnej niebieskiej linii strzałkę skierowaną w prawo (rys. 1.3.6). Płytkę mikroprocesora powinna zostać podłączona do PC przez złącze USB zaraz po otwarciu programu *PlatformIO*. Ładowanie programu trwa dobrą chwilę. Po jego zakończeniu można odłączyć mikroprocesor od PC i jest on już gotowy do użycia. Po pierwszym włączeniu albo po dłuższym czasie nieużywania proces synchronizacji odbiornika GPS może trwać 10 – 15 minut. Transmisja komunikatów APRS rozpoczyna się dopiero po odebraniu kompletnych danych pozycyjnych z satelitów GPS. W przypadku wystąpienia błędów w trakcie ładowania najlepiej jest skasować zawartość pamięci mikrokomputera (za pomocą funkcji „Erase Flash” środowiska programistycznego) i załadować program ponownie.

Każda późniejsza zmiana parametrów konfiguracyjnych wymaga ponownej kompilacji i załadowania programu do pamięci mikrokomputera.



Rys. 1.3.4. Plik *settings.h*



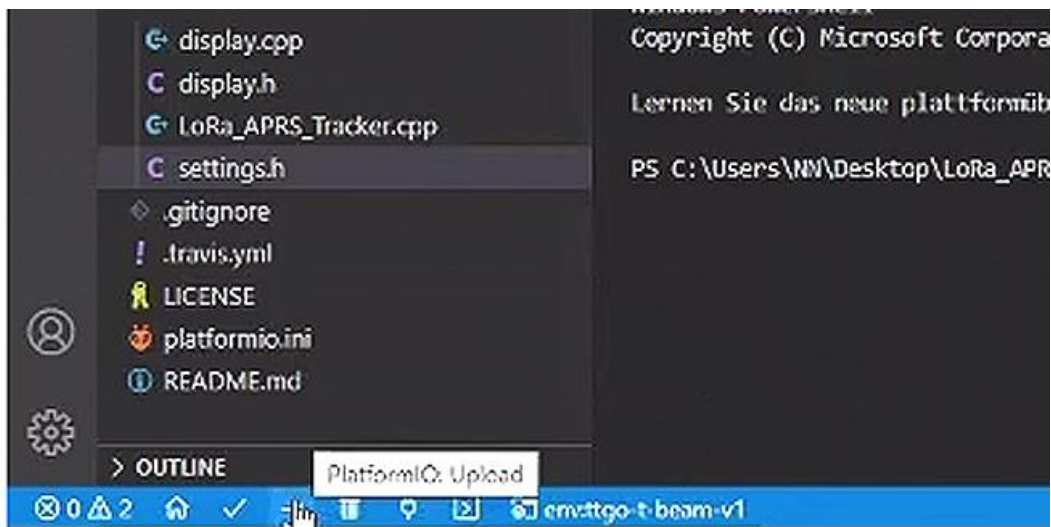
Rys. 1.3.5. Wybór wersji mikrokomputera

Analogicznie przebiega ładowanie oprogramowania bramki internetowej. Wybór modułów mikroprocesorów jest szerszy, ponieważ nie muszą one posiadać odbiornika GPS i pojemnika dla akumulatora (są przeważnie instalowane w miejscach, gdzie nie ma problemów z zasilaniem z sieci). Współrzędne geograficzne stacji są podawane w konfiguracji oprogramowania. Dane konfiguracyjne są zawarte w pliku *is-cfg.json*. Należą do nich znak wywoławczy lub nazwa stacji (w przypadku osób nielicencjonowanych), dane dostępowe do lokalnej sieci WLAN, komunikat APRS, współrzędne geograficzne stacji, hasło dostępu do sieci APRS-IS, adres serwera sieci i numer kanału logicznego (ang. *port*). Jako serwer dostępowy APRS-IS można pozostawić *euro.aprs2.net*. Generator hasła dostępu w oparciu o znak lub nazwę stacji znajduje się w Internecie pod adresem [1.3.5] (rys. 1.3.8). Po załadowaniu programu identycznie jak w pierwszym przypadku należy dodatkowo załadować plik konfiguracyjny *is-cfg.json*. W tym celu po naciśnięciu symbolu główki zagadkowego stworzenia (można widzieć w nim mrówkę) konieczne jest wybranie w lewym oknie eksploratora używanej płytki mikrokomputera (np. *env:ttgo-lora32-v2*) w przykładzie z fot. 1.3.2, następnie należy przejść do punktu „Platform” poniżej i wybrać „Upload Filesystem Image” (rys. 1.3.7). W przypadku pojawienia się na ekranie meldunku błędu najczęściej pomaga ponowne załadowanie pliku konfiguracyjnego. Po wyświetleniu meldunku o braku jakiejś biblioteki trzeba dopisać jej nazwę w pliku *platformio.ini*.

O ile w klasycznym systemie APRS bramki radiowo-internetowe były instalowane w miejscach dogodnych i zapewniających możliwie duże zasięgi, o tyle w sieci APRS-LoRa korzystniejsza jest instalacja możliwie dużej liczby bramek o niedużych zasięgach – nawet przy oknie mieszkania lub na balkonie. Zmniejsza się w ten sposób niebezpieczeństwo kolizji pakietów danych. Stacje stałe powinny z tego

samego względu nadawać komunikaty nawet co 20 minut lub rzadziej. Stacje pracujące z samochodu nie powinny nadawać częściej niż co minutę – najlepiej, aby odstęp czasu zależał od szybkości ruchu (ang. *smart beaconing*). Nie powinny one retransmitować radiowo odebranych komunikatów aby nie zwiększać obciążenia kanałów radiowych. Bramki uruchamiane przez osoby nie posiadające licencji tym bardziej nie powinny pracować nadawczo. Korzystne jest, aby bramki pracowały na okrągło bez przerw. O ile w przypadku stacji nadawczych meldujących radiowo swoje położenia praca bezobsługowa (bez obecności operatora w pobliżu) wymaga uzyskania specjalnej licencji, o tyle bramki czysto odbiorcze mogą pracować bez nadzoru i specjalnych zezwoleń.

W Austrii i w Niemczech w sieci APRS LoRa stosowana jest częstotliwość 433,775 MHz, współczynnik rozpraszania (SF w terminologii LoRa) 12, szerokość pasma 125 kHz, szybkość transmisji 183 bit/s i stopa danych korekcyjnych CR 4/5 co oznacza położenie nacisku na większe zasięgi i niezawodność transmisji. Często stosowanym rozszerzeniem dla stacji indywidualnych jest -7, a dla bramek internetowych -10. Standardowym symbolem dla stacji indywidualnych jest symbol pieszego. Można go oczywiście zmienić na dowolny inny pasujący do sytuacji. W Polsce stosowana jest (w chwili powstania niniejszego skryptu) częstotliwość 434,855 MHz, szerokość pasma 125 kHz, stopa danych korekcyjnych CR 4/7 i współczynnik rozpraszania SF równy 9 co zapewnia szybkość transmisji netto zbliżoną do 1200 bit/s (jak w klasycznym wydaniu APRS). Szczegółowej analizie doboru parametrów transmisji i uzyskiwanych wyników poświęcony jest punkt 1.1. Ponieważ krótkofalarstwo jest służbą eksperymentalną zachęcamy użytkowników do wypróbowania innych zestawów parametrów.



Rys. 1.3.6. Ładowanie programu do pamięci mikrokomputera

Oprogramowanie OE5BPA jest jednym z możliwych programów dla stacji APRS pracujących w systemie LoRa. Jest ono znane i rozpowszechnione nie tylko w Austrii ale i w innych krajach. W Polsce znane jest oprogramowanie SQ9MDD oparte na opracowaniu austriackich krótkofalowców OE1ACM i OE3CJB. Sprawa będzie się z pewnością dalej rozwijać dynamicznie i należy spodziewać się powstawania nowych programów i udoskonalonych wersji obecnych. Postaramy się uwzględnić je w przyszłych publikacjach.

Oprogramowanie SQ9MDD pracuje na modułach *TTGO T-beam* v.0.7 i v.1.0 oraz Lora32. Oferuje ono również funkcjonalność TNC w trybie KISS (rys. 1.3.9).

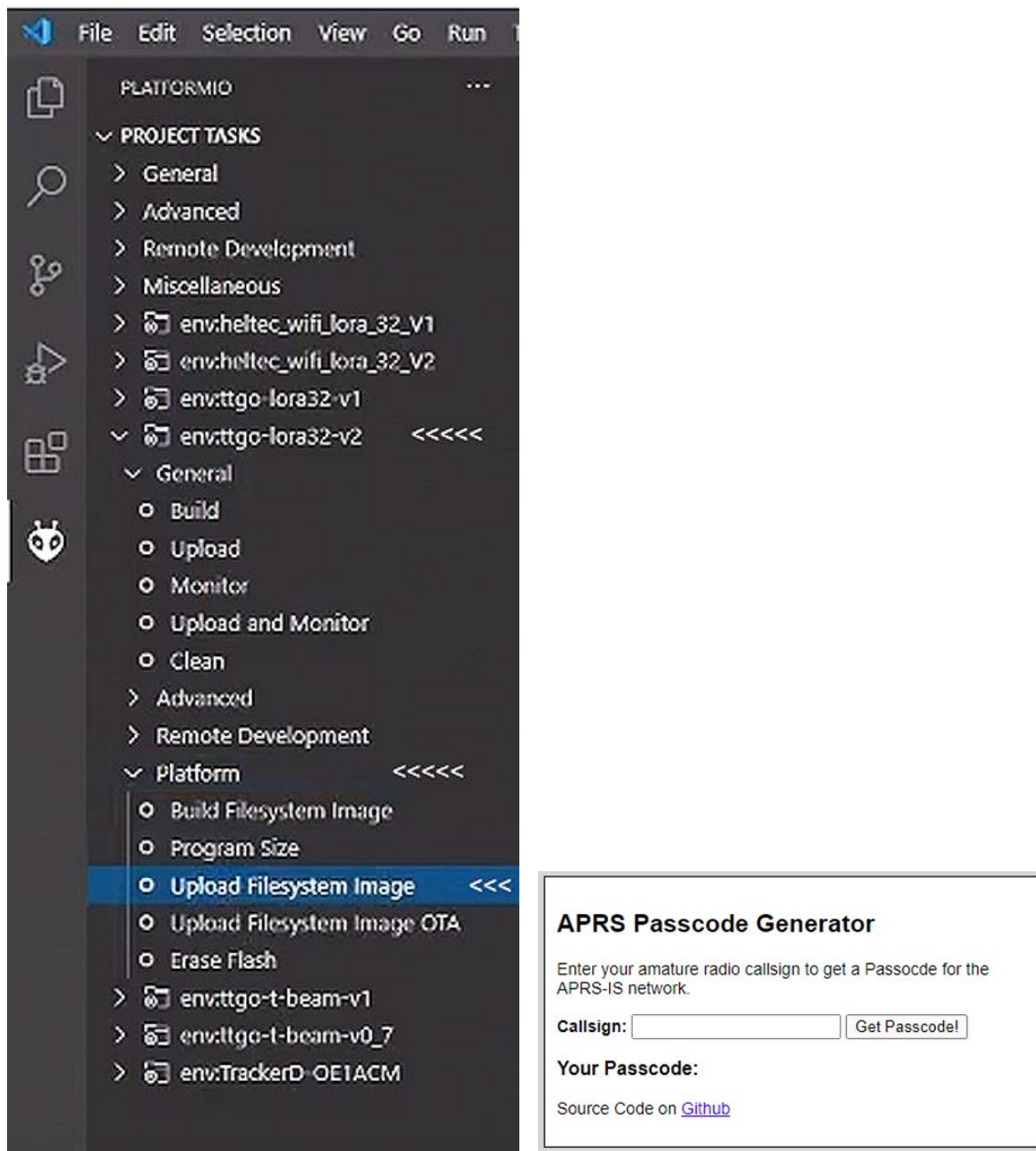
Uruchomienie programu na modułach LoRa wymaga pobrania go z githuba ze strony:

<https://github.com/SQ9MDD/TTGO-T-Beam-LoRa-APRS/releases> a następnie zainstalowania programu ładującego esp-home ze strony <https://github.com/esphome/esphome-flasher/releases>.

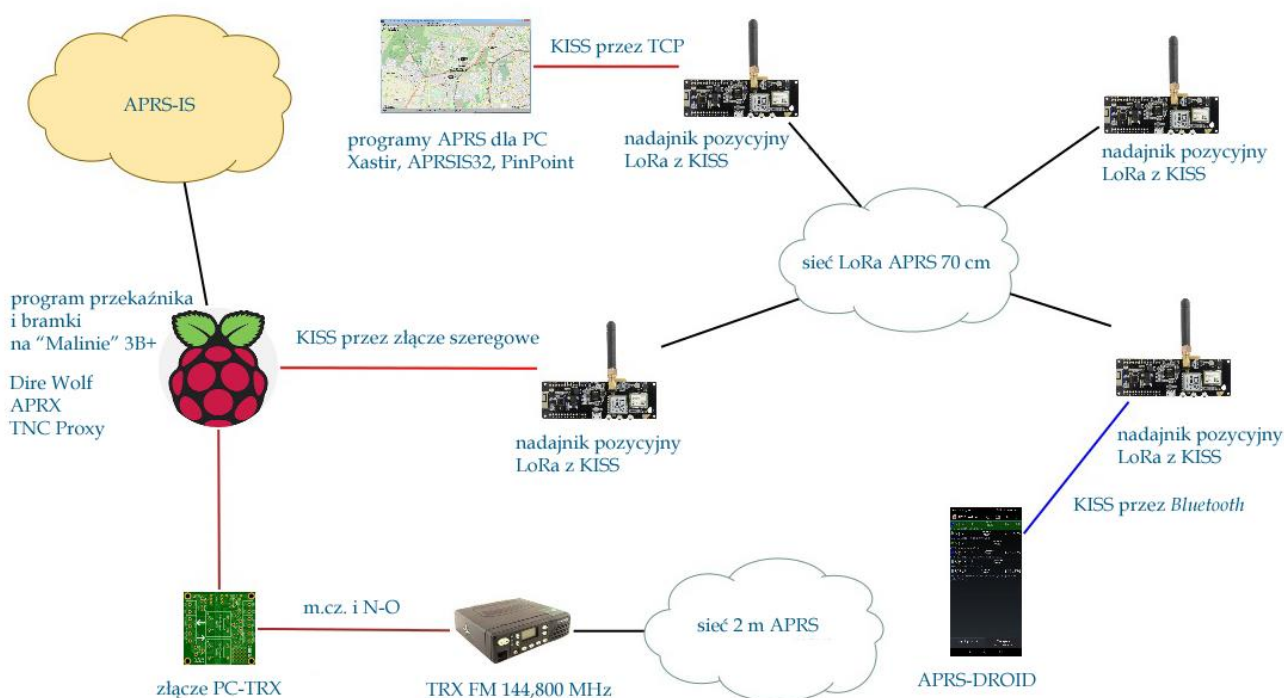
Następnie należy podłączyć moduł LoRa do złącza USB i wywołać program ładujący. Dla zapewnienia komunikacji z mikroprocesorem konieczne może być też zainstalowanie sterownika CP210X. W programie ładującym należy wybrać złącze szeregowo, do którego jest podłączony moduł i plik oprogramowania pobrany z githuba, a na koniec nacisnąć przycisk ekranowy „Flash ESP”.

Do zainstalowania oprogramowania z plików źródłowych konieczne jest, jak w przypadku programu OE5BPA, użycie środowiska *Visual Studio Code* i uzupełnienia *PlatformIO*.

Po otwarciu *PlatformIO* należy wczytać kod źródłowy przy użyciu punktu „Import Arduino Project”, potem nacisnąć na symbol mrówki i na punkt ładowania „Upload”. Uruchomienie programu wymaga połączenia się z modułem i skonfigurowania dostępu do lokalnej sieci bezprzewodowej WiFi przez podanie nazwy sieci i hasła dostępu. Konfiguracja parametrów APRS wymaga podania częstotliwości nadawania i odbioru, szybkości transmisji (domyślnie jest to 300 bodów), wprowadzenia znaku wywoławczego stacji z rozszerzeniem, wybrania symbolu stacji oraz wyboru treści komunikatu i wprowadzenia jego tekstu. Konieczne jest też ustalenie odstępów czasowych między kolejnymi transmisjami.



