

KRZYSZTOF DĄBROWSKI
OE1KDA

MINI- I MIKROKOMPUTERY
W KRÓTKOFALARSTWIE
TOM 2

WIEDEŃ 2021



© Krzysztof Dąbrowski OE1KDA
Wiedeń 2021

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.

Mini- i mikrokomputery w krótkofalarstwie

Tom 2

Krzysztof Dąbrowski OE1KDA

Wydanie 1
Wiedeń, kwiecień 2021

Spis treści

4. Procesory programowane w środowisku Arduino	6
4.1. ATtiny45	6
4.1.1. Radiolatarnia QRSS	6
4.1.2. Program radiolatarni QRSS	9
4.2. ESP8266	11
5. Malina	18
5.1. Wiadomości ogólne	18
5.1.1. System dźwiękowy	22
5.1.2. Pozostałe złącza	23
5.1.3. Automatyczne wywoływanie programów	23
5.1.4 Zdalna obsługa <i>Maliny</i>	25
5.2. Programy dla emisji cyfrowych	26
5.2.1. Emisja WSPR	26
5.3. Programy terminalowe dla emisji cyfrowych	28
5.3.1. Fldigi	28
5.3.2. WSJT-X	30
5.3.3. JS8Call	31
5.3.4. QSSTV	32
5.3.5. Dire Wolf	33
5.4. Odbiorniki programowalne	34
5.4.1. Odbiorcza bramka APRS	36
5.5. APRS	39
5.5.1. Bramka odbiorcza i stacja meteorologiczna	39
5.5.2. Instalacja APRX	43
5.6. Cyfrowy przekaźnik APRS dla łączności kryzysowych	45
5.7. Echolink	47
5.8. Odbiór WSPR na RSPduo i WSJT-X	48
5.9. Odbiór cyfrowego dźwięku	51
5.10. Prosty serwer HTTP	53
5.11. „Hamserver Pi”	54
6. Mikroprzemiennik „PiStar”	55
6.1. Szczegóły konfiguracji w trybie eksperta	62
6.2. Oprogramowanie EA7EE	66
7. Odbiornik programowalny „Kiwi”	69
8. Wielozakresowy odbiornik FT8 na „Red Pitayi”	72
9. Nadajnik telegraficzny z PIC10F206	74
Dodatek A. Elementy q w trasach APRS	76
Dodatek B. Program radiolatarni z Arduino i Si5351	78
Dodatek C. Połączenia w sieciach	80
C1. Protokoły TCP/IP	82
C2. Adresy internetowe	85
C3. Protokół AX25	86
Literatura i adresy internetowe	87

Sommaire

Mini- et microordinateurs pour radio amateurs

4. Microprocesseurs programmables en Arduino IDE	6
4.1. ATtiny45	6
4.1.1. Balise QRSS	6
4.1.2. Code source de la balise QRSS	9
4.2. ESP8266	11
5. Raspberry Pi	18
5.1. Informations de base	18
5.1.1. Carte son USB	22
5.1.2. Autres interfaces	23
5.1.3. Lancer automatiquement des programmes	23
5.1.4. Télécommande de Raspberry Pi	25
5.2. Programmes pour les modes numériques	26
5.2.1. WSPR	26
5.3. Logiciel pour communication en modes numériques	28
5.3.1. Fldigi	28
5.3.2. WSJT-X	30
5.3.3. JS8Call	31
5.3.4. QSSTV	32
5.3.5. Dire Wolf	33
5.4. Récepteurs logicielles	34
5.4.1. Passerelle RX APRS	36
5.5. APRS	39
5.5.1. Passerelle RX et la station météo	39
5.5.2. Installation d'APRX	43
5.6. Digipéteur APRS pour opérations de secours	45
5.7. Echolink	47
5.8. Réception de WSPR avec RSPduo et WSJT-X	48
5.9. Réception de voix numérique	51
5.10. Serveur HTTP simple	53
5.11. „Hamserver Pi“	54
6. Hotspot „PiStar”	55
6.1. Détails de configuration experte	62
6.2. Logiciel de EA7EE	66
7. Récepteur logiciel „Kiwi”	69
8. Récepteur multibande FT8 à „Red Pitaya”	72
9. Émetteur CW avec PIC10F206	74
Annexe A. Éléments q en adresses APRS	76
Annexe B. Programme de balise Arduino et Si5351	78
Dodatek C. Connexions dans les réseaux	80
C1. Protocoles TCP/IP	82
C2. Adresses internet	85
C3. Protocol AX25	86
Bibliographie et les pages web	87

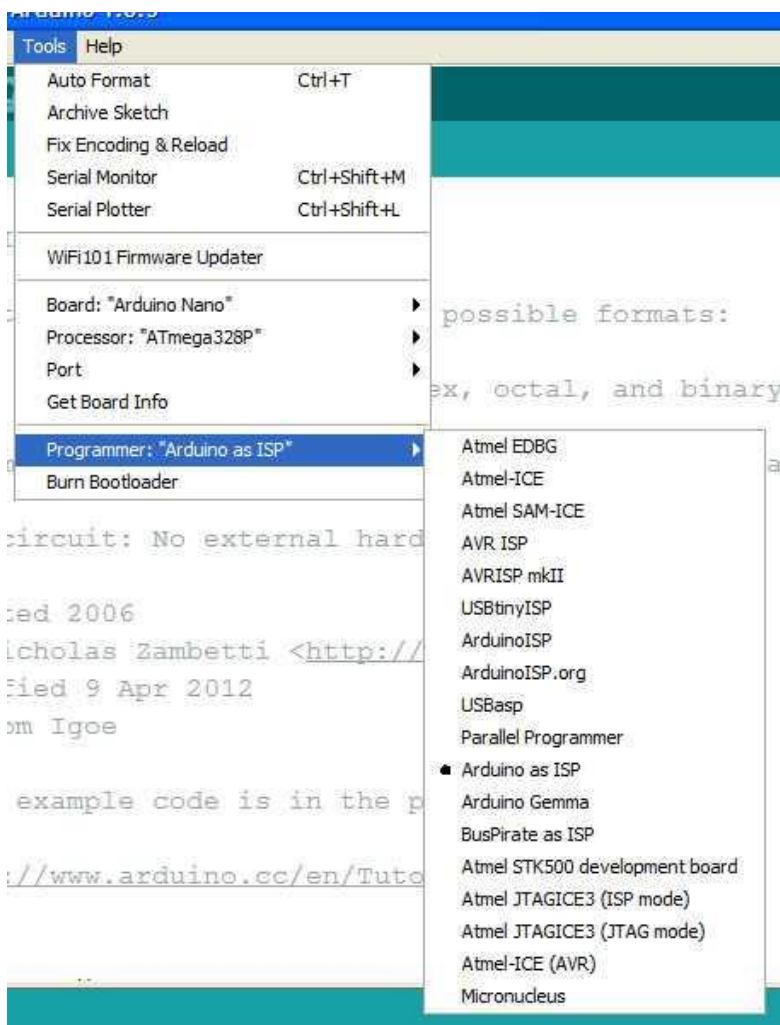
4. Procesory programowane w środowisku Arduino

Środowisko programistyczne Arduino IDE pozwala także na opracowywanie programów dla innych typów procesorów, w tym niektórych mikroprocesorów AVR i ESP. Wyboru wersji Arduino lub typu procesora dokonuje się w menu „Tools” („Narzędzia”).

4.1. ATtiny45

4.1.1. Radiolatarnia QRSS

Do kluczowania nadajnika QRSS – czyli wolnej telegrafii wykorzystano mikroprocesor ATtiny45. Jego sygnał wyjściowy z nóżki 3 kluczuje napięcie na diodzie pojemnościowej włączanej w obwód rezonatora kwarcowego (w układzie nadajnika G0UPL wykorzystano w tym celu spolaryzowaną zaporowo czerwoną diodę elektroluminescencyjną 5 mm), ale można zastosować go do kluczowania amplitudy. Rozwiązanie nadajnika QRSS może być dowolne i różnić się od nadajnika G0UPL. W transmisjach wolnej telegrafii QRSS stosuje się różne długości kropki od 1 sekundy do nawet 30. Proporcje długości pozostałych elementów znaków (kresiek, przerw) pozostają stałe. Sygnały odbierane są obserwowane na wskaźnikach widma programów terminalowych i odczytywane optycznie. Dla ułatwienia odczytu został opracowany wykres ścieżek prowadzących do odbieranego znaku przez wchodzące w jego skład kropki i kreski. Do często spotykanych wariantów należy kluczowanie amplitudy jak w zwykłej telegrafii i kluczowanie częstotliwości z odstępem kilku Hz.

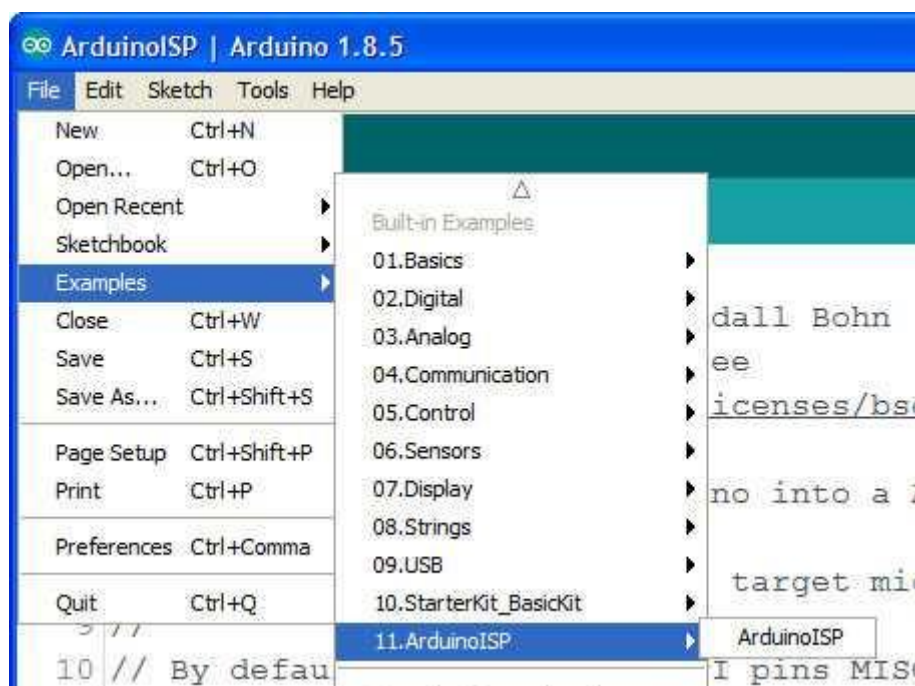


Rys. 4.1.1.1. Wybór programatora

Częstotliwości nadawania w dolnych pasmach KF leżą przeważnie w zakresie 800 – 900 Hz powyżej dolnej granicy pasma, w paśmie 10 MHz pomiędzy 10140,000 – 10140,100 kHz, a w pasmach wyższych w pobliżu częstotliwości 28322 kHz lub jej podharmonicznych w niższych pasmach (21241,5, 14161 kHz itd). Więcej informacji o QRSS podano pod <https://groups.io/g/qrssknights>. W ostatnim czasie przyjmuje się nowy standard: korzystanie z odcinków pasm o szerokości 200 Hz leżących poniżej podzakresów WSPR i przylegających do ich dolnych granic, a więc przykładowo w paśmie 80 m częstotliwość środkowa WSPR wynosi 3570,100 kHz, QRSS 3569,900 kHz, w paśmie 40 m są to odpowiednio 7040,100 i 7039,900 kHz, 30 m – 10140,200 i 10140,000 kHz, 20 m – 14097,100 i 14096,900 kHz, 17 m – 18106,100 i 18105,900 kHz, 15 m – 21096,100 i 21095,900 kHz, 12 m – 24926,100 i 24925,99 kHz, 6 m – 50294,500 i 50294,300 kHz, ale sytuacja w paśmie 10 jest inna: WSPR 28926,200 kHz, QRSS 28125,600 kHz oraz w sieci włoskiej dodatkowo 28322,000 kHz +/- 100 Hz.

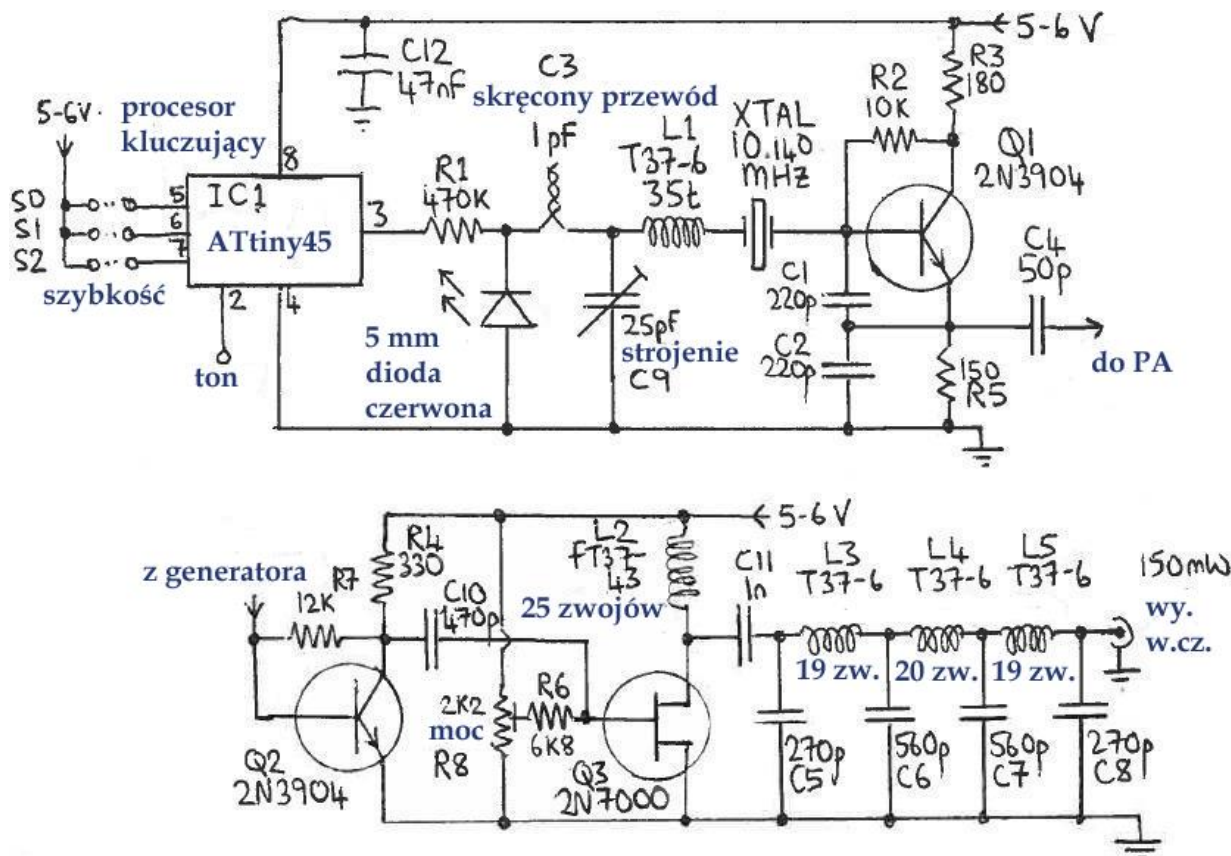
Program dla mikrokomputera ATtiny45 jest napisany w środowisku programistycznym Arduino, w jego języku i zgodnie z konwencjami przyjętymi dla Arduino. Zawiera on więc obie podstawowe funkcje `setup()` i `loop()`, a dla ustalenia odstępów czasu i czasów oczekiwania korzysta z funkcji `millis()` i `delay()`. Znaki alfabetu Morse'a są zapisane bitowo w tabeli w ten sposób, że każde słowo zawiera na początku bity jedności (stanowiące ignorowane przez program wypełniacze), następnie bit zero pełniący dla programu rolę bitu startowego i dopiero następne bity stanowią zakodowany znak telegraficzny. Zero oznacza kropkę a jedynka kreskę. Dzięki tak przyjętemu sposobowi zapisu w słowach o szerokości bajtu można zakodować znaki telegraficzne o różnej długości od jednego do 7 elementów. Wejścia 0, 1 i 2 (odpowiednio na nóżkach 5, 6 i 7) służą do wyboru szybkości telegrafowania.

Środowisko programistyczne Arduino pozwala na załadowanie pobranych z Internetu uzupełnień dla różnych typów mikroprocesorów. Po skompilowaniu programu można go załadować do mikroprocesora docelowego korzystając z osobnego programatora albo z płytki Arduino używanej jako programator ISP („Arduino as ISP”). Należy do niego załadować program programatora ISP. Złącze SPI Arduino musi być połączone z ATtiny45: wyprowadzenie 13 Arduino → 7 ATtiny45, 12 → 6, 11 → 5, 10 → 1, masa na 4 i zasilanie na 8. Najlepiej dokonać tego na płytce prototypowej. Dla innych typów procesorów wyprowadzenia złącza mogą się różnić od podanych. Przed załadowaniem programu w narzędziach należy wybrać typ procesora docelowego z częstotliwością zegarową 1 MHz. Przykłady wykorzystania Arduino jako programatora można znaleźć w punkcie „File|Examples” („Plik|Przykłady”). Do załadowania programu przez Arduino do procesora docelowego trzeba na koniec posłużyć się poleceniem „Burn Bootloader”.



Rys. 4.1.1.2. Przykłady

Na rysunku 4.1.1.3 przedstawiono przykładowy prosty nadajnik QRSS G0UPL kluczowany częstotliwościowo. Bez problemu można jednak znaleźć inne rozwiązania. Zamiast mikroprocesora ATtiny45 można zastosować ATtiny13 mający dwa razy więcej pamięci. Oczywiście można także do kluczowania zastosować Arduino, ale autor pragnął zamieścić przykład programowania innych mikroprocesorów w środowisku Arduino.



Rys. 4.1.1.3. Schemat ideowy nadajnika o mocy wyjściowej 100 – 150 mW

Tabela 4.1.1.1

Elementy L, C dla pasm 80 – 30 m

Element	Pasmo 80 m	Pasmo 40 m	Pasmo 30 m
L1	27 zw., T37-6, żółty	27 zw., T37-6, żółty	27 zw., T37-6, żółty
L2	25 zw., FT37-43, czarny	25 zw., FT37-43, czarny	25 zw., FT37-43, czarny
L3	25 zw., T37-2, czerwony	19 zw., T37-6, żółty	19 zw., T37-6, żółty
L4	27 zw., T37-2, czerwony	21 zw.; T37-6, żółty	20 zw., T37-6, żółty
L5	25 zw., T37-2, czerwony	19 zw., T37-2, żółty	19 zw., T37-6, żółty
C1, C2	680 pF	470 pF	220 pF
C3	1 pF, skręcony przewód	1 pF, skręcony przewód	1 pF, skręcony przewód
C4	47 pF	47 pF	47 pF
C5, C8	470 pF	270 pF	270 pF
C6, C7	1200 pF (1,2 nF)	680 pF	560 pF
C9	25 pF, trymer	25 pF, trymer	25 pF, trymer
C10	470 pF	470 pF	470 pF
C11	1 nF	1 nF	1 nF
C12	47 nF	47 nF	47 nF

Oporniki: R1 – 470 kΩ, R2 – 10 kΩ, R3 – 180 Ω, R4 – 330 Ω, R5 – 150 Ω, R6 – 6,8 kΩ, R7 – 12 kΩ, R8 – potencjometr montażowy 2,2 kΩ.

Tranzystory: Q1, Q2 – 2N3904 lub odpowiednik europejski, Q3 – 2N7000, dioda 5 mm czaerwona, mikroprocesor ATtiny13 lub ATtiny45.

Kondensator C3 o pojemności około 1 pF jest wykonany z dwóch skrzęconych ze sobą odcinków przewodu o długościach 50 mm w izolacji. Po skrzęceniu przewodów ze sobą należy obciąć kondensator tak, aby miał długość 15 mm. Pojemność C3 należy dobrać tak, aby dewiacja częstotliwości wynosiła 4 – 5 Hz.

Tabela 4.1.1.2

Wybór szybkości telegrafowania

Kontakt (nóżka)	12 sł./min	6 sł./min	Kropka 1 s	Kropka 3 s	Kropka 6 s	Kropka 10 s	Kropka 15 s	Kropka 20 s
S2 (7)					X	X	X	X
S1 (6)			X	X			X	X
S0 (5)		X		X		X		X

4.1.2. Program radiolatarni QRSS

```
//
// Radiolatarnia QRSS
//
// Copyright 2012 Hans Summers G0UPL.
//

// W następnej linii podać znak wywoławczy.
const char msg[] = "N0CALL ";

const unsigned int speeds[] = {1, 2, 10, 30, 60, 100, 200, 300};
// Szybkości: 12 słów/min., 6 słów/min., QRSS1, QRSS3, QRSS6, QRSS10, QRSS20, QRSS30

#define KEY 4

//
// Zamiana liter na znaki telegraficzne
//
byte charCode(char c)
{
    switch (c)
    {
        case 'A': return B11111001;    // A  .-
        case 'B': return B11101000;    // B  -...
        case 'C': return B11101010;    // C  -.-.
        case 'D': return B11110100;    // D  -..
        case 'E': return B11111100;    // E  .
        case 'F': return B11100010;    // F  ..-.
        case 'G': return B11110110;    // G  --.
        case 'H': return B11100000;    // H  ....
        case 'I': return B11111000;    // I  ..
        case 'J': return B11100111;    // J  .---
        case 'K': return B11110101;    // K  -.-
        case 'L': return B11100100;    // L  .-..
        case 'M': return B11111011;    // M  --
        case 'N': return B11111010;    // N  -.
        case 'O': return B11110111;    // O  ---
    }
}
```

```

    case 'P': return B11100110;    // P .--.
    case 'Q': return B11101101;    // Q --.-
    case 'R': return B11110010;    // R .-.
    case 'S': return B11110000;    // S ...
    case 'T': return B11111101;    // T -
    case 'U': return B11110001;    // U ..-
    case 'V': return B11100001;    // V ...-
    case 'W': return B11110011;    // W .--
    case 'X': return B11101001;    // X -..-
    case 'Y': return B11101011;    // Y -.--
    case 'Z': return B11101100;    // Z --..
    case '0': return B11011111;    // 0 ----
    case '1': return B11001111;    // 1 .----
    case '2': return B11000111;    // 2 ..---
    case '3': return B11000011;    // 3 ...--
    case '4': return B11000001;    // 4 ....-
    case '5': return B11000000;    // 5 .....
    case '6': return B11010000;    // 6 -....
    case '7': return B11011000;    // 7 --...
    case '8': return B11011100;    // 8 ---..
    case '9': return B11011110;    // 9 ----.
    case ' ': return B00000000;    // Space
    case '/': return B11010010;    // / -..-.
    default: return charCode(' ');
  }
}

//
// Definicje wyprowadzeń
//
void setup() {
  pinMode(KEY, OUTPUT);    // Definicje wyjść
  pinMode(0, INPUT);    // Definicje wejść przełączających szybkość
  pinMode(1, INPUT);
  pinMode(2, INPUT);
}

//
// Funkcja pętli Arduino
//
void loop()
{
  static unsigned long milliLimit;
  unsigned long milliNow;

  milliNow = millis();    // Liczba milisekund od uruchomienia systemu

  if (milliNow >= milliLimit)    // Porównanie z granicą
  {
    milliLimit = milliNow + 100;    // Ustawienie następnej granicy o 0,1 sekundy później
    beacon();    // Wywołanie programu kluczującego
  }
}

//

```

```

// Program kluczujący
//
void beacon()
{
    static unsigned int counter;    // Licznik dla długości bitu, wielokrotność 0,1 sekundy
    static byte pause;             // Odstępy między znakami
    static byte msgIndex = 255;    // Indeks do tekstu komunikatu
    static byte character;         // Kod bieżącego symbolu CW
    static byte key;               // Licznik naciśnięcia klucza 3 – kreska, 1 – kropka
    static byte symbol;            // Indeks bitu w znaku CW
    static byte ditspeed;          // Indeks (0-7) do tabeli szybkości

    counter++;                     // Zliczanie długości kropki
    if (counter == speeds[ditspeed]) // Porównywanie z wybraną szybkością
    {
        counter = 0;              // Zerowanie licznika długości kropki

        if (!pause)
        {
            key--;                 // Przy naciśniętym kluczu zmniejszanie stanu licznika naciśnięcia
                                   // Przy stanie zerowym licznika naciśnięcia i indeksu symboli przerwa między znakami
            if ((!key) && (!symbol)) pause = 2;
        }
        else
            pause--;               // Zliczanie długości przerwy

        if (key == 255)            // Po końcu symboli licznik -> 255
        {
            if (!symbol)           // Gdy licznik symbolu zero pobranie następnego znaku
            {
                msgIndex++;        // Zwiększenie indeksu

                if (!msg[msgIndex]) // Po znalezieniu końca komunikatu (zero)
                {
                    // indeks na jego początek
                    msgIndex = 0;    // i odczyt szybkości z wyprowadzeń 0-2
                    ditspeed = 4 * digitalRead(2) + 2 * digitalRead(1) + digitalRead(0);
                }

                // Pobranie kodu bieżącego znaku
                character = charCode(msg[msgIndex]);

                symbol = 7;          // Ustawienie licznika bitów na najbardziej lewy bit kodu
                                   // Poszukiwanie bitu startowego 0
                while (character & (1<<symbol))
                    symbol--;
            }

            symbol--;               // Obniżanie stanu licznika, przejście w prawo na kolejne bity

                                   // Znak odstępu jest przypadkiem szczególnym, klucz nie naciśnięty
            if (character == charCode(' '))
                key = 0;
            else
            {
                // sprawdzanie wskazywanego bitu
                if (character & (1<<symbol))
                    key = 3;        // 1 oznacza kreskę o długości 3 kropek
            }
        }
    }
}

```

```

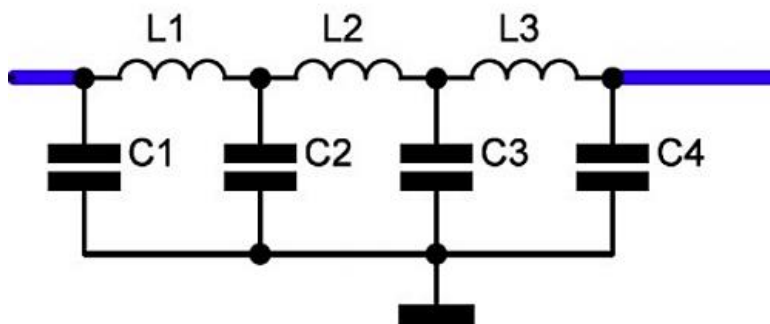
    else
        key = 1;          // 0 oznacza kropkę, o wybranej długości
    }
}

if (key)                  // Sygnał na wyprowadzeniu wyjściowym
    digitalWrite(KEY, HIGH);
else
    digitalWrite(KEY, LOW);
}
}

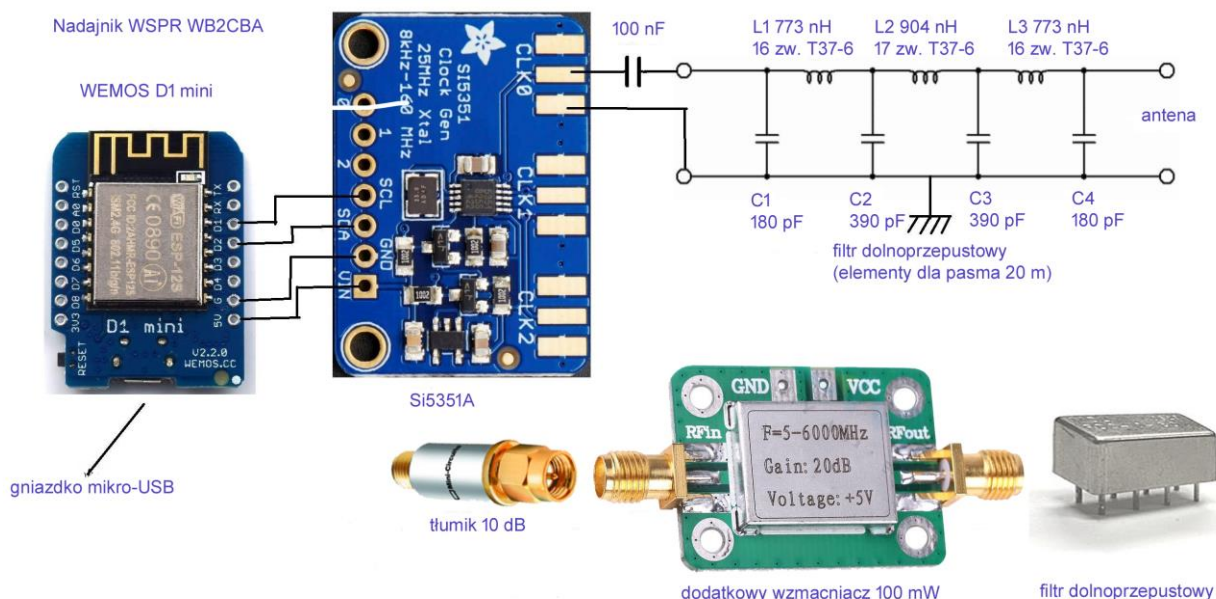
```

4.2. ESP8266

Nadajnik WSPR konstrukcji WB2CBA składa się z trzech podstawowych części: sterującego modułu mikroprocesora z łączem WiFi *ESP8266 D1 mini*, który zapewnia także łączność z Internetem i synchronizację czasu, syntezer Si5351 i wyjściowego filtra dolnoprzepustowego siódmego rzędu. Zamiast *D1 mini* można użyć ESP8266-12E albo dowolnego innego modułu z serii ESP8266 NodeMCU (moduły pracują w normach 802.11 b/g/n i w zależności od otoczenia zapewniają zasięgi WiFi przekraczające 100 m w paśmie 2,4 GHz). W oryginalnym układzie nadawany jest sygnał 10 mW pochodzący z syntezer, ale dodanie wzmacniacza mocy pozwala na uzyskanie 100 mW lub więcej w zależności od jego typu. Na ilustracji przedstawiono przykład wzmacniacza HFM-6G oferowanego przez sklep internetowy Funkamateura www.box73.de. Wzmacniacz pokrywa zakres 5 MHz – 6 GHz i dla wysterowania mocą około 1 mW wymaga użycia tłumika 10 dB na wyjściu syntezer. Na ilustracji pokazano jako przykład tłumik firmy *Mini-Circuits*. Przy zasilaniu napięciem 5 V wzmacniacz pobiera prąd 85 mA. Filtr własnej konstrukcji można też zastąpić przez fabryczny np. wybrany z tabeli 4.2.2. Oprogramowanie nadajnika znajduje się w Internecie pod adresem <https://antrak.org.tr/wp-content/uploads/ESPWSPR-yazilimlar.zip>, a niezbędne biblioteki Arduino pod adresami <https://github.com/etherkit/Si5351Arduino> (dla obsługi syntezer Si5351), <https://github.com/etherkit/JTEncode> (do kodowania WSPR, JT65, JT9, JT4, FT8, FSQ), <https://github.com/PaulStoffregen/Time> (do sterowania czasowego), <https://github.com/Sensorslot/NTPtimeESP> (do synchronizacji czasu). Do zaprogramowania modułu ESP należy skorzystać ze środowiska programistycznego *Arduino IDE*. Przed wpisaniem programu do *D1 mini* konieczne jest podanie w nim własnego znaku wywoławczego, lokatora, mocy, częstotliwości pracy i danych dostępowych do domowej sieci WLAN. Program synchronizuje czas korzystając z serwerów internetowych albo z domowego modemu dostępowego. Nadawanie komunikatów jest sygnalizowane świeceniem niebieskiej diody na module ESP. Częstotliwość pracy powinna być skontrolowana za pomocą odbiornika i ewentualnie skorygowana w programie. Dla ułatwienia kalibracji częstotliwości w zestawie programów WB2CBA zawarty jest program kalibracyjny.