Rys. 1.3.7. Dodatkowy krok przy ładowaniu konfiguracji bramki radiowo-internetowej (po lewej)
 Rys. 1.3.8. Okno generatora hasła dostępu do APRS-IS. Należy wpisać w nim znak wywoławczy lub przyjętą nazwę bramki i nacisnąć przycisk „Get Passcode!” (po prawej). Niektóre ze spotykanych w Internecie generatorów haseł dostępu generują dla znaków stacji z rozszerzeniem inne hasła niż dla znaku podstawowego. Hasła te nie są przyjmowane przez sieć APRS-IS



Rys. 1.3.9. Możliwe sposoby wykorzystania oprogramowania SQ9MDD

1.4. Wyposażenie stacji APRS w systemie „LoRa”

Stacja *APRScube* konstrukcji DL3DCW (<http://aprscube.de>) jest urządzeniem składającym się z gotowych modułów opartych na modułach *m2stack*: podstawowego z mikrokomputerem ESP32, złączem WLAN, kieszenią dla pamięci microSD, 16 MB pamięci programu i dwucalowym wyświetlaczem, modułu radiowego Lora o mocy wyjściowej 60 mW (18 dBm), odbiornika GPS, modułu QRO – wzmacniacza mocy 1 W (30 dBm), bateryjnego, prototypowego i innych. Minimalny zestaw modułów dla APRS składa się z modułu podstawowego, modułu LoRa i modułu GPS.

APRScube jest przewidziana do pracy w warunkach domowych albo przewoźnie w samochodzie. Możliwe jest podłączenie kombinowanego czujnika temperatury, wilgotności i ciśnienia atmosferycznego (ENV II). Jako uzupełnienie dostępny jest statyw z czujnikiem temperatury i wilgotności powietrza. Oba czujniki mają przypisany ten sam adres i dlatego nie mogą być używane równolegle. Do zasilania napięciem 5 V służy gniazdko USB typu C. Nowszy moduł ENV III zawiera czujniki SHT30 i QMP6988.

Tabela 1.4.1

Parametry czujników zawartych w modułach ENV II i ENV III (dane w nawiasach)

Czujnik SHT30		Czujnik BMP280 (QMP6988)	
Temperatura	-40 – 120°C	Ciśnienie	300 – 1100 hPa
Uchyb temperatury	± 0,2°K (dla 0 – 60°C)	Uchyb ciśnienia	±1 hPa (±3,9 hPa)
Wilgotność wzgl.	10 – 90%	Złącze I2C, wtyk PH2.0 czterokontaktowy	
Uchyb wilgotności	2%	Wymiary 24 x 32 x 8 mm (24,2 x 32,2 x 8,1 mm)	

Stacja może pracować jako urządzenie meldujące współrzędne odczytane z odbiornika GPS (ang. *tracker*) i jako bramka internetowa przekazująca do serwera odebrane dane – *iGate*. Przy pracy w domu dane pozycyjne mogą być przekazywane bezpośrednio do serwera APRS-IS przez sieć lokalną WLAN, a przy pracy ruchomej poza domem nadawane radiowo w standardzie *LoRa*. Uruchomienie, konfiguracja i obsługa stacji *APRScube* są nieskomplikowane.

W witrynie <http://aprscube.de> dostępne są aktualne wersje oprogramowania, program ładujący ESP32 *Flash_Download_Tool* i sterownik *CP210X*.

Zmiana konfiguracji stacji w pliku *APRScube.ini* nie wymaga ponownego ładowania całości oprogramowania z PC. Przy użyciu programu *WinSCP* możliwy jest dostęp FTP do *APRScube*, pobranie pliku i ponowne wpisanie go po dokonaniu modyfikacji.



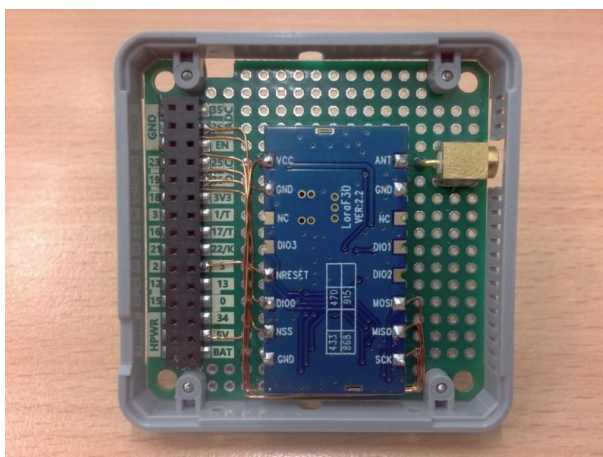
Fot. 1.4.1. APRScube



Fot. 1.4.2. Moduł GPS



Fot. 1.4.3. Kombinowany czujnik ENV II



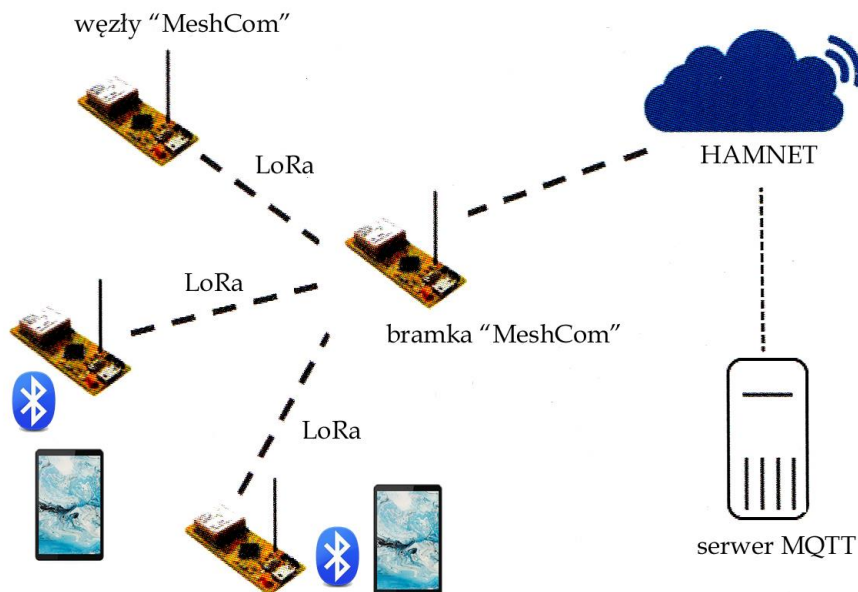
Fot 1.4.4. Moduł wzmacniacza mocy 1 W

2. Sieć systemu „LoRa MeshCom”

Krótkofalarska sieć „LoRa MeshCom” jest opracowywanym przez krótkofalowców austriackich rozwiązaniem pozwalającym na przesyłanie meldunków tekstowych, danych telemetrycznych, pozycyjnych APRS i rozkazów zdalnego sterowania na znaczne odległości mimo małych mocy nadawania. Jej zaletą jest także niskie zapotrzebowanie na energię, dzięki czemu stacje sieci mogą być zasilane bateryjnie. Sieć może okazać się przydatna również w łącznościach kryzysowych i ratunkowych. Sieć *Meshcom* ma topologię siatki, w której poszczególne węzły pośredniczą w komunikacji między dowolnymi innymi węzłami i bramkami łączącymi ją z serwerami MQTT poprzez Hamnet. Różni się więc ona od sieci LoRaWAN mającej topologię gwiazdową, której centralnymi punktami są koncentratory (bramki) i serwery internetowe. W odróżnieniu od sieci LoRaWAN – TNN, Helium, ChirpStack itp. – pracujących głównie w paśmie 868 MHz sieć *MeshCom* korzysta z pasma 433 MHz na zasadach obowiązujących krótkofalowców.

Zasięgi modułów *LoRa* mogą w zależności od zakresu częstotliwości i stosowanych anten przekraczać 10-20 km w terenach wiejskich i 1-5 km w terenach miejskich. W szczególnie korzystnych warunkach sygnały nadawane z mocą 300 mW były odbierane nawet w odległościach 80 – 100 km. Moc wyjściowa modułów nie przekracza jednak przeważnie 100 mW, a ze względu na ich nagrzewanie zalecane jest korzystanie z niższych mocy. Zastosowane liniowe rozpraszanie widma sygnału (ang. *chirp*) znacząco zwiększa odporność na zakłócenia i osiągnięte zasięgi [2.3].

Krótkofalowcy korzystają z modułów na częstotliwość 433 MHz (EU433). W rozwiązaniu austriackim wykorzystano popularne i niedrogie moduły *LoRa* typów *Lilygo-TTGO-T-Beam* (z odbiornikiem GPS, fot. 1.5.1), *Lilygo TTGO Lora* (bez GPS), firm Heltec, DollaTec, Wisblock i podobne.



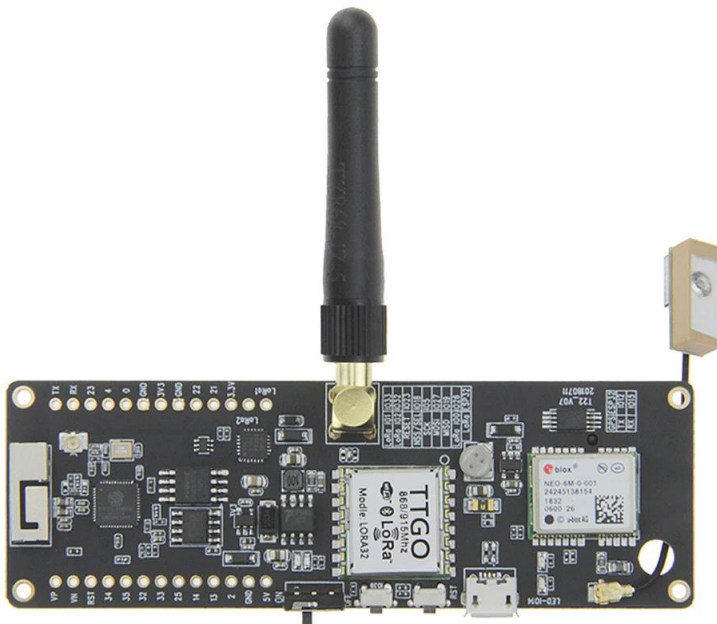
Rys. 2.1. Zasada organizacji sieci. Każdy z węzłów może pośredniczyć w komunikacji między pozostałymi elementami sieci

Stacje użytkowników – węzły (ang. *node*) – lub ich grupy (lokalne chmury radiowe) są połączone z innymi za pośrednictwem Hamnetu. Hamnet staje się więc siecią szkieletową dla stacji indywidualnych wyposażoną w bramki wejściowe LoRa (rys. 2.1). W razie potrzeby węzły użytkowników mogą pośredniczyć w przekazywaniu komunikatów do bramki i w przeciwną stronę. Sieć wokół bramki ma więc topologię mniej lub bardziej pełnej siatki (ang. *mesh*). Jej przydatność i niezawodność rosną w miarę wzrostu liczby czynnych stacji. Bramki różnią się od węzłów jedynie tym, że dysponują połączeniem z Internetem lub z Hamnetem i utrzymują w ten sposób łączność z serwerem MQTT. Jest on w ten sposób połączony ze wszystkimi lokalnymi chmurami tworzącymi dzięki niemu wspólną większą chmurę. Zasadniczo pracuje on więc jak znany z innych sieci reflektor. Meldunki pozycyjne są przekazywane do *aprs.fi*.

W Austrii na trasach prowadzących do bramek hamnetowych używana jest częstotliwość 433,175 MHz (w Niemczech m.in. 439,700 MHz). Szerokość pasma sygnału wynosi obecnie 250 kHz (poprzednio było to 125 kHz), stopa korekcji 4/6, a współczynnik rozpraszania 11 (w Niemczech również 10). Szyfrowanie treści (PSK) jest oczywiście wyłączone.

Moduły wyposażone w odbiornik GPS pozwalają na przesyłanie bieżących danych pozycyjnych, natomiast w modułach nie wyposażonych w odbiorniki można wprowadzić stałą pozycję w konfiguracji. Bezpłatny program *MeshCom 4.0* dla Androida pozwala na połączenie telefonu lub komputera androidowego z modułem przez złącze *Bluetooth* i zobrazowanie odebranych pozycji na mapie albo na wyświetlanie i nadawanie komunikatów tekstowych.

W sieci *MeshCom* stosowane są dwa mechanizmy zabezpieczające przed nieskończoną retransmisją komunikatów w pętlach. Jednym z nich jest analiza numerycznych identyfikatorów komunikatu. Komunikaty o znanym identyfikatorze (a więc już wcześniej retransmitowane przez dany węzeł) nie są ponownie przekazywane dalej. Drugim z nich jest ograniczenie liczby skoków czyli liczby retransmisji między węzłami podobnie jak dla sieci APRS. Ich maksymalna liczba jest ograniczona do siedmiu, ale standardowo krótkofalowcy korzystają z ograniczenia do pięciu, z tym, że każdy z nadawców może wybrać dowolną wartość w podanym zakresie.



Fot. 2.2. Moduł z odbiornikiem GPS Fot. 2.3. Moduł *DollaTek Lora 433* bez odbiornika GPS

Najpopularniejsze obecnie moduły TTGO są wyposażone albo w procesory ESP32 albo nRF52. Do ładowania oprogramowania wewnętrznego [2.9] można skorzystać z programu *ESPflasher* dla Windows. Nie wymaga on instalacji, a jedynie wywołania jego pliku exe.

Oprogramowanie modułu węzła *MeshCom* wymaga wprowadzenia znaku wywoławczego za pomocą polecenia `--setcall...`, w modułach nie posiadających odbiornika GPS trzeba podać współrzędne geograficzne i wysokość n.p.m. za pomocą poleceń `--setlat`, `--setlon` i `--setalt`.

Częstotliwość pracy i parametry rozpraszania widma są wpisane do programu. Polecenie `--info` służy do wywołania informacji o parametrach konfiguracyjnych, a `--help` wywołuje tekst pomocy. Do komunikacji PC z modułem węzła można użyć dowolnego programu terminalowego j.np. *PuTTY*. Szybkość transmisji wynosi 115200 bit/s. W poprzednich wersjach konieczna była instalacja programu *Mesh-tastic* na PC.

Bramki wejściowe Hamnetu wymagają dodatkowo podania w ramach argumentów danych dostępowych dla WiFi (identyfikatora, hasła) i adresu IP serwera MQTT (internetowego albo hamnetowego) i wyboru jednej tych sieci. Bramki korzystają z tego samego oprogramowania, co węzły i różnią się tylko podaniem tych dodatkowych parametrów.

Serwer MQTT (*Message Queing Telemetry Transport*) służy do rozprowadzania komunikatów w sieciach TC/IP przy użyciu protokołu MQTT (opracowanego w 1999 roku na potrzeby łączności satelitarnej i od 2013 roku stosowanego w sieci *Internetu Przedmiotów* – IoT). Pośredniczy on więc w komunikacji między bramkami LoRa w Hamniecie. Dodatkowo zapewnia także połączenia z systemami APRS, DAPNET, Messengera w Hamniecie, austriackiego serwera Tetra (służącego do wymiany komunikatów w tej sieci), itd. dla komunikatów zawierających dane odpowiedniego rodzaju. Rozwiązanie austriackie korzysta z własnego oprogramowania serwera *ÖVSV-MeshCom-Server* zamiast powszechnie stosowanego oprogramowania *Mosquitto*. Pozwala to krótkofalowcom na pełniejszy wpływ na jego funkcjonalność i konfigurację niż w przypadku rozwiązania komercyjnego.