Rys. 4.2.1. Filtr dolnoprzepustowy siódmego rzędu do nadajników małej mocy



Rys. 4.2.2. Nadajnik WB2CBA. Do konstrukcji można dodać wzmacniacz o mocy wyjściowej 100 mW (konieczne jest obniżenie mocy sterującej z syntezy do ok. 1 mW za pomocą tłumika) i fabryczny filtr dolnoprzepustowy

Tabela 4.2.1. Wartości elementów dla dolnoprzepustowego filtra Czebyszewa siódmego rzędu

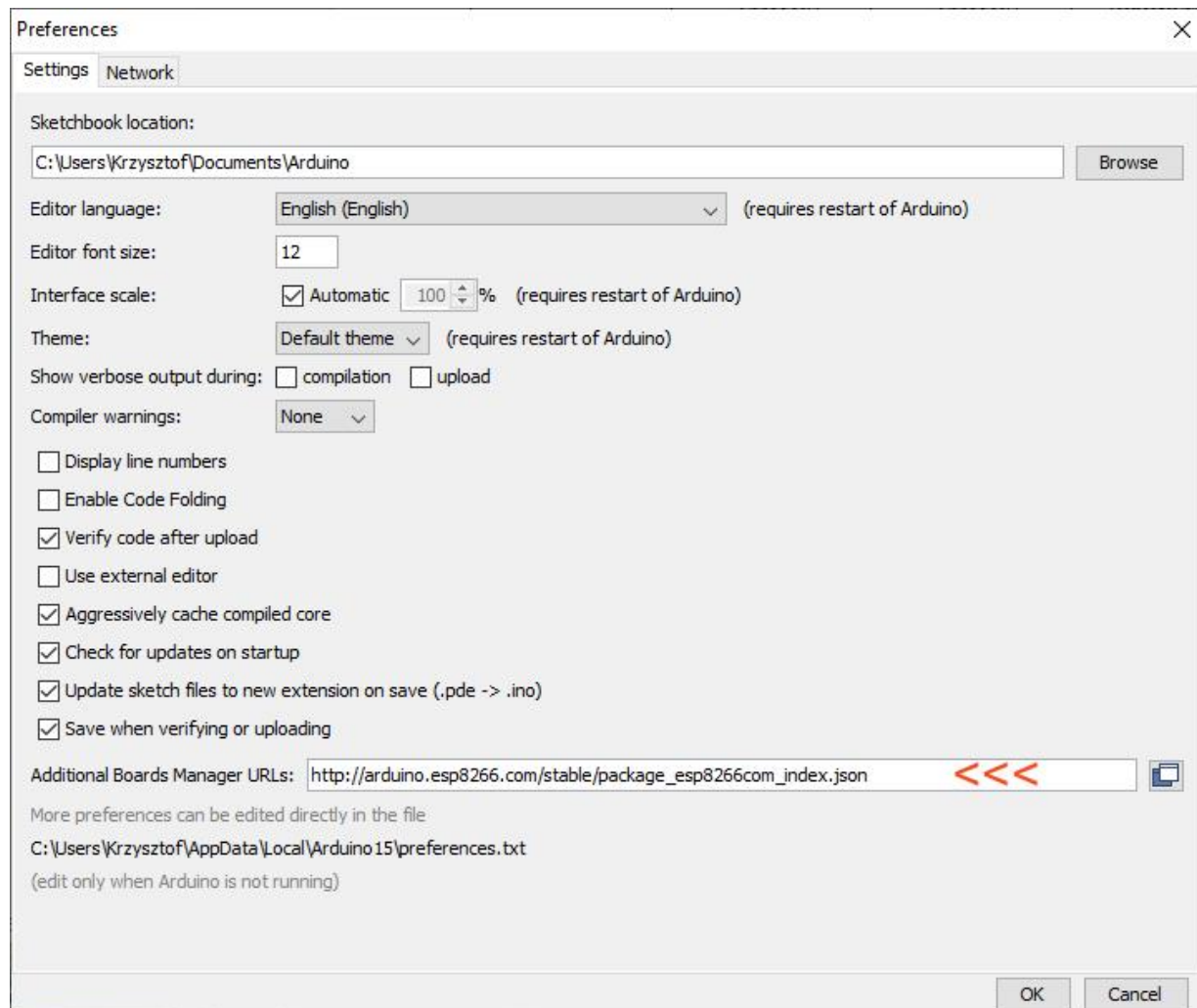
Pasma [m]	C1, C4 [nF]	C2, C3 [nF]	L1, L3 [μF]	L2 [μH]
2190	33	47	56	60
630	9,1	15	16	17
160	0,82	2,2	4,44	5,61
80	0,47	1,2	2,43	3,01
40	0,27	0,68	1,38	1,7
30	0,27	0,56	1,09	1,26
[m]	[pF]	[pF]	[nH]	[nH]
20	180	390	773	904
17	120	270	548	668
15	100	270	444	571
12	100	220	438	515
10	75	180	303	382
6	36	100	197	248
4	27	75	149	187

Tabela 4.2.2. Filtry dolnoprzepustowe firmy Mini-Circuits dla pasm krótkofalowych i 50 – 430 MHz

Filtr	Zakres dla tłumienia < 1 dB [MHz]	Częstotliwość graniczna 3 dB [MHz]	Pasma zaporowe dla tłumienia > 20 dB [MHz]	Pasma zaporowe dla tłumienia > 40 dB [MHz]
PLP-5	0 – 5	6	8 – 10	10 – 200
PLP-10.7	0 – 11	14	19 – 24	24 – 200
PLP-21.4	0 – 22	24,5	32 – 41	41 – 200
PLP-30	0 – 32	35	47 – 61	61 – 200
PLP-70	0 – 60	67	90 – 117	117 – 300
PLP-90	0 – 81	90	121 – 157	157 – 400
PLP-150	0 – 140	155	210 – 300	300 – 600
PLP-200	0 – 190	210	290 – 390	390 – 800
PLP-550	0 – 520	570	750 – 920	920 – 2000

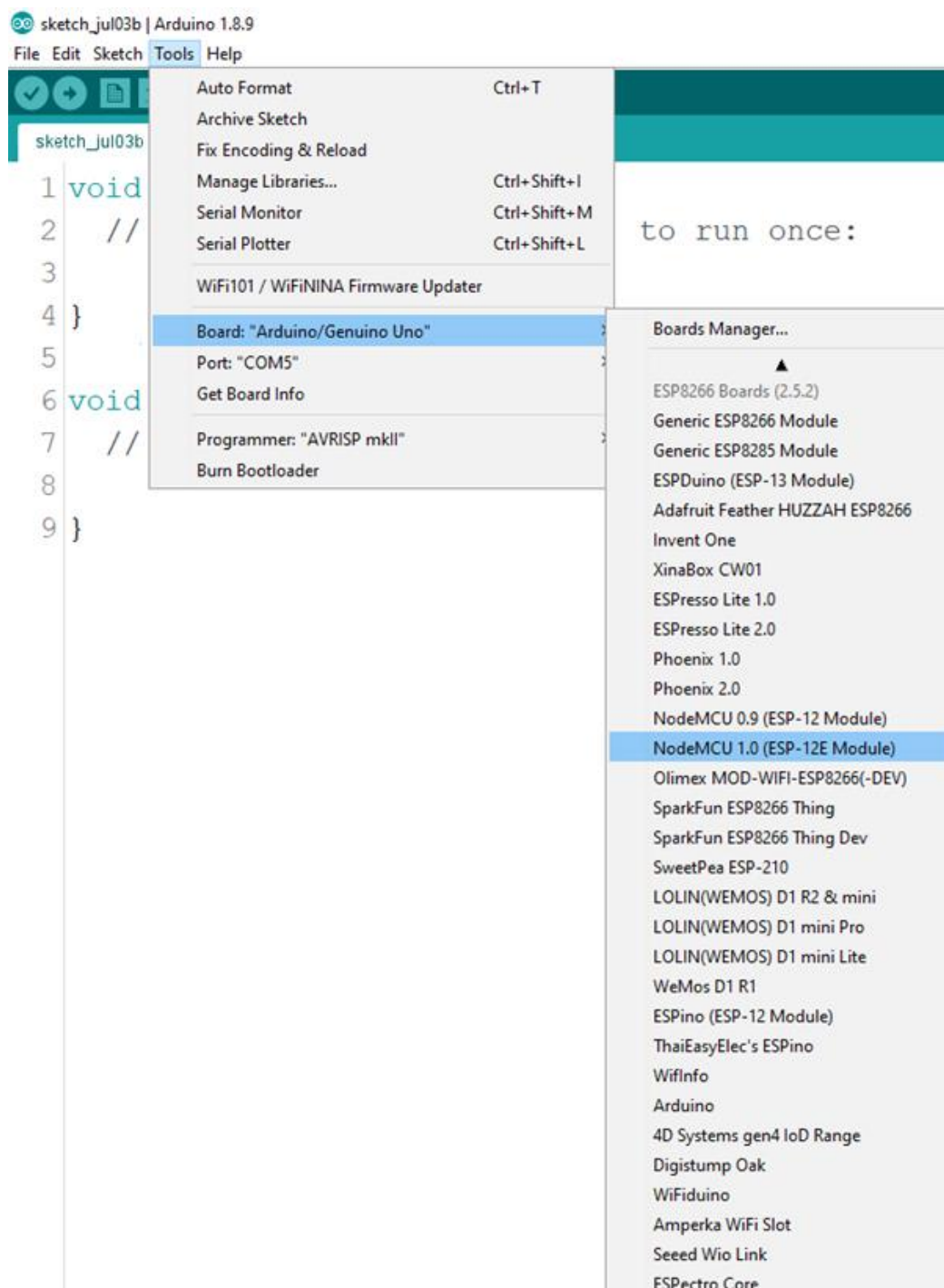
Uwagi:

maksymalna moc doprowadzona do wejścia = 0,5 W, impedancje we./wy. 50 Ω , WFS w paśmie przenoszenia typ. 1,7, WFS w paśmie zaporowym typ. 18, obudowy metalowe hermetyczne A01 ~ 20 x 10 x 10 mm, masa ~ 5 g



Rys. 4.2.3. Dodanie do środowiska Arduino rozszerzenia dla procesorów ESP8266

Dla rozszerzenia środowiska programistycznego Arduino o obsługę procesorów ESP8266 należy w jego konfiguracji („Preferences”) w polu „Additional Boards Manager URLs” wpisać adres internetowy: http://arduino.esp8266.com/stable/package_esp8266com_index.json.



Rys. 4.2.3.a. Wybór płytki procesora

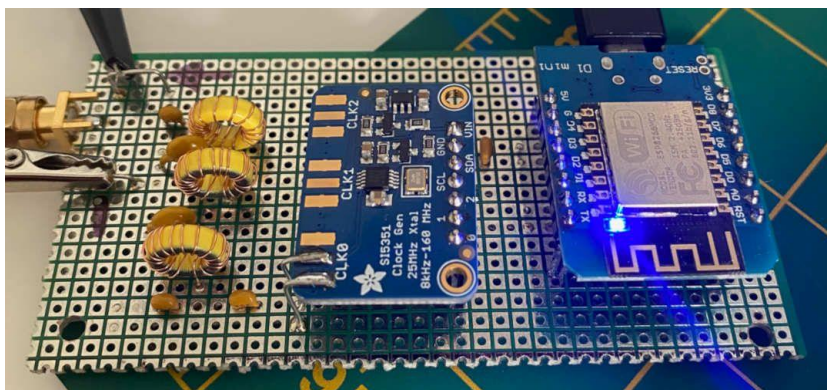
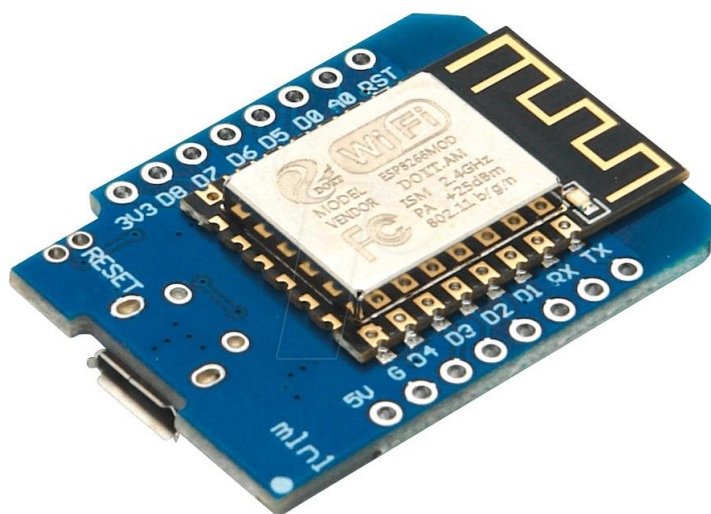
W menu „Narzędzi” („Tools”) do wyboru są wówczas płytki mikrokomputerów z ESP8266, m.in. użyty w nadajniku moduł „NodeMCU 1.0 (ESP-12 Module)” (pod „Board ‘Arduino/Genuino Uno’”). W dalszych punktach menu należy w tym przypadku wybrać: częstotliwość zegarową („CPU Frequency”) 80 MHz, szybkość transmisji („Upload Speed”) 115200 bit/s, pamięć programu („Flash Size”) 4 MB (FS”2 MB OTA ~1019 KB) i programator („Programmer”) AVRIP *mkII*. Moduł ESP podłącza się do PC za pomocą kabla USB z pasującymi wtyczkami.

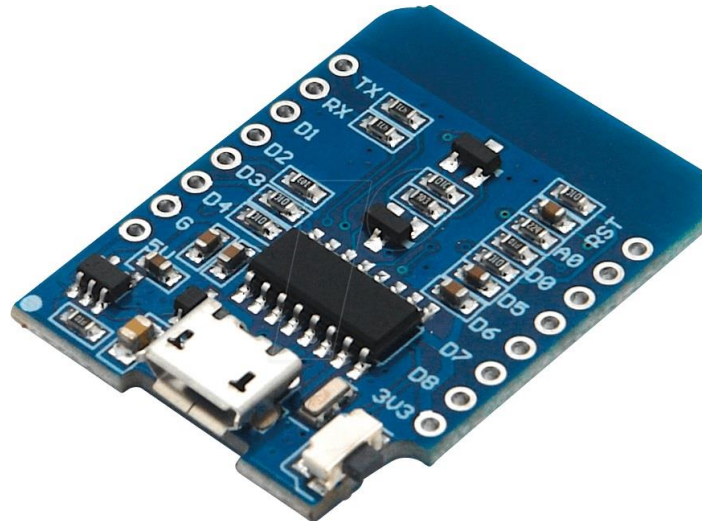
Na rynku dostępnych jest kilka rodzajów modułów ESP8266 różniących się wymiarami, wyprowadzeniami i wyposażeniem. Niektóre z nich są przewidziane do współpracy z Arduino, *Maliną* i *Banana Pi*. Do programowania modułów modeli posiadających gniazdo kontaktowe na krótszym boku potrzebny jest moduł przejściowy z USB np. DEBO ESP8266 USB.

Tabela 4.2.3

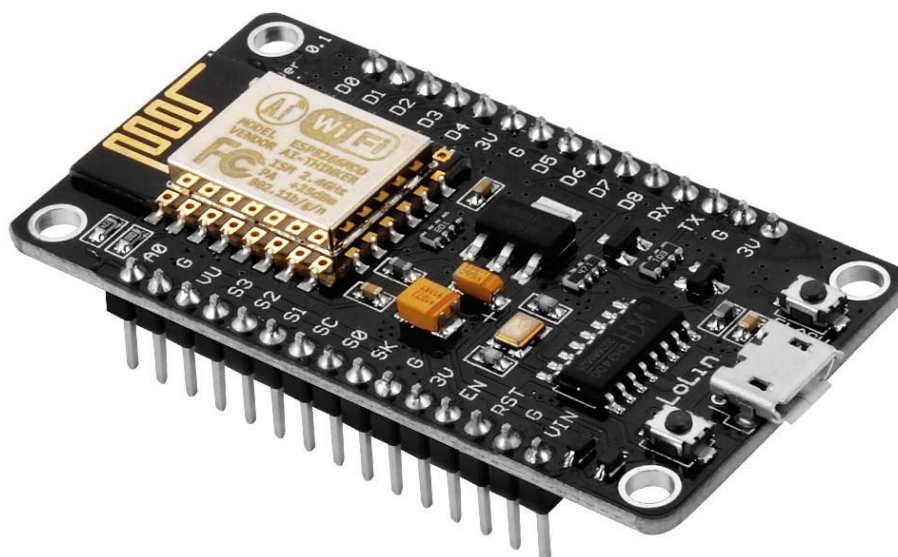
Najważniejsze parametry modułu *NodeMCU ESP8266*

Parametr	Wartość
Procesor	ESP8266 – 32-bitowy <i>Tensilica LX106</i>
Częstotliwość zegarowa	80 lub 160 MHz
Pamięć robocza	64 kB
Pamięć programu	96 kB
Zewnętrzna pamięć SPI	Do 16 MB
Normy WiFi	802.11 b/g/n, pasmo 2,4 GHz, moc 20 dBm
Połączenie sieciowe	Zbiór protokołów TCP/IP (IPv4): TCP, UDP, HTTP, FTP
Złącza danych	UART, I2C, GPIO
Przetwornik a-c	10-bitowy
Przetworniki c-a	2 x 8 bitów, wyjście z modulacją szerokości impulsów,
Napięcie zasilania	3,0 – 3,6 V (5 V przez gniazdko mikroUSB)
Pobór prądu	80 mA
Wymiary	58 x 31 x 12 mm
Masa	1 g

Rys. 4.2.4. Konstrukcja nadajnika WB2CBA na uniwersalnej płytce dziurkowanej. Zastosowano w nim moduł *D1 MINI ESP8266* i płytkę *Si5351* firmy *Adafruit*Fot. 4.2.5. Moduł *D1 MINI ESP8266* użyty w nadajniku WB2CBA posiada 11 wyprowadzeń logicznych, złącza I2C i 1Wire, gniazdko USB, wyjście sygnału z modulacją szerokości impulsów, wejście analogowe 3,3 V i 4 MB pamięci programu, ale nie posiada złącza SPI i kieszeni dla pamięci SD



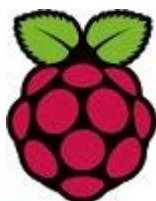
Fot. 4.2.6. Odwrotna strona modułu *D1 MINI ESP8266*



Fot. 4.2.7. Moduł *NodeMCU ESP8266*

5. Malina

5.1. Wiadomości ogólne



„Raspberry Pi” zwany także sympatycznie *Maliną* jest kolejnym mikrokomputerem przeznaczonym w pierwszym rzędzie do celów dydaktycznych, który znalazł szerokie zastosowanie w rozmaitych hobby i oczywiście w krótkofalarstwie. Jest to mikrokomputer wyższej klasy aniżeli Arduino, posiada oparty na Linuksie system operacyjny pozwalający na równoległą pracę wielu programów, wielokrotnie większe pojemności pamięci programu i roboczej, wydajniejszy procesor połączony z procesorem graficznym, szeroki wybór złączy i urządzeń peryferyjnych.

Spośród wielu modeli, które ukazały się na rynku od powstania pierwszej *Maliny* obecnie największe znaczenie mają Pi-3B+, Pi-4B i różne odmiany *Maliny Zero*. W modelu Pi-4B pracuje czterordzeniowy 64-bitowy procesor Cortex-A72 o częstotliwości zegarowej 1,5 GHz, pojemność pamięci roboczej wynosi (zależnie od wariantu) 1, 2, 4 lub 8 GB, do dyspozycji są złącza Ethernetu, dwupasmowe WiFi, Bluetooth, 2 złącza USB 2.0, 2 złącza USB 3.0 oraz mikroHDMI. Jako pamięć programu służą ogólnie dostępne moduły SD o pojemnościach 16, 32 lub 64 GB. Nabywca może się zaopatrzyć w moduł pamięci SD z systemem operacyjnym Raspbian – NOOBS lub pobrać z Internetu obraz (odwzorowanie) pamięci i samodzielnie zaprogramować moduł pamięciowy korzystając z takich programów jak *Win32 Disk Imager* albo *Etcher*. Zalecana jest instalacja Raspbiana na module o pojemności minimum 16 GB i klasie 10 (klasa pamięci informuje o szybkości dostępu). W internecie można znaleźć także odwzorowania pamięci uzupełnione o dodatkowe programy użytkowe, w tym także warianty z programami krótkofalarskimi. Instalacja i uruchomienie programów na *Malinie* wymaga podstawowej znajomości Linuksa. Duża część programów dla *Maliny* jest napisana w języku *Python*. Kompilator *Pythona* należy do podstawowego wyposażenia programowego systemu.

Tabela 5.1.1. Najważniejsze parametry wybranych aktualnych modeli *Maliny*

Parametr	Raspberry Pi-3B+	Raspberry Pi-4B	Raspberry Pi Zero W
Procesor	Cortex A53 64 bity	Cortex A72 64 bity	ARM 1176JZF-S
Częstotliwość zegarowa	1,4 GHz	1,5 GHz	1 GHz
Liczba rdzeni	4	4	1
Układ scalony SoC	BCM2837B0	BCM2711B0	BCM2835
Pamięć robocza RAM	1 GB DDR2	1, 2, 4, 8 GB	512 MB
Pamięć programu	mikroSD	mikroSD	mikroSD
Kontakty GPIO	40	40	40
Złącze Ethernet	X	Gigabit	—
Złącze WiFi	2,4 i 5 GHz, 802.11 b/g/n/ac	2,4 i 5 GHz, 802.11 b/g/n/ac	802.11 n
Złącze Bluetooth	4.2 BLE	5.0 BLE	4.1
Złącze USB2	4	2	1
Złącze USB3	—	2	—
Złącza HDMI	1 x HD HDMI	4 x 4 k HDMI	1 x HD mini-HDMI
Złącze szeregowe kamery – CSI	X	X	X
Złącze szer. wyświetlacza – DSI	X	X	—
Wyjście m.cz. i analogowego sygnału wizji	Wspólne gniazdko m.cz. i wizji	Wspólne gniazdko m.cz. i wizji	—
Pobór prądu	390 mA	560 mA	160 mA
Pobór prądu z HDMI i WiFi	520 mA	710 mA	180 mA
Gniazdo zasilania	mikroUSB	USB-C	mikroUSB

Uwagi:

– Model *Pi Zero* różni się od *Pi Zero W* tym, że nie posiada złączy Ethernetu i Bluetooth.

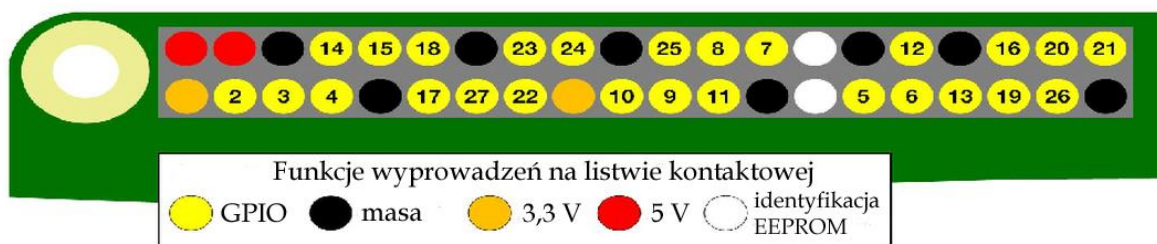
- Układ scalony SoC (*System on Chip*) zawiera procesor główny, procesor graficzny i układy pomocnicze niezbędne do ich połączenia.
- Mikrokomputery *Zero* i *Zero W* mają procesor, układ SoC i pamięć roboczą identyczne jak w modelu Pi-1 (A, A+, B, B+). *Zero* nie ma modemu WiFi, a *Zero WH* ma wlutowaną listwę kontaktową GPIO
- Model Pi-3B różni się od Pi-3B+ tym, że jest wyposażony w złącze WiFi odpowiadające normie 802.11 n, złącze Bluetooth 4.1 LE, układ SoC typu BCM2837, a częstotliwość zegarowa procesora wynosi 1,2 GHz. Pobór prądu wynosi 260 mA, a ze złączami HDMI i WiFi – 360 mA.
- Wspólne gniazdko fonii i wizji w modelach Pi-3 i Pi-4 posiada cztery kontakty, a wtyczka koncentryczna zapadkowa ma cztery pierścienie. Jej szczyt jest połączony z kanałem lewym fonii, drugi pierścień – z prawym, trzeci stanowi masę jak w trójkontaktowych wtyczkach stereofonicznych, a czwarty najbliższy obudowy jest kontaktem dla sygnału wizji (rys. 5.1.2). Żaden z dotychczasowych modeli *Maliny* nie jest wyposażony w wejście mikrofonowe i dlatego jeżeli jest ono potrzebne konieczne jest podłączenie dodatkowego podsystemu dźwiękowego do jednego z gniazd USB. Część modeli radiostacji jest wyposażona we własne podsystemy dźwiękowe, co znacznie upraszcza ich podłączenie do komputerów.

Do aktualizacji zainstalowanego systemu operacyjnego służą polecenia:

```
sudo update && sudo apt upgrade -y && reboot.
```

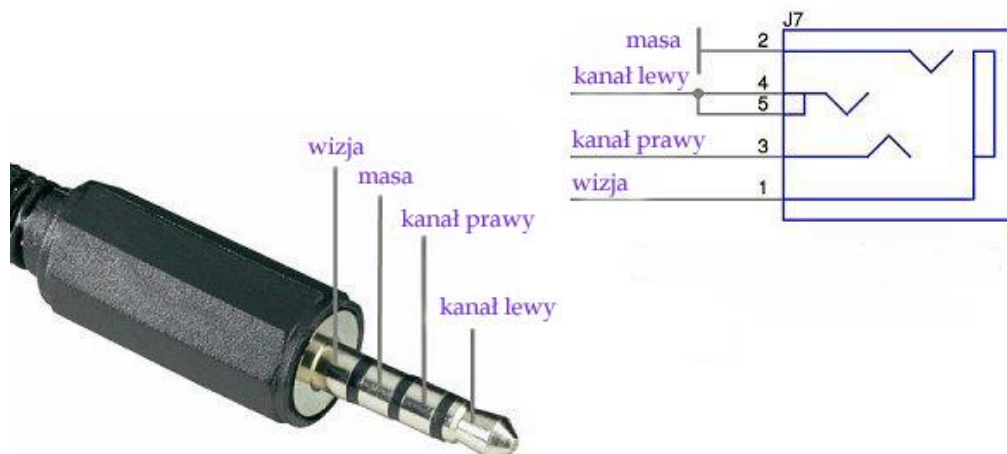
Po zainstalowaniu i po aktualizacji systemu warto zrobić kopię bezpieczeństwa pamięci SD. Moduł można skopiować na PC korzystając z czytnika SD i programu *Win32 Disk Imager* albo *Etcher*. Można także skorzystać z narzędzi Raspbiana otwierając w menu „Accessories” („Narzędzia”) program kopiujący „SD Card Copier” i w jego oknie jako źródło („Copy from device”) wybrać pamięć programu `/dev/mmcblk0` a w polu pamięci docelowej („Copy to device”) pamięć dla kopii, najprawdopodobniej będzie to `/dev/sda`. Po naciśnięciu przycisku *Start* rozpoczyna się kopiowanie, które może trwać nawet do 15 minut. Moduł docelowy musie mieć taką samą pojemność jak źródłowy lub większą.

Listwy kontaktowe na złączu GPIO pozwalają na podłączenie do mikrokomputera płytek rozszerzeń (ang. HAT – *Hardware At Top*) lub innych dowolnych urządzeń peryferyjnych. Ich wybór jest obecnie dosyć obszerny. Niektóre programy krótkofalarskie wykorzystują wyprowadzenia na złączu GPIO jako wyjście generowanych przez *Malinę* zmodulowanych sygnałów wielkiej częstotliwości. W modelu Pi-1 listwa kontaktowa GPIO posiadała 26 kontaktów, a w nowszych została przedłużona do 40 kontaktów przy zachowaniu tego samego porządku na pierwszych 26. *Malina* może być używana do nadawania sygnałów w.cz. zmodulowanych częstotliwościowo (FM), amplitudowo (AM), SSB, SSTV itd.

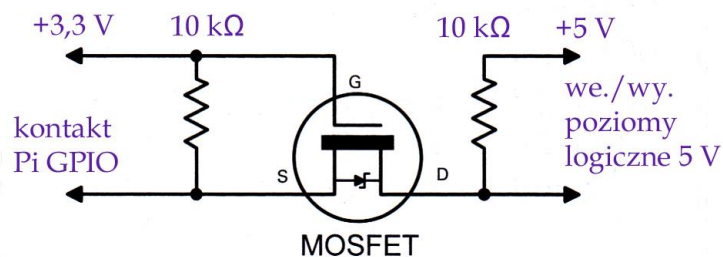


Rys. 5.1.1. Wyprowadzenia na 40-kontaktowej listwie GPIO są identyczne w modelach Pi-2, Pi-3 i Pi-4. Kontakty GPIO2 i GPIO3 są wykorzystywane także przez złącze I2C, GPIO14 i GPIO15 – przez złącze UART, a GPIO7, GPIO8, GPIO9, GPIO10 i GPIO11 – przez złącze SPI. Pozostałe są przeznaczone do dowolnego użytku. GPIO4 jest wyjściem licznika GPCLK0 wykorzystywanym w konstrukcjach nadawczych

Złącza USB pozwalają na podłączenie klawiatury i myszy potrzebnych w trakcie instalacji i uruchamiania programów, zewnętrznych podsystemów dźwiękowych umożliwiających m.in. dodanie wejścia mikrofonowego, odbiorników programowalnych (SDR), mikroprzełączników cyfrowego dźwięku j.np. DV4mini, DVAP i innych urządzeń. W najstarszych modelach do złącza USB podłączany był także modem WiFi. Obecnie stanowi on standardowe wyposażenie *Maliny*.



Rys. 5.1.2. Wyprowadzenia fonii i wizji



Rys. 5.1.3. Dopasowanie poziomów logicznych 3,3 V na złączu GPIO do logiki 5 V

System operacyjny *Maliny* posiada graficzną powierzchnię obsługi (nie miały jej starsze wersje systemu), ale w niektórych przypadkach przydatne jest otwarcie okna wiersza poleceń i podawanie w nim poleceń linuksowych. Do standardowego wyposażenia systemu należą m.in. administrator plików (*File Manager*), administrator programów (*Task Manager*), internetowa przeglądarka *Chromium*, środowiska programistyczne *Javy* i *Pythona*, oprogramowanie *RealVNC* pozwalające na obsługę *Maliny* z komputera PC przez lokalną sieć (konieczne jest zainstalowanie na nim klienta VNC), program terminalowy i wiele innych.

Do zainstalowania dodatkowych programów w wielu wypadkach wystarczy posłużyć się poleceniem `apt-get`:

1) `sudo apt-get update` – aktualizuje spis dostępnych programów, polecenie dla `apt-get` jest poprzedzone przez polecenie `sudo`, które nadaje użytkownikowi uprawnienia administratora (`sudo` = *superuser do*). W systemie z dostępem dla wielu użytkowników konieczne byłoby jeszcze podanie hasła dostępu dla administratora, ale w sytuacji gdy użytkownik korzysta z *Maliny* sam (nie jest ona przewidziana do równoległego korzystania przez wielu użytkowników) uprawnienia administratora uzyskuje się bez hasła.

2) `sudo apt-get install <nazwa programu>` – polecenie instalacji programu o podanej nazwie, przykład `sudo apt-get install apache2` -y powoduje zainstalowanie serwera *apache*.

Zamiast `apt-get` można w nowszych wersjach systemu podawać polecenie `apt`, `sudo apt ...` jest więc równoznaczne z `sudo apt-get ...`

Uwaga: we wszystkich poleceniach Linuksa i nazwach lub oznaczeniach rozróżniane są duże i małe litery. Nazwy wielu plików konfiguracyjnych lub systemowych Rasbiana rozpoczynają się od kropki.

Programy są dostępne w Internecie przeważnie w postaci skompresowanych archiwów. Najczęściej stosowane formaty ze sposobami ich kompresji i rozpakowywania podano w tabeli 5.1.1.

Tabela 5.1.1. Kompresja i rozpakowywanie archiwów linuxowych

Format	Polecenie dla spakowania katalogu	Polecenie rozpakowujące
gzip	tar czvf test.tar.gz <katalog>	tar xzvf test.tar.gz
bzip2	tar cjvf test.tarbz2 <katalog>	tar xjvf test.tar.bz2
xz	tar cJvf test.tar.xz <katalog>	tar xJvf test.tar.xz

Uwagi:

- najszybciej komprimuje gzip,
- najwolniejszym ale dającym najwyższy stopień kompresji jest xz.

W celu odczytania adresu IP *Maliny* w sieci domowej należy przejechać myszą nad symbolem „WiFi” i odczytać adres łącza ethernetowego albo bezprzewodowego w okienku informacyjnym.

W oknie terminalowym (wiersza poleceń) do odczytania adresu służy polecenie *hostname -I* albo *sudo ifconfig*. Pierwsze łączy Ethernetu i WiFi noszą odpowiednio oznaczenia *eth0* i *wlan0*. Złącze pętli wewnętrznej (lo) 127.0.0.1 pozwala na komunikację programów z serwerami pracującymi na tym samym mikrokomputerze. Zależnie od rodzaju i funkcji serwera po adresie podawany jest po dwukropku numer używanego przez niego kanału logicznego (ang. *port*). Polecenie *ifconfig* wyświetla w odpowiedzi wiele dodatkowych przydatnych informacji, ale jeżeli jest to niepożądane można zakres informacji ograniczyć podając oznaczenie łącza np. *sudo ifconfig eth0*.

Aby pracujący na mikrokomputerze serwer był zawsze dostępny w domowej sieci pod tym samym adresem IP należy w modemie internetowym (ang. *router*) zarezerwować dla niego adres statyczny. Sposób rezerwacji zależy od modelu modemu i jego oprogramowania, a więc trzeba go odczytać z okna obsługi albo z instrukcji. Przydzielający adresy IP serwer DHCP rozpoznaje urządzenia po ich adresach MAC (ang. Media Access Control) i przydziela w takim przypadku zarezerwowany adres IP.

Dla skorzystania z następnych punktów rozdziału i skutecznego korzystania z *Maliny* konieczna jest pewna podstawowa znajomość Linuksa. Autor nie może tutaj zamieścić wyczerpującego kursu Linuksa ze względu na jego objętość i wobec tego odsyła czytelników do odpowiedniej literatury. Większości podstawowych czynności można wprowadzić dokonać z poziomu graficznej powierzchni obsługi i nie powinno to przysparzać trudności, ale obsługa systemu z poziomu wiersza poleceń (okna terminalowego) rozszerza w istotny sposób te możliwości. Konieczna jest wówczas jednak znajomość struktury i znaczenia katalogów oraz konceptu uprawnień zarówno użytkownika jak i samych programów. Programy, pliki tekstowe i inne dowolnego rodzaju mogą być przykładowo wywoływane lub przeglądane przez dowolnych użytkowników, ale ich modyfikacja może już wymagać uprawnień administratora (*root*) albo właściciela pliku. Z kolei program mający uprawnienia administratora ma znacznie szersze możliwości działania i może je w swoim ograniczonym zakresie przekazywać zwykłym użytkownikom nie mającym uprawnień administratora. Oprócz tych dwóch poziomów uprawnień istnieje jeszcze poziom dla grupy użytkowników. Dla każdego z rodzajów użytkowników przydzielane są oddzielnie uprawnienia do odczytu pliku, zapisu (modyfikacji) i wykonywania jeżeli jest to program wykonywalny albo skrypt rozkazowy *shella*. Do zmiany uprawnień w zakresie dozwolonym dla użytkownika służy polecenie *chmod*. Właściciel pliku lub programu ma oczywiście pełne prawa dostępu do niego i modyfikacji. W spisach zawartości katalogów uprawnienie podawane są w postaci ciągu *-rwxrwxrwx* przy czym pierwsze trzy pozycje odpowiadają uprawnieniom właściciela, drugie członków grupy roboczej, a trzecie wszystkich pozostałych użytkowników. W miejscu nieprzyznanych uprawnień występują myślniki np. *-rwxr-xr-x* pozwala wszystkim poza właścicielem jedynie na odczyt i wykonywanie programu (przyznanie uprawnień wykonywania bez odczytu nie jest możliwe). Uprawnienia można także zapisywać w postaci liczby ósemkowej, przytoczonemu przykładowi odpowiadałby zapis *755*. Dla plików nie będących programami np. tekstowych często przyznawane są uprawnienia *644*, pozwalające właścicielowi na odczyt i zmianę zawartości pliku, a pozostałym użytkownikom tylko na odczyt (jeżeli jest to pożądané). W przypadku prywatnie używanego mikrokomputera poziom uprawnień dla grupy jest przeważnie nieistotny, natomiast w przypadku programów instalowanych na poziomie administratora właścicielem jest użytkownik-administrator noszący standardowo nazwę *root* i stąd istotne

jest odróżnienie uprawnień właściciela i pozostałych użytkowników (przeważnie tylko jednego jedynego użytkownika).



Fot. 5.1.4. Model 3B+



Fot. 5.1.5. Model Zero W2 z włutowaną listwą

5.1.1. System dźwiękowy

System dźwiękowy Linuksa nosi nazwę ALSA i jest zasadniczo zbiorem sterowników urządzeń fizycznych, sterowników skompilowanych i umieszczonych w aktualnym jądrze Linuksa (lub jego pochodnych). Na *Malinie* często występuje w postaci zestawu *BlueALSA*. Wariant ten umożliwia korzystanie ze złącza Bluetooth, ale nie zawiera serwera dźwięku *PulseAudio*. Serwer ten jest szczególnie przydatny w udostępnianiu strumieni cyfrowego dźwięku innym programom. W ograniczonym zakresie pozwala on nawet na zmianę częstotliwości próbkowania dźwięku. Serwer dźwiękowy zarządza systemem ALSA. Do najbardziej rozpowszechnionych należą obecnie *PulseAudio* i *Jack*, z tym, że pierwszy z nich jest bardziej rozpowszechniony, zwłaszcza na komputerach biurkowych. Drugi z nich jest przewidziany raczej do zawodowych zastosowań w muzyce. Ze względu na występujące między nimi konflikty użytkownik powinien wybrać jeden z nich, najbardziej pasujący do konkretnej sytuacji.

Instalacja *PulseAudio* na *Malinie* wymaga:

1. Uaktualnienia spisu dostępnych programów za pomocą polecenia
`sudo apt update`
2. Instalacji przy użyciu polecenia
`sudo apt install -y pulseaudio`
3. Sprawdzenia prawidłowości instalacji przez zapytanie o wersję programu
`pulseaudio -- version`
4. Dobrze jest też zainstalować graficzną powierzchnię obsługi serwera:
`sudo apt install -y pavucontrol`

5.1.2. Pozostałe złącza

Modele poprzedzające Pi-4 są wyposażone w dwa złącza szeregowo typu UART, z których pierwsze (sprzętowe) dysponuje pełną funkcjonalnością, a drugie symulowane programowo (miniUART) – tylko ograniczoną. Złącza te występują w obecnych wersjach systemu operacyjnego pod oznaczeniami `/dev/serial0` i `/dev/serial1` zamiast jak poprzednio pod `/dev/tty0` i `/dev/tty1`. Przyporządkowanie ich oznaczeń zawiera plik `/boot/config.txt`. Zalecane jest używanie nazwy `serial0` dla złącza korzystającego z wyprowadzeń 14 i 15 na listwie GPIO. W praktyce oznacza to, że oznaczenie `serial0` odpowiada złączu programowemu w modelach z włączonym kanałem Bluetooth, a sprzętowemu w pozostałych przypadkach. Możliwa jest także zamiana złączy przez dodanie na końcu pliku `/boot/config.txt` za pomocą edytora `nano` na jego końcu linii `dtoverlay=pi3-miniuart-bt` i `core_freq=250`.

Model Pi-4 posiada cztery dodatkowe sprzętowe złącza UART noszące numery 2 – 5. Są one dostępne pod oznaczeniami `/dev/tty/AMA1` – `/dev/tty/AMA4`, a ich przyporządkowanie jest również podane w pliku `/boot/config.txt`.

Spis złączy szeregowych wywołuje się za pomocą polecenia:

```
ls -l /dev | grep serial
```

Polecenie `cat /dev/serial0` kieruje na monitor wszystkie dane docierające do złącza `serial0`.

5.1.3 Automatyczne wywoływanie programów

W znacznej części jeśli nie w większości przypadków mikrokomputery w rodzaju *Maliny* są wykorzystywane do zastosowań działających automatycznie bez ingerencji użytkownika (oczywiście po zainstalowaniu, skonfigurowaniu i uruchomieniu programów). W tych przypadkach pracują one bez podłączonych klawiatury, myszy i monitora. Ręczne wywoływanie programów za każdym razem po włączeniu byłoby więc mocno niewygodne nawet gdyby chodziło tylko o naciśnięcie myszą lub palcem symbolu na ekranie. Na szczęście Linuks daje możliwości automatycznego wywoływania programów („autostartu”) bezpośrednio lub za pomocą skryptów *shella*. Skrypty *shella* mogą zawierać dowolne polecenia systemu, zapytania, warunki itp. i w związku z tym dają możliwości zareagowania na różne sytuacje, które mogą wystąpić w trakcie uruchamiania. Przykładowo program korzystający z dostępu do Internetu przez WiFi musi poczekać ze startem do nawiązania połączenia z modemem internetowym i otrzymania od niego adresu IP. Sprawdzenie tego warunku i ewentualne oczekiwanie na jego spełnienie łatwo jest zapewnić przy użyciu odpowiednio przygotowanego skryptu *shella*. Bez tego wywołanie programu mogłoby nastąpić za wcześniej i nie mógłby on pracować zgodnie z oczekiwaniami.

W najprostszym przypadku można uruchamiany program dopisać do pliku `rc.local` za pomocą polecenia wywołującego edytor `nano`:

```
sudo nano /etc/rc.local
```

Przykładowo dla wywołania serwera odbiornika RTL-SDR należy w pliku dopisać linię

```
/usr/local/bin/rtl_tcp -a 192.168.1.88
```

W pliku należy podać pełną ścieżkę dostępu do wywoływanego programu. W przytoczonym przykładzie ostatnim parametrem jest adres IP w domowej sieci (jeżeli został on wcześniej na stałe przypisany w modemie internetowym). Zamiast bezpośredniego wywołania programu do pliku można dodać wywołanie skryptu *shella* wykonującego więcej i bardziej złożonych czynności związanych z uruchomieniem programu. Umieszczenie na końcu linii znaku `&` powoduje wykonywanie polecenia w tle zamiast oczekiwania jego zakończenia przed przejściem dalej.

Bardziej uniwersalnym rozwiązaniem jest podanie w linii wywoławczej zmiennej `$_IP` zawierającej adres IP (wpisywany do niej po nawiązaniu połączenia z siecią)

```
/usr/local/bin/rtl_tcp -a $_IP &
```

Przykładowy skrypt *shella* oczekujący na otrzymanie adresu i przedłużający w miarę potrzeby czas oczekiwania może wyglądać następująco:

```
counter=1
until [ [ $_IP ] ] ; do
    _IP=$(hostname -I)
    sleep 1
    echo „\`date -u\`:Czekam na adres IP – sekund(y):” ( ( counter++ ) )
    if [ $counter -gt 30 ]
```



```

then
    echo „\`date -u\` :Brak połączenia z siecią. Koniec oczekiwania”
exit
fi
done
/usr/local/bin/rtl_tcp -a $_IP &

```

Skrypt oczekuje w pętli na otrzymanie adresu IP. W przykładzie w każdym przebiegu pętli czas oczekiwania wynosi sekundę, a cały czas jest ograniczony do 30 sekund. Po bezowocnym oczekiwaniu skrypt kończy pracę bez uruchomienia serwera.

Drugą możliwością automatycznego uruchamiania programów jest skorzystanie z administratora usług *Systemd*. Programy uruchamiane w tle są traktowane jak usługi w sposób zbliżony do znanego z Windows. Usługi te mogą być w prosty sposób uruchamiane i zatrzymywane za pomocą poleceń:

```

sudo systemctl start <program>.service
sudo systemctl stop <program>.service

```

Zamiast oznaczenia <program> podawana jest nazwa uruchamianego lub zatrzymywanego programu, oczywiście bez nawiasów.

Polecenia można podawać także w skróconej postaci np.

```

sudo systemctl stop <program>

```

Informacje o uruchamianym programie należy wprowadzić do pliku */etc/systemd/system*

Przykładowo dla wywołania serwera RTL-SDR plik deklarujący uruchamianą jednostkę (usługę) zawiera następującą treść:

```

[Unit]
Description=RTL-SDR Server
Wants=network-online.target
After=network-online.target
[Services]
ExecStart=/bin/sh -c '/usr/local/bin/rtl_tcp -a (hostname -I) '
WorkingDirectory=/home/pi
StandardOutput=inherit
StandardError=inherit
Restart=always
User=pi
[Install]
WantedBy=multi-user.target

```

Pierwsza sekcja zawiera opis usługi i wymagania, które muszą być spełnione przed i po uruchomieniu. Następna sekcja zawiera informacje o zachowaniu się usługi i jej wywołanie w linii *ExecStart*. Przy wywoływaniu programu jako usługi nie można skorzystać ze zmiennej *\$_IP* dlatego też w nawiasie występuje zapytanie o adres IP. Samo polecenie wygląda tak, jakby było podane w wierszu poleceń i jest wykonywane przez interpretator plików *shella*. Odczytuje on informacje zawarte w pojedynczym cudzysłowie. Zapytanie o adres IP brzmi tym przypadku *hostname -I*. Po linii wywołania podany jest katalog roboczy oraz kanały wyjścia i meldunków błędów. *Inherit* oznacza w tym przypadku dostosowanie się do ustawień ogólnych. Normalnie są więc one wydawane na monitor. Po zatrzymaniu serwera startuje on ponownie zgodnie z treścią linii *Restart*. Ostatnia linia zapewnia dalszą pracę Linuksa.

Trzecią możliwością jest plik autostartu */etc/xdg/lxsession/LXDE-pi/autostart*. Użytkownik może dopisać w nim dowolne programy, które mają być uruchamiane automatycznie.

W oknie terminalowym należy wywołać edytor nano:

```

sudo nano /etc/xdg/lxsession/LXDE-pi/autostart

```

Należy w nim dodać nową linię powyżej linii *@xscreensaver* i wpisać do niej polecenie wywołania programu w postaci takiej jak podawana w wierszu poleceń. Po dopisaniu danych do pliku należy nacisnąć kombinację Ctrl-X, odpowiedzieć Y na zapytanie i nacisnąć klawisz Enter. Po ponownym uruchomieniu *Maliny* program jest wywoływany automatycznie.

Do automatycznego uruchamiania programów posiadających graficzną powierzchnię obsługi można także założyć w głównym katalogu użytkownika (a więc `/home/pi` dla użytkownika *pi*) katalog `.config/autostart` korzystając z poleceń

```
cd
```

```
mkdir .config/autostart
```

i dla każdego uruchamianego w ten sposób programu należy założyć plik o rozszerzeniu `.desktop` i zawartości wzorowanej na poniższej:

```
# Plik .config/autostart/jmacs.desktop
```

```
[Desktop Entry]
```

```
Type=Application
```

```
Name=jmacs
```

```
Comment=Start jmacs in LXTerminal
```

```
Exec=lxterminal -e jmacs
```

Linia `Exec` zawiera polecenie wywołujące program. W tym przykładzie otwierane jest okno edytora *jmacs*.

Dla zwykłego uruchomienia programu wystarczy założenie odnośnika do odpowiedniego pliku `.desktop` w `/usr/share/applications`:

```
cd .config/autostart
```

```
ln -s /usr/share/applications/lxterminal.desktop
```

Katalog `.config/autostart` jest katalogiem prywatnym każdego z użytkowników.

5.1.4. Zdalna obsługa *Maliny*

W sytuacjach kiedy *Malina* ma pracować autonomicznie bez klawiatury, myszy i monitora dostęp do niej w celu zainstalowania nowych programów, dokonania zmian w konfiguracji itp. jest możliwy w protokole SSH.

Na graficznej powierzchni obsługi należy wybrać punkt „Preferences” („Ustawienia”) „Raspberry Pi Configuration” („Konfiguracja Raspberry Pi”), znaleźć zakładkę „Interfaces” („Złącza”), włączyć SSH i potwierdzić przyciskiem OK.

Drugą możliwością jest podanie w oknie terminalowym następujących poleceń:

```
sudo raspi-config
```

przejsć do pozycji 5 („Interfacing Options” – „Konfiguracja połączeń”), potem w punkcie P2 włączyć SSH.

W następnym oknie należy wybrać „Yes” w celu włączenia serwera i na zakończenie nacisnąć OK. Na PC należy następnie zainstalować program terminalowy IP. Najbardziej znanym przedstawicielem tej kategorii jest PuTTY. W polu adresowym PuTTY podaje się adres IP *Maliny* i kanał logiczny 22. Odpowiednim programem terminalowym dla Androida jest *VX ConnectBot*.

Połączenia przez SSH są połączeniami tekstowymi typu terminalowego i służą do komunikacji z *Maliną* na poziomie wiersza poleceń, co w większości przypadków jest wystarczająco dobre i praktyczne. W przypadku niektórych programów, zorientowanych na obsługę z powierzchni graficznej wygodniejsze może być połączenie VNC. W konfiguracji *Maliny* należy uruchomić wchodzący w skład systemu serwer RealVNC, a na PC zainstalować klienta RealVNC (www.realvnc.com). Do nawiązania połączenia klienta z *Maliną* należy podać w polu adresowym klienta jej adres IP. Przy pierwszej próbie nawiązania połączenia przez klienta PC wyświetlany jest meldunek o nie rozpoznaniu serwera. Po naciśnięciu przycisku „Kontynuuj” („Continue”) dane serwera zostają zapisane i połączenie staje się możliwe. Po nawiązaniu połączenia należy tak samo jak w przypadku bezpośredniego dostępu przez klawiaturę podać nazwę użytkownika i hasło dostępu. Standardowo dla świeżo uruchomionego systemu użytkownik nosi nazwę *pi*, a hasłem jest *raspberrypi*. Co najmniej hasło należy jednak zmienić jak najszybciej na własne, aby nie ułatwiać dostępu osobom niepożądanym.

W celu włączenia serwera z graficznej powierzchni obsługi na *Malinie* należy wybrać punkt „Preferences” („Ustawienia”) „Raspberry Pi Configuration” („Konfiguracja Raspberry Pi”), znaleźć zakładkę „Interfaces” („Złącza”) i włączyć VNC.

Alternatywą do VNC może być *Team Viewer*.

5.2. Programy dla emisji cyfrowych

5.2.1. Emisja WSPR

Program *WsprryPi* umożliwia uruchomienie radiolatarni WSPR małej mocy. Na wyjściu GPIO4 listwy kontaktowej wyprowadzony jest sygnał w.cz. o mocy około 10 mW generowany za pomocą wyjściowego układu modulacji szerokości impulsów (GPCLK0). Bezpośrednie połączenie anteny z wyjściem jest niekorzystne z wielu względów. Po pierwsze na wyjściu *Maliny* generowany jest bogaty w (nieparzyste) harmoniczne sygnał prostokątny. Powodowałoby to transmisję sygnałów zakłócających na częstotliwościach harmonicznym. W przypadku ogólnym mogą one leżeć poza pasmami amatorskimi i powodować zakłócenia innych służb. Dla otrzymania czystego sygnału wielkiej częstotliwości (w.cz.) konieczne jest użycie kilkuczłonowego filtra dolnoprzepustowego. Połączenie anteny bezpośrednio (nawet przez filtr dolnoprzepustowy) do wyprowadzenia procesora niesie ze sobą niebezpieczeństwo jego uszkodzenia przez zaindukowane w antenie napięcia w.cz. (np. wskutek wyładowań burzowych) albo przez zaindukowane ładunki statyczne (maksymalne dopuszczalne napięcie na wyprowadzeniach GPIO wynosi 3,3 V). Dlatego też między wyjście procesora i antenę powinno się włączyć separator tranzystorowy – wzmacniacz wielkiej częstotliwości, który dodatkowo oprócz izolacji anteny od procesora zapewnia zwiększenie mocy wyjściowej nadajnika, przykładów do 100 mW lub więcej. W przypadku uszkodzenia nadajnika konieczna jest tylko wymiana tranzystora w stopniu końcowym, a nie całego, znacznie droższego mikrokomputera. Przykłady konstrukcji filtrów i wzmacniacza mocy podano w poprzednich rozdziałach, a także w tomach 56 i 57 „Biblioteki”.

Najbardziej znanym programem nadawczym WSPR jest *WsprryPi*. Jest on dostępny bezpłatnie w Internecie. Program pracuje na *Malinach* 1 – 4 i *Zero* w zakresie częstotliwości do 250 MHz. W celu zainstalowania należy otworzyć okno wiersza poleceń korzystając z kombinacji Ctrl-Alt-T i podać w nim polecenie:

```
git clone https://github.com/g4wnc/WsprryPi.git
```

Następnie należy przejść do nowoutworzonego katalogu za pomocą polecenia `cd WsprryPi`.

Dla skompilowania programu należy podać polecenie `make`.

Ostatnim krokiem jest instalacja za pomocą polecenia `sudo make install`.

Program posiada tryb próbny, w którym generuje on ton ułatwiający sprawdzenie prawidłowości pracy i właściwego dostrojenia nadajnika. W celu jego wywołania należy w oknie wiersza poleceń podać polecenie (w przykładzie dla pasma 20 m): `sudo ./wspr -test-tone 14.097e6`.

Do zakończenia transmisji służy kombinacja Ctrl-C. W przypadku prób w innym paśmie należy podać w poleceniu wywoławczym odpowiednią częstotliwość. Jakość (czystość) nadawanego sygnału można sprawdzić na oscyloskopie, amplitudę można także zmierzyć woltomierzem w.cz. Można także jako przyrządu kontrolnego użyć odbiornika dostrojonego do częstotliwości pracy. Pozwala to także na wprowadzenie korekty dostrojenia w programie, tak aby trafić w podzakres WSPR.

Po sprawdzeniu poprawności pracy można już wywołać program radiolatarni za pomocą polecenia:

```
sudo ./wspr -repeat -offset -selfcalibration OE1KDA JN88 10 20m .
```

Oczywiście podany w przykładzie znak wywoławczy i lokator należy zastąpić własnymi. Przedostatni parametr jest mocą nadajnika w dBm i trzeba go także dostosować do własnej sytuacji, przy 100 mW jest to 10 dBm. Ostatni parametr wywołania podaje pasmo pracy nadajnika.

Wywołanie programu z parametrem `-h` powoduje wyświetlenie pomocy informującej również o innych parametrach wywołania. W celu zapewnienia kalibracji czasu i częstotliwości *Malina* powinna być stale połączona z Internetem. Korzysta ona wówczas z usługi czasu NTP.

Meldunki odbioru przez inne stacje można znaleźć w witrynie www.wsprnet.org.

Ręcznego uruchamiania programu po każdorazowym włączeniu mikrokomputera można uniknąć zakładając skrypt wywołujący w katalogu `/etc/int.d`. W podanym przykładzie nosi on nazwę `wspr20` gdyż uruchamia nadawanie w paśmie 20 m.

Przykładowa treść skryptu wygląda jak następuje (podane w nim dane stacji należy zastąpić własnymi):

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          FWsprryPI - WSPR auf 20m 10mW, DL7VDX
# Required-Start:
# Required-Stop:
```

```

# Default-Start:      2 3 4 5
# Default-Stop:       0 1 6
# Short-Description:  WSPR z automatyczną kor. częst. NTP
# Description:         WSPR konfiguracja dla 20m, 10mW, JO62, NTP
Time/QRG-kor., nadawanie co 10 minut
### END INIT INFO

# Akcje
case "$1" in
    start)
        echo "starte WSPR Bake auf 20m mit 10mW aus JO62, NTP
Time&QRP korrektur"
        # START
        sudo /home/pi/WsprrryPi/wspr --repeat --self-calibration DL7VDX
JO62 10 20m 0 0 0 0
        ;;
    stop)
        echo "stoppe WSPR Bake auf 20m"
        # STOP
        sudo killall wspr
        ;;
    restart)
        echo "... restart WSPR Bake"
        # RESTART
        ;;
esac

```

exit 0

Następnie należy przyznać mu uprawnienia:

```
sudo chmod +x /etc/init.d/wspr20
```

i wpisać do pliku *update-rc.d*

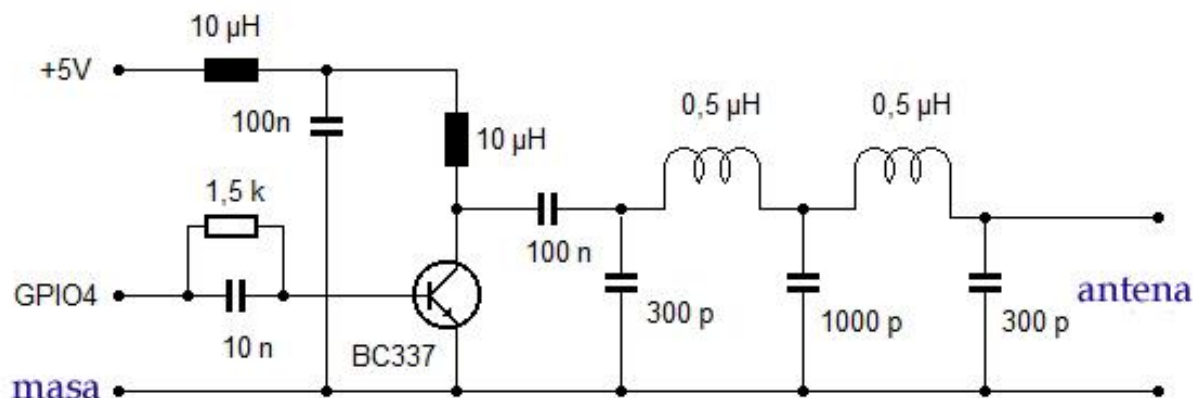
```
sudo update-rc.d wspr20 defaults.
```

Oprócz automatycznego wywołania możliwe jest też jego wywołanie i zatrzymanie pracy za pomocą poleceń (odpowiednio):

```
sudo /etc/init.d/wspr20 start
```

i

```
sudo /etc/init.d/wspr20 stop.
```



Rys. 5.2.1.1. Przykładowy schemat wzmacniacza 100 mW z filtrem dla pasma 30 m (źródło <https://www.elektronik-labor.de/HF/WSPR.html>)



Fot. 5.2.1.2. Konstrukcja wzmacniacza mocy na płytce dziurkowanej (źródło <https://www.elektronik-labor.de/HF/WSPR.html>)

Dla wywołania programu jako usługi należy założyć plik *wsprrypi.service* w katalogu

/etc/systemd/system

nano wsprrypi.service

i wpisać do niego następujące dane

[Unit]

Description=Wsprrypi

After=multi-user.target

[Service]

Type=forking

User=root

ExecStart=/usr/bin/screen -d -m -S wspr wspr -r OE1KDA JN88 10 20m

ExecStop=/usr/bin/killall -w -s 2 wspr

WorkingDirectory=root

[Install]

WantedBy=multi-user.target

Po zapisaniu pliku w pamięci SD użytkownik ma do dyspozycji następujące polecenia:

sudo systemctl start wsprrypi – dla uruchomienia usługi

sudo systemctl stop wsprrypi – dla wyłączenia usługi

sudo systemctl status wsprrypi – dla sprawdzenia jej pracy

sudo systemctl enable wsprrypi – wywołanie po starcie systemu

sudo systemctl disable wsprrypi – wyłączenie automatycznego startu

5.3. Programy terminalowe dla emisji cyfrowych

5.3.1. Fldigi

Zalecane jest aktualizowanie spisu programów dostępnych do instalacji i w miarę możliwości korzystanie z aktualnej wersji systemu operacyjnego. W przypadku pojawienia się meldunków o błędach w trakcie instalacji warto na początek upewnić się czy w poleceniach, nazwach programów, numerach, ewentualnych parametrach nie ma literówek. W przypadku gdy ślady po nieudanej instalacji uniemożliwiają dalsze próby można pozostałości po instalowanym programie usunąć za pomocą polecenia *sudo apt purge <nazwa programu>*.

Fldigi jest programem terminalowym służącym do komunikacji emisjami PSK31, PSK63, Olivią, MFSK, DominoEX, Contestią, MT63, systemem dalekopisowym Hella, Throb, Thor, RTTY, FSQ i pochodnymi od nich. Jego instalację można podzielić na kilka następujących kroków:

1. *sudo apt update && sudo apt upgrade && reboot* – aktualizacja systemu operacyjnego, jeżeli jest potrzebna. Jeżeli system był niedawno aktualizowany krok ten można pominąć. Polecenie jest podawane w oknie wiersza poleceń.

2. Powiększenie obszaru pamięci wirtualnej na dysku SD. W tym celu należy otworzyć w edytorze *nano* plik */etc/dphys-swapfile* za pomocą polecenia *sudo nano /etc/dphys-swapfile* i w linii

CONF_SWAPSIZE=100 dokonać zmiany na CONF_SWAPSIZE=1024. W celu zapisania zmiany na dysku należy nacisnąć kombinację Ctrl-X i następnie klawisz *Enter*. Następnie należy ponownie wystartować *Malinę* aby zaczęła korzystać z powiększonego obszaru. Po zakończeniu instalacji Fldigi można ponownie powrócić do ustawienia CONF_SWAPSIZE=100 edytując plik w ten sam sposób.

3. W oknie terminalowym (wiersza poleceń) należy następnie podać polecenie `sudo nano /etc/sourceslist`, znaleźć w nim linię rozpoczynającą się od `#deb-src` i skasować w niej początkowy krzyżyk oznaczający komentarz. Po zapisaniu zmian za pomocą kombinacji Ctrl-X i naciśnięciu klawisza „Enter” należy ponownie wystartować *Malinę*.

4. Dla uproszczenia dalszego procesu instalacji należy zainstalować zestaw *aptitude* za pomocą polecenia `sudo apt install -y aptitude` i zaktualizować spis zawartości za pomocą polecenia `sudo aptitude update`.

5. Następnie trzeba zainstalować dodatkowe pliki wymagane przez Fldigi za pomocą polecenia `sudo aptitude build-dep fldigi`. Na wyświetlane w trakcie instalacji pytania należy odpowiadać „Y”.

6. Zasadniczo zestaw bibliotek *libxft-dev* powinien zostać zainstalowany w punkcie 5, ale w przypadku gdyby tak się nie stało albo gdyby instalacja nie była kompletna należy je uzupełnić za pomocą polecenia

`sudo aptitude install libxft-dev pavucontrol`.

7. Po zakończeniu powyższych prac przygotowawczych można przystąpić do instalacji fldigi. W oknie terminalowym (wiersza poleceń) należy podawać kolejno następujące polecenia:

`cd ~`

`wget http://www.w1hkj.com/files/fldigi/fldigi-4.1.08.tar.gz` – numer pobieranej wersji należy zastąpić przez numer aktualnej

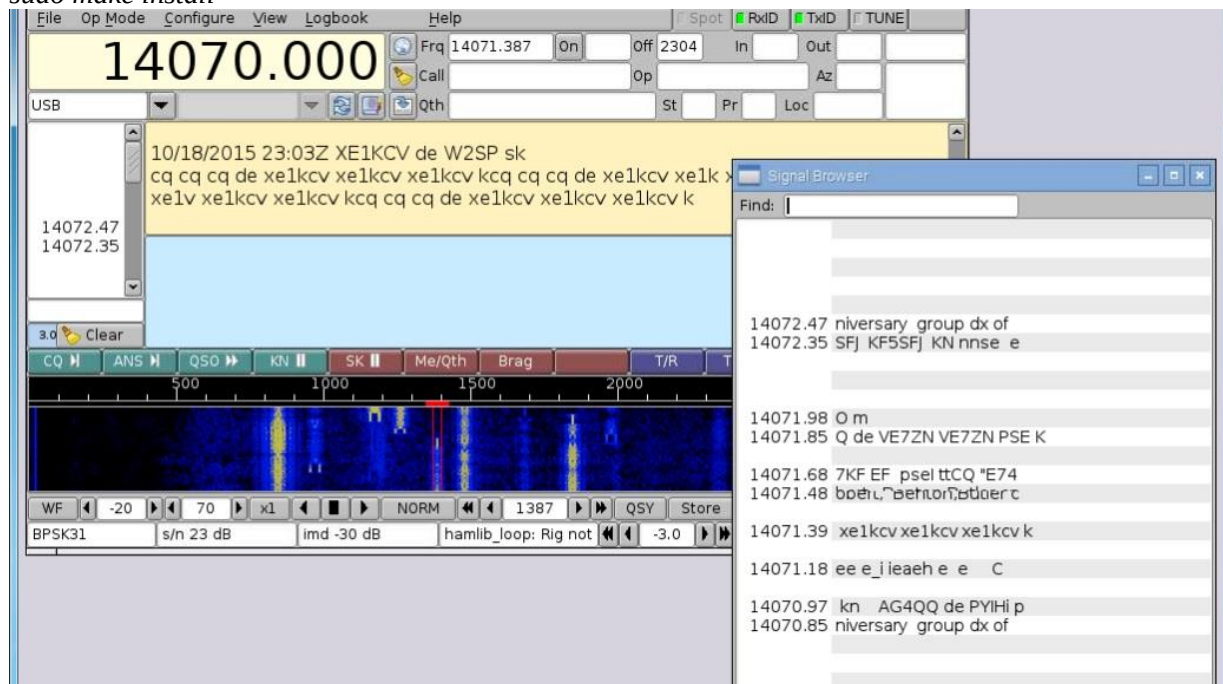
`tar xvfz fldigi-4.1.08.tar.gz` – w poleceniu rozpakowującym także należy podać numer aktualnej wersji

`cd fldigi-4.1.08` – przejście do katalogu fldigi

`./configure`

`make`

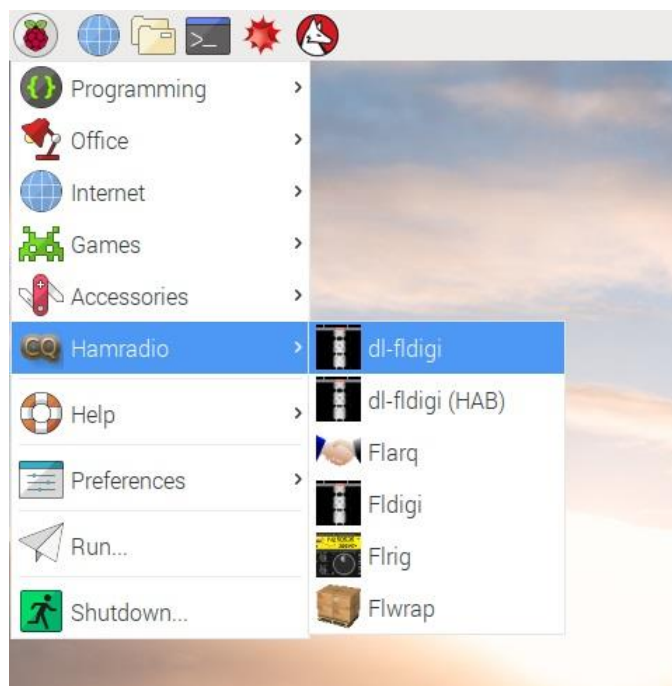
`sudo make install`



Rys. 5.3.1.1. Okno główne Fldigi na *Malinie*

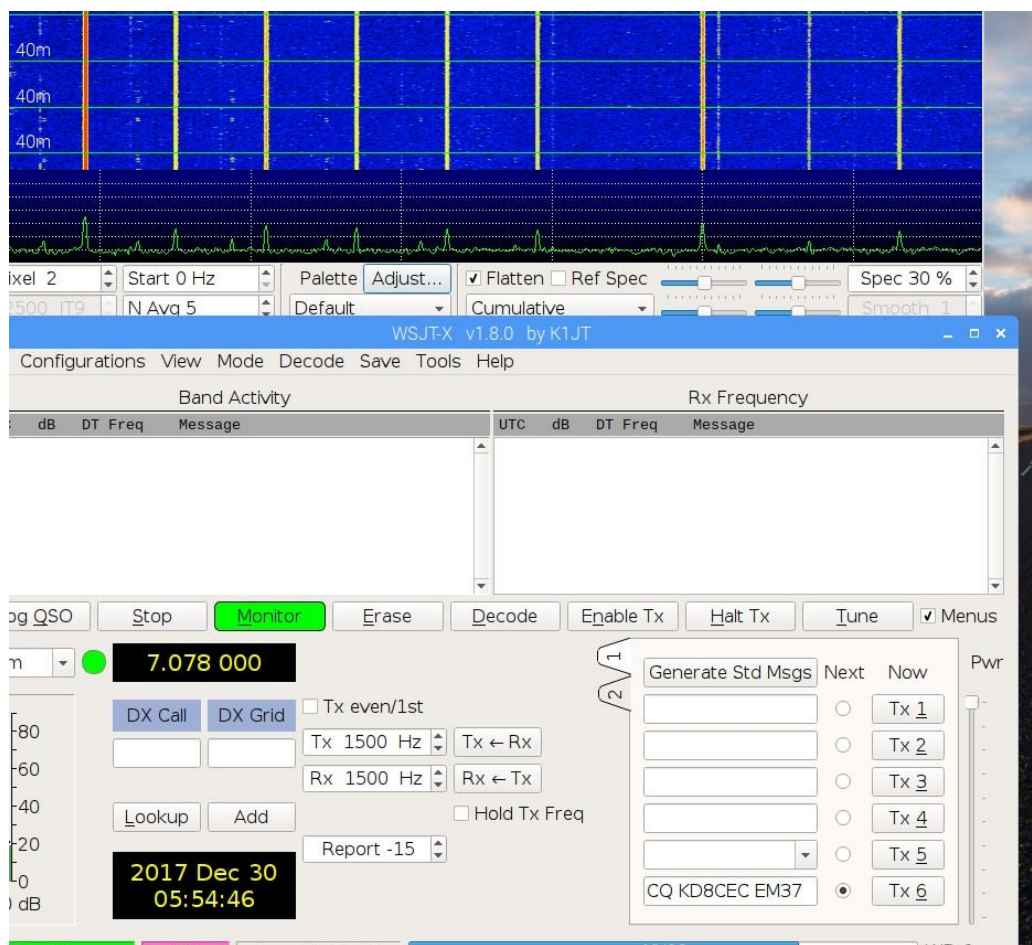
Po przeprowadzeniu instalacji konieczna jest konfiguracja Fldigi. Dodatkowy system dźwiękowy USB powinien być włączony do gniazdka USB przez wywołaniem Fldigi, aby można go było podać w konfiguracji programu (w zakładce „Audio” „USB Audio Device...”).

Objętość pamięci wirtualnej należy ponownie przestawić na CONF_SWAPSIZE=100.



Rys. 5.3.1.2. Wywołanie programów z zestawu Fldigi

5.3.2. WSJT-X



Rys. 5.3.2.1. Okno główne WSJT-X dla Maliny

WSJT-X jest programem terminalowym służącym do łączności przy użyciu słabych sygnałów emisjami należącymi do grupy WSJT, takimi jak JT65, FT8, FT4 itd. Obecnie FT8 dominuje nie tylko na falach krótkich, ale cieszy się popularnością w łącznościach naziemnych w pasmach 2 m i 70 cm, a także w łącznościach przez satelitę QO-100. Aktualne wersje programu znajdują się w witrynie <https://physics.princeton.edu/pulsar/k1jt/wsjsx.htm>.

W celu zainstalowania wsjtx z pobranego archiwum należy:

1. Zainstalować dodatkowe pliki wymagane przez program podając w oknie terminalowym (wiersza poleceń) kolejno polecenia