Tabela 2.1

Wybrane moduły radiowe LoRa 433 MHz

Producent	Typ	Wyposażenie
Lilygo	TTGO T-Beam (V1.1)	Łącze LoRa SX127x, WiFi, Bluetooth, ekran OLED 0,96 cala I2C, odbiornik GPS typu NEO-6M, gniazdko antenowe SMA, pojemnik na akumulator 18650
Lilygo	TTGO Lora (V2.0, V2.1)	Łącze LoRa SX127x, WiFi, Bluetooth, ekran OLED 0,96 cala, procesor ESP32, brak GPS, kieszeń microSD, gniazdko antenowe μ .FL lub SMA, gniazdko mikroUSB (zas. 5 V), gniazdko zasilania z akumulatora litowego 2,7 – 3,6 V
Heltec Automation	WiFi LoRa 32 (V2)	Łącze LoRa SX1276/1278, WiFi, Bluetooth, ekran OLED 0,96 cala 128 x 64 pkt., procesor ESP32, gniazdko mikro-USB (zas. 5 V), kabel do zasilania (bat. 3,7 V), brak GPS, gniazdko antenowe μ .FL
Lilygo	TTGO T-Echo	Łącze LoRa SX1262, Bluetooth, ekran rInk 1,54 cala, gniazdko antenowe μ .FL, procesor nRF5284, odbiornik GPS typu L76K

2.1. Przykładowe zastosowania sieci

Do przykładowych możliwości wykorzystania sieci należą:

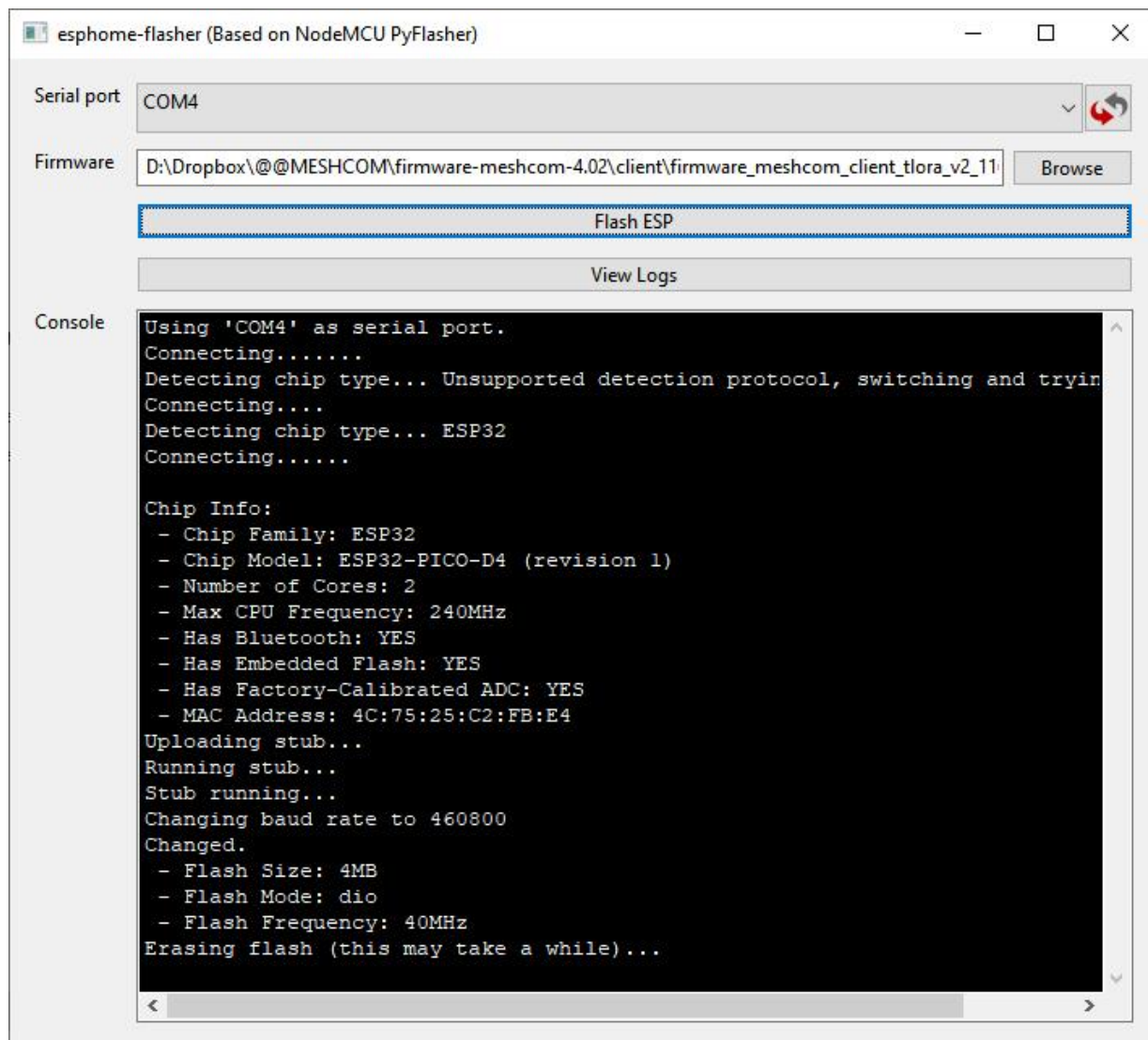
- komunikacja między dowolnymi dwoma stacjami w sieci (lokalnej chmurze radiowej) – w stylu *WhatsApp* przez program androidowy,
- komunikacja z grupą,
- nadawanie tekstów z komputerów androidowych (wiadomości zawierające do 228 znaków),
- nadawanie informacji pozycyjnych,
- transmisja różnych danych telemetrycznych, meteorologicznych, informacji o burzach i wyładowaniach elektrycznych, obecne wersje oprogramowania pozwalają na podłączenie czujników meteorologicznych BMP280 lub BME280,
- ostrzeżenia o niebezpiecznych zjawiskach (podtopienia, silne wiatry, silne opady, burze itd.),
- połączenie z funkcją SMS sieci DMR IPSC2,
- komunikacja z modułami telegraficznymi Morserino,
- łączności satelitarne przez transponder QO-100 WB w kanale 2409,25 MHz,
- zdalnie połączenie dostępnych publicznie przycisków alarmowych (policja, pogotowie, straż), normalnie w sieciach 868 MHz, ale w sytuacjach alarmowych przełączane do sieci amatorskiej i amatorskiego serwera MQTT,
- połączenia ze stacjami pracującymi emisją JS8Call itp.,
- zdalne sterowanie przemiennikami amatorskimi lub ich włączanie i wyłączanie,
- zawiadamianie i przywoływanie jak w amatorskiej sieci DAPNET (POCSAG),
- lokalny dostęp do Hamnetu (wolna transmisja danych),
- i wiele innych.

Oczywiście są to tylko propozycje i zakres zastosowań można dowolnie uzupełniać. Każdy z użytkowników sieci może dowolnie wybierać interesujące go możliwości i aspekty.

2.2. Instalacja i uruchomienie oprogramowania

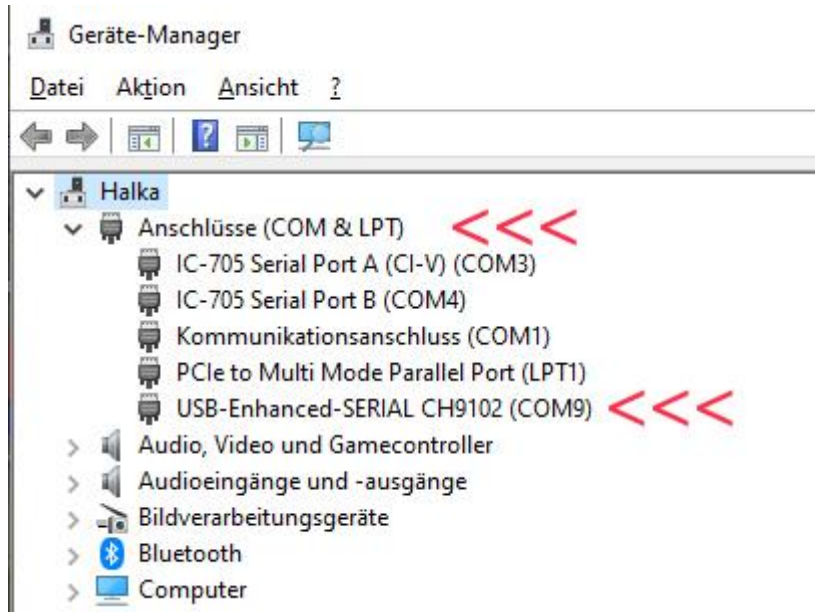
1. Za pierwszym razem konieczne jest pobranie programu *ESPHome-Flasher* służącego do wpisania oprogramowania do modułu *LoRa*.

Z adresu <https://github.com/esphome/esphome-flasher/releases/tag/1.4.0> pobierany jest plik *ESPHome-Flasher-1.4.0-Windows-x64.exe*.



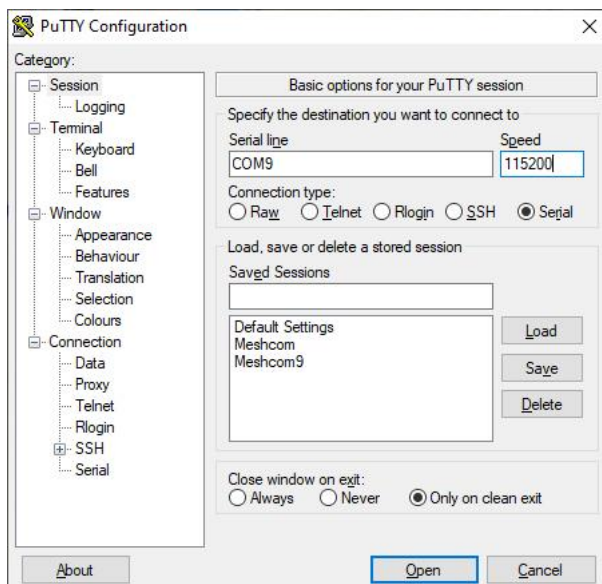
Rys. 2.2.1. Ładowanie oprogramowania przy użyciu programu *ESPHome-Flasher*

Program nie wymaga instalacji i skorzystanie z niego polega jedynie na wywołaniu pliku *ESPHome-Flasher-1.4.0-Windows-x64.exe*. (lub o podobnej nazwie w zależności od wersji programu). Konieczne jest podanie numeru złącza COM używanego przez moduł *LoRa* i znalezienie na dysku pliku oprogramowania modułu (po naciśnięciu ekranowego przycisku „Browse” – „Przełóżaj”). Numer jest widoczny w *Menadżerze Urządzeń* po podłączeniu modułu do złącza USB komputera. Numery złączy są przydzielane przez system Windows i mogą się różnić od podanego w przykładach. Jeżeli po podłączeniu modułu *LoRa* w menadżerze nie pojawia się dodatkowe złącze COM konieczne jest zainstalowanie sterownika CP210x lub CP910x – zależnie od modelu. Sterowniki te są dostępne w Internecie i najłatwiej je znaleźć za pomocą wyszukiwarki. Załadowanie pliku do pamięci mikroprocesora wymaga naciśnięcia przycisku *Flash ESP* na ekranie.

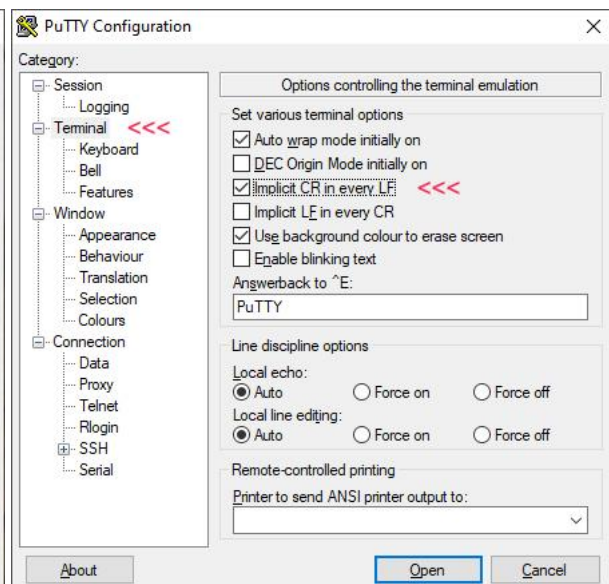


Rys. 2.2.2. Złącze COM widoczne jest w *Menadżerze Urządzeń* w spisie złączy COM i LPT. W tym przykładzie jest to złącze COM9

2. Pobranie pliku oprogramowania węzła sieci dla modułu LoRy z witryny <https://icssw.org/download-category/meshcom-4-0-client-firmware/>. Dostępne są w niej różne wersje oprogramowania dla poszczególnych modeli płytek LoRy. Należy zwrócić uwagę na wybranie pasującego pliku, gdyż w przeciwnym przypadku program będzie działał tylko częściowo albo nie będzie działał w ogóle z powodu różnic w wyposażeniu modułów. Obecnie do wyboru są wersje dla modułów T-Beam, T-Lora, Heltec i Rak4631 (patrz p. 4). Przy zakupie modułów należy koniecznie wybierać moduły standardu EU433 – czyli pracujące w paśmie 433 MHz. W chwili powstawania obecnego skryptu aktualna jest wersja 4.23.
3. Po załadowaniu oprogramowania do mikroprocesora należy go ponownie wystartować.



Rys. 2.2.3. Parametry transmisji



Rys. 2.2.4. Dodawanie znaków CR do LF

4. Dla modułu RAK Wisblock (RAK-4631-C + RAK 19007) należy pobrać plik z rozszerzeniem *uf2*. Po podłączeniu modułu do PC za pomocą kabla z wtyczką USB-C należy dwukrotnie szybko nacisnąć przycisk obok gniazda USB-C. Po zapaleniu się zielonej diody moduł jest widoczny dla Windows jako zewnętrzny nośnik. Należy na niego skopiować pobrany plik *uf2*. Po zakończeniu kopiowania moduł startuje automatycznie.

5. Do komunikacji komputera z modulem *LoRy* wystarczy dowolny prosty program terminalowy w rodzaju *PuTTY*. Konieczne jest wybranie w nim połączenia szeregowego (punkt „Serial”), pasującego złącza COM (w przykładzie COM9) i szybkości transmisji 115200 bit/s (rys. 2.2.3). Prawidłowe wyświetlanie odpowiedzi mikroprocesora w oknie terminalowym *PuTTY* wymaga też zaznaczenia w konfiguracji punktu dodającego powrót wózka (CR) do każdego znaku nowej linii (LF) – rys. 2.2.4.

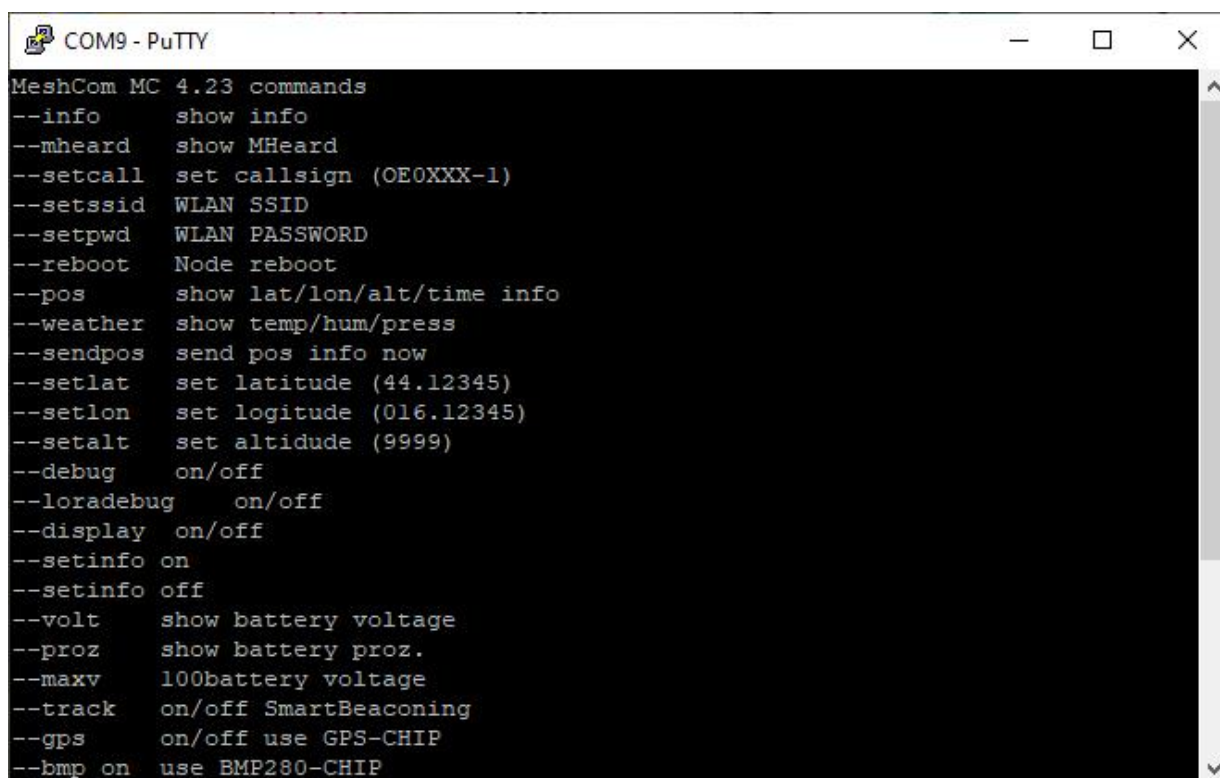
6. Wszystkie polecenia podawane przez złącze szeregowo do modułu *LoRy* są poprzedzone podwójnym myślnikiem. Na początek konieczne jest wprowadzenie własnego znaku wywoławczego z obowiązkowym rozszerzeniem. Zasadniczo dopuszczalnym zakresem rozszerzeń jest -1 – -99, ale w przypadku konieczności zapewnienia kompatybilności ze standardem APRS zakres zawęża się do -1 – -15. Rozszerzenie -12 jest zarezerwowane dla bramek internetowych lub hamnetowych połączonych z serwerem MQTT. Dane pozycyjne do serwera *aprs.fi* są transmitowane tylko wtedy, gdy znak stacji zawiera rozszerzenie.