```
sudo apt install libqt5multimedia5-plugins libqt5serialport5
```

```
sudo apt install -y libqt5sql5-sqlite libfftw3-single3
```

2. W przeglądarce internetowej otworzyć stronę

<https://physice.princeton.edu/pulsar/k1jt/wsjsx.html>

3. Znaleźć na stronie aktualne archiwum dla *Maliny* i pobrać je. Nazwa pliku kończy się na *_armhf.deb*.

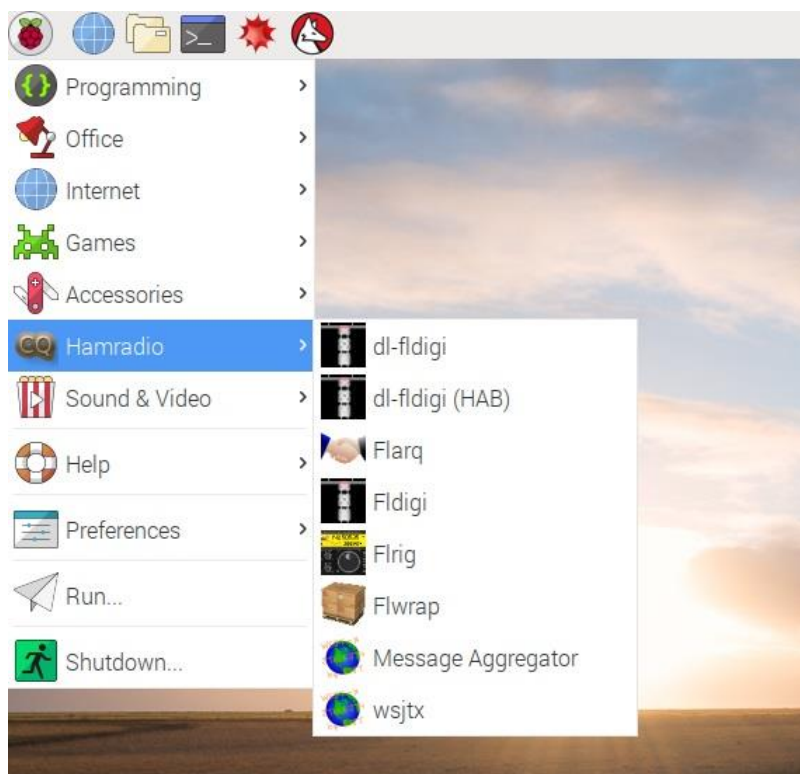
4. W oknie wiersza poleceń podać kolejno

```
cd ~
```

```
sudo dpkg -i ~/Downloads/wsjtx_2.1.2_armhf.deb
```

– numer wersji należy zastąpić przez aktualny.

Korzystanie z programu wymaga wprowadzenia danych konfiguracyjnych i informacji o stacji. Wśród nich należy wybrać podsystem dźwiękowy USB podobnie jak w przypadku Fldigi. W przypadku aktualizacji już zainstalowanego programu należy wykonać kroki 2 – 4.



Rys. 5.3.2.2. Wywołanie WSJT-X

5.3.3. JS8Call

JS8Call jest programem pochodnym od wsjtx i pozwala na prowadzenie dialogów przy wykorzystaniu emisji FT8. Instalacja plików z nim związanych przebiega w ten sam sposób jak w p. 1 wsjtx, o ile nie zostały już wcześniej zainstalowane. Następnie

1. W przeglądarce internetowej należy otworzyć stronę

files.js8call.com/latest.html

2. Pobrać archiwum programu podając w oknie wiersza poleceń polecenia:

```
cd ~
```

```
sudo dpkg -i ~/Downloads/js8call*
```

Po zainstalowaniu się programu konieczna jest jak zwykle jego konfiguracja.

5.3.4. QSSTV

QSSTV jest cieszącym się uznaniem programem linuksowym do łączności telewizyjnych z wolnym wybieraniem linii – SSTV. Program i instrukcję do niego można pobrać z witryny

<http://users.telenet.be/on4qz>

Kolejne kroki instalacji należy:

1. W oknie terminalowym przejść do głównego katalogu użytkownika

```
cd ~
```

2. Zainstalować pliki wymagane przez QSSTV

```
sudo apt update
```

```
sudo apt install -y g++ libfftw3-dev qt5-default libpulse-dev
```

```
sudo apt install -y libhamlib-dev libasound2 libv4l-dev
```

```
sudo apt install -y libopenjp2-7 libopenjp2-7-dev
```

3. Otworzyć w przeglądarce internetowej stronę

<http://users.telenet.be/on4qz>

i pobrać z niej aktualną wersję programu

4. Zainstalować program korzystając z poleceń:

```
cd ~
```

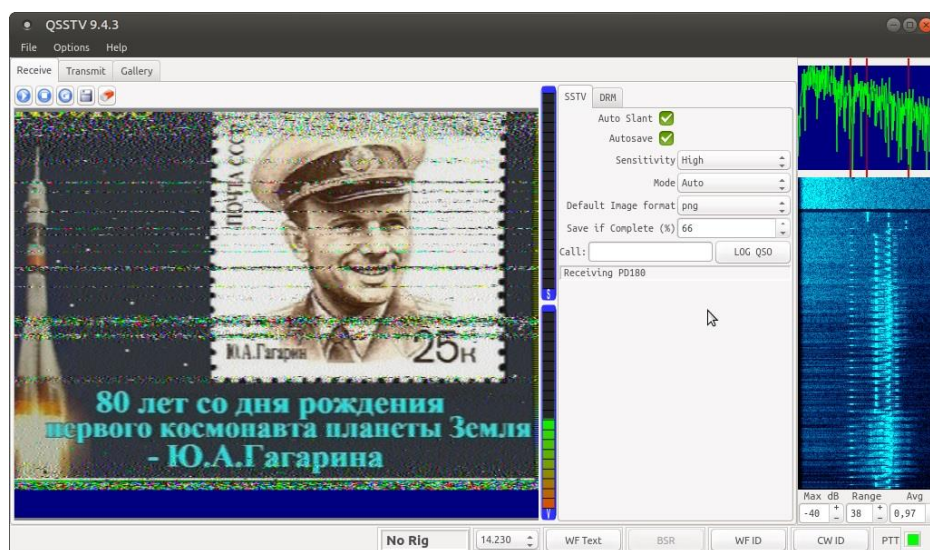
```
unzip ~/Downloads/qsstv*
```

```
cd ~/qsstv*
```

```
qmake
```

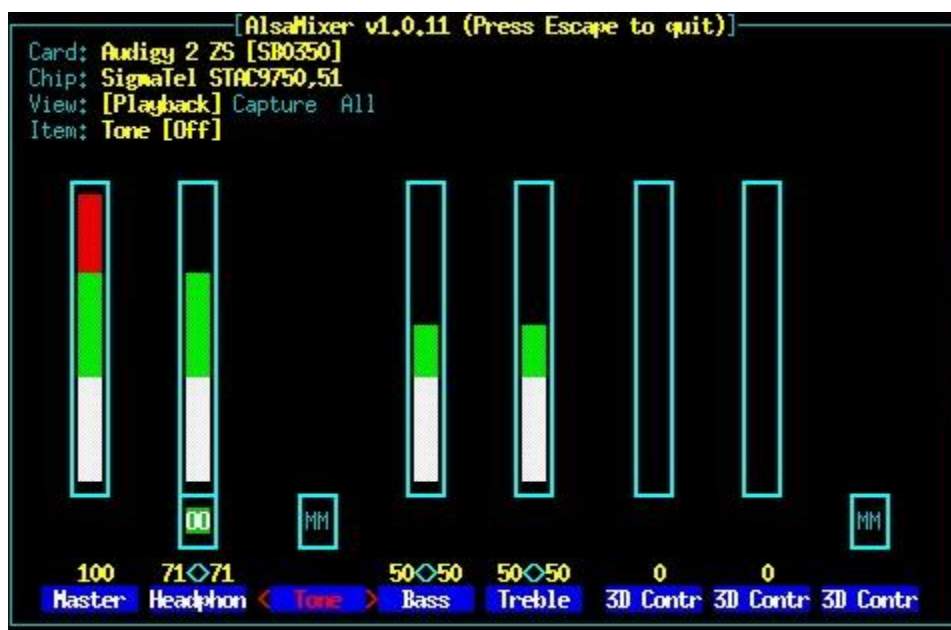
make – wykonanie polecenia trwa dość długo

```
sudo make install
```



Rys. 5.3.4.1. Odbiór obrazów w QSSTV

Podobnie jak w poprzednich programach należy w konfiguracji oprócz danych stacji i operatora wybrać system dźwiękowy USB. Korzystanie z niego wymaga uruchomienia miksera dźwięku ALSA. Do kłucowania radiostacji można posłużyć się zdalnym sterowaniem CAT albo włączyć automatyczne kłucowanie nadajnika VOX. QSSTV posiada funkcje kalibracji częstotliwości zegarowej systemu dźwiękowego dla skorygowania pochylenia odbieranych obrazów.



Rys. 5.3.4.2. Alsamiksers

5.3.5. Dire Wolf

Dire Wolf jest programowym sterownikiem węzła *packet-radio* – modemem TNC. Program dzieli otrzymany strumień danych na pakiety AX.25, a następnie generuje tony m.cz. odpowiadające wartościom logicznego zera i logicznej jedynki. Tony te modulują następnie nadajnik SSB lub FM. Przy odbiorze następuje dekodowanie tonów, wydzielenie z pakietów AX.25 danych użytkowych i złożenie ich dla otrzymania pierwotnej informacji. Programowy modem TNC może pracować na modelach Pi-3 i Pi-4. Jest on obecnie wykorzystywany przez oprogramowanie APRS: nadawanie informacji pozycyjnych, przekaźnik cyfrowy APRS, bramkę radiowo-internetową itd., a także przez programy terminalowe dla innych systemów łączności j.np. Winlinku. W przypadku korzystania z programów APRS przedstawiających pozycje stacji na mapie (*Xastir* itp.) konieczne jest użycie pamięci SD o większej pojemności, co najmniej 32 GB.

W celu zainstalowania programu należy po kolei :

1. Przejść do głównego katalogu użytkownika
`cd ~`

2. Pobrać kod źródłowy i skompilować program, aby otrzymać najaktualniejszą wersję

```
sudo apt install -y libasound2-dev libudev-dev
```

```
git clone https://www.github.com/wb2osz/direwolf
```

```
cd direwolf
```

```
sudo make install
```

```
make install-conf
```

```
make install-rpi
```

W celu skonfigurowania programu trzeba otworzyć w edytorze *nano* plik *direwolf.conf*

```
sudo nano ~direwolf.conf
```

Należy przejść do sekcji „FIRST AUDIO DEVICE PROPERTIES” i w linii #ADEVICE plughw: 1,0 usunąć początkowy krzyżyk, następnie w sekcji „CHANNEL 0 PROPERTIES” w linii „MYCALL NOCALL” należy wprowadzić dużymi literami własny znak wywoławczy. W tej samej sekcji w linii MODEM należy podać szybkość transmisji 300, 1200 albo 9600.

W sekcji „DIGIPEATER PROPERTIES” w linii #DIGIPEAT 0 0 ... należy też usunąć początkowy krzyżyk.

Operatorzy mieszkający w okolicach, w których nie ma jeszcze bramek internetowo-radiowych APRS mogą skonfigurować *Dire Wolfa* jako serwer bramki iGate i w ten sposób zwiększyć zasięg sieci. Operatorzy zamieszkali w pobliżu działającej bramki raczej nie powinni jej dublować. W zależności od zainteresowania innych stacji można natomiast uruchomić bramkę na innej częstotliwości alternatywnej (np. 432,500 MHz zamiast standardowej 144,800 MHz). Włączenia bramki dokonuje się przez usunięcie

krzyżyka poprzedzającego linię IGSERVER. W linii IGLOGIN należy po usunięciu początkowego krzyżyka podać własny znak wywoławczy i wygenerowane na jednym z wymienionych wcześniej serwerów internetowych hasło dostępu do sieci APRS-IS. Za zakończenie modyfikacji należy je zapisać naciskając kombinację Ctrl-X, następnie „Enter” i odpowiadając „Y” na pytanie przy zamykaniu edytora *nano*.

Przy okazji instalacji programu instalowane są również instrukcje do niego dostępne pod

`/usr/local/share/doc/direwolf`.

Spośród wielu programów klientów APRS najbardziej rozpowszechnione obecnie są YAAC i Xastir. Oba przedstawiają położenie odebranych stacji na mapach.

Instalacja YAAC wymaga

1. Przejścia w przeglądarce internetowej pod adres <http://www.ka2ddo.org/ka2ddoYAAC.html>.

2. Pobrania ze strony aktualnej wersji programu.

3. Zainstalowania wymaganych programów pomocniczych za pomocą polecenia

```
sudo apt install -y openjdk-8-jre librtx-java
```

4. Instalacji YAAC

```
cd ~
```

```
mkdir yaac
```

```
cd yaac
```

```
unzip ../Downloads/YAAC.zip.
```

Przed skonfigurowaniem YAAC należy uruchomić *Dire Wolfa* aby YAAC mógł się z nim połączyć.

Potem należy uruchomić YAAC przez wywołanie:

```
java -Xmx512m -jar YAAC.jar
```

W wywołaniu Javy pierwszy z parametrów rezerwuje 512 MB pamięci heapu aby YAAC mógł tam składać potrzebne mapy. Parametr `-jar` informuje Javę, że ma wykonywać plik o rozszerzeniu `.jar`. Ostatnim parametrem jest nazwa pliku YAAC. Po pierwszym wywołaniu uruchamiany jest asystent konfiguracji pomagający w jej przeprowadzeniu. Wśród danych konfiguracyjnych znajduje się oczywiście własny znak wywoławczy i lokalizacja stacji. W zakładce dodania i konfiguracji złączy („Add and configure interfaces”) należy wybrać złącze programowe AGWPE z domyślnymi parametrami. W przypadku braku dostępu do Internetu YAAC korzysta z map zapisanych lokalnie.

5.4. Odbiorniki programowalne

W warunkach amatorskich korzystnym rozwiązaniem odbiornika programowalnego jest układ oparty o procesor RTL2832 (RTL-SDR). Odbiorniki te w postaci paluszków wtykanych do złącza USB komputera były pierwotnie przewidziane do komputerowego odbioru telewizji, ale stosunkowo szybko zostały rozszyfrowane przez krótkofalowców i po opracowaniu odpowiednich sterowników są używane do odbioru różnych stacji, w tym stacji amatorskich w zakresie od 25 MHz wzwyż. Oferowane są również odbiorniki posiadające dodatkowy konwerter częstotliwości pozwalający na odbiór fal krótkich. Dla uzyskania dobrego i niezakłóconego odbioru odbiorniki powinny być umieszczone możliwie jak najdalej od komputera i możliwie jak najbliżej anteny. Dla uniknięcia długich i nieporęcznych połączeń kablowych można odbiornik razem z mikrokomputerem służącym jako serwer odbiorczy umieścić w miejscu zapewniającym dogodne warunki odbioru, a połączenie serwera z komputerem odbiorczym zrealizować poprzez lokalną sieć WiFi. Na komputerze odbiorczym należy zainstalować program SDR# i wybrać w nim odbiornik „RTL-SDR (TCP)”.

Instalacja serwera przebiega jak następuje:

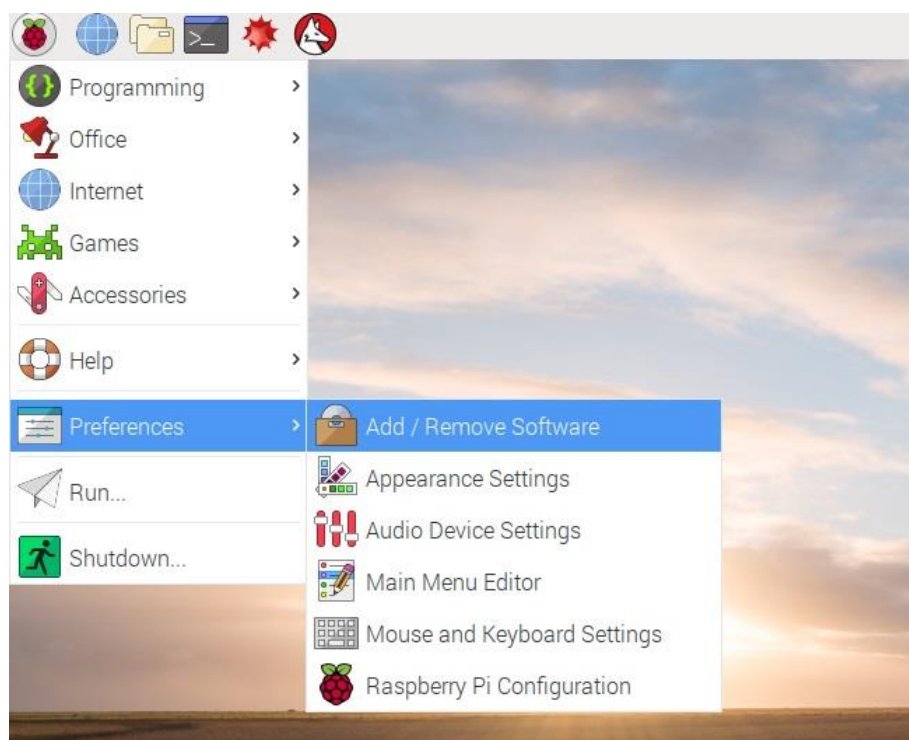
1. W menu *Maliny* wybrać pozycję „Preferences” („Ustawienia”) – „Add/Remove software” („Dodaj/Usuń programy”).

2. W polu poszukiwania programu wpisać *rtl-sdr* i nacisnąć przycisk „Enter”.

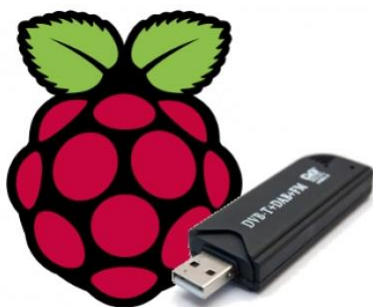
3. Po krótkim czasie wyświetla się lista znalezionych programów, w której należy znaleźć właściwy program (dla RTL2832U).

4. Po wybraniu programu nacisnąć na symbol po lewej stronie w celu potwierdzenia instalacji i na przycisk „Apply” („Zastosuj”) dla jej przeprowadzenia. Program instalacyjny pyta użytkownika o jego hasło dostępu.

5. Po zakończeniu instalacji należy jeszcze raz nacisnąć na przycisk „Apply” („Zastosuj”) w celu zamknięcia okna. Po naciśnięciu „OK” okno także się zamyka.



Rys. 5.4.1. Instalacja programów w Raspbianie



Po ponownym uruchomieniu *Maliny* należy do gniazdka USB włożyć odbiornik i w oknie wiersza poleceń podać polecenie `rtl_test -t` w celu sprawdzenia prawidłowego działania. Test trwa kilka minut, a więc należy poczekać do jego zakończenia i wyświetlenia informacji na ekranie. Jeżeli system nie znalazł odbiornika należy wyłączyć i ponownie uruchomić *Malinę*. Gdyby po kilku próbach znalezienie nie było możliwe może to oznaczać, że używany odbiornik jest niekompatybilny z typowymi modelami.

Po prawidłowym zakończeniu testu należy uruchomić serwer za pomocą polecenia `rtl_tcp -a <adres IP>`. Adres IP *Maliny* można

znaleźć podając w oknie wiersza poleceń rozkaz `hostname -I`.

Jak już podano powyżej w ustawieniach programu SDR# należy wybrać odbiornik dostępny przez lokalną sieć, a mianowicie „RTL-SDR (TCP)”, po czym w polu połączenia wprowadza się adres IP serwera. Po podłączeniu anteny do odbiornika można dostroić go do pożądanego pasma i sprawdzić jego pracę.

W celu uruchomienia serwera po każdym starcie *Maliny* można przygotować skrypt autostartu jak to podano w przykładzie powyżej albo uruchomić na niej serwer VNC i korzystać z dostępu przez VNC z komputera odbiorczego w celu podania potrzebnego polecenia.

Program odbiorczy GQRX współpracuje z wieloma typami odbiorników programowalnych, m.in.: Funcube Dongle Pro+, RTL-SDR, Airspy, Airspy HF+, Lime SDR, Pluto SDR, SDRplay i innymi dostarczającymi kwadraturowych strumieni danych. Program wymaga znacznej mocy przetwarzania i dlatego można korzystać z niego tylko na modelach Pi-3 i Pi-4.

1. Instalacja wymaga podania w oknie terminalowym poleceń:

```
sudo apt install -y gnuradio libvolk1-bin libusb-1.0-0 gr-iqbal
```

```
sudo apt install -y qt5-default libqt5svg5 libportaudio2
```

2. Otwarcia w przeglądarce strony

<http://gqrx.dk/download>

i pobrania z niej archiwum programu.

3. Utworzenia katalogu dla plików rozpakowanych z archiwum i przejścia do niego

`mkdir ~/gqrx` – symbol ~ oznacza katalog główny użytkownika czyli `/home/pi`

`cd ~`

4. Rozpakowania archiwum (nazwy mogą się różnić w zależności od aktualnej wersji programu)

`sudo tar -xvf ~/Downloads/gqrx-sdr-2.11.5-linux-rpi3.tar.xz --strip-components=1`

5. Zainstalowania *rules.d* za pomocą polecenia

`sudo cp ~/gqrx/udev/*.rules /etc/udev/rules.d`

6. Wybrania właściwego profilu dla cyfrowej obróbki danych

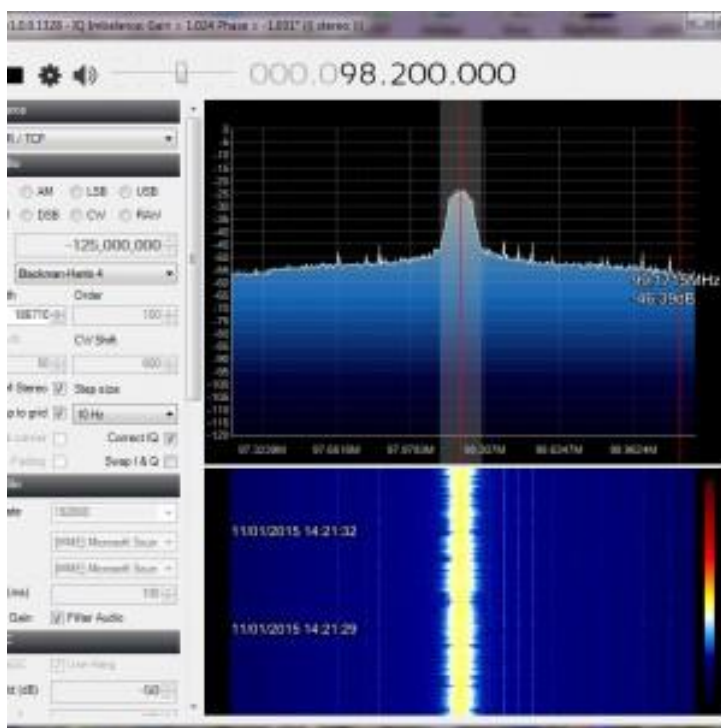
`sudo volk_profile`

Przed wywołaniem Gqrx należy podłączyć odbiornik i podsystem dźwiękowy do gniazd USB *Maliny*.

Obciążenie procesora zależy od szerokości pasma wyświetlanego w oknie.

Programem sprawdzonym i cieszącym się powodzeniem jest *Linrad*. Program nie jest dostępny w postaci skompilowanej i wymaga instalacji z plików źródłowych. Pliki są dostępne w witrynie <http://sm5bsz.com/linuxdsp/linrad.htm>.

Po uruchomieniu odbiornika na RTL-SDR można zainstalować dodatkowo program *PiAware* dekodujący lotnicze meldunki systemu ADS-B nadawane na częstotliwości 1090 MHz. Jest on dostępny w Internecie pod adresem <https://piaware.flightcdn.com/piaware-sd-card-5.0.img.zip>.



Rys. 5.4.2. Program odbiorczy SDRsharp

5.4.1. Odbiorcza bramka APRS

Bramka składa się z odbiornika RTL-SDR podłączonego do *Maliny*.

1. Przed zainstalowaniem programu można w miarę potrzeby zaktualizować Raspbiana:

`sudo apt-get update`

`sudo apt-get upgrade`

`sudo apt-get dist-upgrade`

`sudo rpi-update`

`sudo reboot`

2. Następnie należy usunąć oprogramowanie *Puls Audio*

`sudo apt-get remove --purge pulseaudio`

`sudo apt-get autoremove`

`rm -rf /home/pi/.asoundrc /home/pi/.pulse`

`sudo reboot`

3. Kolejnym krokiem jest modyfikacja pliku `/etc/modprobe.d/raspi-blacklist.conf`

`sudo nano /etc/modprobe.d/raspi-blacklist.conf`

w którym należy dodać podane poniżej linie zapewniające, że system nie pobierze standardowych sterowników odbiornika

```
blacklist dvb_usb_rtl28xxu
```

```
blacklist rtl_2832
```

```
blacklist rtl_2830
```

4. Następnie należy ponownie wystartować *Malinę*

```
sudo reboot
```

5. Założyć nowy katalog dla oprogramowania odbiornika

```
mkdir ~/rtl
```

6. W katalogu tym trzeba zainstalować oprogramowanie

```
cd ~/rtl
```

```
sudo apt-get install git build-essential cmake libusb-1.0-0-dev
```

```
sudo git clone git://git.osmocom.org/rtl-sdr.git
```

```
cd rtl-sdr
```

```
mkdir build
```

```
cd build
```

```
cmake .. -DINSTALL_UDEV_RULES=ON
```

```
make
```

```
sudo make install
```

```
sudo ldconfig
```

7. Kolejnym krokiem jest instalacja programu Sox ułatwiającego uruchamianie i diagnozę odbiornika

```
sudo apt-get install sox.
```

Po zainstalowaniu programu można sprawdzić prawidłowe działanie odbiornika podając polecenie *rtl_test*. W trakcie diagnozy użytkownik powinien otrzymać same meldunki OK.

Następnie można sprawdzić prawidłowość odbioru za pomocą poniższych lub podobnych poleceń (należy zmodyfikować je podając właściwe częstotliwości stacji):

```
rtl_fm -M wbfm -f 93.1M | play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```

```
rtl_fm -M fm -f 144.800M | play -r 24k -t raw -e s -b 16 -c 1 -V1 -
```

```
rtl_fm -M fm -f 162.500M | play -r 24k -t raw -e s -b 16 -c 1 -V1 -
```

Powodują one kolejno odbiór stacji UKF (w przykładzie na częstotliwości 93,1 MHz), odbiór APRS na częstotliwości 144,800 MHz i odbiór satelitów meteorologicznych NOAA na częstotliwości 162,5 MHz. Oczywiście można podać mniejszą lub większą liczbę podobnych poleceń lub opuszczyć np. odbiór satelity przy braku odpowiedniej anteny. Za każdym razem program powinien zapewnić prawidłowy odbiór i dźwięk słyszalny w głośniku.

8. Następnie należy zainstalować dekodery *multimonNG* przy użyciu poleceń

```
cd ~/rtl
```

```
sudo apt-get install qt4-qmake libpulse-dev libx11-dev
```

```
git clone https://github.com/EliasOenal/multimonNG.git
```

```
cd multimonNG
```

```
mkdir build
```

```
cd build
```

```
qmake ../multimon-ng.pro
```

```
make
```

```
sudo make install
```

9. Potem przychodzi kolej na zainstalowanie oprogramowania bramki radiowo-internetowej

```
cd ~/rtl
```

```
sudo apt-get install python-pkg-resources
```

```
git clone https://github.com/asdil12/pymultimonaprs.git
```

```
cd pymultimonaprs
```

i skompilować je

```
sudo python setup.py build
```

```
sudo python setup.py install
```

10. Dla otrzymania dostępu do sieci APRS-IS można wygenerować kod (hasło) dostępu w jednej z podanych uprzednio witryn albo lokalnie korzystając z następujących poleceń (wpisując w nich własny znak wywoławczy):

```
cd ~/rtl/pymultimonaprs
```

```
./keygen.py yourcallsignKey for yourcallsign: xxxxxx
```

Na wszelki wypadek warto też zrobić kopię bezpieczeństwa pliku `/etc/pymultimonaprs.json` i wprowadzić polecenie odbioru APRS:

```
sudo cp /etc/pymultimonaprs.json /etc/pymultimonaprs.old
```

```
sudo nano /etc/pymultimonaprs.json
```

```
rtl_fm -f 144.800M -s 22050 -p 2 -g 42.0 - | multimon-ng -a AFSK1200 -A -t raw -
```

Gdzie `-p` oznacza poprawkę zmierzoną w trakcie kalibracji (o czym dalej), a `-g` wzmocnienie toru odbiorczego. Należy je dobrać w razie potrzeby.

11. Do sprawdzenia pracy programu bramki służy polecenie

```
sudo pymultimonaprs
```

12. Dla automatycznego uruchamiania bramki można utworzyć skrypt wywoławczy

```
sudo nano /etc/init.d/pymultimonaprs
```

o następującej treści

```
#!/bin/sh
```

```
### BEGIN INIT INFO
```

```
# Provides: pymultimonaprs
```

```
# Required-Start: $all
```

```
# Required-Stop: $syslog
```

```
# Default-Start: 2 3 4 5
```

```
# Default-Stop: 0 1 6
```

```
# Short-Description: start/stop of pymultimonaprs
```

```
### END INIT INFO
```

```
case "$1" in
```

```
start)
```

```
    sudo pymultimonaprs --syslog &
```

```
;;
```

```
stop)
```

```
    sudo killall pymultimonaprs
```

```
;;
```

```
*)
```

```
    echo "Usage: /etc/init.d/pymultimonaprs {start|stop}"
```

```
    exit 1
```

```
;;
```

```
esac
```

```
exit 0
```

Po zapisaniu skryptu należy umożliwić jego wywołanie

```
sudo chmod +x /etc/init.d/pymultimonaprs
```

```
sudo update-rc.d pymultimonaprs defaults
```

i uruchomić

```
sudo /etc/init.d/pymultimonaprs start
```

13. Kalibracja częstotliwości odbiornika RTL-SDR wymaga zainstalowania specjalnego programu kalibracyjnego

```
cd ~/rtl
```

```
sudo apt-get install libtool autoconf automake libfftw3-dev
```

```
git clone https://github.com/asdil12/kalibrate-rtl.git
```

```
cd kalibrate-rtl
```

```
git checkout arm_memory
```

```
./bootstrap
```

```
./configure
```

```
make
```

```
sudo make install
```


14. W podanym w Internecie przykładzie jako wzorzec do kalibracji wybrano stację w paśmie 900 MHz, a konkretnie wykrytą przez program stację bazową GSM pracującą w kanale 7. Przykład ten należy dostosować do lokalnej sytuacji.

`kal -s GSM900`

Jego częstotliwość zostaje następnie wykorzystana do obliczenia poprawki.

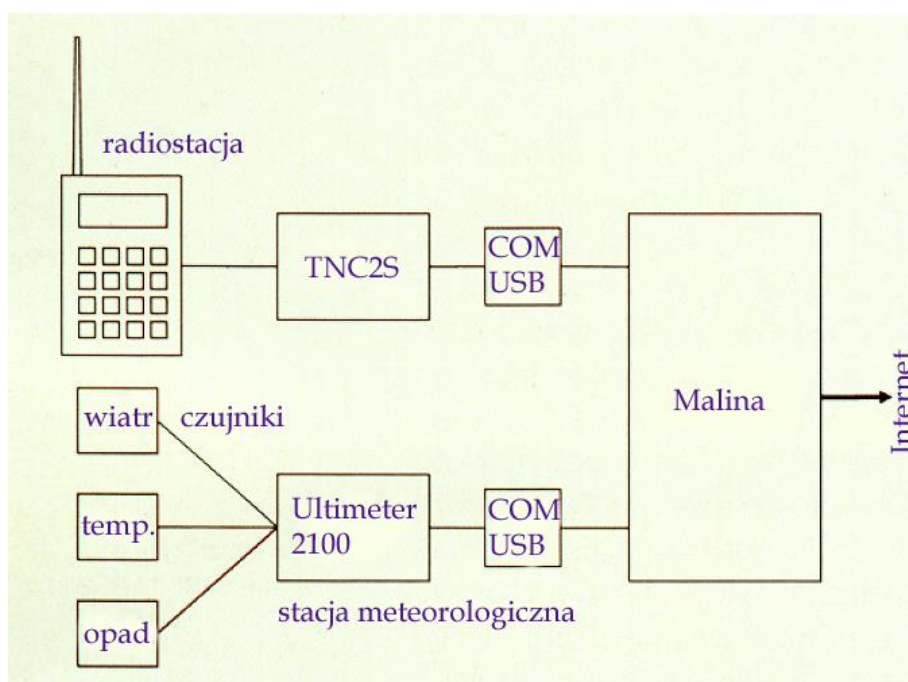
`kal -c 7 -v`

W cytowanym przykładzie obliczona została poprawka 2,306, skąd po zaokrągleniu wynikała wartość 2 podana wyżej w przykładzie dostrojenia do częstotliwości APRS.

5.5. APRS

5.5.1. Bramka odbiorcza i stacja meteorologiczna

Opisana z numerze 5/2017 CQDL bramka odbiorcza APRS iGate na *Malinie* pozwala na podłączenie do niej również stacji meteorologicznej Ultimeter 2100.



Rys. 5.5.1.1. Schemat blokowy stacji

Bramka jest wyposażona w oprogramowanie APRSX autorstwa OH2MQK. Oprogramowanie to pozwala na nadawanie różnych tekstów radiolatarni i na odczyt przygotowanych tekstów z pliku. Autorzy DL2GKM i DG8AL użyli wprawdzie stacji meteorologicznej Ultimeter 2100, ale po odpowiedniej modyfikacji kodu można wykorzystać stacje dowolnych typów. Uproszczony schemat blokowy pokazano na ilustracji 5.5.1.1. Ze względu na to, że zarówno stacja meteorologiczna jak i modem TNC2 są wyposażone w złącza szeregowo COM konieczne było zastosowanie przejściówek COM-USB. TNC2 pracuje w tym rozwiązaniu w trybie KISS.

Standardowo numeracja i oznaczenia złączy USB zależą od kolejności ich inicjalizacji, ale można im przypisać stałe oznaczenia modyfikując odpowiednio plik *30-usb-rules*. Fragment pliku przedstawiono poniżej (oznaczenia ATTRS mogą być różne od podanych w zależności od producenta przejściówki):

```

SUBSYSTEM=="tty",
ATTRS (idVendor)=="0403",
ATTRS (idProduct)=="6001",
SYMLINK+="ttyUSB_tnc2s"
SUBSYSTEM=="tty",
ATTRS (idVendor)=="067b",
ATTRS (idProduct)=="2303",
  
```

`SYMLINK+= "ttyUSB_u2001"`

Numer w nazwie pliku został tak dobrany aby miał on najwyższy priorytet odczytu.

Ustalono fabrycznie numery przejściówek można odczytać za pomocą polecenia

`dmesg | grep usb`

Złącze dla TNC2 otrzymało w pliku nazwę `/dev/ttyUSB_tnc2s`, a złącze dla stacji meteorologicznej `/dev/ttyUSB_u2100`.

Konieczne jest też uzupełnienie pliku `/etc/aprx.conf`:

```
<beacon>
  beaconmode aprsis
  cycle-size 10m
  beacon symbol „R&“ $myloc
  comment „RX iGate OE1KDA Inzersdorf“
</beacon>
<beacon>
  beaconmode aprsis
  cycle-size 5m
  beacon srccall OE1KDA-6 file /tmp/wxbeacon.txt
</beacon>
```

Podany fragment pliku zawiera dwie sekcje rozpoczynające się od oznaczenia `<beacon>`. W pierwszej z nich znajduje się konfiguracja radiolatarni bramki iGate, a w drugiej konfiguracja tekstu radiolatarni meteorologicznej. Druga z sekcji zawiera wskazanie pliku informacji meteorologicznych. Odstęp między transmisjami danych meteorologicznych wynosi w tym przykładzie 5 minut.

Dane odbierane ze stacji meteorologicznej są po sformatowaniu zapisywane w pliku `/tmp/wxbeacon.txt`.

Plik napisany jest w języku Python:

```
import serial
import time

u2001 = serial.Serial("/dev/ttyUSB_u2001", 2400, timeout=1)
count = 2
while True:
    record = u2001.readline()
    record = record[2:]
    packet = "!4821.04N/01608.01E"
    if len(record) > 0:
        count = count - 1

#Wiatr
try:
    value = int((float(int(record[44:48], 16))+0.5)/10)
    WindSpeed = "%03d" % value
except:
    WindSpeed = "/... "

#Kierunek wiatru
try:
    value = (int(float(int((record[4:8], 16))*360/255+0.5)
    WindDir = "%03d" % value
except:
    WindDir = "_... "

#Temperatura
try:
    value = int((float(int(record[8:12], 16))+0.5)/10)
    Temperature = "t%03d" % value
except:
    Temperature = "t... "
```

```

#Opad 1h
try:
    value = int(float(int(record[12:18], 16))+0.5)
    Rain24h = "p%03d" % value
except:
    Rain24h = "p..."

#Barometr
try:
    value = int(float(int(recordp16:20], 16))+0.5)
    Barometer = "b%05d" % value
except:
    Barometer = "b....."

#Wilgotność
try:
    value = int((float(int(record[24:28], 16))+0.5) % value)
    Humidity = "h%02d" % value
except:
    Humidity = "h.."

#Całkowity opad
try:
    value = int(-float(int(record[40:44], 16))+0.5)
    RainTotal = "P%03d" % value
except:
    RainTotal = "P..."

#Porywy
try:
    value = int((-float(int(record[0:4], 16))+0.5)/10)
    Gust = "g%03d" % value
except:
    Gust = "g..."

#Opad 24h
Rain1h = "r..."

#Złożenie pakietu
packet += WindDir
packet += WindSpeed
packet += Gust
packet += Temperature
packet += Rain1h
packet += Rain24h
packet += RainTotal
packet += Humidity
packet += Barometer
packet += "Próba Ultimetru z Rpi 3B.\n"
if count < 1:
    f = open("/tmp/wxbeacon.txt", "w")
    f.write(packet);
    f.close
    print packet

```

count = 60

Po otwarciu złącza USB program w nieskończonej pętli odpytuje dane ze stacji meteorologicznej, formatuje je i co pięć minut zapisuje w pliku. Stacja meteorologiczna pracuje w trybie gromadzenia danych (ang. *Data Logger Mode*) i w rytmie sekundowym wysyła pakiet danych o długości 48 bajtów. Program otwiera złącze szeregowo */dev/ttyUSB_u2100* z szybkością transmisji 2400 bit/s i ograniczeniem czasu oczekiwania do 2 sekund. Zapobiega to zawieszeniu się programu w przypadku niedotarcia danych. Dane po wczytaniu są pobierane z odpowiednich części komunikatu i przetwarzane z tekstu na postać liczbową przy użyciu funkcji *int("...", 16)*. Potem następują przeliczenia zależne od rodzaju wielkości i dane są składane w komunikacie odpowiadającym standardowi APRS. Przypadek braku danych jest przechwytywany z użyciem polecenia *try* i zamiast danych wpisywane są wielokropki "...". Na koniec do komunikatu dodawane są współrzędne i reszta danych APRS. Dane te są wpisywane do pliku */tmp/wxbeacon.txt*, z którego są odczytywane przez APRX.

Aby uniknąć nadmiernego zużycia pamięci SD przez częsty wielokrotny zapis lepiej jest plik ten założyć na dysku w pamięci RAM. Dysk ten jest zakładany za każdym razem po uruchomieniu *Maliny* za pomocą następujących poleceń:

```
mkdir /tmp/ramdisk
```

```
mount -t tmpfs -o size=50k none /tmp/ramdisk
```

```
chmod 777
```

Polecenia te są dopisane do pliku *start-aprx* przed linią *echo "... starte wxbeacon"*

Ścieżkę dostępu do pliku *wxbeacon.txt* należy odpowiednio zmodyfikować zastępując podawaną wyżej */tmp/wxbeacon.txt*.

Dla zapewnienia współpracy APRX z przytoczonym programem w Pythonie zbierającym dane meteorologiczne konieczne jest uzupełnienie pliku startowego APRX */usr/local/bin/start-aprx*.

```
#!/bin/bash
```

```
#
```

```
killall aprx
```

```
rm /var/run/aprx.pid>/dev/null
```

```
#
```

```
echo „Uruchomienie TNC 2S w trybie KISS“
```

```
#
```

```
# Ustawienie szybkości transmisji
```

```
#
```

```
stty -F /dev/ttyUSB_tnc2s speed 19200
```

```
sleep 5
```

```
#
```

```
# Przełączenie TNC w tryb KISS
```

```
#
```

```
echo -ne „\r\033IDL2GKM\r” > /dev/ttyUSB0
```

```
echo -ne „\r\033s0\r” > /dev/ttyUSB0
```

```
echo -ne „\r\033mu\r” > /dev/ttyUSB0
```

```
echo „...TNC w trybie KISS!”
```

```
sleep 5
```

```
#
```

```
# Połączenie z modemem i uruchomienie APRX
```

```
#
```

```
kissattach /dev/ttyUSB_tnc2s packet
```

```
kissparms -p packet -t 700 -s 200 -r 32 -l 100 -f n
```

```
sleep 1
```

```
#
```

```
echo „Konfiguracja złącza do stacji meteo”
```

```
#
```

```
stty -F /dev/ttyUSB_u2001 speed 2400
```

```
echo „  starte wxbeacon „
```

```
# start programu w tle
```

```
python /usr/local/bin/wxbeacon.py&
```

```
sleep 5
```

```
#
echo „...Modem połączony. StartAPRX”
#
aprx
# odbiór i wyświetlanie pakietów
axlisten
#
exit 0
#
```

Skrypt startowy jest wywoływany w pliku */etc/rc-local*. Nadawane komunikaty można obserwować w Internecie w witrynie *aprs.fi*.

Możliwe jest także zmodyfikowanie programu w Pythonie tak, aby czytywał czujniki podłączone bezpośrednio do listwy GPIO.

5.5.2. Instalacja APRX

Program APRX wykonuje trzy powiązane ze sobą zadania: pracę jako bramka radiowo-internetowa APRS (iGate), jako przekaźnik cyfrowy (ang. *digipeater*) i jako bramka D-PRS – APRS.

Do zainstalowania APRX na Malinie służy polecenie

```
sudo apt-get install aprx -y
```

Konfigurację programu zawiera plik */etc/aprx.conf*. Do najważniejszych zawartych w nim danych konfiguracyjnych należą:

```
mycall OE1KDA-11
```

```
myloc lat 0000.00 N lon 00000.00 E
```

```
<aprsis>
```

```
login $mycall
```

```
# server.euro.aprs2.net
```

```
server.rotate.aprs2.net
```

```
passcode *****
```

```
# heartbeat-timeout 1m      # Puls co minutę, można także usunąć znak komentarza i jako parametr
podać 0 dla wyłączenia
```

```
# filter „ „
```

```
</aprsis>
```

```
<logging>
```

```
pid /var/run/aprs.pid
```

```
rflog /var/log/aprsx/aprx-rf.log
```

```
aprsxlog /var/log/aprx/aprx.log
```

```
erlangfile /var/run/aprx.state
```

```
erlang-loglevel LOGDAEMON
```

```
erlanglog /var/log/erlang.log
```

```
erlang-log1min
```

```
</logging>
```

```
<interface>
```

```
serial-device /dev/ttyAMA0 19200 8n1 KISS      # Dla TNC
```

```
# alias WIDE1-1
```

```
callsign $mycall
```

```
tx-ok false      # Dla bramki czysto odbiorczej, dla nadawczo-odbiorczej true
```

```
</interface>
```

```
<beacon>
```

```
beaconmode APRSIS
```

```
cycle-size 60m
```

```
beacon symbol „I&“ $myloc comment „RX iGate Raspberry Pi“
```

```
</beacon>
```


Sekcja <aprsis> dotyczy połączenia z siecią APRS-IS. Kod dostępu *passcode* jest generowany m.in. w podanych uprzednio witrynach internetowych. Zamiast aktywnego w tym przykładzie połączenia z serwerem *rotate.aprs2.net* można łączyć się z serwerem europejskim, ale zalecane jest korzystanie z serwera wybierającego połączenie rotacyjnie. Polecenie *heartbeat* nie jest obowiązkowe. Możliwe, ale nie obowiązkowe jest także dodanie filtra komunikatów.

W sekcji <logging> określone są miejsca składowania dzienników pracy APRX i plik PID. Plik *aprx.log* zawiera meldunki dotyczące pracy programu, a plik *aprx-rf.log* komunikaty odebrane i retransmitowane.

Sekcja <interface> definiuje parametry połączenia z TNC (zewnętrznym albo na płycie rozszerzeń HAT) lub modemem dźwiękowym. Dla modemu dźwiękowego zamiast podania parametrów łącza szeregowego może być to polecenie *ax25-device \$mycall* itp. Możliwe jest też korzystanie z kilku złączy pod różnymi znakami wywoławczymi. Dla każdego z nich można oddzielnie włączyć transmisję pakietów albo nie.

Sekcja <beacon> zawiera definicje tekstów i odstępów między transmisjami dla radiolatarni.

Dla TNC na płycie rozszerzeń *Maliny* należy:

1. W pliku */boot/cmdline.txt* usunąć linię
`console=ttyAMA0, 115200 kgdboc=ttyAMA0, 115200`
 jeśli plik i linia istnieją.
 2. W pliku */boot/config.txt* dodać linię
`enable_uart=1,`
 a dla Pi3 także linie
`dtoverlay=pi3-miniuart-bt`
`core_freq=250`
 3. Do pliku */lib/systemd/system/hciattach.service* dodać linie konfigurujące UART służący do komunikacji z TNC
- ```
[Unit]
ConditionPathIsDirectory=/proc/device-tree/soc/gpio@7e200000/bt_pins
Before=bluetooth.service
After=dev-ttyS0.device
[Service]
Type=forking
ExecStart=/usr/bin/hciattach /dev/ttyS0 bcm43xx 921600 noflow -
[Install]
WantedBy=multi-user.target
```

Dla modemu dźwiękowego korzystającego z systemu dźwiękowego włączonego do gniazda USB należy:

1. Usunąć *pulseaudio*  
`sudo apt-get remove pulseaudio`
2. Zainstalować modem dźwiękowy i obsługę protokołu ax25  
`sudo apt-get install soundmodem`  
`sudo apt-get install ax25-tools ax25-apps`
3. Skonfigurować obsługę ax25 i sterownika *soundmodem* podając w pliku */etc/ax25/axports*  

```
#name callsign speed paclen window description
APRS OE1KDA-11 1200 255 2 APRS-iGate
```

 W pliku */etc/ax25/soundmodem.com*:  

```
<?xml version="1.0"?>
<modem>
 <configuration name="APRS">
 <chaccess txdelay="150" slottime="100" ppersist="40" fulldup="0" txtail="10"/>
 <audio type="alsa" device="plughw:0,0" halfdup="1" capturechannelmode="Mono"/>
 <ptt file="none" gpio="0" hamlib_model="" hamlib_params=""/>
 <channel name="Channel 0">
```

```

<mod mode="afsk" bps="1200" f0="1200" f1="2200" diffenc="1"/>
<demod mode="afsk" bps="1200" f0="1200" f1="2200" diffdec="1"/>
<pkt mode="MKISS" ifname="sm0" hwaddr="DG3OBU-10" ip="10.6.0.1"
netmask="255.255.255.0" broadcast="10.6.0.255"/>
</channel>
</configuration>
</modem>

```

4. Po skonfigurowaniu *ax25* i *soundmodemu* należy ustawić siłę głosu. Do tego celu można wykorzystać program *alsamixer*. Dla uniknięcia przesterowania wejścia należy wywołać oscyloskop programu („Scope”) i ustawić występowanie na około 70%. Zasadniczo konieczne jest wyłączenie wzmacniacza wstępnego na wejściu mikrofonowym, ale zależy to od konkretnego użytego modelu systemu dźwiękowego USB.

Dokonane ustawienia zapamiętuje się za pomocą polecenia

```
sudo alsactl store.
```

Automatyczny start programu uzyskuje się dopisując do pliku */etc/default/aprx* linię *STARTAPRX="yes"*

lub odpowiednio modyfikując jeśli było w niej wpisane „no”.

Wywołanie sterownika *soundmodem* można po prostu dopisać do liku */etc/rc.local*  
*/usr/sbin/soundmodem /etc/ax25/soundmodem.conf -R -M >/dev/null 2>/dev/null &*  
*sleep 1*

W bramce może pracować radiostacja FM dowolnego typu na pasmo 2 m albo w przypadku pracy na częstotliwości alternatywnej – na pasmo 70 cm. Można także skonstruować prostą radiostację opartą na module *DRA818V* lub *DRA818U*. Konstrukcje takie są opisane m.in. w tomie 57 „Biblioteki”. W praktyce rozwiązania korzystające z TNC spisują się lepiej i niezawodniej od opartych na *soundmodemie*, ale jest to w jakimś stopniu zależne od modelu mikrokomputera i wersji oprogramowania.

Oprócz APRX można w bramkach APRS zainstalować *Xastir*.

## 5.6. Cyfrowy przekaźnik APRS dla łączności kryzysowych

Cyfrowa stacja przekaźnikowa (ang. *digipeater*) konstrukcji *SQ9MDD* (<http://hamspirit.pl/SQ9MDD/>) jest przewoźnym elementem infrastruktury sieci APRS. Została ona opracowana z myślą o instalowaniu w razie potrzeby na terenach objętych klęską żywiołową jako wsparcie urządzeń przenośnych i przewoźnych. Dzięki możliwości połączenia się z urządzeniem za pomocą sieci WiFi możliwa jest równoległa praca operatora z wykorzystaniem programów *APRSI32* lub *APRS Droida*.

Urządzenie zostało wbudowane do hermetycznej i łatwej do transportu skrzynki wyposażonej w gniazda antenowe i zasilania. Składa się ono z zasilacza 12 V/15 A, *Maliny* Pi3B+, złącza separująco-sterującego, wyświetlacza, systemu dźwiękowego ze złączem USB oraz gniazd i wyłączników.

Pracą przekaźnika steruje program *APRX*, a zamiast sprzętowego modemu TNC użyto modemu programowego *Dire Wolf* korzystającego z podsystemu dźwiękowego USB. Do kluczowania nadawanie-odbior wykorzystano wyprowadzenie na listwie GPIO mikrokomputera. Skrypty konfiguracyjne przeмиennik i obsługujące wyświetlacz dotykowy zostały napisane w Pythonie. Programy komunikują się ze sobą dzięki skryptowi *TNC Proxy*. Skrypty te są dostępne na Githubie.

Skrypt *TNC Proxy* znajduje się pod adresem

<https://github.com/sq5t/tnc-proxy>

Biblioteka *AX25*

<https://github.com/sq5t/ax25lib>

*Direwolf*

<https://github.com/wb2osz/direwolf>

Program *APRX*

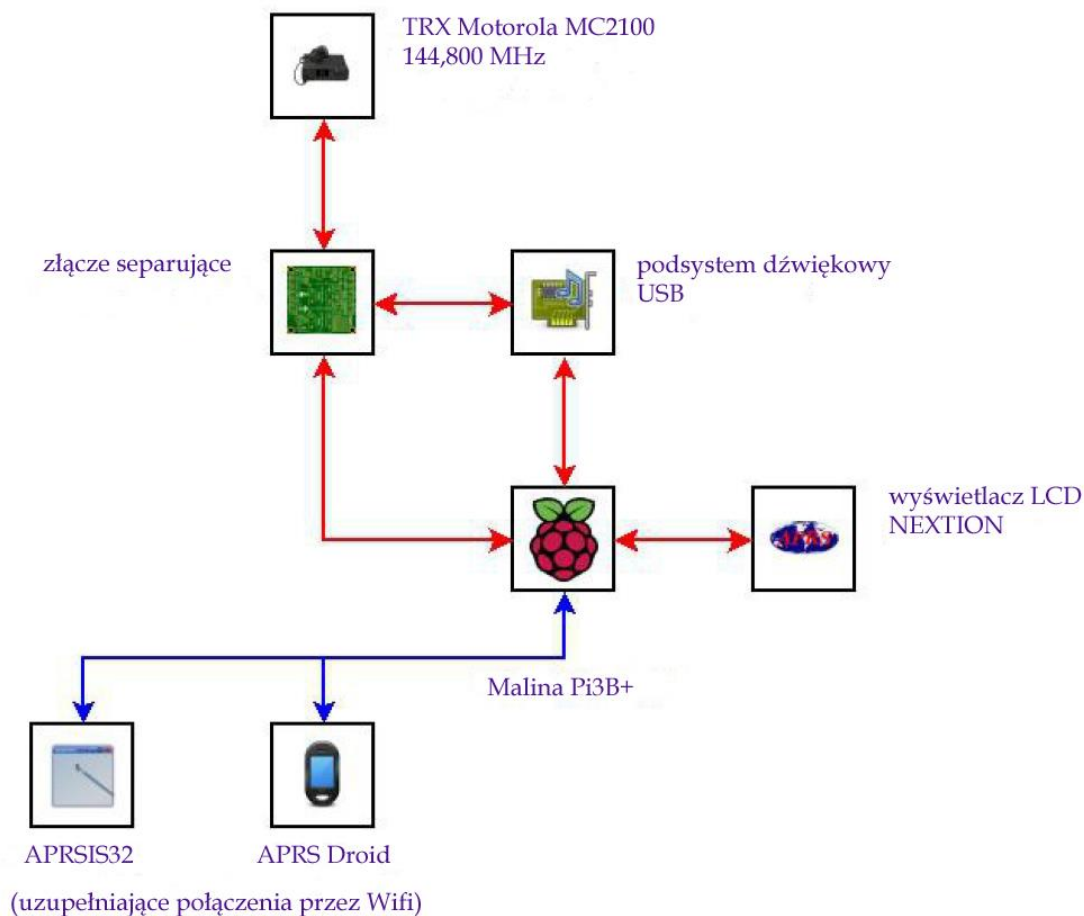
<https://thelifeofkenneth.com/aprx/>

Opis instalacji i konfiguracji przekaźnika

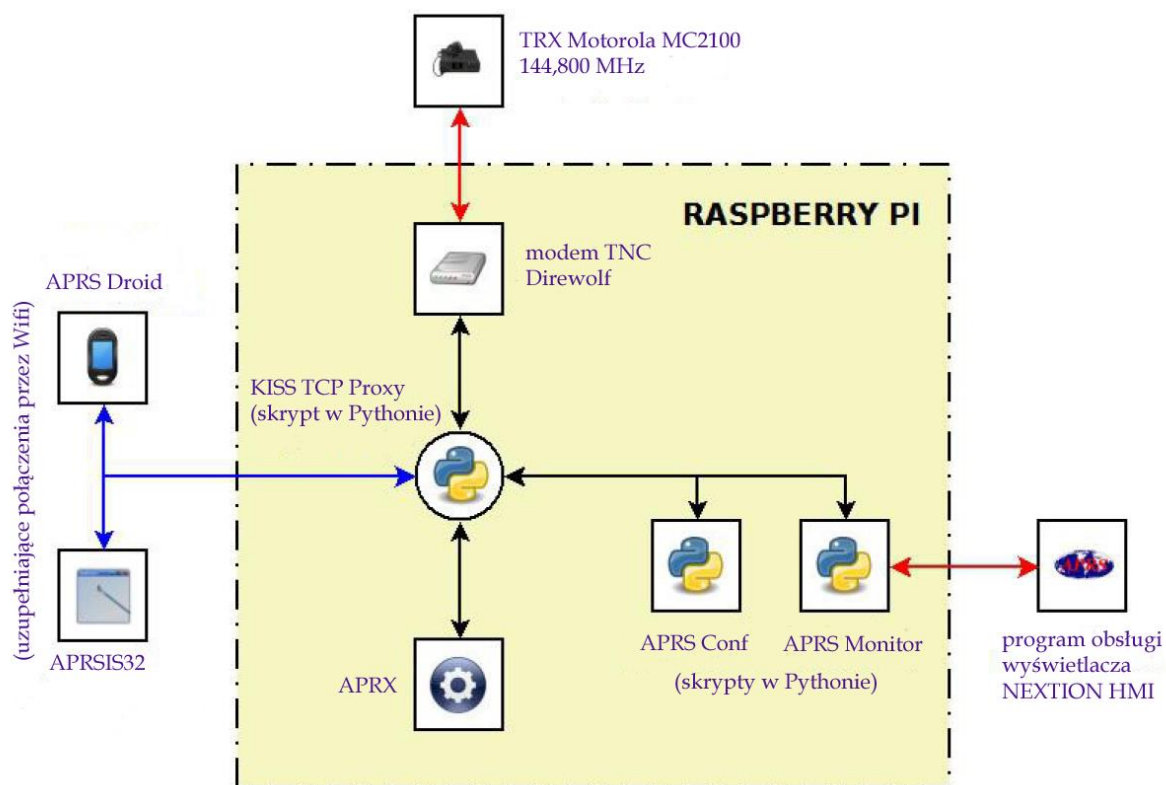
<http://tech4.pl/SQ9MDD/?p=626>

Całość zebrano także w witrynie

<https://github.com/SQ9MDD/EmComm-Off-Grid-DIGI-Project>



Rys. 5.6.1. Schemat blokowy stacji przekaznikowej SQ9MDD



Rys. 5.6.2. Diagram przepływu danych



Fot. 5.6.3. Widok konstrukcji

## 5.7. Echolink

Przełącznik echolinkowy na *Malinie* jest oparty o oprogramowanie *SvxLink* ([www.svxlink.org](http://www.svxlink.org)). Oprogramowanie przełącznika można pobrać z witryny:

<https://github.com/sm0svx/svxlink/wiki/InstallSrcHwRpi>.

Do połączenia fonicznego z radiostacją służy podsystem dźwiękowy USB, a do kluczowania N-O wykorzystywane jest wyjście GPIO17. Sygnał blokady szumów COR (w radiostacjach Motoroli CSQ) jest doprowadzony do wejścia GPIO3. Poziomy sygnałów m.cz. najlepiej jest ustawić w mikserze *alsamikser*.

Instalacja *SvxLinku* składa się z następujących kroków:

1. Aktualizacja systemu (jeśli konieczna) i instalacja bibliotek

```
sudo apt-get update && sudo apt-get upgrade
```

```
sudo apt-get install g++ make libsigc++-1.2-dev libgsm1-dev libpopt-dev tcl8.5-dev libgcrypt-dev
libspeex-dev libasound2-dev alsa-utils libqt4-dev
```

```
sudo apt-get install libsigc++ cmake groff
```

2. Instalacja Echolinku

```
sudo adduser svxlink
```

```
wget https://github.com/sm0svx/svxlink/archive/master.tar.gz
```

```
tar xvf master.tar.gz
```

```
cd svxlink-master/src
```

```
mkdir build && cd build
```

```
sudo cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc
```

```
DLOCAL_STATE_DIR=/var -DUSE_QT=NO ..
```

```
make
```

```
make doc
```

```
sudo make install
```

```
sudo ldconfig
```

3. Skopiowanie katalogu *etc/svxlink* do */etc/svxlink*

```
sudo cp -r etc/svxlink/* /etc/svxlink/
```

4. Wprowadzenie własnego znaku, hasła dostępu i informacji o stacji w konfiguracji

5. Sprawdzenie i ewentualne dopasowanie wejścia blokady szumów i wyjścia N-O na liście GPIO

6. Skonfigurowanie automatycznego startu *SvxLinku*

```
sudo cp opt/svx.sh /opt/svx.sh
```

```
sudo chmod +x /opt/svx.sh
```

7. Dodania w pliku *./config/lxsession/LXDE-pi/autostart* linii

```
@lxterminal -l -e "/opt/svx.sh"
```

przed linią wywołania wygaszacza ekranu.

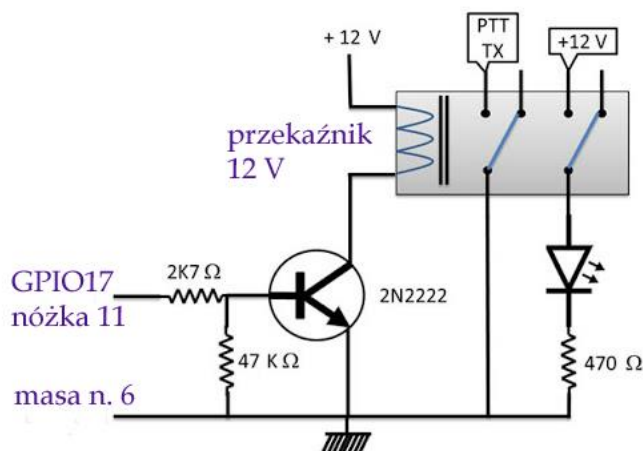
W zestawie jest również skrypt służący do programowego startowania *Maliny* bez konieczności wyłączania zasilania lub wywoływania startu na ekranie. Jego uruchomienie wymaga następujących poleceń:

```
sudo cp opt/reset.py /opt/reset.py
sudo cp lib/systemd/system/reset.service to /lib/systemd/system/reset.service
chmod 644 /lib/systemd/system/reset.service
systemctl enable reset.service
systemctl start reset.service
```

W pliku */etc/rc.local* należy też sprawdzić i ewentualnie dopasować wejście GPIO.

Programowe startowanie komputera może być przydatne w fazie testów nie tylko dla przemiennika echolinkowego ale także przy korzystaniu z dowolnego innego oprogramowania.

Oprogramowanie *SvxLink* wchodzi także w skład zestawu Hamserver używanego na serwerach Hamnetu.



Rys. 5.7.1. Kluczowanie nadajnika. Dla wielu modeli radiostacji wystarcza jedynie tranzystor wykonawczy



Fot. 5.7.2. Radiostacja na module cyfrowej obróbki sygnałów DRA818 (V lub U w zależności od zakresu) dla domowych węzłów Echolinku albo bramek APRS

## 5.8. Odbiór WSPR na RSPduo i WSJT-X

W rozwiązaniu odbiorczym WSPR dla *Maliny*, opisanym w nrze 1/2020 *Funkamateura* użyto odbiornika programowalnego RSPduo (lub innego modelu z tej serii), programu terminalowego WSJT oraz programowego połączenia odbiornika z dekodern WSPR za pośrednictwem programowego wirtualnego kabla. Mikrokomputer Pi3B++ lub Pi4 jest wyposażony w 5-calowy monitor HDMI. Pozwala to na dłuższy nieprzerwany odbiór komunikatów WSPR i śledzenie warunków propagacji bez blokowania

radiostacji i komputera PC potrzebnych do innych zastosowań. Ważną zaletą jest niski pobór prądu w porównaniu z PC i radiostacją krótkofalową, niższa cena i mniejsze zużycie drogiego sprzętu. Rozwiązanie można dostosować innych modeli odbiorników (RTL-SDR itd.) i może ono także służyć do odbioru innych emisji obsługiwanych przez WSJT. Odbiornik RSPduo jest wyposażony w 12-bitowy przetwornik analogowo-cyfrowy i daje lepsze rezultaty aniżeli RTL-SDR mający przetwornik 8-bitowy.

1. Pierwszym krokiem jest instalacja sterownika *2.13 API/HWDRIVER* do RSPduo, dostępnego w witrynie [www.sdrplay.com/downloads](http://www.sdrplay.com/downloads):

Plik *SDRplay\_RSP\_API-Rpi-2.13.1.run* należy zapisać w pamięci, przyznać prawa wykonywania go i wywołać:

```
wget http://www.sdrplay.com/software/SDRplay_RSP_API-RPi-2.13.1.run
chmod 777 SDRplay_RSP_API-RPi-2.13.1.run
./SDRplay_RSP_API-RPi-2.13.1.run
```

Dla RTL-SDR byłoby to polecenie `sudo apt-get install rtl-sdr librtlsdr-dev`.