Do wpisania znaku służy polecenie `--setcall`, przykładowo `--setcall OE1KDA-9`. Po 10 sekundach od wpisania znaku program automatycznie startuje od nowa.

7. Użytkownicy korzystający z modułów *LoRy* nie posiadających odbiornika GPS powinni także wprowadzić własne współrzędne geograficzne korzystając w tym celu z poleceń `--setlat`, przykładowo dla szerokości geograficznej OE1KDA `--setlat 16.35`, `--setlon`, przykładowo dla długości geograficznej OE1KDA `--setlon 48.137` i dla wysokości n.p.m. `--setalt`, przykładowo `--setalt 200`.

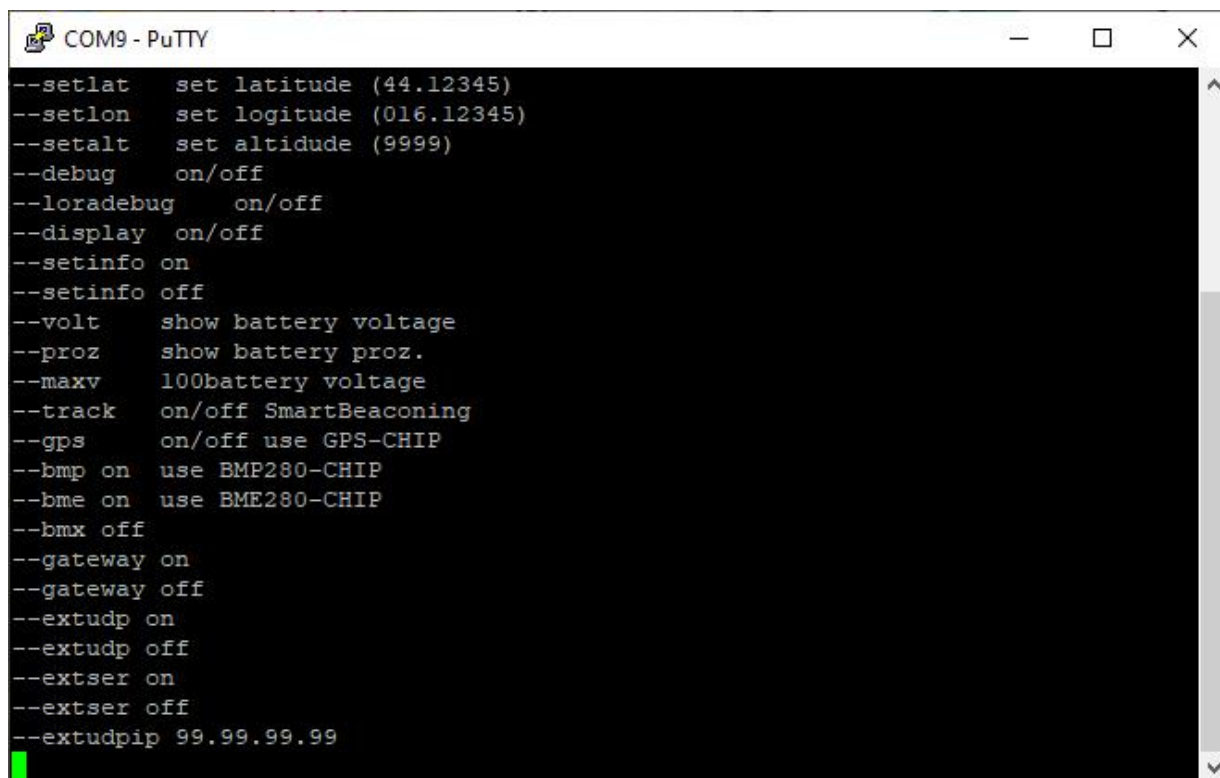
8. Częstotliwość pracy 433,175 MHz, szerokość pasma 250 kHz, stopa korekcji 4/6 i współczynnik rozpraszania 11 (= SF2048) są standardowo wpisane do programu i nie mogą być zmieniane przez użytkownika. W stosunku do poprzednich ustawień 125 kHz, 12, 4/8 uzyskuje się większą szybkość transmisji przy około 4-krotnym obniżeniu czułości.

9. Do sprawdzenia wprowadzonych danych służy polecenie `--info`. Polecenie `--help` informuje o dalszych dostępnych poleceniach i ich składni. Przykłady na ilustracjach dotyczą konfiguracji węzła sieci radiowej w wersji oprogramowania 4.23. Bramka do serwera MQTT nie została włączona.



```
COM9 - PuTTY
MeshCom MC 4.23 commands
--info      show info
--mheard    show MHeard
--setcall   set callsign (OE0XXX-1)
--setssid   WLAN SSID
--setpwd    WLAN PASSWORD
--reboot    Node reboot
--pos       show lat/lon/alt/time info
--weather   show temp/hum/press
--sendpos   send pos info now
--setlat    set latitude (44.12345)
--setlon    set logitude (016.12345)
--setalt    set altitude (9999)
--debug     on/off
--loradebug on/off
--display   on/off
--setinfo   on
--setinfo   off
--volt      show battery voltage
--proz      show battery proz.
--maxv      100battery voltage
--track     on/off SmartBeaconing
--gps       on/off use GPS-CHIP
--bmp       on use BMP280-CHIP
```

Rys. 2.2.5. Tekst pomocy w oknie *PuTTY*

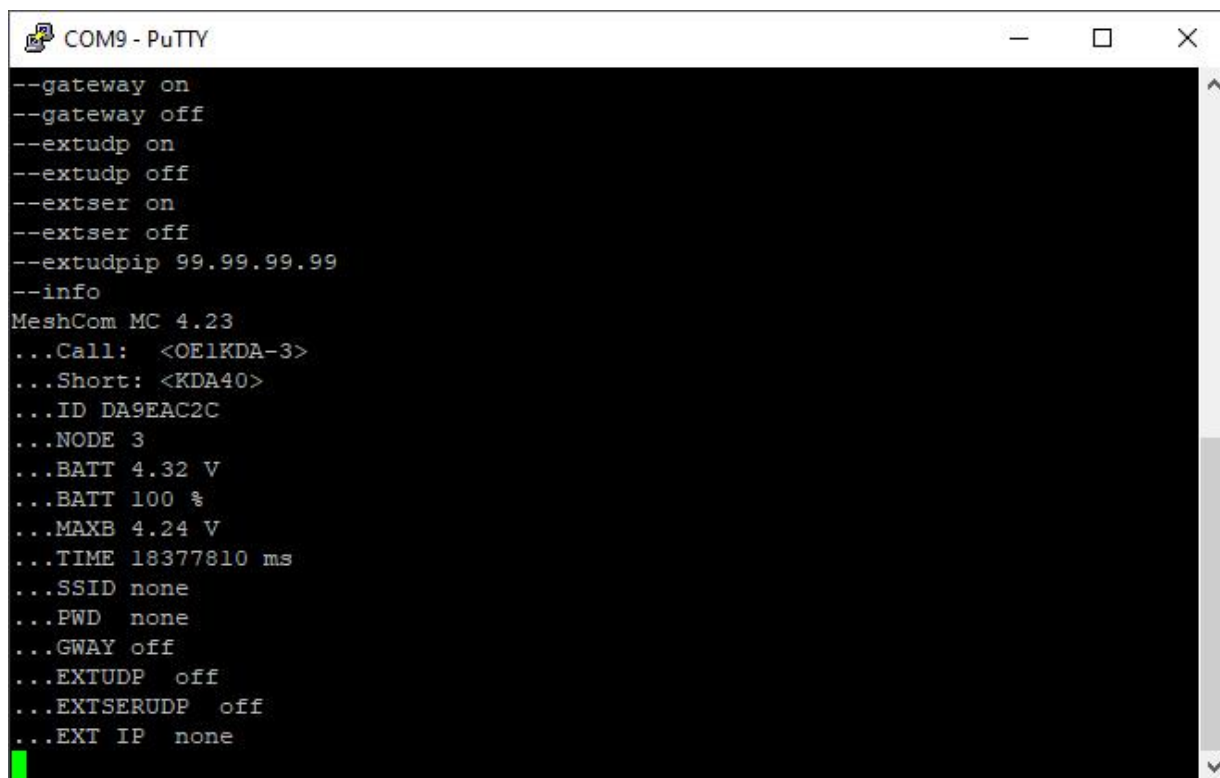


```

COM9 - PuTTY
--setlat  set latitude (44.12345)
--setlon  set logitude (016.12345)
--setalt  set altitude (9999)
--debug   on/off
--loradebug  on/off
--display on/off
--setinfo on
--setinfo off
--volt    show battery voltage
--proz    show battery proz.
--maxv    100battery voltage
--track   on/off SmartBeaconing
--gps     on/off use GPS-CHIP
--bmp on  use BMP280-CHIP
--bme on  use BME280-CHIP
--bmx off
--gateway on
--gateway off
--extudp  on
--extudp  off
--extser  on
--extser  off
--extudpip 99.99.99.99

```

Rys. 2.2.6. Dalszy ciąg tekstu pomocy



```

COM9 - PuTTY
--gateway on
--gateway off
--extudp  on
--extudp  off
--extser  on
--extser  off
--extudpip 99.99.99.99
--info
MeshCom MC 4.23
...Call: <OE1KDA-3>
...Short: <KDA40>
...ID DA9EAC2C
...NODE 3
...BATT 4.32 V
...BATT 100 %
...MAXB 4.24 V
...TIME 18377810 ms
...SSID none
...PWD none
...GWAY off
...EXTUDP off
...EXTSERUDP off
...EXT IP none

```

Rys. 2.2.7. Informacja o stacji skonfigurowanej jako węzeł radiowy

10. Po sprawdzeniu prawidłowości konfiguracji węzeł może rozpocząć regularną pracę w eterze. Spis odbieranych stacji jest wywoływany za pomocą polecenia `--mh` (`--mheard`). Oprócz tego na początek interesujące są polecenia:

`--sendpos` – nadanie geograficznego położenia stacji w oparciu o dane wprowadzone w sposób podany powyżej albo odczytane z odbiornika GPS;

:tekst – nadanie tekstu wprowadzonego przez program terminalowy i złącze szeregowo. Należy zwrócić uwagę, że jest on poprzedzony dwukropkiem.

```

COM9 - PuTTY
- /
-----
MHeard call | date | time | typ | source hardware | mod | rssi | snr |
-----
MHeard call | date | time | typ | source hardware | mod | rssi | snr |
-----
MHeard call | date | time | typ | source hardware | mod | rssi | snr |
-----
/MSG:2F..not sent
/MSG:2F..not sent
mMSG:6D..not sent
hMSG:68..not sent
/MSG:2F..not sent
/MSG:2F..not sent
mMSG:6D..not sent
hMSG:68..not sent
-----
MHeard call | date | time | typ | source hardware | mod | rssi | snr |
-----
OE1KDA-9 | 2023-01-01 | 05:42:47 | POS | TLORA_V2_1_lp6 | 3 | -61 | 8 |
-----

```

Rys. 2.2.8. Spis odbieranych stacji (--mheard)

11. Konfiguracja bramki dostępowej do serwera MQTT wymaga podania dodatkowo poleceń:

- setssid – wprowadza nazwę lokalnej sieci WLAN,
- setpwd – wprowadza hasło dostępu do lokalnej sieci,
- setinet – wybór dostępu do serwera MQTT przez Internet, jest to ustawienie domyślne,
- sethamnet – wybór dostępu do serwera MQTT przez Hamnet, pod warunkiem istnienia odpowiedniej infrastruktury hamnetowej w otoczeniu; austriacki serwer MQTT jest w Internecie dostępny pod adresem IP 89.185.97.38, w austriackim Hamnecie serwer – pod adresem IP 44.143.8.143. W Polsce na pewno będą to inne adresy, chociaż dostęp internetowy do serwera austriackiego może być interesujący w początkowej fazie rozwoju sieci w kraju;
- gateway on – włączenie bramki.

Bramka w lokalnej sieci korzysta z adresu przyznanego przez serwer DHCP.

Pełny zestaw poleceń dostępnych przez złącze szeregowo podano w punkcie 2.6.

12. Instalacja oprogramowania dla płytki Heltec w wersji V3 wymaga pobrania z witryny <https://icssw.org/esp32s3-python-tool-installation/> plików *bootloader.bin*, *boot_app0.bin*, *partitions.bin* i pliku oprogramowania wewnętrznego (dokładna nazwa zależy od wersji oprogramowania) *FW_CL_heltec_v3_4.23.bin*.

Zapis danych w pamięci płytki odbywa się za pomocą następujących czterech poleceń dla *Windows-PowerShell*a (rzeczywisty numer złącza COM należy odczytać w Menadżerze Urządzeń, w przykładzie jest to złącze COM3):

```
python.exe -u esptool --chip esp32s3 -port COM3 write_flash --flash_mode keep --flash_size detect 0x0 bootlader.bin
```

```
python.exe -u esptool --chip esp32s3 -port COM3 write_flash --flash_mode keep --flash_size detect 0xe000 boot_app0.bin
```



```
python.exe -u esptool --chip esp32s3 -port COM3 write_flash --flash_mode keep --flash_size detect 0x8000 partitions.bin
python.exe -u esptool --chip esp32s3 -port COM3 write_flash --flash_mode keep --flash_size detect 0x10000 FW_CL_heltec_v3_4.23.bin
```

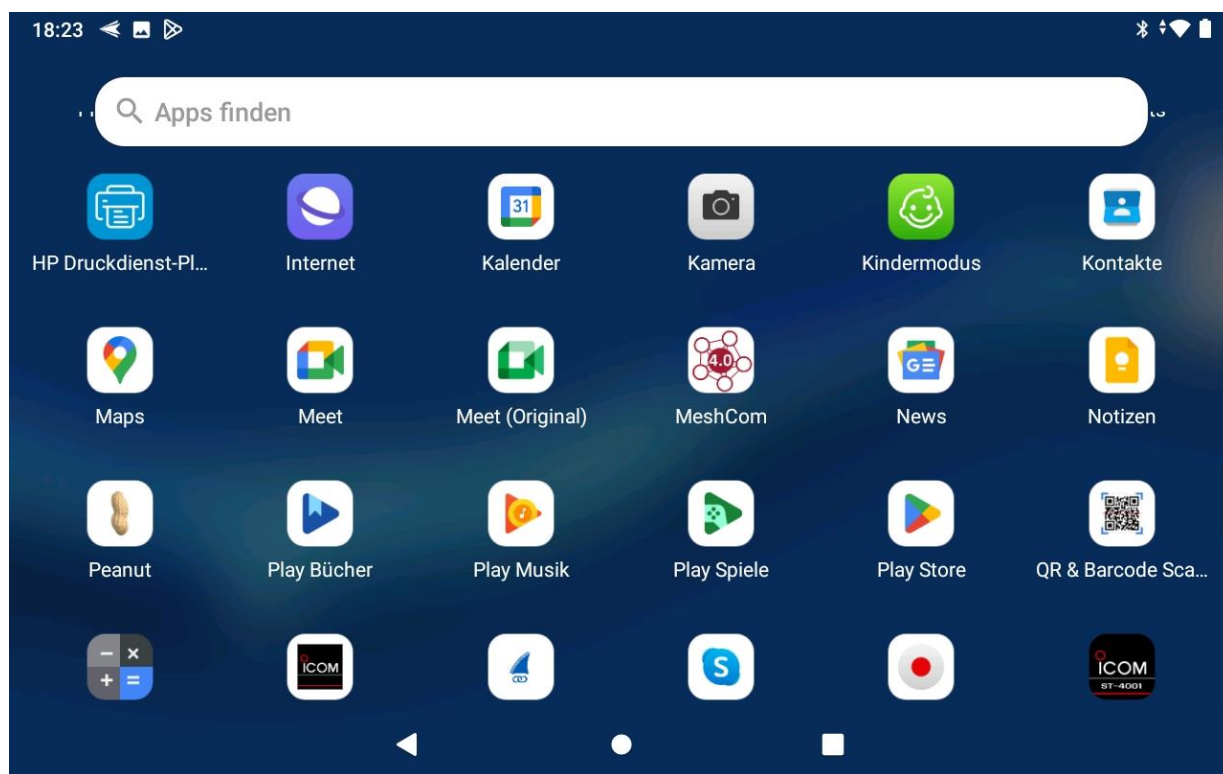
Po wpisaniu całości komunikacja z płytką Heltec V3 może się odbywać identycznie jak dla pozostałych modeli przy użyciu programu terminalowego *PuTTY* i programu androidowego *MeshCom*.

2.3. Androidowy program *MeshCom*

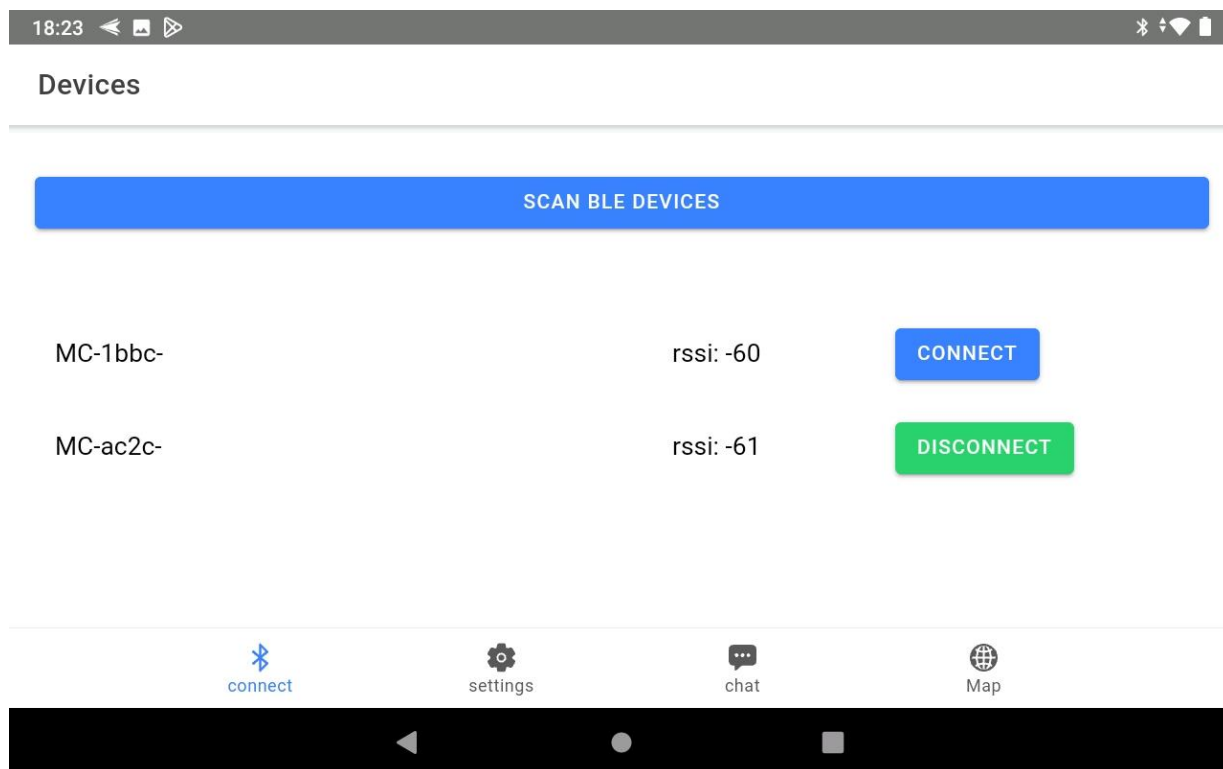
Program komunikacyjny *MeshCom* dla komputerów i telefonów androidowych jest dostępny na stronie <https://icssw.org/meshcom-app/>. Program umożliwia wymianę krótkich wiadomości w stylu *whatsapp* i śledzenie położenia stacji sieci na mapie. Jego instalacja jest przydatna tylko jeżeli użytkownik chce korzystać z wymiany wiadomości w tym sposób. Komunikacja komputera z modulem *LoRy* odbywa się przez złącze *Bluetooth*. Przed skorzystaniem z programu konieczne jest *bluetoothowe* sparowanie komputera androidowego z modulem *LoRy*. W obecnych wersjach oprogramowania kodem dostępowym BT jest 000000. W przypadku zainstalowania nowej wersji oprogramowania modułu *LoRy* lub nowej wersji *MeshComu* konieczne może być skasowanie danych dostępowych BT i ponowne sparowanie urządzeń.

Obsługa *MeshComu* jest nieskomplikowana, a korzystanie z jego funkcji w dostatecznym stopniu wyjaśniają przedstawione na ilustracjach 2.3.1 – 2.3.6 ujęcia z ekranu. Przykłady odnoszą się do wersji 4.2.

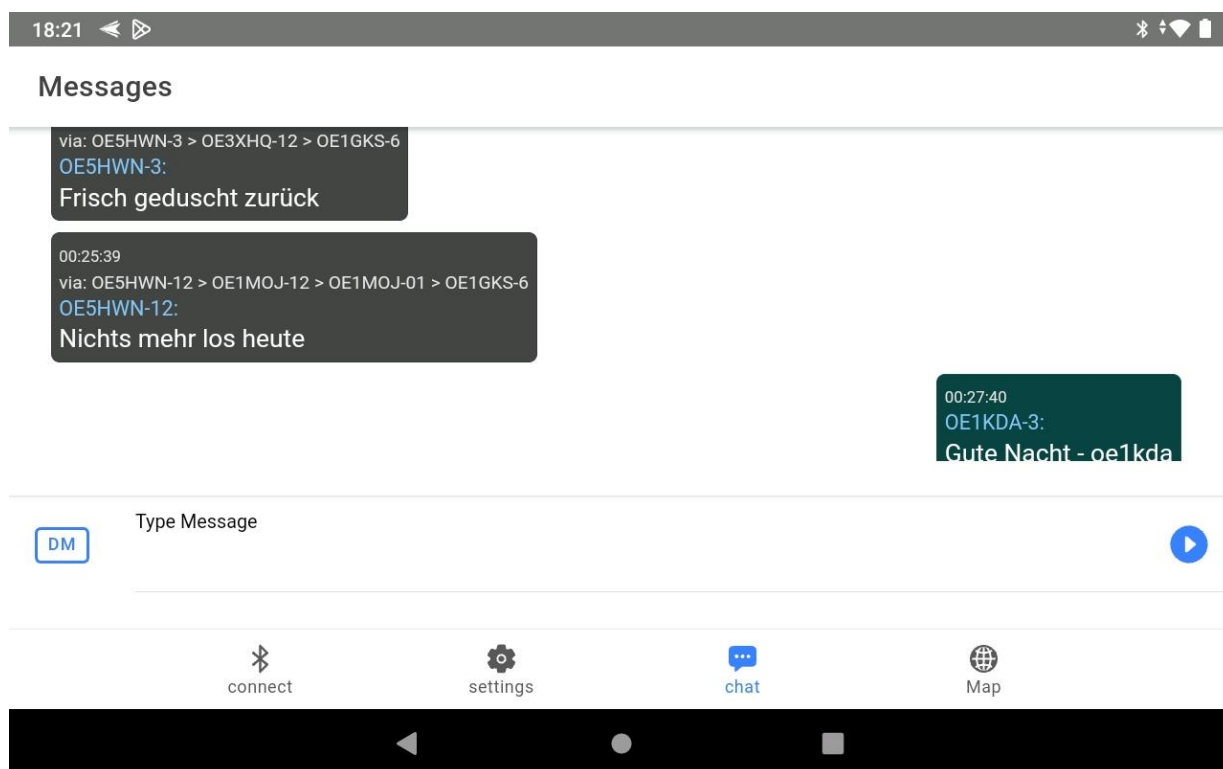
W widocznym na ilustracji 2.3.4 polu wiadomości tekstowych można także wpisywać polecenia dla modułu *LoRy* poprzedzone jak w poprzednim przypadku podwójnym myślnikiem. Pełny zestaw poleceń zawiera tabela w punkcie 2.6.



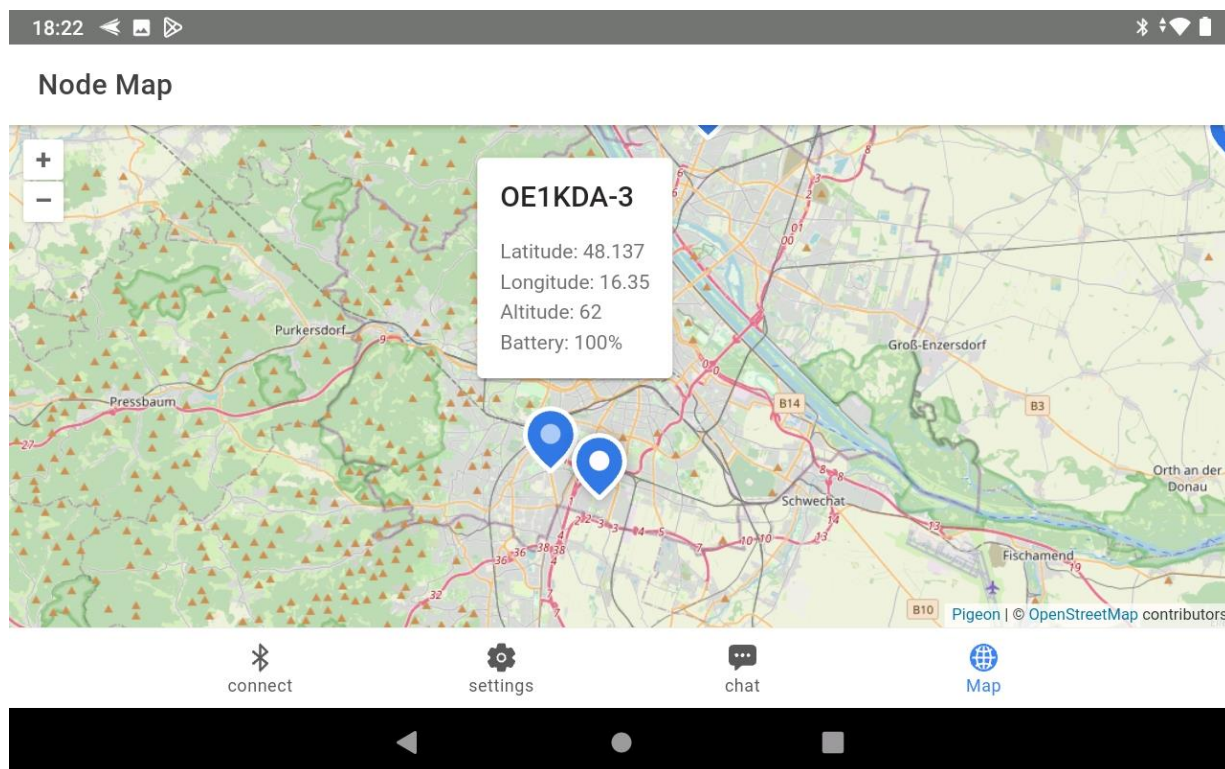
Rys. 2.3.1. Wywołanie programu *MeshCom* na komputerze androidowym (symbol na czwartej pozycji w drugiej linii symboli)



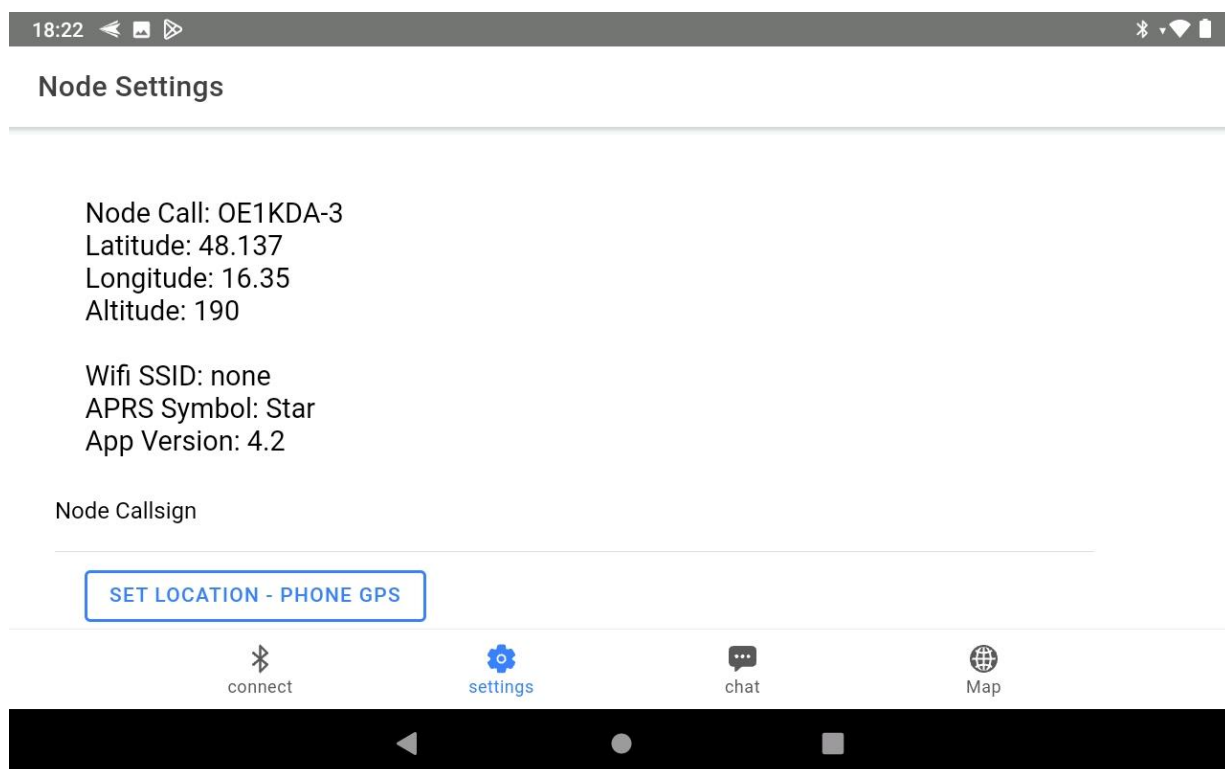
Rys. 2.3.2. Okno połączenia BT. Należy wybrać pożądane urządzenie i nacisnąć na ekranie przycisk „Connect”. Przy pierwszym połączeniu (parowaniu) otwierane jest okienko, w którym należy podać kod dostępu do modułu *LoRy* – 000000. Do rozłączenia służy przycisk „Disconnect”. W przykładzie program jest połączony z urządzeniem z drugiej linii



Rys. 2.3.4. Okno komunikacji tekstowej. Wiadomości albo poprzedzone podwójnym myślnikiem polecenia wpisuje się do pola z podpisem „Type Message” przy użyciu klawiatury ekranowej. Do wysłania służy niebieski przycisk ze strzałką



Rys. 2.3.5. Położenie stacji na tle mapy. Dziwnym trafem wysokość podana w konfiguracji modułu Lory 190 m została gdzieś po drodze potraktowana jako wysokość w stopach i ponownie przeliczona – stąd wyszła na ekranie nieprawidłowa wartość 62 m. Okienko informacyjne otwiera się po naciśnięciu lokalizacji stacji na ekranie



Rys. 2.3.6. Okno konfiguracji programu. Tutaj wyświetlana jest prawidłowa wysokość n.p.m.

2.4. Wymiana komunikatów z innymi systemami

W wymianie meldunków pomiędzy stacjami sieci *MeshCom* i innymi systemami krótkofalarskimi przyjęto następujące formaty komunikatów:

1. Meldunki (komunikaty) klientów APRS do węzłów *MeshCom*

Dowolny klient APRS może przysyłać meldunki do węzłów *MeshCom*, ale pod warunkiem, że adresat jest w tym czasie czynny w sieci *MeshCom*. Długość wiadomości jest ograniczona do 180 znaków alfanumerycznych. Reszta jest obcinana. Znak adresata musi zawierać rozszerzenie, a więc dopuszczalny jest znak OE1KDA-1, ale nie OE1KDA.