2. Korzystanie z odbiornika wymaga zainstalowania dodatkowych bibliotek:

```
sudo apt-get install git build-essential automake cmake
sudo apt-get install libpulse-dev libgtk-3-dev freeglut3 freeglut3-dev
sudo apt-get install libhamlib2 libhamlib-utils libhamlib++-dev
sudo apt-get install libhamlib-dev libhamlib-doc python-libhamlib2
```

3. Po poprzednich krokach następuje instalacja *SoapySDR*. Jest to złącze programowe pozwalające na korzystanie przez inne programy z odbiorników RTL-SDR, HackRF i SDRplay. Dzięki niemu program dekodujący uzyskuje dostęp do odbiornika przez standaryzowane złącze programowe API (*Application Programming Interface*) bez konieczności indywidualnego dopasowywania programu.

```
git clone https://github.com/pothosware/SoapySDR.git
```

```
cd soapySDR
```

```
mkdir build && cd build
```

```
cmake ../-DCMAKE_BUILD_TYPE=Release
```

```
make -j4
```

```
sudo make install
```

```
sudo ldconfig
```

```
cd..
```

```
cd..
```

4. Dla SDRduo konieczna jest jeszcze instalacja *SoapySDRPlay*:

```
git clone
```

```
git clone https://github.com/pothosware/SoapySDRPlay.git
```

```
cd soapySDRPlay
```

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make -j4
```

```
sudo make install
```

```
cd..
```

```
cd..
```

Dla odbiornika RTL-SDR należy w pierwszych dwóch liniach zamiast *SoapySDRPlay* wpisać nazwę *SoapyRTLSDR*.

5. W celu sprawdzenia działania odbiornika należy podłączyć go za pomocą kabla do złącza USB i podać polecenie:

```
SoapySDRUtil --info
```

6. W dalszym ciągu konieczna jest jeszcze instalacja GQRX i dekodera WSPR. GQRX jest znanym już od dłuższego czasu programem odbiorczym korzystającym z wielu typów odbiorników i dekodującym różne rodzaje emisji: AM, FM, WFM i SSB. Jego instalacja wymaga zainstalowania programu wraz z bibliotekami za pomocą polecenia:

```
sudo apt-get install gqrx-sdr
```

W razie gdyby znaleziona wersja nie pracowała poprawnie można ją zaktualizować za pomocą poleceń:



```
wget https://github.com/csete/gqrx/releases/download/v2.11.5/gqrx-sdr-2.11.5-linux-rpi3.tar.xz
tar -xvzf gqrx-sdr-2.11.5-linux-rpi3.tar.xz
```

Numer i oznaczenie wersji należy oczywiście zastąpić aktualnym w momencie instalacji.

W programie należy wybrać minimalne możliwe pasmo przenoszone czyli dla RSPduo 250 kHz. Dla przyspieszenia reakcji graficznej powierzchni obsługi można włączyć sterownik *OpenGL* podając polecenie `sudo raspi-config` i włączając sterownik w ustawieniach.

7. Instalacja WSJT-X wymaga pobrania z witryny <https://physics.princeton.edu/pulsar/k1jt/wsjt.html>, zainstalowania go i wywołania. Przed dalszym użyciem konieczna jest jego konfiguracja wymagająca m.in. podania znaku wywoławczego, lokalizacji, wyboru systemu dźwiękowego itd.

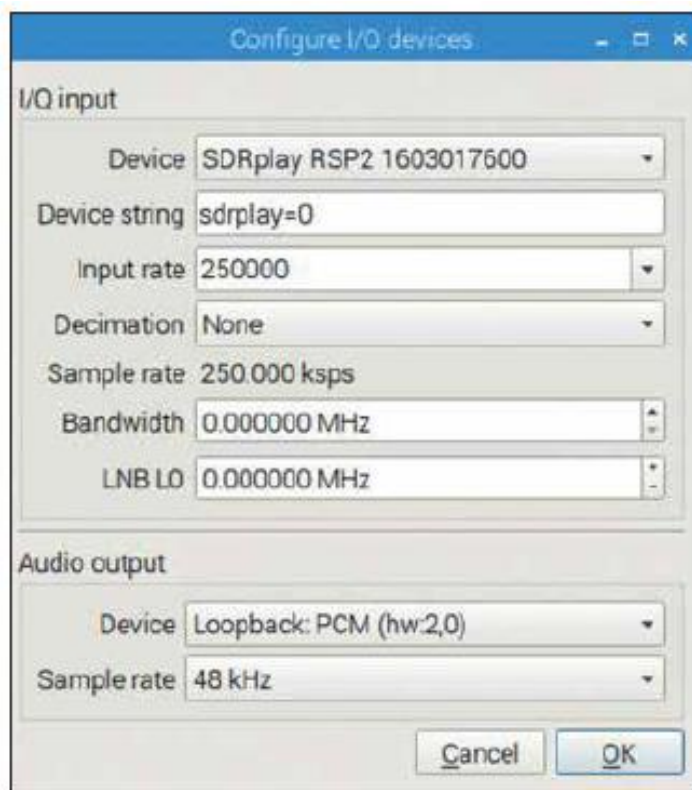
8. Dla powiązania odbiornika przez GQRX z WSJT konieczna jest też konfiguracja wirtualnego kabla dla sygnałów audio. W tym celu należy w oknie terminalowym Raspbiana (oknie wiersza poleceń) podać polecenie

```
sudo modprobe snd_aloop
```

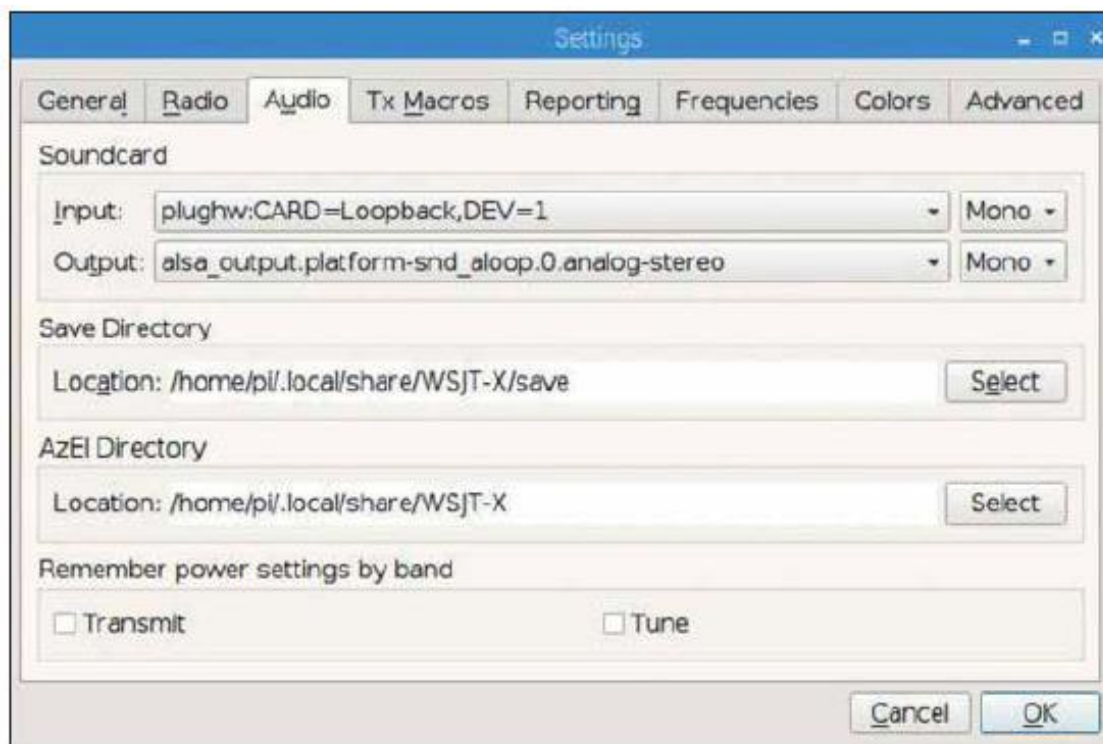
Następnie należy włączyć odbiornik, wywołać program detekcyjny GQRX i wybrać w nim wirtualny podsystem dźwiękowy jako wyjściowy. Odbiornik należy dostroić do jednej z częstotliwości WSPR i tą samą częstotliwość (to samo pasmo) należy wybrać w WSJT po jego wywołaniu. W konfiguracji WSJT należy ten sam wirtualny podsystem dźwiękowy wybrać jako wejście.

Jeżeli wszystko jest w porządku w oknie GQRX powinno być widoczne widmo sygnałów, a w oknie WSJT – zdekodowane komunikaty.

Autor opisu Dimitri Jeljussejew twierdzi wprawdzie, że GQRX nie daje się wywoływać automatycznie po włączeniu całości, i że konieczne jest pozostawienie na stałe podłączonej klawiatury, ale warto z tym poeksperymentować korzystając z pliku *autostart LXDE* albo pliku *autostart* z katalogu */config* na podane powyżej sposoby. Umożliwiłoby to odłączenie klawiatury i myszy po zakończeniu prac instalacyjno-uruchomieniowych.



Rys. 5.8.1. Wybór systemu dźwiękowego jako wyjściowego GQRX



Rys. 5.8.2. Wybór wejściowego systemu dźwiękowego w WSJT

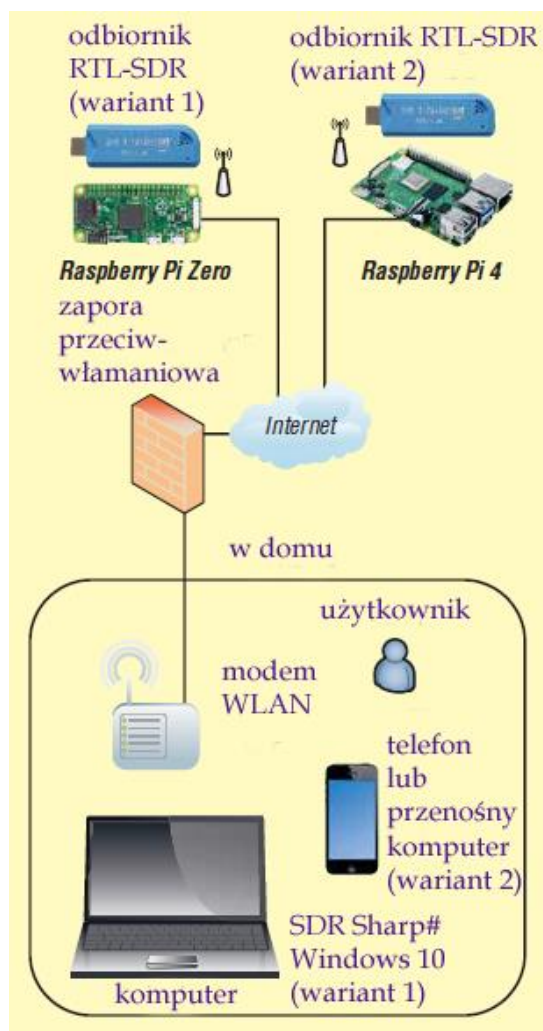


Fot. 5.8.3. Widok konstrukcji

### 5.9. Odbiór cyfrowego dźwięku

Opisane przez DL1YBL w numerze 6/2020 Funkamateura rozwiązanie pozwala na odbiór cyfrowego dźwięku nadawanego w systemach DMR, C4FM, NXDN, P25 za pomocą odbiornika RTL-SDR lub odbiorników innych typów podłączonych do *Maliny* i słuchanie zdekodowanych transmisji na komputerach lub telefonach przez Internet lub w ramach domowej sieci WiFi. Pozwala to na zainstalowanie odbiornika w miejscu korzystnym dla odbioru radiowego. Rozwiązanie jest niedrogie i energooszczędne. Oprogramowanie można zainstalować na module SD z funkcjonującym już systemem Raspbian albo pobrać z Internetu odwzorowanie pamięci i przepisać je na moduł SD za pomocą wymienianych już programów służących do tego celu. W wariantcie pierwszym odbiornik jest podłączony do

modelu *Pi Zero*. Do odbioru transmisji analogowych na PC można zastosować program *sdr-sharp* dostępny w witrynie [www.rtl-sdr.com/tag/sdrsharp](http://www.rtl-sdr.com/tag/sdrsharp). W konfiguracji połączenia TCP należy podać numer kanału logicznego (ang. *port*) 1234. Szerokość obserwowanego pasma wynosi kilkaset kHz. Odbiór cyfrowego głosu wymaga uzupełnienia o wokoder AMBE. Zamiast wokodera sprzętowego spotykanego w wielu modelach radiostacji w opisanym rozwiązaniu zastosowano wokoder programowy.



Rys. 5.9.1. Warianty rozwiązania



Fot.5.9.2. Wariant 1

W charakterze wokodera służy biblioteka *Mbelib* (<https://github.com/szechyjs/mbelib>) stanowiąca część oprogramowania *DSD+ 1.101*. Dla DMR, NXDN, APCO P25 i kilku innych mało używanych przez krótkofalowców systemów cyfrowego głosu daje ona bardzo dobre wyniki, natomiast dla D-Stara są one bardzo złe. Dekodowanie transmisji TETRA konieczne jest zainstalowanie dodatkowej wtyczki programowej. Bibliotekę *Mbelib* należy zainstalować w katalogu *DSD+* założonym w katalogu, w którym znajduje się *SDR#*. Do tego katalogu należy też skopiować bibliotekę *SDRsharpDSD.dll*. Do odbioru cyfrowego głosu należy w pliku konfiguracyjnym *SDR#* dodać linię

```
<add key="DSD" value="SDRSharp.DSD.DSDPlugin., SDRSharp.DSD" />
```

a do dekodowania TETRA także linię

```
<add key="Tetra" value="SDRSharp.Tetra.TetraPlugin, SDRSharp.Tetra" />
```

Połączenie wokodera z programem odbiorczym zapewnia wirtualny kabel *VB-Cable Driver*. Można pobrać go z witryny [www.vb-audio.com/Cable](http://www.vb-audio.com/Cable).

Instalacja odbiornika RTL-SDR na *Malinie Zero* przebiega następująco:

```
apt install rtl-sdr librtlsdr-dev
```

```
apt-get install git cmake libusb-1.0-0.dev build-essential
```

```
git clone git://git.osmocom.org/rtl-sdr.git
```

```
cd rtl-sdr/
mkdir build
cd build/
cmake ../
make sudo make install
sudo ldconfig
cd ..
```

```
cp rtl-sdr.rules /etc/udev/rules.d
```

Po zainstalowaniu można przeprowadzić próbę podając polecenie

```
rtl_tcp -a 192.168.176.46 &
```

Podany w nim (przykładowy) adres IP można odpytać za pomocą polecenia *ifconfig*.

Do automatycznego wywołania można założyć plik *etc/systemd/system/rtl\_stick.service* o następującej zawartości:

```
#####
```

```
[Unit]
```

```
Description= RTL_Stick Service
```

```
After=multi-user.target network-online.target
```

```
Wants=network-online.target
```

```
[Service]
```

```
Type=idle
```

```
ExecStart=/bin/bash -c "/usr/bin/rtl_tcp -a $(/sbin/ip -o -4 addr show dev eth0 | awk '{ print $4; }' | cut -d/ -f1) "
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
#####
```

W celu uruchomienia usługi należy podać polecenia

```
systemctl enable rtl_stick.service
```

```
systemctl daemon-reload
```

W wariancie drugim konieczne jest użycie modelu Pi4 i zainstalowanie na nim programu odbiorczego *Open Web RX* (<https://github.com/ha7ilm/openwebrx>). Nowsze wersje programu powstałe przy współpracy DD5JFK obsługują odbiorniki RTL-SDR, HackRF, SDRplay, AirSpy, Lime-SDR i PlutoSDR (<https://www.openwebrx.de/download/rpi.php>). Pozwalają one na korzystanie z kilku odbiorników równoległe. Program jest wyposażony w dekodery cyfrowych transmisji DMR, YSF i POCSAG, a także w dekodery emisji grupy WSJT-X (FT8, FT4, WSPR, JT65, JT9). Open Web SDR dysponuje serwerem HTTP, z którym można nawiązać połączenie za pomocą przeglądarki internetowej z dowolnego komputera lub telefonu pracującego pod dowolnym systemem operacyjnym. W wariancie tym możliwe jest dekodowanie emisji D-STAR, DMR, NXDN i C4FM, a także odbiór analogowych transmisji FM.

Istotną zaletą wariantu pierwszego są jego niskie koszty, komfort pracy oferowany przez program odbiorczy SDR#, możliwość dekodowania TETRA i automatyczne rozpoznawanie systemu cyfrowego.

W wariancie drugim na komputerze użytkownika nie potrzeba żadnych dodatkowych programów.

W wariancie tym nie ma możliwości automatycznego rozpoznawania emisji, a także dekodowania systemów TETRA i P25, ale za to można dekodować emisje z rodziny WSJT-X. Jakość głosu jest jednak gorsza niż w wariancie pierwszym.

## 5.10. Prosty serwer HTTP

Prosty serwer http może okazać się przydatny do testowania własnych stron internetowych albo hamnetowych, do udostępniania w Internecie lub Hamnecie wyników pomiarowych albo innych obserwacji, a także do różnych innych eksperymentów związanych nie tylko z krótkofalarstwem. Jednym z takich bezpłatnych serwerów dla *Maliny* jest *lighttpd* ([www.lighttpd.net](http://www.lighttpd.net)). Jego instalacja wymaga następujących kroków:

1. Utworzenia katalogu `/var/www` używanego przez wiele serwerów dla danych stron http:

```
sudo mkdir /var/www
```

2. Założenia specjalnego użytkownika `www-data` dla serwera. Istnieje on już w wielu wersjach Raspbiana, ale w przypadku jego braku należy najpierw założyć grupę `www-data`

```
sudo addgroup www-data
```

a następnie użytkownika

```
sudo adduser --ingroup www-data www-data
```

3. Przyznania użytkownikowi uprawnień właściciela katalogu `/var/www`

```
sudo chown -R www-data:www-data /var/www
```

4. Zainstalowania serwera i rozszerzeń php

```
sudo apt-get update
```

```
sudo apt-get install lighttpd php5-cgi
```

5. Po zakończeniu instalacji automatycznie tworzony jest skrypt wywołujący serwer po włączeniu *Maliny*. Podanie w oknie przeglądarki internetowej na PC lub innym dowolnym komputerze adresu IP *Maliny* w sieci lokalnej powoduje wyświetlenie w niej tymczasowej pierwszej strony serwera.

6. Aby umożliwić korzystanie ze stron php należy włączyć moduł *fastcgi*

```
sudo lighty-enable-mod fastcgi
```

```
sudo lighty-enable-mod fastcgi-php
```

7. Ponownego załadowania zmodyfikowanej konfiguracji serwera

```
sudo /etc/init.d/lighttpd force-reload
```

Serwer startuje zawsze automatycznie po włączeniu *Maliny*.

8. Dla zwiększenia pojemności pamięci przeznaczonej dla serwera można katalog `/var/www` przenieść z pamięci SD na zewnętrzny nośnik podłączony do złącza USB. W poniższym przykładzie jest on logicznie podłączony (zamontowany) do systemu jako `/media/usb1` (`sudo mkdir /media/usb1`, `sudo mount /dev/sda1 /media/usb1`)

```
sudo mv /var/www /media/usb1
```

9. Na nośniku należy założyć symboliczny odnośnik do katalogu