2. Meldunki (komunikaty) od węzłów *MeshCom* do klientów APRS mają format
:<adres docelowy>:<tekst>

Przykładowo

:OE1KDA-7:Pozdrowienia

Należy zwrócić uwagę na obowiązkowe dwukropki.

3. Meldunki (komunikaty) do APRS2SOTA wymagają zarejestrowania znaku nadawcy u G0LGS (<https://www.sotaspots.co.uk/>).

Zarejestrowani nadawcy mogą wówczas nadawać komunikaty adresowane do APRS2SOTA mające następujący format:

:APRS2SOTA:<numer rej. szczytu> <częstotliwość> <emisja> [znak] [uwagi]

Np.

:APRS2SOTA:OE/OO-303 145.500 FM OE5JFE/P Spot via Meshcom

4. Poczta elektroniczna do Internetu. Wiadomości mają następujący format:

:EMAIL:<adres odbiorcy> <tekst>

Należy zwrócić uwagę na znak odstępu między adresem odbiorcy i tekstem wiadomości.

2.5. Urządzenia dodatkowe

1. Dodatkowy przycisk zwierany do masy po naciśnięciu.

W modułach T-Beam jest to przycisk środkowy spośród trzech zawartych na płycie.

W modułach Heltec V2 i V3 jest podłączony do wyprowadzenia nr 0.

W modułach T-Lora i ESP32-E22 jest połączony do wyprowadzenia nr 12.

2. Czujniki meteorologiczne BMP280 (pomiar ciśnienia i temperatury) i BME280 (pomiar ciśnienia, temperatury i wilgotności względnej).

W modułach T-Beam, T-Lora T3_V1.6.1 Heltec V2 i V3 podłączenia do wyprowadzeń magistrali I2C jak dla wyświetlacza. Napięcie zasilania jest pobierane z wyprowadzenia 3,3 V.

Do włączenia czujników służą odpowiednio polecenia

--*bmp on*, lub

--*bme on*.

Konieczne jest ponowne wystartowanie modułu. Odczyt danych meteorologicznych następuje po podaniu polecenia --*wheather* przez złącze szeregowo albo za pomocą programu androidowego.

Do całkowitego wyłączenia czujników służy polecenie --*bmx off*.

3. Odbiorniki GPS. Dopuszczalne są jedynie odbiorniki zasilane napięciem 3,3 V. Konieczne jest podłączenie masy, napięcia zasilania 3,3 V oraz wyprowadzeń TX i RX.

Odbiornik GPS6MV2 HW-248 – UBLOX NEO-6M-0-001

Moduł Heltec V2

RX <-- wyprowadzenie 2 (TX z punktu widzenia modułu),

TX --> wyprowadzenie 17 (RX z punktu widzenia modułu),

Moduł Heltec V3

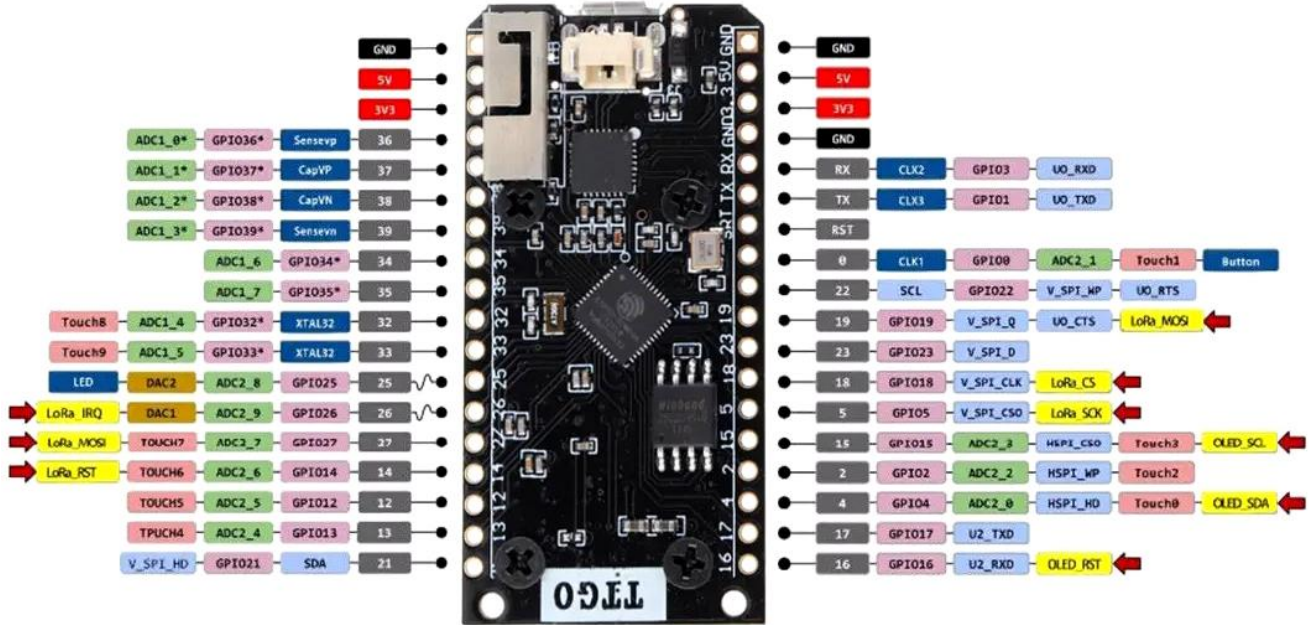
RX <-- wyprowadzenie 37 (TX z punktu widzenia modułu),

TX --> wyprowadzenie 36 (RX z punktu widzenia modułu),

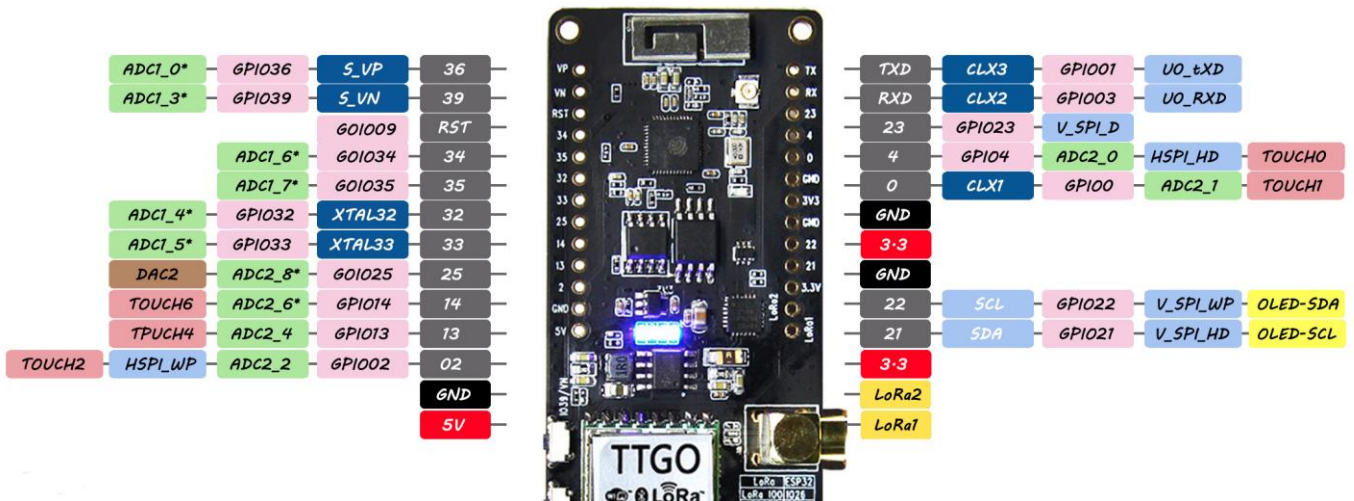
Moduł ESP32-E22

RX <-- wyprowadzenie 16 (TX z punktu widzenia modułu),

TX --> wyprowadzenie 17 (RX z punktu widzenia modułu),
 Moduł T-Lora T3_V1.6.1
 RX <-- wyprowadzenie 13 (TX z punktu widzenia modułu),
 TX --> wyprowadzenie 15 (RX z punktu widzenia modułu).



Rys. 2.5.1. Numeracja wyprowadzeń na płytce T-Lora. Wygląd płytki może się różnić w zależności od producenta



Rys. 2.5.2. Numeracja wyprowadzeń na płytce T-Beam. Wygląd płytki może się różnić w zależności od producenta

2.6. Zestaw poleceń MeshComu

Tabela 2.6.1

Zestaw poleceń podawanych wyłącznie przez złącze szeregowo

Polecenie	Znaczenie
--help	Wywołanie pomocy
--info	Wywołanie informacji o bieżących ustawieniach
--pos	Wywołanie informacji o współrzędnych geograficznych stacji
--mheard (--mh)	Wywołanie spisu stacji odebranych od momentu uruchomienia programu
--setcall <znak>	Wprowadzenie znaku wywoławczego z rozszerzeniem. Po 10 sekundach od wpisania znaku następuje automatyczny start programu
--setssid <nazwa>	Podanie nazwy lokalnej sieci WiFi
--setpwd <hasło>	Podanie hasła dostępu do lokalnej sieci
--sethamnet	Dostęp do serwera MQTT przez Hamnet. Po 10 sekundach następuje automatyczny start programu
--setinet	Dostęp do serwera MQTT przez Internet. Jest to ustawienie domyślne. Po 10 sekundach od podania polecenia następuje automatyczny start programu
--gateway on	Włączenie bramki – dostępu do serwera MQTT przez sieć WiFi
--gateway off	Wyłączenie bramki
--setinfo off	Wyłączenie wyświetlania komunikatów LoRa/UDP w programie terminalowym. Ustawienie domyślne
--setinfo on	Włączenie wyświetlania powyższych komunikatów

Tabela 2.6.2

Zestaw poleceń dostępnych na złączach szeregowym i Bluetooth

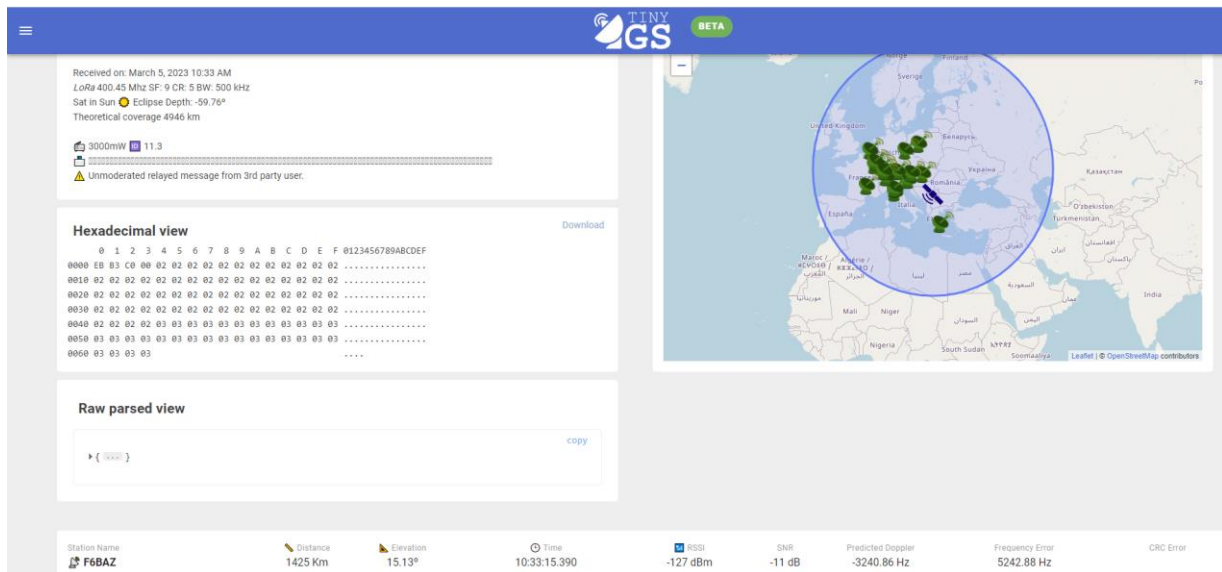
Polecenie	Znaczenie
--volt	Wyświetlanie napięcia w voltach na wyświetlaczu modułu
--proc	Wyświetlanie stanu zasilania (naładowania akumulatora) w procentach
--maxv 99.9	Podanie maksymalnego napięcia naładowanego akumulatora. Istotne dla wskazań procentowych
--button on	Aktywacja przycisku w modelach T-Beam, T-Lora, ESP32/E22, Heltec V2/V3
--button off	Wyłączenie przycisku j.w. Gdy przycisk nie jest podłączony należy wyłączyć odpytywanie jego stanu.
--all	Wyświetlanie na wyświetlaczu meldunków pozycji aż do nadejścia nowego meldunku
--msg	Wyświetlanie tylko odebranych meldunków
--display on/off	Włączenie lub wyłączenie wyświetlacza modułu LoRa
--sendpos	Nadawanie meldunku zawierającego własne współrzędne
--setlat xx.xxxxxx	Wprowadzenie długości geograficznej w stopniach (przy braku odbiornika GPS). Wartości dodatnie – długości wschodnie
--setlon xxx.xxxxxx	Wprowadzenie szerokości geograficznej w stopniach jak powyżej. Wartości dodatnie – szerokości północne
--setalt xxxxx	Wprowadzenie wysokości w metrach n.p.m.
--track on	Włączenie wskazań dla inteligentnej (zależnej od szybkości ruchu) transmisji współrzędnych
--track off	Wyłączenie powyższych wskazań
--wx	Włączenie wskazań meteorologicznych. Warunkiem jest podłączenie czujnika BMP280 lub BME280
--gps on	Włączenie odczytu odbiornika GPS. Dotyczy to dodanego odbiornika
--gps off	Wyłączenie odczytu odbiornika GPS. Zmniejsza zużycie prądu z akumulatora
--reboot	Ponowne załadowanie i wystartowanie programu

2.7. Aktywność

MeshCom 4.0 Server														
NOT REGISTERED (D.05.25) 2023-07-11 18:10:34														
MeshCom NODES														
GATEWAYS	NR.	LASTTIME	CALL	VIA GW	HOP	SNR	RSSI	HW	FW	LAT	LON	ALT	BAT	LAST GPS
NODES	1	2023-07-11 18:10:25	9V1LH-12	9V1LH-12				TLORA T3_V1.6.1	4.20	01.3770	103.9417	42m	100%	0:13:51
ACTIVITY	2	2023-07-11 18:10:23	BOT GATE	BOT GATE										
INTERLINK	3	2023-07-11 18:10:21	DB0WTH-99	DB0WTH-99				HELTEC V2	4.13	49.4080	09.8247	465m	100%	0:13:31
UDP-GW	4	2023-07-11 17:32:24	DL45DR-44	DB0WTH-99	4	-2	-101	E22	4.23	49.4978	09.7673			0:37:56
	5	2023-07-11 17:22:30	DL45DR-5	DB0WTH-99	3	-2	-102	TBEAM T22_V1.1	4.23	49.4978	09.7662	152m	78%	0:47:45
EXTERN	6	2023-07-11 18:10:30	DD1WKS-12	DD1WKS-12				TLORA T3_V1.6.1	4.23	49.4268	06.8320			0:25:53
LOG	7	2023-07-11 17:42:48	DD7MH-1	DD7MH-12	4	6	-50	RAK4631	4.23	48.1650	12.6167	462m	74%	0:27:39
	8	2023-07-11 18:10:11	DD7MH-12	DD7MH-12				RAK4631	4.20	48.1650	12.6167	466m	93%	0:27:34
DENY	9	2023-07-11 17:42:34	DD7MH-3	DD7MH-12	4	7	-55	RAK4631	4.23	48.1652	12.6167	477m	43%	0:27:53
SITE-INFO	10	2023-07-11 18:03:47	DG1MPX-17	DG1MPX-99	4	10	-36	TBEAM T22_V1.1	4.10	48.1032	11.5888	560m	100%	0:06:47
	11	2023-07-11 18:10:25	DG1MPX-99	DG1MPX-99				TBEAM T22_V1.1	4.11	48.1037	11.5888	545m		0:08:41
WIKI	12	2023-07-11 18:10:31	DG3SBI-99	DG3SBI-99				TBEAM T22_V1.1	4.23	48.8417	09.7098			0:25:39
	13	2023-07-11 18:10:31	DG5NET-99	DG5NET-99				TLORA T3_V1.6.1	4.23	49.5152	10.8783	299m		0:10:02
MeshCom	14	2023-07-11 17:46:21	DG9FFM-1	DG9FFM-12	5	13	-38	TBEAM 1268	4.23	49.7567	09.5142	155m	100%	0:24:07
	15	2023-07-11 18:10:29	DG9FFM-12	DG9FFM-12				HELTEC V2	4.23	49.7567	09.5142	160m		0:19:58
OE1KBC	16	2023-07-11 17:48:07	DG9FFM-3	DG9FFM-12	5	6	-101	HELTEC V2	4.23	49.7598	09.5105	160m		0:22:21
OE1KFR	27	2023-07-11 18:13:49	OE1EMP-13	OE1EMP-13				TBEAM T22_V1.1	4.23	48.1332	16.4667	160m	87%	0:18:31
	28	2023-07-11 18:14:06	OE1GKS-12	OE1GKS-12				TBEAM T22_V1.1	4.23	48.1480	16.3208	242m	98%	0:14:45
	29	2023-07-11 18:08:47	OE1GKS-4	OE1GKS-12	5	11	-70	TBEAM T22_V1.1	4.23	48.1480	16.3208	242m	100%	0:05:19
	30	2023-07-11 17:27:55	OE1KDA-3	OE1GKS-12	5	-16	-127	TLORA T3_V1.6.1	4.23	48.1370	16.3500	189m	100%	0:45:56
	31	2023-07-11 16:36:15	OE1UHA-5	OE1GKS-12	5	-11	-121	TLORA T3_V1.6.1	4.21	48.1668	16.3185		4%	outdated

Rys. 2.7.1. Fragment spisu czynnych stacji na pulpicie serwera *MeshCom* pod adresem internetowym <https://srv08.oevsv.at/meshcom/>. Linia 30 dokumentuje obecność stacji OE1KDA-3. Na ilustracji opuszczono linie 17 – 26

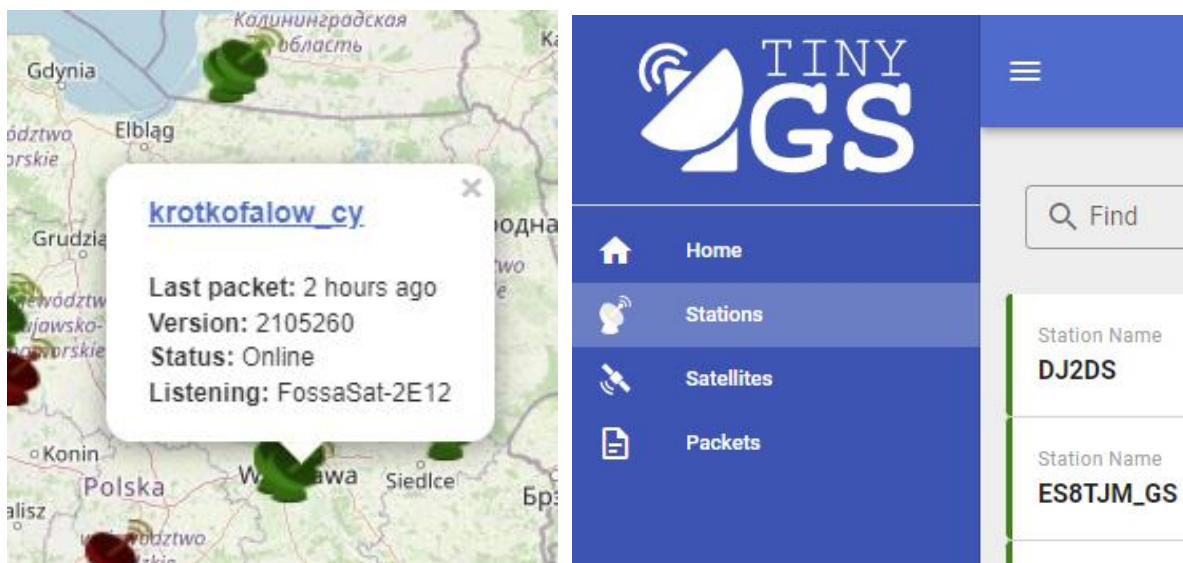
3. Odbiór satelitów „LoRa”



Rys. 3.1. Witryna *tinysgs.com*: informacje o jednym z satelitów



Rys. 3.2. Stacje odbiorcze na terenie Polski w dniu 12 lipca 2023 w godzinach popołudniowych

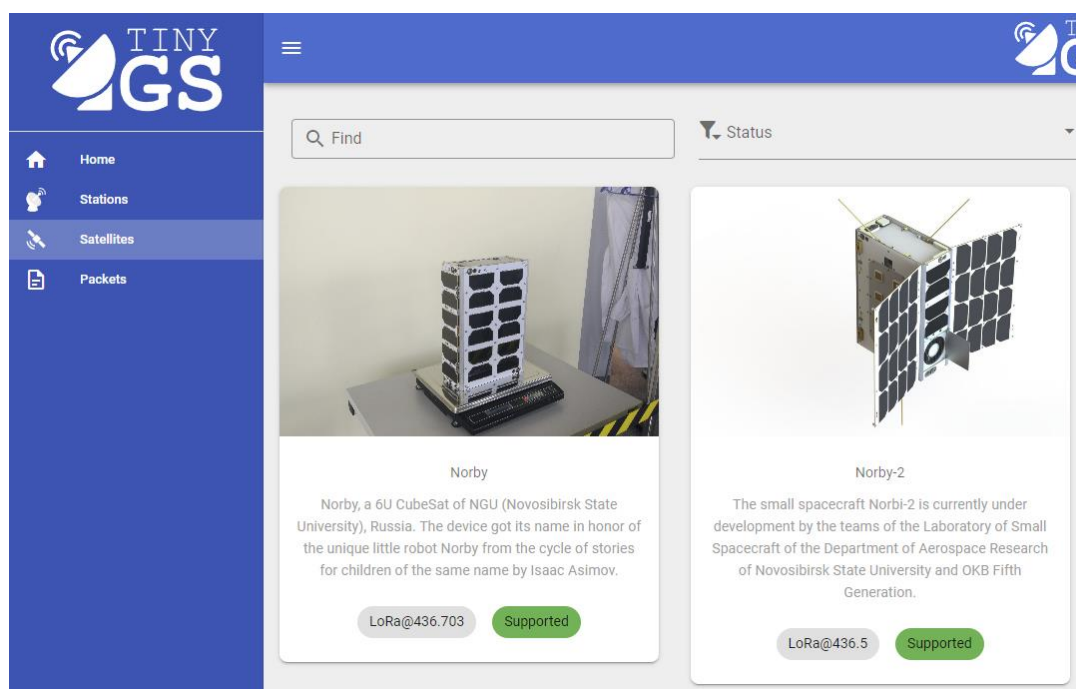


Rys. 3.3. Po naciśnięciu symbolu stacji na mapie wyświetlane są dodatkowe informacje

Rys. 3.4. Menu boczne witryny, wybrany spis stacji

Witryna [3.1] prezentuje światową sieć stacji naziemnych *TinyGS* odbierających satelity nadające dane telemetryczne w systemie LoRa, sondy meteorologiczne i nadajniki zainstalowane na pokładach innych obiektów latających. Sieć ta zapewnia ciągły odbiór danych z satelitów niezależnie od ich położenia na orbicie i nie narażając organizatorów eksperymentów na koszt utrzymywania profesjonalnych stacji naziemnych. Oprócz pokazanej na ilustracjach mapy stacji menu boczne strony (otwierane za pomocą symbolu trzech poziomych kresek) umożliwia wywołanie spisów stacji, satelitów i odebranych ostatnio pakietów. Po naciśnięciu znaku stacji w spisie otwierane jest okno zawierające informacje o jej aktywności, dane statystyczne spis ostatnio odebranych pakietów, mapa wskazująca położenie stacji i mapa nieba pokazująca aktualne położenie satelitów.

Spis satelitów pozwala na wywołanie szczegółowych informacji o każdym z nich. Na ostatniej stronie („Packets”) wyświetlany jest spis ostatnio odebranych komunikatów. Naciśnięcie komunikatu powoduje wyświetlenie informacji o satelicie i szczegółów dotyczących pakietu danych.



Rys. 3.5. Fragment spisu satelitów

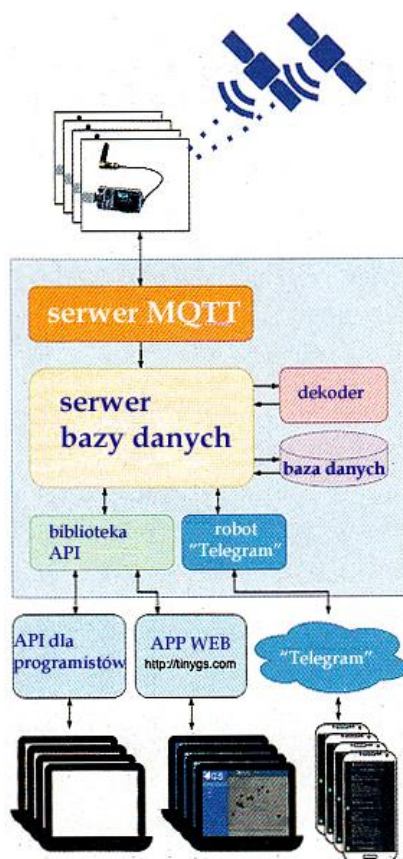
Tanie i nieskomplikowane wyposażenie stacji opiera się na modułach ESP32 LoRa (433 MHz), takich jak Heltec WiFi LoRa32 V1 i V2, TTGO Lora32 V1 i V2, T-BEAM z wyświetlaczami OLED 0,96 cala i inne ESP32 z wyświetlaczami wyposażone w nadawczo-odbiorcze moduły SX126X lub SX127X, oraz w łącznie WLAN umożliwiające połączenie internetowe z siecią *TinyGS*.

Oprogramowanie stacji odbiorczej i program ładujący z PC do mikroprocesora przez złącze USB są dostępne w witrynie [3.2]. W celu skonfigurowania go należy za pomocą przeglądarki internetowej połączyć się z serwerem HTTP modułu pod adresem 192.168.4.1. Oprogramowanie stacji można załadować także korzystając ze środowiska *PlatformIO*. Po pierwszym załadowaniu i skonfigurowaniu programu możliwe jest włączenie funkcji automatycznej aktualizacji („AutoUpdate”). Możliwa jest też instalacja przy wykorzystaniu środowiska programistycznego „Arduino IDE”. Na zakończenie konieczne jest zameldowanie stacji w sieci.

Załączone do modułów krótkie pionowe antenki są oczywiście niewystarczające. Konieczne jest użycie zewnętrznej anteny dipolowej, pionowej w rodzaju GP [3.3], kilkuelementowej anteny Yagi, anteny dwuzwojowej albo dowolnej innej anteny satelitarnej na pasmo decymetrowe. Najlepiej gdy znajduje się ona na dachu, ale dobre rezultaty (dzięki odbiciom od ścian budynków) dają też anteny balkonowe. Odebrane dane telemetryczne są przekazywane internetowo do serwera *MQTT*. Są one dostępne w Internecie pod adresem [3.1].

Obecnie możliwy jest m.in. odbiór satelitów FEES, FEES2, FossaSat 2E1 – 2E6, Gossamer, Norbi i SATLLA-2A/B. Częstotliwości pracy różnią się dla poszczególnych satelitów, ale m.in. są to 400,45 MHz, 401,7 MHz, 436,703 MHz, 436,9 MHz i 437,6 MHz. Niektóre z satelitów pracują w paśmie 915 MHz. Możliwe jest przekazywanie do sieci danych pochodzących z dowolnych satelitów LoRa i od innych obiektów latających. Dane te mogą być nadawane zresztą nie tylko w systemie *LoRa* ale i dowolnymi innymi rodzajami modulacji odbieralnymi przez moduły *LoRa*, a więc FSK, GFSK, MSK, GMSK i OOK.

Oprogramowane jest wyposażone w tryb testowy. W celu sprawdzenia pracy odbiornika konieczne jest użycie drugiego modułu *LoRa* jako źródła sygnału. Nadawane przez to źródło próbne pakiety danych nie są przekazywane do sieci.



Rys. 3.6. Struktura sieci *TinyGS* (źródło <https://tinygs.com>)



Fot. 3.7. TTGO w obudowie

Architekturę sieci przedstawiono na rysunku 3.6. Dane z satelitów są odbierane przez dużą liczbę stacji naziemnych i następnie przekazywane przez WLAN do internetowego serwera MQTT (ang. *message queing telemetry transport*). Protokół ten jest używany w transmisji danych telemetrycznych między różnymi urządzeniami także w przypadku sieci o ograniczonych możliwościach i większych opóźnieniach w transmisji. Serwer MQTT otrzymuje oprócz danych telemetrycznych informacje o satelitach, takie jak ich nazwa, częstotliwość pracy i rodzaj modulacji. Jest on połączony z głównym serwerem sieci, na którym następuje dekodowanie i zapis (archiwizacja) danych.

Dane odebrane przez własną stację i przez wszystkie inne zameldowane w sieci są udostępniane w witrynie *TinyGS*. Użytkownicy mogą selekcyjonować dane według stacji odbiorczych, satelitów i wybierać pakiety danych. Dane te mogą pochodzić również od innych obiektów np. balonów i zawierać informacje o ich lokalizacji.

Użytkownicy mogą także wywoływać informacje o aktualnie odbieralnych satelitach i o stacjach odbiorczych wraz z ich współrzędnymi geograficznymi, statystykami odbioru itp. Oprócz tego można wyświetlać położenie odbieralnych satelitów na tle mapy.

4. Próbnik połączeń „LoRy”

Skonstruowane z myślą o zastosowaniach krótkofalarskich urządzenie, opisane w „Funkamateurze” 3/2023, jest oparte na module *ESP32 LoRa V2* firmy „Heltec” i służy do sprawdzania działania i jakości połączeń radiowych w systemie „LoRa” w paśmie 433 MHz. Może ono przydać się konstruktorom urządzeń łącznościowych i programistom opracowującym ich oprogramowanie albo do badań zasięgu sieci.

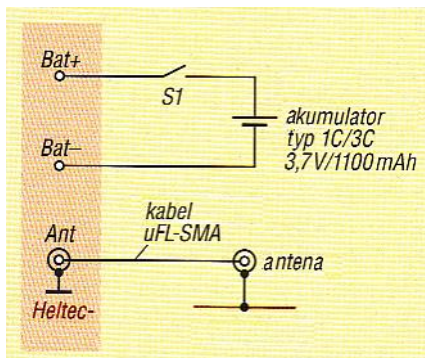
Przewidziano także możliwość wykorzystania układu do nauki telegrafii. Konieczne jest podłączenie do kontaktu GPIO25 wtórnika emiterowego, na dowolnym tranzystorze npn, sterującego słuchawką o oporności $2 \times 16 \Omega$ (rys.4.3). Generowany jest ton o częstotliwości 650 Hz.

Próbnik FTD może być sterowany przez łącza *Bluetooth* (BT) albo USB z komputera lub inteligentnego telefonu. Ustawione parametry i odbierane dane są wyświetlane na ekranie telefonu, komputera i na wyświetlaczu próbnika. Moduł jest wyposażony w wyświetlacz organiczny OLED o przekątnej 0,96 cala i złącze WiFi. Podobnie jak w module RA-02 występuje tutaj gniazdko antenowe μ FL/IPEX. Oprócz tego jest on wyposażony w gniazdko JST-PH do podłączenia akumulatora, który można ładować przez gniazdko USB (rys. 4.1).

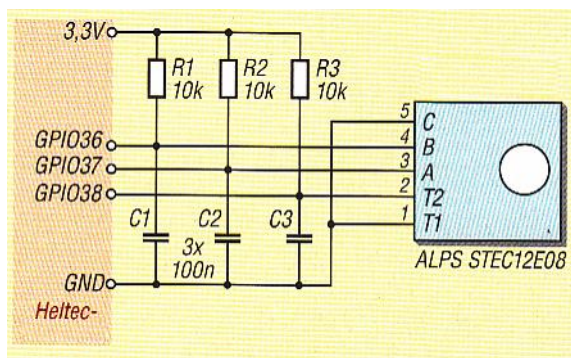
W konstrukcji przewidziano możliwość dołączenia do kontaktów GPIO36–38 obrotowego 24-pozycyjnego kodera obrotowego typu ALPS STEC 12E08 (rys. 4.2). Jego wykorzystanie jest przewidziane dopiero w przyszłych wersjach oprogramowania.

Oprogramowanie w postaci archiwum jest dostępne w Internecie w witrynie „Funkamateura” pod adresem https://www.funkamateur.de/tl_files/downloads/hefte/2023/dl2mee_field-test-device.zip.

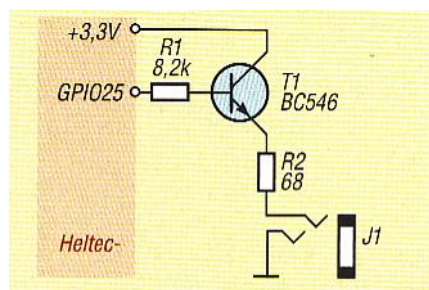
Do załadowania oprogramowania służy program narzędziowy *ESPTool.py* dla PC dostępny w witrynie <https://docs.espressif.com/projects/esptool/en/latest/esp32/esptool>.



Rys. 4.1. Podłączenie akumulatora i anteny (przeciwwaga anteny – pasek miedziany na płytce)



Rys. 4.2. Podłączenie kodera



Rys. 4.3. Podłączenie słuchawek przez wtórnik emiterowy Fot. 4.4. Widok ogólny



Na komputerze ładującym konieczne jest zainstalowanie sterownika CP2102 firmy *Silicon Labs*. W większości przypadków jest on już zainstalowany w związku z wcześniejszymi potrzebami. Po załadowaniu do procesora ESP32 i wywołaniu oprogramowanie rozpakowuje się samoczynnie i wykonuje niezbędne kroki instalacyjne. W przypadku wystąpienia błędów w czasie instalacji należy odłączyć

zasilanie od modułu, odłączyć go od złącza USB i po upływie 20 sekund po podłączeniu przerwanych połączeń wystartować ESP32 ponownie. Po uruchomieniu oprogramowania warto wprowadzić w konfiguracji własny znak wywoławczy.

Program można obsługiwać przez złącze USB albo beprzewodowo przez złącze BT przy użyciu dowolnego programu terminalowego z komputera albo telefonu androidowego, np. bezpłatnego programu *Serial Bluetooth Terminal App*. Szybkość transmisji wynosi domyślnie 115200 bodów. Korzystanie ze złącza BT wymaga uprzedniego sparowania urządzeń.

Użytkownik ma do dyspozycji następujące funkcje: wprowadzenie znaku wywoławczego (*#RP znak*), odbiór i nadawanie pakietów „LoRy” (polecenia *#ME* dla włączenia monitorowania i *#PL* dla zmiany parametrów), nadawanie tekstu radiolatarni (*#BI* odstęp czasu; *#BT* tekst komunikatu i *#BAKE* dla włączenia transmisji), funkcję stacji przekaźnikowej (*#REP*), monitorowania dwóch kanałów (*#MO...* dla wprowadzenia częstotliwości i nazw kanałów; *#MONI* dla włączenia), pracę w trybie APRS (*#APRS...*) i jako trener do nauki telegrafii (*#CW...*). Podanie poleceń bez parametrów powoduje wyświetlenie pomocy i informacji o niezbędnych parametrach.

Dla włączenia transmisji komunikatów APRS tak, aby pozycja była widoczna na mapie w witrynie *aprs.fi* służy następująca (przykładowa) sekwencja poleceń:

<i>#RP OE1KDA</i>	- wprowadzenie znaku wywoławczego
<i>#L 18</i>	- moc nadawania 18 dBm
<i>#APRS DEF</i>	- parametry APRS
<i>#APRS ANZ</i>	- położenie stacji
<i>APRS PUT</i>	- nadanie pakietu.

Dane pozycyjne są zapisane na stałe w programie i należy je dopasować do stanu rzeczywistego.

Do nauki telegrafii służy (przykładowo) następujące polecenia:

<i>#CW A 10</i>	- nadanie 10 skrótów CW
<i>#CW F 2</i>	- nadawanie grup po 5 znaków przez 2 minuty
<i>#CW M 4</i>	- nadawanie grup mieszanych.

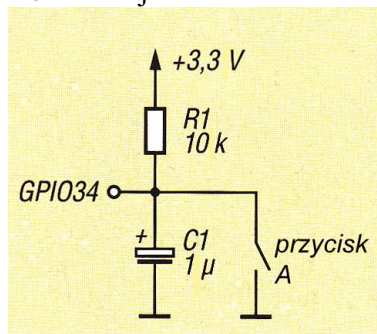
Konstrukcja i wyposażenie płytki Heltec V3 różni się znacznie od konstrukcji w wersji V2. Oznacza to, że dopasowanie oprogramowania do wersji V3 może okazać się czasochłonne. W związku z trudnościami z zakupem płytek w wersji V2 autorzy programu opracowali wersję dla płytek LILYGO-LoRa32-T3_V1.6.1. Oferuje ona prawie pełną funkcjonalność wersji pierwotnej.

Oprogramowane to jest dostępne w witrynie Funkamateura na stronie:

FUNKAMATEUR > Downloads/Archiv > Downloads zum Heft,

a plik nosi nazwę *FTDFirmwareTTGO.zip*.

Konieczne jest zainstalowanie dodatkowego przycisku podłączonego do wyprowadzenia GPIO34.



Rys. 4.5. Sposób podłączenia przycisku

Znajdujący się na płytce procesor jest trochę gorzej ekranowany aniżeli procesor na płytce Helteca V2 i może powodować słaby gwizd przy odsłuchu telegrafii. Autorzy planują opracowanie dodatkowego filtra, który eliminowałby to zjawisko.

Dodatek A

Instalacja Pythona i Esptool

Zaprogramowanie modułów *LoRy* typu Heltec w wersji V3 wymaga uprzedniego zainstalowania *Pythona* i programu ładującego *ESPTOOL*. Użytkownicy pozostałych omówionych modułów tego nie potrzebują.

Instalacja – o ile nie została wykonana wcześniej z innych powodów – składa się z następujących kroków:

1. Instalacja *Pythona* dla Windows (pakiet PIP jest instalowany automatycznie z Pythonem):
Z adresu <https://meshtastic.discourse.group/t/tutorial-setting-up-a-windows-machine-for-use-with-meshtastic-python/2872>
Należy koniecznie zaznaczyć pole „Add Python ... to PATH”.
2. Uruchomienie *PowerShella* i sprawdzenie wersji Pythona przez podanie w jego oknie polecenia
`py --version`
3. Aktualizacja *PIP* i sprawdzenie jego wersji przez podanie w oknie *PowerShella* poniższych poleceń
`py -m pip install --upgrade pip`
`pip --version`
4. Instalacja i aktualizacja programu narzędziowego *ESPTOOL*
`pip install --upgrade esptool`
`pip install --upgrade pytap2`

Literatura i adresy internetowe

Roczniki 2022 – 2023 Funkamateura

Roczniki 2021 – 2023 QSP

„LoRaWAN-Knoten im IoT”, Claudius Kühnel, Elektor, Akwizgran [Aachen] 2021, ISBN 978-3-89576-467-7

- [1.1] www.arduino.cc – główna witryna projektu „Arduino”
- [1.2] sklep.avt.pl – sklep internetowy wydawnictwa AVT
- [1.3] www.swiatradio.com.pl – witryna internetowa „Świata Radio”
- [1.4] www.kg-gps.de
- [1.5] www.dragino.com – witryna producenta nakładek „LoRa” dla „Arduino”
- [1.6] aprs.fi – obserwacja położenia stacji APRS i ich komunikatów
 - [1.1.1] <https://unsigned.io> – kalkulator szybkości transmisji i czułości kanału „LoRy”
 - [1.3.1] aprs-map.info – wyświetlanie lokalizacji stacji na mapie
 - [1.3.2] https://github.com/lora-aprs/LoRa_APRS_Tracker – oprogramowanie nadajnika pozycji OE5BPA
 - [1.3.3] https://github.com/lora-aprs/LoRa_APRS_iGate – oprogramowanie bramki internetowej OE5BPA
 - [1.3.4] platformio.org – oprogramowanie *PlatformIO* IDE i *VSCode*
 - [1.3.5] <https://apps.magicbug.co.uk/passcode/> – generator hasła dostępu do APRS-IS
 - [1.3.5a] <https://aprs.do3sw.de/> – generator hasła dostępu do APRS-IS
 - [1.3.6] „Stacja APRScube“, Krzysztof Dąbrowski, OE1KDA, „Świat Radio” 7-8/2022, str. 48
 - [1.3.7] „iGate und Tracker bei DB0SL”, Kurt Moraw, DJ0ABR, CQDL 7/2022, str. 45
- [2.1] „LoRa MeshCom”, Kurt Baumann, OE1KBC, QSP 1/2022, str. 14
- [2.2] „LoRa MeshCom”, Kurt Baumann, OE1KBC, QSP 2/2022, str. 11
- [2.3] „Nadajnik APRS w systemie LoRa”, Krzysztof Dąbrowski, OE1KDA, Świat Radio 12/2019 str. 30
- [2.4] „Telemetry”, tom 33 z serii „Biblioteka polskiego krótkofalowca”
- [2.5] meshtastic.org – oprogramowanie wewnętrzne modułów
- [2.6] <https://gitforwindows.org>
- [2.7] <https://github.com/LilyGO/TTGO-T-Beam#start-of-content> – krótkofalarskie oprogramowanie dla modułów TTGO/T-Beam z GPS
- [2.8] <https://wiki.oevsv.at/wiki/MeshCom> – informacje w języku niemieckim
- [2.9] <https://wiki.oevsv.at/wiki/MeshCom/eshCom-Firmware> – oprogramowanie wewnętrzne modułów dopasowane do potrzeb krótkofalarskich
- [2.10] <https://de.wikipedia.org/wiki/MQTT> – opis protokołu MQTT
- [2.11] www.icsww.org – witryna projektu „MeshCom”
- [2.12] „Amateurfunk 2023”, „Software für den Funkamateure 2023”, VTH-Verlag
- [2.13] „LoRa APRS – eine Einführung”, Karsten Heddenhausen, DC7OS, CQDL 4/2-23, str. 52
- [2.14] oe1kbc@oevsv.at – prowadzący projekt „LoRa MeshCom”, referat projektów narodowych i międzynarodowych OEVS
- [3.1] <https://tinygs.com> – witryna TinyGS
- [3.2] <https://github.com/G4lile0/tinyGS>
- [3.3] <https://github.com/G4lile0/tinyGS/wiki>
- [3.4] „TinyGS – winzige Bodenstationen für LoRa-Satelliten“, Andreas Bilsing, DM4TG, CQDL 4/2023, str. 44

W serii „Biblioteka polskiego krótkofalowca” dotychczas ukazały się:

- Nr 1 – „Poradnik D-STAR”, wydanie 1 (2011), 2 (2015), 3 (2019) i 4 (2021)
- Nr 2 – „Instrukcja do programu D-RATS”
- Nr 3 – „Technika słabych sygnałów” Tom 1
- Nr 4 – „Technika słabych sygnałów” Tom 2
- Nr 5 – „Łączności cyfrowe na falach krótkich” Tom 1
- Nr 6 – „Łączności cyfrowe na falach krótkich” Tom 2
- Nr 7 – „Packet radio”
- Nr 8 – „APRS i D-PRS”
- Nr 9 – „Poczta elektroniczna na falach krótkich” Tom 1, wydanie 1 (2012)
- Nr 10 – „Poczta elektroniczna na falach krótkich” Tom 2, wydanie 1 (2012)
- Nr 11 – „Słownik niemiecko-polski i angielsko-polski” Tom 1
- Nr 12 – „Radiostacje i odbiorniki z cyfrową obróbką sygnałów” Tom 1
- Nr 13 – „Radiostacje i odbiorniki z cyfrową obróbką sygnałów” Tom 2
- Nr 14 – „Amatorska radioastronomia”
- Nr 15 – „Transmisja danych w systemie D-STAR”
- Nr 16 – „Amatorska radiometeorologia”, wydanie 1 (2013) i 2 (2017)
- Nr 17 – „Radiolatarnie małej mocy”
- Nr 18 – „Łączności na falach długich”
- Nr 19 – „Poradnik Echolinku”
- Nr 20 – „Arduino w krótkofalarstwie” Tom 1
- Nr 21 – „Arduino w krótkofalarstwie” Tom 2
- Nr 22 – „Protokół BGP w Hamnecie”
- Nr 23 – „Technika słabych sygnałów” Tom 3, wydanie 1 (2014), 2 (2016) i 3 (2017)
- Nr 24 – „Raspberry Pi w krótkofalarstwie”
- Nr 25 – „Najpopularniejsze pasma mikrofalowe”, wydanie 1 (2015) i 2 (2019)
- Nr 26 – „Poradnik DMR” wydanie 1 (2015), 2 (2016) i 3 (2019), nr 326 – wydanie skrócone (2016)
- Nr 27 – „Poradnik Hamnetu” wydanie 1 (2015) i 2 (2021)
- Nr 28 – „Budujemy Ilera” Tom 1
- Nr 29 – „Budujemy Ilera” Tom 2
- Nr 30 – „Konstrukcje D-Starowe”
- Nr 31 – „Radiostacje i odbiorniki z cyfrową obróbką sygnałów” Tom 3
- Nr 32 – „Anteny łatwe do ukrycia”
- Nr 33 – „Amatorska telemetria”, wydanie 1 (2017) i 2 (2022)
- Nr 34 – „Poradnik systemu C4FM”, wydanie 1 (2017), 2 (2019) i 3 (2021)
- Nr 35 – „Licencja i co dalej” Tom 1
- Nr 36 – „Cyfrowa Obróbka Sygnałów”
- Nr 37 – „Telewizja amatorska”
- Nr 38 – „Technika słabych sygnałów” Tom 4, wydanie 1 (2018), 2 (2020) i 3 (2022)
- Nr 39 – „Łączności świetlne”
- Nr 40 – „Radiostacje i odbiorniki z cyfrową obróbką sygnałów” Tom 4
- Nr 41 – „Licencja i co dalej” Tom 2
- Nr 42 – „Miernictwo” Tom 1
- Nr 43 – „Miernictwo” Tom 2
- Nr 44 – „Miernictwo” Tom 3
- Nr 45 – „Testy sprzętu” Tom 1
- Nr 46 – „Testy sprzętu” Tom 2
- Nr 47 – „Licencja i co dalej” Tom 3
- Nr 48 – „Jonosfera i propagacja fal”
- Nr 49 – „Anteny krótkofalowe” Tom 1, wydanie 1 (2020) i 2 (2023)
- Nr 50 – „Anteny ultrakrótkofalowe” Tom 1, wydanie 1 (2020) i 2 (2022)
- Nr 51 – „Anteny krótkofalowe” Tom 2, wydanie 1 (2020) i 2 (2023)
- Nr 52 – „Anteny ultrakrótkofalowe” Tom 2, wydanie 1 (2020) i 2 (2023)
- Nr 53 – „Anteny mikrofalowe”

- Nr 54 – „Proste odbiorniki amatorskie” Tom 1
- Nr 55 – „Proste odbiorniki amatorskie” Tom 2
- Nr 56 – „Proste nadajniki amatorskie” Tom 1
- Nr 57 – „Proste nadajniki amatorskie” Tom 2
- Nr 58 – „Mini- i mikrokomputery w krótkofalarstwie” Tom 1
- Nr 59 – „Mini- i mikrokomputery w krótkofalarstwie” Tom 2
- Nr 60 – „DX-y w C4FM”
- Nr 261 – „Poradnik DMR” Tom 1, z nru 26, wydanie 1 (2021)
- Nr 262 – „Poradnik DMR” Tom 2, z nru 26, wydanie 1 (2021)
- Nr 63 – „Testy sprzętu” Tom 3
- Nr 64 – „Poczta elektroniczna na falach krótkich”, z nrów 9 i 10, wydanie 2 (2022)
- Nr 65 – „Testy sprzętu” Tom 4
- Nr 66 – „Mieszanka firmowa” Tom 1
- Nr 67 – „Mieszanka firmowa” Tom 2
- Nr 68 – „System LoRa”