```
sudo ln -s /media/usb1/www /var/www
```

Serwer korzysta w wyniku tego z nośnika USB tak jakby katalog znajdował się w pamięci SD.

Do celów specjalnych można wywoływać *lighttpd* z dodatkowymi parametrami:

-f <nazwa pliku> – wywołanie z podanym (alternatywnym) plikiem konfiguracyjnym. Standardowo `/etc/lighttpd/lighttpd.conf`

-m <katalog> – użycie podanego katalogu zawierającego moduły. Standardowo `/usr/lib/lighttpd`

-p – wyświetlenie pliku konfiguracyjnego w wewnętrznym formacie

-t – test pliku konfiguracyjnego

-D – serwer nie pracuje w tle

-v – podanie numeru wersji

-V – wyświetlenie spisu dodatkowych funkcji zawartych w aktualnej wersji programu

-h – wywołanie krótkiego tekstu pomocy.

### 5.11. „Hamserver Pi”

Pakiet oprogramowania „Hamserver Pi” dostępny w internecie pod adresem <https://db0gw-i.ampr.org/download/category/28-hamserverpi> jest odwzorowaniem pamięci dla *Maliny* zawierającym dodatkowo do systemu operacyjnego serwer HTTP z wyszukiwarką i statystyką odwiedzin, system CMS do zarządzania stronami, serwer FTP, bramkę *iGate* APRS-RX dla APRS serwer dostępowy do sieci APRS-IS, serwer *Mumble* dla prowadzenia konferencji fonicznych, serwer poczty elektroicznej, serwer VoIP, serwer radiowy dla odbiorników RTL-SDR, oprogramowanie *SvxLink* dla Echolinku, *SvxServer* do połączenia przemienników fonicznych przez sieć TCP/IP, oprogramowanie klienta VPN i serwer pośredniczący dla Echolinku („Proxy”). Całość jest przeznaczona do wykorzystania w sieci Hamnetu i ze względu na brak dostatecznych zabezpieczeń nie jest zalecane korzystanie z niego w Internecie. Uruchomienie i konfiguracja całości jest stosunkowo prosta co ułatwia uruchamianie nowych węzłów hamnetowych.

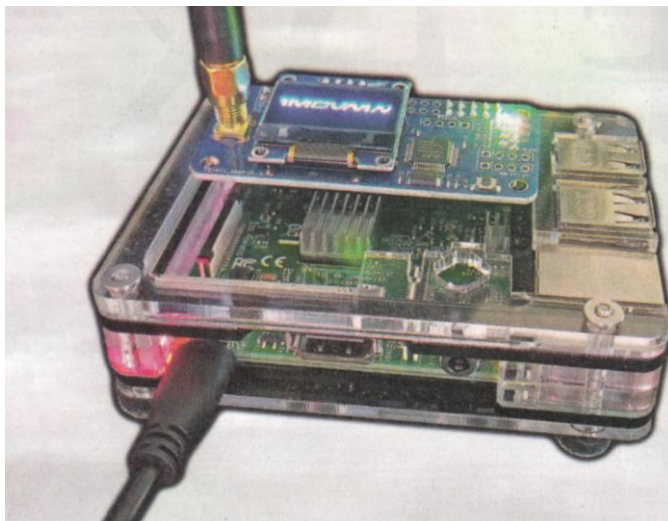


## 6. Mikroprzeziennik „PiStar”

Oferta gotowych przezienników domowych (ang. *hotspot*) jest wprawdzie dostatecznie obfita, ale miłośnicy własnych konstrukcji mogą sami skonstruować niedrogie urządzenia oparte o *Malinę* i dodatkową płytkę rozszerzeń (ang. *PiHAT*) MMDVM. Po zainstalowaniu na mikrokomputerze oprogramowania *Pi-Star* otrzymuje się mikroprzeziennik o tej samej nazwie. Płytkę MMDVM posiada gniazdko wtykane na listwę kontaktów mikrokomputera. Informacje o dostępnych konstrukcjach MMDVM i oprogramowaniu *Pi-Star* można znaleźć w Internecie m.in. pod adresem [6.2], wystarczy też podać nazwę w wyszukiwarce internetowej. Instrukcja do programu *Pi-Star* znajduje się pod adresem [6.3].

Mikroprzeziennik pracuje w systemach cyfrowego głosu D-STAR, DMR, C4FM, NXDN i APCO P25 oraz w systemie przywoławczym POCSAG. Możliwa jest także konfiguracja przeziennika skrośnego z dostępem radiowym w jednym systemie, a połączeniem internetowym w innym. Pozwala to przykładowo na pracę w sieci YSF (C4FM) przy użyciu radiostacji DMR. Ostatnio dużą popularność zyskało oprogramowanie „PiStar” autorstwa EA7EE. Jest ono szczególnie zalecane do pracy w sieci C4FM.

Wśród modeli gotowych przezienników albo zestawów do własnej konstrukcji opartych na „Pi-Starze” znajdują się m.in. Jumbo Spot, Zumspot i NEXTGEN XD4.



Fot. 6.1. Konstrukcja mikroprzeziennika

Po zaopatrzeniu się w *Malinę* i pasującą płytkę MMDVM konieczne jest pobranie programu *Pi-Star* w odpowiedniej wersji spod adresu [6.2] i przepisania go na moduł pamięci mikro SD o pojemności co najmniej 16 GB. Do wyboru są również wersje dla innych typów mikrokomputerów. Oprogramowanie wraz systemem operacyjnym jest dostępne w postaci odwzorowania (obrazu) pamięci mikrokomputera, dlatego też skopiowanie go za pomocą zwykłej funkcji kopiowania systemu Windows nie wchodzi w grę. Konieczne jest użycie specjalnego programu kopiującego. Do najpopularniejszych z nich należy *Win32Diskimager* ([6.5], [6.6]) ale dobrze znany jest również *Etcher* ([6.4]). Oprócz wersji dla Windows istnieją także wersje *Etchera* dla Linuksa i MacOS.

Po włożeniu modułu pamięci do kieszeni *Maliny* uruchamiany jest najpierw jej system operacyjny, a następnie „Pi-Star”. Program próbuje początkowo nawiązać połączenie z Internetem przez WiFi, ale oczywiście brakuje mu jeszcze właściwych danych dostępowych. Na zakończenie tej fazy program przechodzi w tryb pracy punktu dostępowego, co pozwala na połączenie się z nim przez komputer PC lub androidowy. Mikroprzeziennik jest dostępny w spisie sieci na PC pod nazwą PI-STAR. Po znalezieniu go w spisie należy nacisnąć na ekranie przycisk *Połącz*. Hasłem dostępu jest *raspberry*.

Po nawiązaniu połączenia należy wywołać powierzchnię obsługi podając w przeglądarce internetowej w polu adresowym adres 192.168.50.1. Wygląd głównej strony przedstawiono na ilustracji 6.2 (w zależności od modelu MMDVM okno to i kolejne przedstawione dalej mogą się różnić od pokazanych na ilustracjach). Należy wybrać na niej punkt konfiguracji (*Configuration*) znajdujący się u góry po prawej stronie okna. Zmiany wprowadzone w każdym z bloków konfiguracyjnych trzeba potwierdzić je za pomocą leżącego poniżej przycisku *Apply Changes* (*Zastosuj*).



W przedstawionym na ilustracji 6.3 panelu konfiguracji trybu pracy (*Control software*) należy wybrać tryby MMDVM HOST i SIMPLEX MODE dla pracy simpleksowej. Do współpracy z modelami starszymi jak DVAP czy DVRPTR służy tryb *DstarRepeater*.

W konfiguracji pracy serwera MMDVM (rys. 6.4) wybiera się system D-Star lub inny z tam wymienionych w zależności od potrzeb. Należy także włączyć wyświetlacz wybierając właściwy jego typ, np. OLED. W konfiguracji ogólnej (rys. 6.5) konieczne jest wpisanie własnego znaku wywoławczego.

Hostname: pi-star Pi-Star: 3.4.11 / Dashboard: 20180523

## Pi-Star Digital Voice Dashboard for M6CEB

Dashboard | Admin | Configuration

Modes Enabled	
D-Star	DMR
YSF	P25
YSF XMode	NXDN

Network Status	
D-Star Net	DMR Net
YSF Net	P25 Net
YSF2DMR	NXDN Net
YSF2NXDN	YSF2P25

Radio Info	
Trx	TX DMR Slot 2
Tx	434.000000 MHz
Rx	434.000000 MHz
PW	HS_Hat:v1.3.6

DMR Repeater	
DMR ID	2342107
DMR CC	1
TS1	disabled
TS2	enabled
TG 2350/not linked	

DMR Master	
BM United Kingdom 2..	

Gateway Activity								
Time (BST)	Mode	Callsign	Target	Src	Dur(s)	Loss	BER	
20:59:12 May 25th	DMR Slot 2	2342682	TG 2350	Net	TX			
20:58:40 May 25th	DMR Slot 2	2341779	TG 2350	Net	23.2	0%	0.0%	
20:56:38 May 25th	DMR Slot 2	KB9BUG	TG 2350	Net	37.2	0%	0.0%	
20:53:29 May 25th	DMR Slot 2	2344941	TG 2350	Net	6.2	0%	0.0%	
20:52:42 May 25th	DMR Slot 2	2353143	TG 2350	Net	1.9	0%	0.1%	
20:51:13 May 25th	DMR Slot 2	KC9DIO	TG 2350	Net	0.8	0%	0.0%	
20:50:15 May 25th	DMR Slot 2	2341938	TG 2350	Net	0.8	0%	0.2%	
20:47:35 May 25th	DMR Slot 2	2344455	TG 2350	Net	3.4	0%	0.0%	
20:47:23 May 25th	DMR Slot 2	2352739	TG 2350	Net	1.6	0%	0.1%	
20:46:07 May 25th	DMR Slot 2	2342875	TG 2350	Net	0.5	0%	0.0%	
20:45:11 May 25th	DMR Slot 2	2400251	TG 2350	Net	5.1	21%	0.0%	
20:45:01 May 25th	DMR Slot 2	2351743	TG 2350	Net	1.2	30%	0.0%	
20:35:16 May 25th	DMR Slot 2	KE5CDK	TG 2350	Net	0.5	0%	0.0%	
20:34:44 May 25th	DMR Slot 2	2352439	TG 2350	Net	0.8	0%	0.0%	
20:34:20 May 25th	DMR Slot 2	2353374	TG 2350	Net	0.5	0%	0.0%	
20:26:48 May 25th	DMR Slot 2	6551023	TG 2350	Net	0.5	0%	0.0%	
20:26:24 May 25th	DMR Slot 2	2344822	TG 2350	Net	2.3	15%	0.0%	
20:26:10 May 25th	DMR Slot 2	2344285	TG 2350	Net	1.6	0%	0.0%	
20:13:59 May 25th	DMR Slot 2	2345103	TG 2350	Net	4.8	22%	0.0%	
20:09:34 May 25th	DMR Slot 2	2341975	TG 2350	Net	8.0	0%	0.0%	

Local RF Activity							
Time (BST)	Mode	Callsign	Target	Src	Dur(s)	BER	RSSI

Pi-Star / Pi-Star Dashboard, © Andy Taylor (MW0MWZ) 2014-2018.  
 ircDDBGateway Dashboard by Hans-J. Barthen (DL5DI),  
 MMDVMDash developed by Kim Huebel (DG9VH),  
 Need help? Click here for the Facebook Group  
 or Click here to join the Support Forum  
 Get your copy of Pi-Star from here.

Rys. .6.2. Okno główne „PiStara”

Pi-Star: 3.4.11 / Dashboard: 20180523

## Pi-Star Digital Voice - Configuration

Dashboard | Admin | Expert | Power | Update | Backup/Restore | Factory Reset

Gateway Hardware Information				
Hostname	Kernel	Platform	CPU Load	CPU Temp
pi-star	4.9.35-v7+	Pi 3 Model B (1GB) - Sony, UK	0 / 0 / 0	45.6°C / 114.1°F

Control Software	
Setting	Value
Controller Software:	<input type="radio"/> DStarRepeater <input checked="" type="radio"/> MMDVMHost (DV-Mega Minimum Firmware 3.07 Required)
Controller Mode:	<input checked="" type="radio"/> Simplex Node <input type="radio"/> Duplex Repeater (or Half-Duplex on Hotspots)

Apply Changes

MMDVMHost Configuration

Rys. 6.3. Konfiguracja trybu pracy („Control Software”)

Operator musi wybrać także częstotliwość pracy mikroprzebiegnika i wpisać ją do konfiguracji. Tą samą częstotliwość trzeba też zaprogramować w radiostacji. Większość modeli MMDVM pracuje w paśmie 70 cm, ale występują również rozwiązania pokrywające pasmo 2 m.

Dalszymi, ale nie niezbędnymi danymi są geograficzna lokalizacja stacji (długość i szerokość geograficzna), nazwa miasta, kraj i lokator. Wybór publicznej dostępności (pole „Public”) pozwala na korzystanie z przebiegnika również innym stacjom, natomiast po zaznaczeniu wariantu stacji prywatnej może z niej korzystać tylko właściciel (stacja o tym samym znaku wywoławczym). Konfiguracja APRS może na początek pozostać niezmieniona. Do ostatnich parametrów konfiguracji ogólnej należą strefa czasowa i język obsługi.

MMDVMHost Configuration				
Setting		Value		
DMR Mode:	<input checked="" type="checkbox"/>	RF Hangtime: 20	Net Hangtime: 20	
D-Star Mode:	<input type="checkbox"/>	RF Hangtime: 20	Net Hangtime: 20	
YSF Mode:	<input type="checkbox"/>	RF Hangtime: 20	Net Hangtime: 20	
P25 Mode:	<input type="checkbox"/>	RF Hangtime: 20	Net Hangtime: 20	
NXDN Mode:	<input type="checkbox"/>	RF Hangtime: 20	Net Hangtime: 20	
YSF2DMR:	<input type="checkbox"/>			
YSF2NXDN:	<input type="checkbox"/>			
YSF2P25:	<input type="checkbox"/>			
MMDVM Display Type:	Nextion	Port: Modem	Nextion Layout: ON7LDS	

Apply Changes

Rys. 6.4. Konfiguracja serwera MMDVM („MMDVM Host”)

W konfiguracji serwera następuje włączenie poszczególnych systemów cyfrowego głosu, a także możliwości łączności skrośnych DMR2YSF, DMR2NXDN itd.

W konfiguracji D-Stara (rys. 6.6) w polu RPT1 wprowadzany jest własny znak operatora z dodatkiem litery B jeśli przebiegnik pracuje w paśmie 70 cm lub C – dla pasma 2 m. Pole RPT2 jest wypełniane automatycznie i zawiera ten sam znak stacji z dodatkiem litery G. Możliwe jest także podanie domyślnego reflektora i modułu oraz serwera APRS. Zestaw parametrów konfiguracyjnych różni się dla poszczególnych systemów cyfrowego głosu. W konfiguracji dla DMR podawane są również identyfikator DMR otrzymany przy rejestracji w sieci, serwer Brandmeistera (BM), z rozwijanej listy, przy korzystaniu z tej sieci i kod CC dla dostępu radiowego. Zalecane jest korzystanie z krajowego, czyli najbliższego, serwera BM i standardowego kodu CC1. Konfiguracja zapory przeciwwłamaniowej powinna wyglądać tak jak na rys. 6.7. W konfiguracji YSF podawany jest reflektor wejściowy i ewentualnie serwer APRS, w konfiguracji P25 aewrwer wejściowy i identyfikator NAC, a w konfiguracji NXDN serwer wejściowy i identyfikator RAN.

Ostatnią istotną sprawą jest konfiguracja bezprzewodowego dostępu do sieci domowej – dojścia do Internetu i przez to do odpowiedniej amatorskiej sieci cyfrowego głosu. Konieczne jest podanie nazwy sieci domowej i hasła dostępu. Bezbiełność tych danych najlepiej sprawdzić przez wyłączenie i włączenie mikroprzebiegnika. Powinien on wtedy nawiązać połączenie z siecią. Adres IP mikroprzebiegnika w sieci domowej powinien być widoczny na jego wyświetlaczu.

General Configuration	
Setting	Value
Hostname:	pi-star Do not add suffixes such as .local
Node Callsign:	M6CEB
CCS7/DMR ID:	2342107
Radio Frequency:	434.000.000 MHz
Latitude:	53.752983 degrees (positive value for North, negative for South)
Longitude:	-2.875540 degrees (positive value for East, negative for West)
Town:	Preston, IO83NS
Country:	Country, UK
URL:	http://www.qrz.com/db/M6CEB <input checked="" type="radio"/> Auto <input type="radio"/> Manual
Radio/Modem Type:	MMDVM_HS_Hat (DB9MAT & DF2ET) for Pi (GPIO)
Node Type:	<input checked="" type="radio"/> Private <input type="radio"/> Public
System Time Zone:	Europe/London
Dashboard Language:	english_uk

Apply Changes

Rys. 6.5. Okno konfiguracji ogólnej

D-Star Konfiguration	
Einstellung	Wert
RPT1 Rufzeichen:	OE8  B
RPT2 Rufzeichen:	OE8  G
ircDDBGateway Passwort:	.....
Standard Reflektor:	DCS009  A <input checked="" type="radio"/> Startup <input type="radio"/> Manual
APRS Host:	austria.aprs2.net
ircDDBGateway Sprache:	Deutsch
Zeit Ansagen:	<input checked="" type="checkbox"/>
Use DPlus for XRF:	<input type="checkbox"/> Note: Update Required if changed

Speichern

Rys. 6.6. Okno konfiguracji łączności D-Starowych. W przykładzie podany jest austriacki reflektor DCS009A, który należy zastąpić zgodnym z własnymi upodobaniami. W radiostacji należy wybrać duplex DUP- i odstęp częstotliwości 0,000 MHz

DMR Configuration	
Setting	Value
DMR Master:	BM_United_Kingdom_2341
BrandMeister Network:	Repeater Information   Edit Repeater (BrandMeister Selfcare)
DMR Colour Code:	1
DMR EmbeddedLCOnly:	<input type="checkbox"/>
DMR DumpTADData:	<input checked="" type="checkbox"/>

Apply Changes

Firewall Configuration	
Setting	Value
Dashboard Access:	<input checked="" type="radio"/> Private <input type="radio"/> Public
ircDDBGateway Remote:	<input checked="" type="radio"/> Private <input type="radio"/> Public
SSH Access:	<input checked="" type="radio"/> Private <input type="radio"/> Public
Auto AP:	<input checked="" type="radio"/> On <input type="radio"/> Off Note: Reboot Required if changed
uPNP:	<input checked="" type="radio"/> On <input type="radio"/> Off

Apply Changes

Rys. 6.7a. Konfiguracja DMR i zapory przeciwwłamaniowej



DMR Konfiguration	
Einstellung	Wert
DMR Master:	DMRGateway
BrandMeister Master:	BM_Switzerland_2281
BrandMeister Netzwerk:	Repeater Information   Edit Repeater (BrandMeister Selfcare)
DMR+ Master:	DMR+_IPSC2-OE- DMO
DMR+ Netzwerk:	Options=StartRef=4198;RelinkTime=120;UserLink=1;TS1_1=110;TS1_2=20;TS2_3=232;
XLX Master:	XLX_950
XLX Master Aktiv:	<input type="checkbox"/>
DMR Color Code:	1
DMR EmbeddedLCOnly:	<input type="checkbox"/>
DMR DumpTADData:	<input checked="" type="checkbox"/>

Speichern

Rys. 6..7b. Konfiguracja DMR dla pracy równoległej w BM i IPSC2 (DMR+)

Po włączeniu w konfiguracji dla DMR bramki międzysystemowej „DMRGateway” w polu „DMR Master” możliwe jest równoległe połączenie z sieciami BM, IPSC2 (DMR+) i reflektorami XLX przez wybór w dalszych polach odpowiednich serwerów „Master”.

W polu „DMR DumpTADData” podawany jest znak stacji. Jest on wyświetlany przez radiostacje mające tę możliwość.

Yaesu System Fusion Konfiguration	
Einstellung	Wert
YSF Startup Host:	YSF55693 - AT C4FM Austria - YSF
APRS Host:	austria.aprs2.net

Speichern

Rys. 6.7c. Konfiguracja C4FM. W przykładzie podano kółeczko (ang. *room*) austriackie, które należy zastąpić przez zgodne z własnymi upodobaniami, np. kółeczko YSF Polska. Włączenie funkcji „WiresX Passthrough” umożliwia przekazywanie do sieci poleceń używanych w trybie „Wires” włączanym za pomocą klawisza „X” lub „DX” zależnie od typu radiostacji. Tryb ten nie ma nic wspólnego z połączeniami w sieci „Wires-X” i służy do zdalnego sterowania przemiennikami w celu nawiązywania dalszych połączeń

Wlan Konfiguration	
WiFi Info	
WiFi Regulatory Domain (Country Code) : CH	
Network 0	Delete    SSID : HUAWEI P30  PSK :
Network 1	Delete    SSID : OE8VIK  PSK :
Network 2	Delete    SSID : OE8VIK-WLAN  PSK :
Scan for Networks (10 secs)   Add Network   Save (and connect)	

Rys. 6.8. Konfiguracja bezprzewodowego dostępu do sieci domowej

Wybranie w polu „DMR Master” jednego z wariantów skrótnych „DMR2YSF” itd. umożliwia łączności skrótnie z daną siecią przy użyciu radiostacji DMR. Włączenie transmisji DMR i DMR2YSF wymaga wyłączenia bezpośrednich łączności YSF i innych możliwości skrótnych. Najlepiej wyłączyć na ten czas także pozostałe możliwości bezpośrednich połączeń z innymi sieciami. Po włączeniu połączeń skrótnych na ekranie wyświetlane są okna konfiguracji DMR i YSF. W konfiguracji DMR jako „DMR

Master” lepiej jest wybrać „DMR2YSF” ale możliwe jest też wybranie „DMR Gateway”. W pierwszym przypadku po stronie DMR można używać dowolnej grupy, a w drugim dodatkowo do oznaczeń reflektorów YSF lub FCS należy podać jako prefiks grupę 7. W konfiguracji YSF należy wybrać pożądany reflektor, gdyż wybór ich przez radio nie jest możliwy. Oprócz tego wybierany jest serwer APRS.

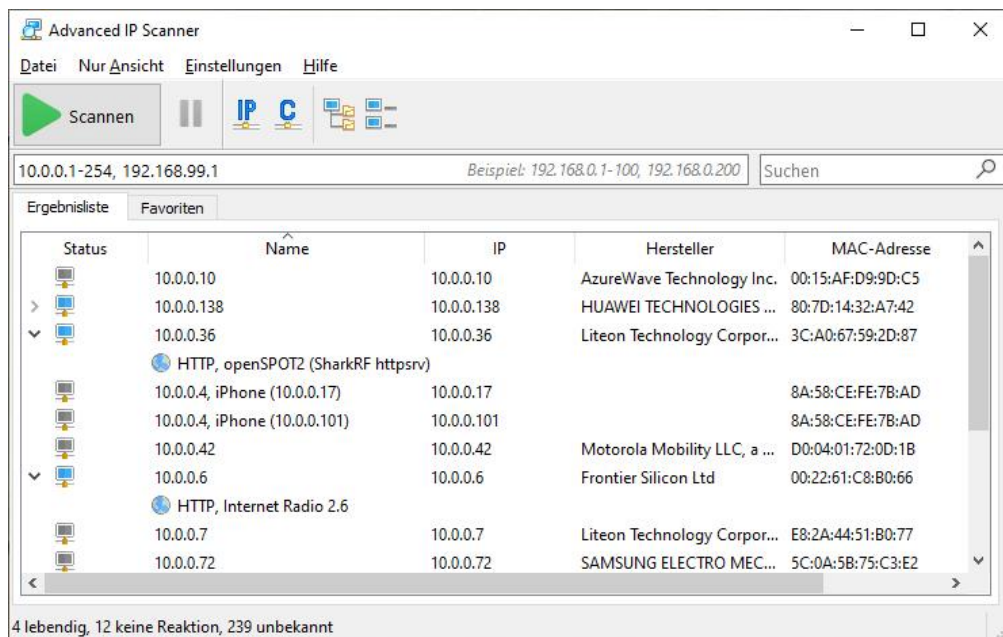
W przypadku połączenia skrośnego z siecią NXDN należy w oknie konfiguracji DMR wybrać DMR2NXDN lub „DMR Gateway”, przy czym pierwsza możliwość jest mniej problematyczna. W przypadku wybrania „DMR Gateway” wybierane grupy rozmówców NXDN muszą być poprzedzone prefiksem 7. W konfiguracji NXDN można podać w polu początkowego serwera pozycję „None” lub wybrać serwer. W polu NXDN RAN podawany jest identyfikator stacji (RAN = *Random Access Number*).

Analogicznie możliwa jest konfiguracja skrośna z C4FM na DMR, NXDN lub P25 odpowiednio przez wybór w polu „YSF Startup Host” pozycji YSF2DMR itd. Pozostałe połączenia skrośne, a najlepiej i bezpośrednio (nie tylko DMR) powinny być w tym czasie wyłączone. W konfiguracji YSF oprócz serwera podawany jest dostęp do sieci APRS, a w powiązanej z nią konfiguracji DMR wybierany jest serwer „DMR Master” i grupa rozmówców. Nie można jej zmienić potem bezpośrednio z radiostacji.

Analogicznie dla łączności skrośnych YSF – NXDN należy w oknie YSF podać jako serwer YSF2NXDN i serwer APRS, a w konfiguracji NXDN identyfikator NXDN i używany serwer.

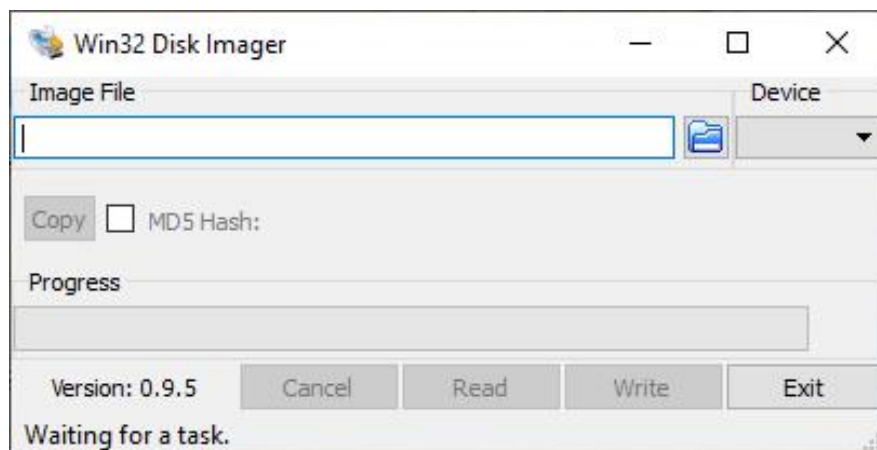
Dla połączeń skrośnych w konfiguracji YSF podawany jest serwer YSFP25 i serwer dostępowy do APRS a w konfiguracji P25 identyfikator P25 (NAC) i serwer dostępowy.

Korzystanie z systemu przywoławczego POCSAG wymaga wybrania w oknie konfiguracyjnym serwera, podania znaku wywoławczego i częstotliwości pracy sieci (standardowo 439,987500 MHz) i hasła dostępowego do sieci.

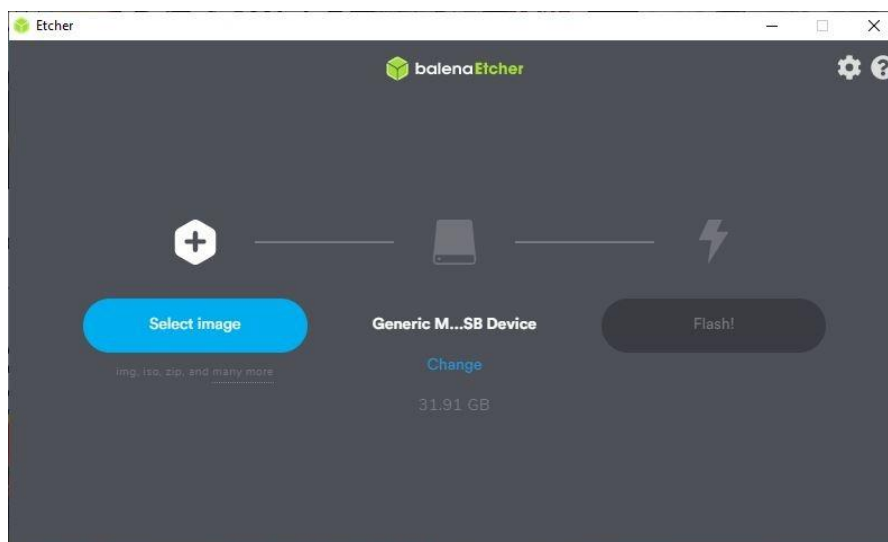


Rys. 6.9. „Advanced IP Scanner” dla Windows – okno główne

W poszukiwaniu adresów IP w sieci domowej pomocny jest bezpłatny program „Advanced IP Scanner” przeszukujący i analizujący sieć bezprzewodową (rys. 6.9). Można pobrać go spod adresu [6.7]. Odczytany z wyświetlacza „Pi-Stara” lub z okna programu analizującego adres IP można wprowadzić do pola adresowego przeglądarki internetowej w przypadku przeprowadzania korekt konfiguracji lub dla obserwacji pracy przemiennika.



Rys. 6.10. „Win32 Disk Imager” służy do zapisu obrazu pamięci na module SD. W polu „Image File” wybierany jest plik na twardym dysku PC, a w polu „Device” – pamięć SD jako dysk zewnętrzny



Rys. 6.11. Program kopiujący „Etcher”



Fot. 6.12.



## 6.1. Szczegóły konfiguracji w trybie eksperta

Poniższe przykłady konfiguracji pochodzą z instrukcji opracowanej przez OE8VIK i opublikowanej w witrynie [ham-dmr.at](http://ham-dmr.at).



Rys. 6.1.1. Przełączenie na tryb eksperta. W trybie eksperta możliwa jest modyfikacja zawartości różnych plików systemowych „Pi-Stara”

General	
RptAddress	127.0.0.1
RptPort	62032
LocalAddress	127.0.0.1
LocalPort	62031
RuleTrace	0
Daemon	1
Debug	0
RFTIMEOUT	20
NetTimeout	20

Speichern

Log	
DisplayLevel	0
FileLevel	1
FilePath	/var/log/pi-star
FileRoot	DMRGateway

Speichern

Voice	
Enabled	1
Language	en_GB
Directory	/usr/local/etc/DMR_Audio

Speichern

Rys. 6.1.2. Konfiguracja bramki międzysystemowej DMR GW wywołana w górnej linii „Quick Edit”. W tym i w poniższych przykładach wystarczy zmienić jedynie niektóre indywidualne dane j.np. serwery sieci, hasła dostępu, identyfikatory DMR, znaki wywoławcze i inne informacje o stacji pozostawiając resztę danych tak jak były

DMR Network 1	
Enabled	1
Address	128.65.196.21
Port	62031
TGRewrite	2,8,2,8,1
PCRewrite	2,84000,2,84000,1001
TypeRewrite	2,9990,2,9990
SrcRewrite	2,84000,2,8,1001
PassAllPC	2
PassAllTG	2
Password	passw0rd
Debug	0
Name	BM_Switzerland_2281

Speichern

Rys. 6.1.3. Konfiguracja sieci od strony Brandmeistra. Należy wybrać nazwę najbliższego serwera BM. Od 1 marca 2021 zamiast ogólnego hasła *passw0rd* stosowane są indywidualne hasła użytkowników ustalone w prywatnym obszarze użytkownika na serwerze BM

DMR Network 2	
Enabled	1
Address	89.185.97.34
Port	55555
TGRewrite	2,9,2,9,1
PCRewrite	2,4000,2,4000,1001
Password	PASSWORD
Debug	0
Id	2328000
Name	DMR+ _IPSC2-OE-DMO
Options	StartRef=4198;RelinkTime=1

Speichern

Rys.6.1.4. Konfiguracja sieci od strony IPSC2. Dla prywatnych mikroprzemienników (anh. *hotspot*) przewidziane są serwery noszące w nazwie oznaczenie DMO. W IPSC2 nie są wymagane indywidualne hasła dostępu

Info	
Latitude	0.000
Longitude	0.000
Location	DVMEGA DSTAR
Description	Austria
URL	http://ham-dmr.at
RXFrequency	432612500
TXFrequency	432612500
Enabled	0
Power	1
Height	0

Speichern

Rys. 6.1.5. Bliższe informacje o stacji

XLX Network	
Enabled	0
File	/usr/local/etc/XLXHosts.txt
Port	62030
Password	passw0rd
ReloadTime	60
Slot	2
TG	6
Base	64000
Startup	950
Relink	60
Debug	0
Id	2328000

Speichern

Rys. 6.1.6. Konfiguracja dostępu do sieci reflektorów XLX

DMR Network 3	
Enabled	0
Name	HBLink
Address	1.2.3.4
Port	5555
TGRewrite	2,11,2,11,1
Password	PASSWORD
Location	0
Debug	0
TGRewrite0	2,11,2,11,1

Speichern

Rys. 6.1.7. Konfiguracja trzeciej sieci DMR – HBLinku

Pi-Star:3.4.16 / Dashboard:20180806

## Pi-Star Digital Voice - Expert Editors

[Tableau](#) | [Admin](#) | [Aktualisieren](#) | [Upgrade](#) | [Datensicherung/Wiederherstellung](#) | [Konfiguration](#)

**Quick Edit:** DStarRepeater | ircDDBGateway | TimeServer | MMDVMHost | DMR GW | YSF GW | P25 GW | NXDN GW

**Full Edit:** DMR GW | PiStar-Remote | WiFi | BM API | DAPNET API | System Cron | RSSI Dat

**Tools:** CSS Tool | SSH Access

Rys. 6.1.8. Dalsza modyfikacja pliku DMR GW

Następnie należy w linii „Full Edit” wybrać DMR GW i w sekcji [DMR Network 2] dopisać linie podane poniżej w kolorze czerwonym i podać własny identyfikator DMR:

[DMR Network 2]

Enabled=1

**Address=89.185.97.34**

Port=55555

TGRewrite0=2,1,2,1,7

TGRewrite1=2,10,2,10,80

TGRewrite2=2,100,2,100,100

TGRewrite3=2,232,2,232,1

TGRewrite4=2,8181,2,8181,9

TGRewrite5=2,8191,2,8191,9

TGRewrite6=2,9,2,9,1

**TGRewrite=2,9,2,9,1 jako ostatni parametr!!!**

PCRewrite0=2,5055,2,5055,6

PCRewrite1=2,4000,2,4000,1001

PassAllPC=2

Password=PASSWORD

Debug=0

Id=23280xxxx (własny identyfikator DMR)

Name=DMR+\_IPSC2-OE-DMO

Options="StartRef=4198;RelinkTime=120;UserLink=1;TS1\_1=110;TS1\_2=20;TS2\_3=232;TS2\_4=8189;TS2\_5=8191;TS2\_6=8184;"

Wprowadzone dane należy zapisać naciskając przycisk na ekranie.



Rys. 6.1.9. Otwarcie konfiguracji MMVMHosts

Spomiędzy dużej liczby parametrów, których większość może zachować wartości domyślne przytoczono jedynie wybrane grupy wymagające wprowadzenia indywidualnych danych.

General	
Callsign	OE8
Id	2328
Timeout	3600
Duplex	0
RFModeHang	300
NetModeHang	300
Display	OLED
Daemon	1
Speichern	

Rys. 6.1.10. W oknie należy wprowadzić własny znak wywoławczy, identyfikator DMR, skorygować ograniczenie czasowe *Timeout* i wybrać rodzaj wyświetlacza. *RFModehang* i *NetModeHang* należy ustawić na 300

Info	
RXFrequency	437900000
TXFrequency	437900000
Power	1
Latitude	0.000
Longitude	0.000
Height	0
Location	Jumbospot2
Description	Austria
URL	http://ham-dmr.at
Speichern	

Rys. 6.1.11. Dalszymi danymi indywidualnymi są częstotliwość pracy, moc nadajnika, lokalizacja i dalsze szczegóły

OLED	
Type	3
Brightness	0
Invert	0
Scroll	0
Rotate	0
Cast	0
LogoSaver	1

Speichern

Rys. 6.1.12. W sekcji OLED należy zmienić typ na 3 i LogoSaver na 1 jeżeli podłączony jest wyświetlacz OLED, np. w Jumbospocie

DMR	
Enable	1
Beacons	0
BeaconInterval	60
BeaconDuration	3
ColorCode	1
SelfOnly	0
EmbeddedLCOnly	0
DumpTADData	1
CallHang	3
TXHang	4
ModeHang	20
>>>>>> OVCM	0
Id	2328036

Speichern

Rys. 6.1.13. W sekcji DMR istotny może być bit OVCM (w przedostatniej linii). Jego ustawienie na zero pozwala na odbiór transmisji przez radiostacje Motoroli, natomiast jedynka uniemożliwia odbiór (przynajmniej przez radiostacje wyposażone w starsze wersje oprogramowania wewnętrznego). Drugim miejscem, w którym wartość bitu może ulegać zmianie są serwery Brandmeistera, gdzie może on przyjmować wartość jeden dla transmisji przekazywanych do sieci albo dla transmisji nadchodzących z sieci i nadawanych radiowo albo dla obu kierunków, albo też może być ustawiony na stałe na zero. Nowsze wersje oprogramowania wewnętrznego Motoroli reagują już prawidłowo, ale mogą nie być dostępne dla wszystkich modeli radiostacji

## 6.2. Oprogramowanie EA7EE

Do pracy w sieci C4FM lepiej korzystać z oprogramowania autorstwa EA7EE. Uwzględnia ono drugą wersję systemu *Fusion* (YCS System Fusion II). W poprzednio opisanej wersji oficjalnej można się wprawdzie połączyć z serwerem YCS, ale do dyspozycji jest jedynie grupa DG-ID własnego kraju. Wybór innej grupy jest niemożliwy. Niemożliwe jest też zaprenumerowanie na stałe innych grup DG-ID. Na wszystko to pozwala natomiast wersja EA7EE. Wersja ta jest dostępna pod adresem: [http://ref075.dstargateway.org/1/Pi-Star\\_RPi\\_V4.12\\_20-EA7EE-C4FM.zip](http://ref075.dstargateway.org/1/Pi-Star_RPi_V4.12_20-EA7EE-C4FM.zip). Oczywiście z biegiem czasu zmianie ulegnie numer wersji dlatego też powyższy należy traktować jako przykładowy.

W przypadku zastępowania dotychczas używanej wersji „Pi-Stara” przez hiszpańską zaleca się zrobienie kopii bezpieczeństwa dotychczasowej instalacji albo też zainstalowanie wersji EA7EE na innym module pamięci SD.

Po wybraniu punktu „Datensicherung/Wiederherstellen“ („Kopia bezpieczeństwa/Powrót do poprzedniego stanu”) można załadować dotychczasowe ustawienia i poddać je potrzebnym modyfikacjom. Po



wczytaniu dotychczasowej konfiguracji i jej aktualizacji należy też ręcznie zabezpieczyć konfigurację DMR, a zwłaszcza przyporządkowanie grup rozmówców do sieci BM lub IPSC2. Operatorzy, którzy dotąd nie korzystali z „Pi-Stara” mogą przeskoczyć te kroki.

Pi-Star:4.1.2 / Tableau: 20200918-ea7ee

## Pi-Star Digital Voice - Konfiguration

Admin ● Expert ● Strom Aktualisieren Datensicherung/Wiederherstellung Werkseinstellung

Tableau

Gateway Hardware Information				
Rechnername	Kernel	Plattform	CPU Last	CPU Temp
jumbo2	4.19.97+	Pi Zero W Rev 1.1 (512MB)	1m:1.09 / 5m:0.7 / 15m:0.42	42.2°C/108°F

Kontrollsoftware	
Einstellung	Wert
Kontroller Software:	<input type="radio"/> DStarRepeater <input checked="" type="radio"/> MMDVMHost (DV-Mega Minimum Firmware 3.07 Required)
Kontroller Mode:	<input checked="" type="radio"/> Simplex Node <input type="radio"/> Duplex Repeater (or Half-Duplex on Hotspots)

Speichern

Rys. 6.2.1. Strona konfiguracyjna Pi-Stara w wersji EA7EE

Dla przeprowadzenia konfiguracji należy wybrać tryb eksperta i w polu „Full Editors” („Nieograniczona edycja”) wybrać *DMR Gateway*. Wszystkie widoczne w oknie parametry należy zaznaczyć myszą i skopiować do nowego dokumentu w Wordzie albo w innym edytorze tekstów. Potem można nacisnąć na pozycję „Konfiguration” („Konfiguracja”)

### EA7EE Yaesu System Fusion Configuration

Einstellung	Wert
Startup Mode:	YSF ▼
YSF Startup Host:	YSF23201 - AT-C4FM-Austria - (YCS232) ▼
Startup DG-ID:	0
YCS Network:	Options=
APRS Enable:	<input type="checkbox"/>
APRS Callsign:	OE8V
aprs.fi ApiKey:	
Beacon Time:	
Re-Link Time:	0
UPPERCASE Hostfiles:	<input checked="" type="checkbox"/> Note: Update Required if changed
WiresX Passthrough:	<input checked="" type="checkbox"/>
Hotspot Follow User:	<input checked="" type="checkbox"/>
DMR Enable:	<input type="checkbox"/> Note: Update Required if changed
ESS DMR Id:	23280
DMR Master:	BM_Europe_2001 ▼
DMR Startup Host:	None ▼
Password:	*****
DMR+ Netzwerk:	Options=
FCS Enable:	<input checked="" type="checkbox"/> Note: Update Required if changed
FCS Startup Host:	FCS00118 - Spain ▼
FCS Network:	Options=

Speichern

Rys. 6.2.2. Konfiguracja ogólna dla uruchamiania w systemie YSF. W polu „YSF Startup Host” zamiast podanego w przykładzie austriackiego kółeczka należy wybrać kółeczko zgodne z upodobaniami i ewentualnie wpisać stałe używane grupy DG-ID oddzielone przecinkami ale bez odstępów. Należy włączyć funkcje zaznaczone na czerwono. Parametry dla systemu DMR są konfigurowane w innym miejscu. Można nie korzystać z reflektorów FCS gdyż są one coraz rzadziej używane

Po zapisaniu powyższych danych należy wybrać pozycję „Expert” i nieograniczoną edycję, a pod nią pozycję *DMR Gateway*.

Do okna trzeba wpisać uratowane poprzednio ręcznie ustawienia DMR z pliku w Wordzie po skasowaniu widocznych tam ustawień domyślnych tak aby okno było puste.

Późniejsze zmiany wybranych grup DG-ID są możliwe po wybraniu w :Uproszczonej edycji” („Quick Editors”) ustawień „YSFGateway”. Zmian grup dokonuje się w sekcji „YSF Network” w otoczonej na rysunku ciemniejszą obwódką polu „Options”.

YSF Network	
Enable	1
Port	42000
Hosts	/usr/local/etc/YSFHosts.txt
ReloadTime	60
ParrotAddress	127.0.0.1
ParrotPort	42012
YSF2DMRAddress	127.0.0.1
YSF2DMRPort	42013
YSF2NXDNAddress	127.0.0.1
YSF2NXDNPort	42014
YSF2P25Address	127.0.0.1
YSF2P25Port	42015
Startup	23201
Options	20,32,88,98
NXDNEnable	0
NXDNStartup	400
NXDNHosts	/usr/local/etc/TGList_NXDN.t
P25Enable	0
P25Startup	400
P25Hosts	/usr/local/etc/TGList_P25.txt
StartupDGID	0

Rys. 6.2.3. Zmiana stale używanych grup w polu „Options”

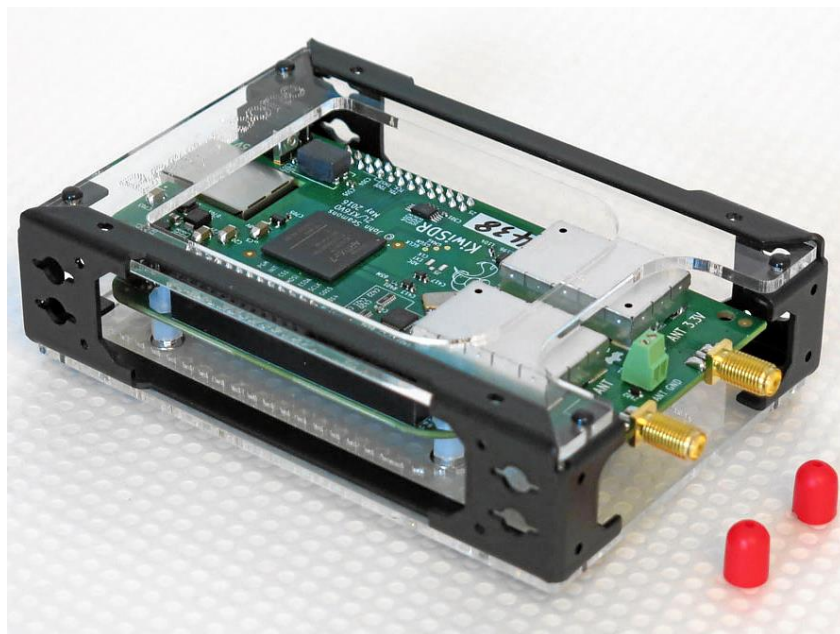
## 7. Odbiornik programowalny „Kiwi”

Odbiornik programowalny „KiwiSDR” ([www.kiwisdr.com/KiwiSDR](http://www.kiwisdr.com/KiwiSDR)) pokrywa zakres 10 kHz – 30 MHz i jest przystosowany do odbioru emisji AM (także z detekcją synchroniczną), wąskopasmowej AM, CW, SSB, i wąskopasmowej FM stosowanej w radiokomunikacji. Część odbiorcza jest połączona z mikrokomputerem *BeagleBone* podobnym do *Maliny*. Oprogramowanie zawiera serwer *http* „Open-WebRX”. Oznacza to, że na komputerach odbiorczych nie trzeba instalować żadnego dodatkowego oprogramowania i mogą one pracować pod dowolnym systemem operacyjnym. Wystarczy przeglądarka internetowa Chrome, FireFox, Safari albo Opera. Odbiornik można dzięki temu umieścić w miejscu dogodnym do odbioru i połączyć z siecią domową kablem ethernetowym LAN z modemem internetowym albo bezprzewodowo przez WiFi. W sieci lokalnej odbiornik jest dostępny pod adresem *kiwisdr.local:8073*. Zaależnie od konfiguracji modemu internetowego (ang. *router*) konieczne może być udostępnienie w nim kanału logicznego 8073 dla połączenia TCP. Na komputerach pracujących pod systeme iOS konieczne jest zainstalowanie usługi mDNS „Bonjour” z iTunes. W razie gdyby połączenie z serwerem odbiornika pod tą nazwą nie udało się można znaleźć jego adres w lokalnej sieci i podać go w polu adresowym przeglądarki.

Odbiornik pracuje z bezpośrednią przemianą analogowo-cyfrową i jest wyposażony w 14-bitowy przetwornik a-c o częstotliwości próbkowania 65 MHz. Posiada on dwa gniazda antenowe SMA, jedno dla anteny odbiorczej (z filtrem dolnoprzepustowym 30 MHz na wejściu), a drugie dla anteny GPS z wyjściem napięcia zasilania 3,3 V. W zestawie jest też odbiornik GPS i antena.

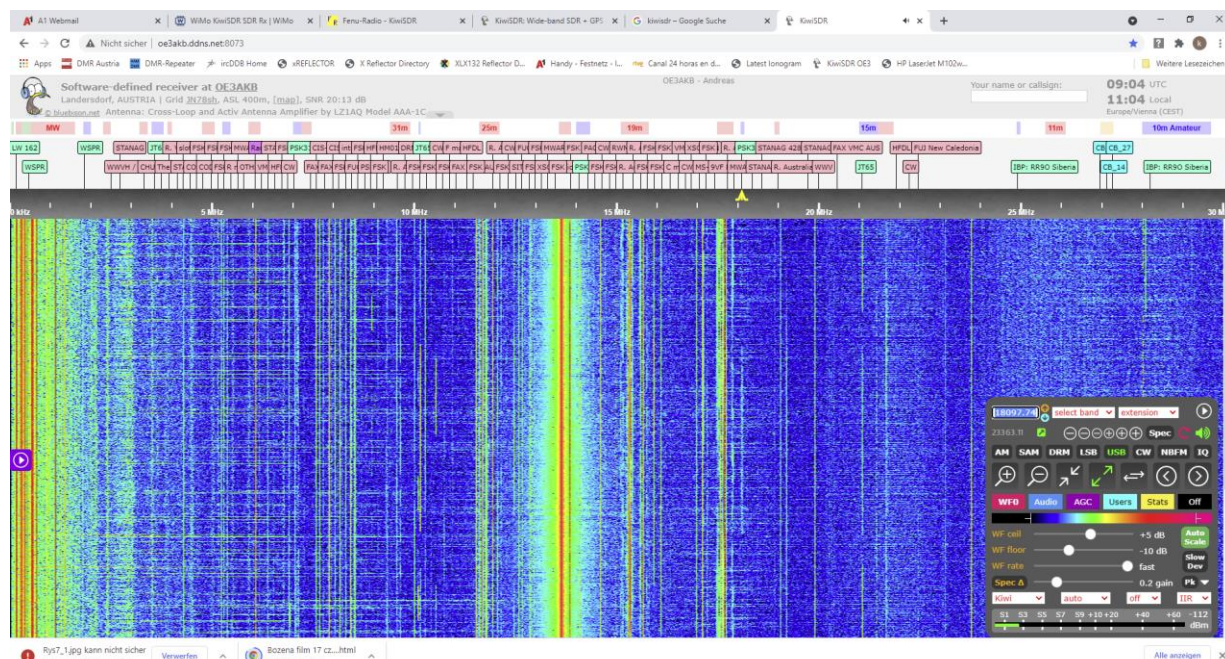
Serwer *http* pozwala na równoległy dostęp czterem użytkownikom i każdy z nich może korzystać z pełnego zakresu 30 MHz lub z dowolnie wybranego podzakresu. Dostrajanie do częstotliwości i wybór interesującego zakresu następuje na wskaźniku wodospadowym. Odbiornik można także udostępnić w Internecie ale wymaga to zarejestrowania go w światowej sieci odbiorników „KiwiSDR” w witrynie <http://sdr.hu/register>, gdzie dostępny jest też aktywny spis odbiorników czynnych na całym świecie. Korzystanie w zakresie prywatnym nie wymaga rejestracji.

Kod źródłowy programu i schemat odbiornika są ogólnie dostępne w Internecie pod adresem [kiwisdr.com/KiwiSDR](http://kiwisdr.com/KiwiSDR) a także pod [https://github.com/jks-prv/Beagle\\_SDR\\_GPS](https://github.com/jks-prv/Beagle_SDR_GPS). Publiczne udostępnienie odbiornika w Internecie wymaga skorzystania z usługi noip ([www.noip.com](http://www.noip.com)), dyndns lub podobnej. Program nie korzysta z monitora, klawiatury ani myszy podłączonych bezpośrednio do mikrokomputera.

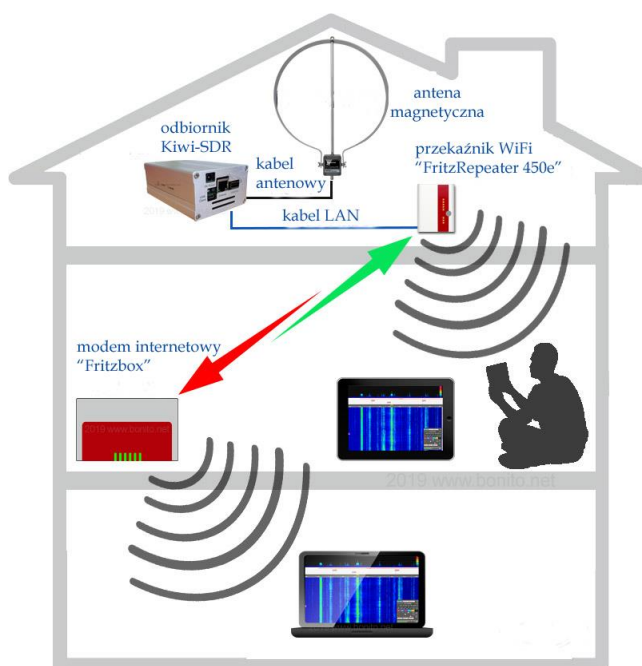


Fot. 7.1. KiwiSDR w obudowie

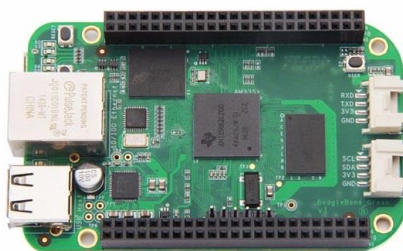




Rys. 7.2. Okno KiwiSDR w przeglądarce internetowej



Rys. 7.3. KiwiSDR w sieci domowej (źródło: bonito.net)



Fot. 7.4. Mikrokomputer BeagleBone

Mikrokomputer *BeagleBone* posiada procesor ARM335x AMR Cortex-A8 taktowany z częstotliwością 1 GHz, 512 MB pamięci roboczej DDR3 RAM, 4 GB pamięci programu eMMC, koprocessory graficzny

i matematyczny (NEON), gniazdo USB, gniazdo Ethernetu, boczne listwy kontaktowe GPIO (2 x 46 kontaktów), gniazda I2C i UART-u. Jako system operacyjny można zainstalować Linuks Ubuntu, Debian, Androida i inne. Na listwach kontaktowych znajdują się także wyprowadzenia SPI. Osiem spośród 65 wyjść logicznych może być używane jako wyjścia z modulacją szerokości impulsów. Mikrokomputer posiada 7 wejść analogowych doprowadzonych do wspólnego wejścia 12-bitowego przetwornika a-c.



## 8. Wielozakresowy odbiornik FT8 na „Red Pitayi”

Odbiornik opracowany przez PE3ES (*Funkamateur* 4/2019) pracuje na ośmiu wybranych podzakresach od 160 do 6 m. Jego oprogramowanie opiera się na dekodерze FT8 Pavla Demina i zawiera modyfikacje pozwalające na wykorzystanie dwóch anten i dwóch zestawów pasm: dziennego i nocnego. „Red Pitaya” ([www.redpitaya.com](http://www.redpitaya.com)) była wprawdzie początkowo przewidziana do celów pomiarowych, ale krótkofalowcy znaleźli możliwości wykorzystania jej w łącznościach radiowych. Szczegółowe omówienie „Pitayi” i jej wykorzystania do celów krótkofalarskich zawiera tom 44 „Biblioteki polskiego krótkofalowca”. Zmodyfikowana wersja Linuksa autorstwa Pavla Demina znajduje się w witrynie [www.alpinelinux.org](http://www.alpinelinux.org). Przy objętości około 100 MB mieści się ona bez trudu wraz programami użytkowymi na typowych modułach pamięciowych microSD 8 lub 16 GB. Po rozpakowaniu pliku należy skopiować go za pomocą Eksploratora Windows na sformatowany w systemie plików FAT32 moduł pamięci SD. Nie należy korzystać z programów kopiujących odwzorowania pamięci (obrazy) takich jak *Win32 Disk Imager*.



Fot. 8.1. Red Pitaya

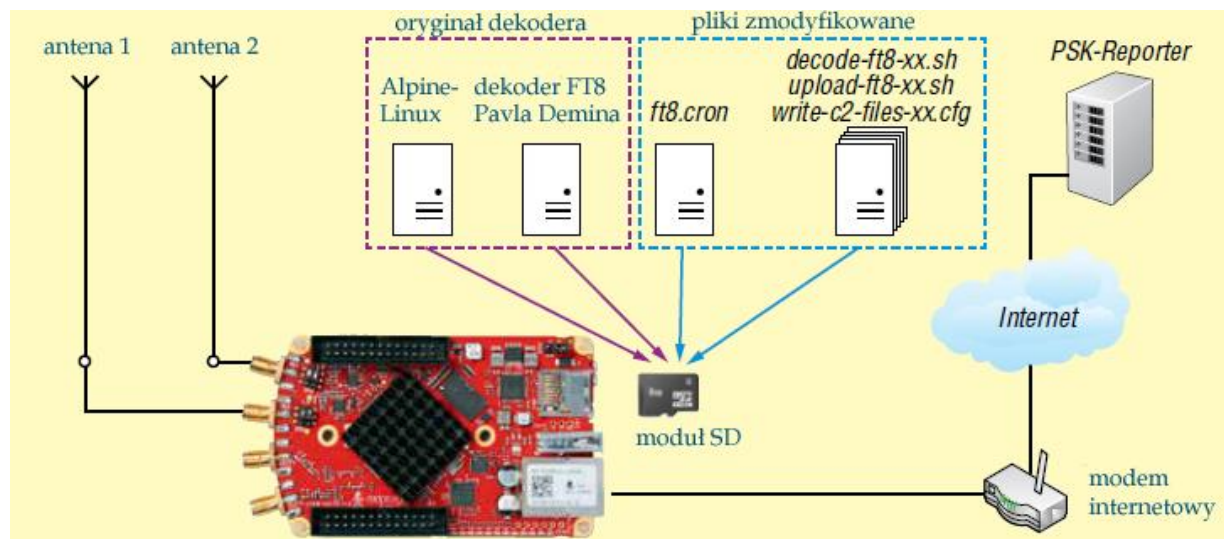
W celu automatycznego wywoływania programu po starcie systemu należy jego plik wsadowy *start.sh* skopiować z katalogu *apps/<nazwa programu>* do najwyższego katalogu w pamięci. Do obsługi systemu i wywoływania dowolnych programów służy graficzna powierzchnia obsługi dostępna przez HTTP. „Red Pitayę” można połączyć z domową siecią lokalną przez WiFi lub za pomocą kabla ethernetowego. Instrukcje do korzystania z *Alpinelinuxa* znajdują się pod adresem <http://pavel-demin.github.io/red-pitaya-notes>. Po wprowadzeniu w przeglądarce adresu IP „Pitayi” wyświetlane są punkty menu pozwalające na wywołanie odbiorników programowalnych, WSPR i FT8 oraz innych narzędzi. Po uruchomieniu programu odbiorczego FT8 (w menu jest on wprawdzie nazywany *transceiverem*, ale w rzeczywistości służy do odbioru) skrypt *ft8.cron* wywołuje dalsze skrypty: *decode-ft8.sh* i *upload-ft8.sh*. Dla sprawdzenia prawidłowej pracy programu można nawiązać z systemem połączenie SSH korzystając na PC z programu PuTTY. W jego oknie po podaniu polecenia *top* wyświetla się spis wszystkich uruchomionych procesów.

„Red Pitaya” dysponuje dwoma wejściami w.cz. co pozwala na podłączenie dwóch anten odbiorczych na różne pasma amatorskie. Ich przełączaniem steruje skrypt *upload-ft8.sh*. Dla zapewnienia obserwacji propagacji w dzień na pasmach wyższych, a w nocy na niższych konieczne jest uzupełnienie pliku *write-c2-files.cfg*. Zmodyfikowane przez PE3ES pliki są dostępne w witrynie [www.funkamateur.de](http://www.funkamateur.de) w obszarze pobrań („Downloads”) w części związanej z kolejnymi numerami pisma („Downloads zum Heft”) pod numerem kwietniowym 2019. Są to pliki *ft8.cron*, *decode-ft8-1D.sh*, *write-c2-files-1D.sh* i *upload-ft8-1.cfg* dla odbioru dziennego przez antenę 1, *ft8.cron*, *decode-ft8-1N.sh*, *write-c2-files-1N.sh* i *upload-ft8-1.cfg* dla odbioru nocnego przez antenę 1, *ft8.cron*, *decode-ft8-2D.sh*, *write-c2-files-2D.sh* i *upload-ft8-2.cfg* dla odbioru dziennego przez antenę 2 oraz *ft8.cron*, *decode-ft8-2N.sh*, *write-c2-files-*

`2N.sh` i `upload-ft8-2.cfg` dla odbioru nocnego przez antenę 2. Dla każdej z anten w pliku `upload-ft8` podany jest inny znak wywoławczy.

Najważniejsze adresy internetowe:

- <http://pavel-demin.github.io/red-pitaya-notes/sdr-transceiver-ft8>
- <http://pavel-demin.github.io/red-pitaya-notes/sdr-transceiver-wspr>



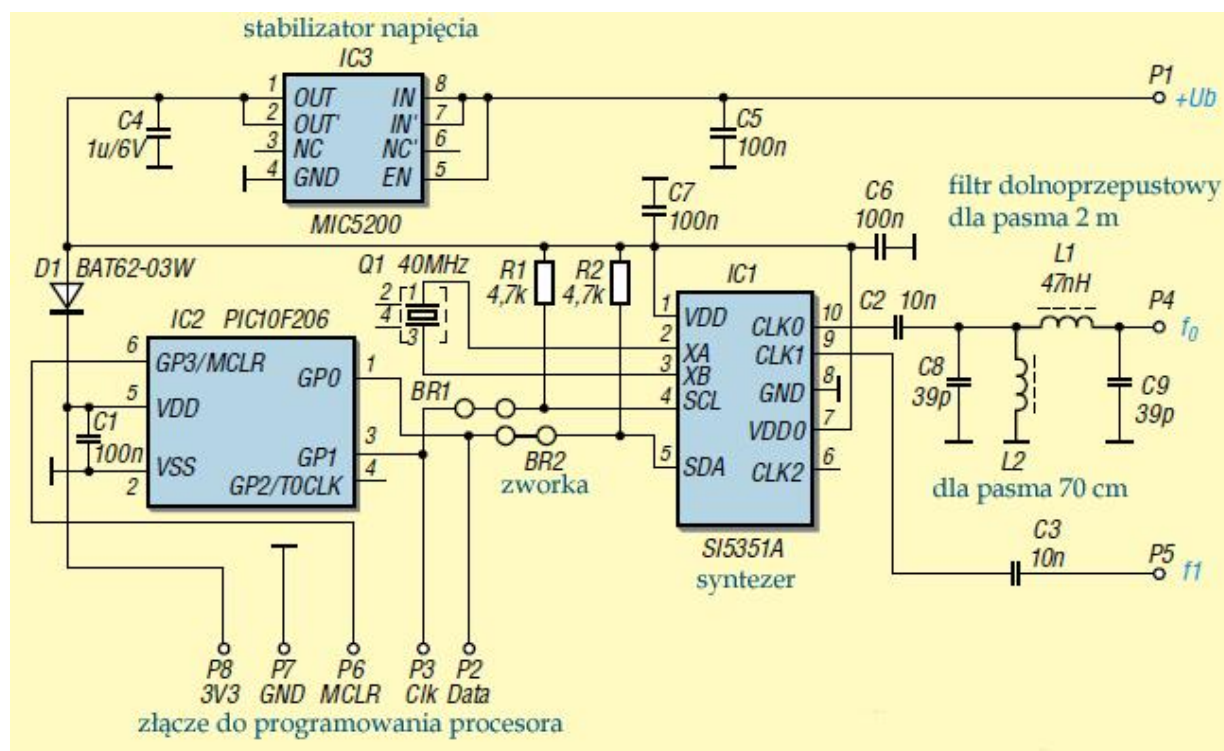
Rys. 8.2. Schemat blokowy współpracy „Red Pitayi” z dwoma antenami i serwerem PSK-Reporter w Internecie. Pliki w niebieskiej ramce są do pobrania z witryny *Funkamateura*, we fioletowej są częścią rozwiązania Pavla Demina

## 9. Nadajnik telegraficzny z PIC10F206

Konstrukcja nadajnika DC6HL została opisana w numerze 12/2018 Funkamateura. Jest to nadajnik małej mocy zawierający mikroprocesor PIC10F206 sterujący przez złącze I2C scalony syntezer Si5351A. Przy zasilaniu napięciem 3,5 – 15 V pobiera on prąd 22 mA i dostarcza do anteny mocy w.cz. 12 dBm. Poziom trzeciej harmonicznej wynosi 0 dBm, a więc konieczne jest stosowanie na wyjściu filtru dolnoprzepustowego. DC6HL eksperymentalnie korzystał z niego również na trzeciej harmonicznej w paśmie 70 cm po dodaniu cewki L2 do filtru dolnoprzepustowego. Niski pobór prądu pozwala na zasilanie nadajnika z akumulatora litowego albo nawet z baterii 9 V.

Nadajnik pokrywa zakres 313 kHz – 200 MHz, a więc można go stosować na wszystkich krótkofalowych pasmach amatorskich oraz na pasmach 6, 4 i 2 m. Częstotliwość kwarcu Q1 generatora sterującego syntezerem może leżeć w granicach 10 – 40 MHz, z tym że typowo są to częstotliwości 25 lub 27 MHz. Nadajnik transmituje telegrafią zakodowaną w mikroprocesorze komunikat i dzięki temu może służyć jako „lis” dla radiopelengacji amatorskiej albo jako radiolatarnia małej mocy do różnych eksperymentów. Oprócz tego mikroprocesor wpisuje do syntezeru ustawienia za każdym razem po włączeniu zasilania.

Syntezer zawiera dwa synchronizowane fazowo generatory VCO (PLLA i PLLB) pracujące w zakresie 600 – 900 MHz i programowalne dzielniki pozwalające na uzyskanie ułamkowych stosunków podziału i w ten sposób pożądaných częstotliwości wyjściowych. W układzie używane jest wyjście CLK0 syntezeru, a sygnał CLK1 może służyć do dowolnych innych celów (pomiarowych, pomocniczych itp.) Układ scalony IC3 dostarcza stabilizowanego napięcia 3,3 V przy napięciu zasilania 3,5 – 15 V.



Rys. 9.1. Schemat ideowy nadajnika z filtrem dolnoprzepustowym dla pasma 2 m

Dioda D1 zabezpiecza resztę układu przed dostaniem się wyższego napięcia w czasie programowania mikroprocesora. Może być to dioda Schottkiego dowolnego typu.

Częstotliwość zegarowa mikroprocesora jest generowana przez wewnętrzny generator RC i wynosi 4 MHz, a w generatorze wzorcowym syntezeru użyto nietypowo kwarcu 40 MHz. Mikroprocesor posiada 512 bajtów pamięci programu i 24 bajty pamięci roboczej i dlatego parametry konfiguracyjne dla syntezeru są zawczasu obliczone na PC w trakcie tworzenia pliku w formacie Intel-.HEX do zaprogramowania procesora. Program konfiguracyjny *Bakensender.exe* dla PC znajduje się w obszarze pobrań w witrynie [www.funkamateur.de](http://www.funkamateur.de) wśród plików dla numeru 12/2018 („Downloads zum Heft”). W oknie programu należy w odpowiednich polach wpisać częstotliwość pracy radiolatarni, współczyn-

nik kreacji częstotliwości i tekst komunikatu, nacisnąć przycisk „Morse Code berechnen” („Przekształć na alfabet Morse’a”) a następnie nacisnąć przycisk ekranowy „Save HEX” („Zapisz plik wyjściowy”). Do zaprogramowania mikroprocesora służy programator *PicstartPlus* albo *PICkit3* wraz z oprogramowaniem MPLAB 8. *PICkit3* pozwala na zaprogramowanie procesora bezpośrednio w układzie przez widoczne na schemacie złącze.

Tabela 9.1

Elementy filtru dolnoprzepustowego dla niektórych pasm amatorskich

Pasmo	C1	C8	L2	C9	L1	Uwagi
70 cm	2,7 pF	8,2 pF	10 nH	—	2,7 pF	Kondensator zamiast L1
2 m	10 nF	39 pF	—	39 pF	47 nH	
10 m	10 nF	150 pF	—	150 pF	270 nH	
80 m	10 nF	1 nF	—	1 nF	2,2 $\mu$ H	

Tabela 9.2

Wykaz elementów

Element	Wartość	Element	Wartość
C1, C5, C6, C7	100 nF	Q1	40 MHz
C2, C3	10 nF	R1, R2	4,7 k $\Omega$
C4	1 $\mu$ F/6 V		
D1	BAT62-03W		
IC1	Si5351A		
IC2	10F206		
IC3	MIC5200-3,3YM		

## Dodatek A

### Elementy q w trasach APRS

Elementy q podawane w trasach pakietów APRS wyświetlanych przez serwery APRS-IS, na tle map w witrynach *aprs.fi* albo *aprsdirect.com* służą do identyfikacji wejścia do sieci APRS-IS, do wykrywania pętli w trasach pakietów i do ich eliminacji, zapewniają kompatybilność z oprogramowaniem bramek radiowo-internetowych i programów klientów.

Obecnie stosowane są następujące elementy q:

Podawane przez serwer:

- qAC – pakiet został odebrany bezpośrednio od klienta przez zaufane (bezpieczne) łącze (FROMCALL = zameldowanie, ang. *login*). Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem serwera,

- qAX – pakiet został odebrany bezpośrednio od klienta, ale przez łącze niesprawdzone (FROMCALL = zameldowanie). Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem serwera.

Element ten jest dodatkiem do elementu TCPIP lub TCPXX,

- qAU – pakiet został odebrany bezpośrednio od klienta w protokole UDP. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem serwera,

- qAo – pakiet został odebrany w kanale logicznym przeznaczonym wyłącznie dla klienta i pole FROMCALL nie odpowiada zameldowanej stacji. Pakiet zawiera albo element I albo qAR, a wskazana bramka internetowa IGate jest zgodna z zameldowaną stacją,

- qAO – pakiet został odebrany w kanale logicznym przeznaczonym wyłącznie dla klienta i pole FROMCALL nie odpowiada zameldowanej stacji,

- qAS – pakiet został odebrany od innego serwera lub został utworzony na tym serwerze. W tym ostatnim przypadku jest to pakiet radiolatarni serwera (nie jest to zalecane). Następujący po nim znak wywoławczy z rozszerzeniem odpowiada zameldowanej stacji lub adresowi IP najbliższego zidentyfikowanego serwera,

- qAr – pakiet wraz z elementem I został odebrany od bramki IGate za pośrednictwem innego serwera. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem bramki radiowo-internetowej,

- qAR – pakiet wraz z elementem I został odebrany od bramki IGate bezpośrednio albo poprzez sprawdzone, godne zaufania łącze. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem bramki radiowo-internetowej.

Podawane przez program klienta:

- qAR – pakiet pochodzący z kanału radiowego jest podawany do sieci APRS-IS przez bramkę radiowo-internetową. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem bramki radiowo-internetowej,

- qAO – pakiet odebrany radiowo jest podawany do sieci APRS-IS przez czysto odbiorczą bramkę radiowo-internetową. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem bramki radiowo-internetowej. Nie zalecane jest instalowanie bramek wyłącznie odbiorczych na standardowych częstotliwościach radiowych APRS (nie są one widoczne dla użytkowników radiowych), a w przypadku bramki nadawczo-odbiorczej należy pakiety otrzymane z sieci nadawać radiowo tylko do bezpośrednio odbieranych stacji,

- qAZ – pakiet pochodzi z serwera albo z bramki radiowo-internetowej i nie powinien być przekazywany dalej. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem bramki radiowo-internetowej lub serwera,

- qAI – pakiet śledzący trasę (ang. *trace*), każdy z serwerów na jego trasie dodaje swoje dane do niego. Następujący po nim znak wywoławczy z rozszerzeniem jest znakiem stacji źródłowej pakietu.

Serwery wymagają jednoznacznego zameldowania ze strony bramek radiowo-internetowych i innych serwerów. Zapobiega to omyłkowemu rozpoznawaniu pętli.

Elementy q powinny być używane jedynie po stronie APRS-IS aby nie obciążać niepotrzebnie kanałów radiowych.

#### Przykłady

Przykład dla pakietu bez elementu q docierającego do systemu przez zaufane łącze:

```
OE1KDA>APRS,TCPIP*:zawartość
```

Pakiet opuszczający serwer przy wyłączonym śledzeniu:

```
OE1KDA>APRS,TCPIP*,qAC,OE1KDA-JS:zawartość
```



Pakiet opuszczający serwer przy włączonym śledzeniu:

OE1KDA>APRS,TCPIP\*,qAI,OE1KDA,OE1KDA-JS:zawartość

Przykład dla pakietu odebranego przez radio i podawanego do APRS-IS:

OE1KDA>APRS,WIDE1\*:zawartość

Pakiet opuszczający serwer przy wyłączonym śledzeniu:

OE1KDA>APRS,WIDE1\*,qAR,OE1KDA-11:zawartość

Pakiet opuszczający serwer przy włączonym śledzeniu:

OE1KDA>APRS,WIDE1\*,qAR,OE1KDA-11OE1KDA-JS:zawartość.



Rys. 1.1. Dane dla stacji OE1KDA-11 odbieranej przez bramkę systemu Lora na mapie *aprs.fi*

## Dodatek B

### Program radiolatarni z Arduino i Si5351

```
// -----
// program arduino_wspr_10m.ino
// dl8pb, jan 2016
//
// Przykład najprostszej realizacji transmisji WSPR
// Używana jest tabela symboli wygenerowana na PC przez program WSPRcode.exe
// i zapisana w pliku tekstowym, a z niego przejęta do programu,
// należy tylko oddzielić symbole przecinkami.
// Program jest przeznaczony do celów dydaktycznych.
// Transmisja zaczyna się po naciśnięciu przycisku przy obserwacji zegara
// Początek o prarzystych minutach.
// Nadajnik automatyczny pod http://www.qrp-labs.com
//
//
// Program korzysta z
// biblioteki Etherkit Si5351 autorstwa Jasona Milldruma, NT7S
// biblioteki wire.h arduino
//
// Wyposażenie
// Arduino UNO
// Płytki Ultimate Arduino; http://www.qrp-labs.com/uarduino.html
// LED + 2,7 k na wyjściu logicznym 13 podłączona do masy, sygn. transmisji
// Przycisk z wypr. 12 zwieranie do masy na początek transmisji
// Zegar GPS do obserwowania: http://www.qrp-labs.com/clockkit.html
//
// Podziękowania także dla Hansa, G0UPL!!
// -----

#include <si5351.h> // biblioteka eatherkit Si5351
#include <Wire.h> // biblioteka wire (i2c)

#define MULT 1ULL // wyłącznie do celów demonstracji

#define BAND 281260000ULL // cHz , różnica w stosunku do 28 124 600
Hz, dla systemu dźwiękowego
#define SHIFT 10000ULL // cHz , przesunięcie w stos. okna wspr 200
Hz, tutaj 100Hz
#define SYMB0 219ULL // cHz
#define SYMB1 73ULL // cHz
#define SYMB2 73ULL // cHz
#define SYMB3 219ULL // cHz
#define DURATION 683ULL // ms

// tabela symboli utworzona przez WSPRcode.exe
// tutaj komunikat: "DL0RI JO40 23"
// -----
byte symbols[162] = {
 1,3,2,2,2,2,0,2,1,0,0,2,3,3,1,2,2,2,3,0,2,3,2,1,1,1,3,0,0,2,\
 2,2,2,2,1,0,2,3,0,3,2,0,2,2,0,2,1,0,1,1,0,2,1,3,2,1,0,2,0,3,\
 3,0,1,2,2,0,2,3,1,0,1,2,1,0,3,0,3,2,0,3,2,2,1,2,1,3,2,2,2,1,\
 1,0,1,2,3,0,2,2,3,0,2,2,0,2,3,2,0,1,0,0,1,3,3,0,3,3,0,2,1,1,\
 2,3,0,0,0,3,1,3,0,0,2,0,0,3,2,1,2,0,1,1,2,0,2,0,0,2,0,3,3,0,\
 3,2,3,1,2,2,2,1,1,0,0,2
};

// -----

const unsigned long long f0 = BAND + SHIFT - MULT * SYMB0;
```

```

const unsigned long long f1 = BAND + SHIFT - MULT * SYMB1;
const unsigned long long f2 = BAND + SHIFT + MULT * SYMB2;
const unsigned long long f3 = BAND + SHIFT + MULT * SYMB3;
unsigned long long frequency[4] = {f0,f1,f2,f3}; // tabela częstotl. WSPR

Si5351 si5351; // obiekt typu si5351

// -----funkcja setup() -----

void setup() {

 pinMode(13, OUTPUT); // LED, nadawanie
 pinMode(12, INPUT_PULLUP); // przycisk startu
 digitalWrite(13, HIGH); // dioda świeci

 // si5351 crystal setup
 si5351.init(SI5351_CRYSTAL_LOAD_8PF, 27000000ULL); // kwarc 27 MHz
 // output strength >> 2 mA -> 7 dBm and 8 mA -> 14 dBm RF power.
 si5351.drive_strength(SI5351_CLK0, SI5351_DRIVE_8MA);
 // clk0 off
 si5351.output_enable(SI5351_CLK0, 0);

}

// -----funkcja loop() -----
void loop() {

 if (!digitalRead(12)){ // oczekiwanie na sygnał startu na wejściu 12
 wspr_send_10m();
 }

}

// ----- funkcja wspr_send_10m() -----
void wspr_send_10m(void) {

 // włączenie CLK0
 si5351.output_enable(SI5351_CLK0, 1);
 for (byte i=0; i< 162; i++){
 blinker(); // do testów, nadawanie
 // si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA); ustawienie CLK0 na
 // nadawanie ze stałą częstotliwością PLL
 si5351.set_freq(frequency[symbols[i]], 0ULL, SI5351_CLK0); //
 // nadawanie na częstotliwości symbolu
 delay(DURATION);
 } // for

 // clk0 OFF
 si5351.output_enable(SI5351_CLK0, 0);
 digitalWrite(13, HIGH); // do testów, gotowość do transmisji
} // wspr_send_10m

// ----- funkcja blinker() -----
void blinker(void){ // tylko do testów, nadawanie
 digitalWrite(13,!digitalRead(13));
}

```

## Dodatek C

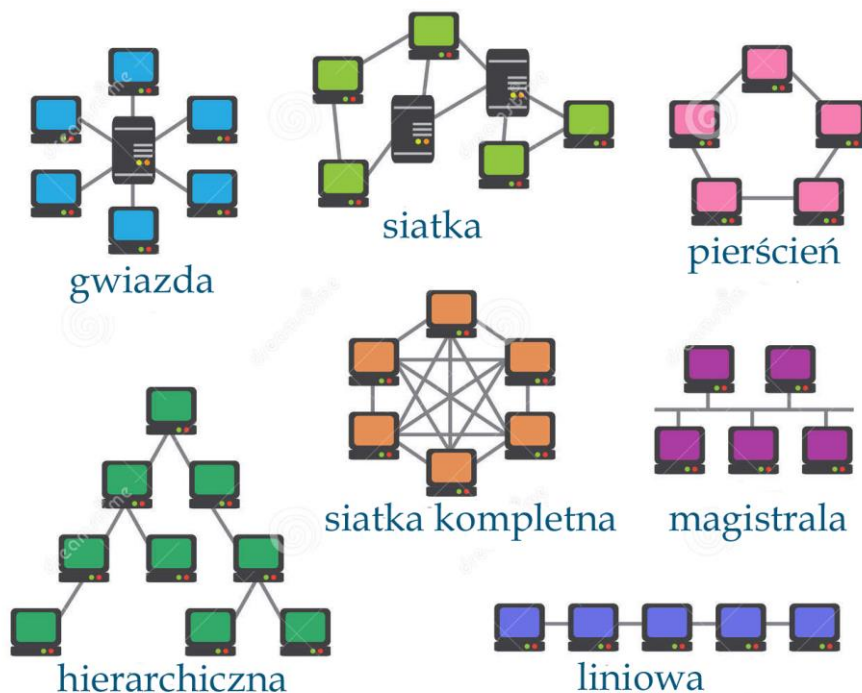
### Połączenia w sieciach

Sieć komputerowa jest systemem wzajemnie powiązanych stacji roboczych, urządzeń peryferyjnych i innych urządzeń albo też mówiąc inaczej zbiorem komputerów połączonych siecią komunikacyjną. Jej celem jest zapewnienie bezpiecznej efektywnej i swobodnej komunikacji. W miarę rozwoju sieci komputerowych konieczne było opracowanie standardów zapewniających kompatybilność sprzętową i programową. Rozróżniane są następujące najważniejsze typy sieci: sieci obszerne WAN (ang. *Wide Area Network*) łączące ośrodki na dużych odległościach, sieci obejmujące duże ośrodki miejskie MAN (ang. *Metropolitan Area Network*), lokalne sieci w ramach dużych ośrodków o urządzeniach zlokalizowanych w odległościach do kilkuset metrów LAN (ang. *Local Area Network*) oraz sieci urządzeń znajdujących się w nieznacznych odległościach (do kilkunastu lub kilkudziesięciu metrów) PAN (ang. *Private Area Network*). Popularnie na określenie tych ostatnich jest używana również nazwa LAN.

Ze względu na typ transmisji sieci można podzielić na następujące rodzaje:

- sieć kolizyjna, w której poszczególne węzły przed rozpoczęciem nadawania sprawdzają czy kanał (linia) jest wolny i rozpoczynają nadawanie. Możliwe jest jednak wystąpienie kolizji pakietów. Przykładami mogą być Ethernet (802.3), AX.25 lub WiFi (802.11).
- sieć typu krążącego żetonu – buławy – (ang. *Token Ring*), w której węzły otrzymują zezwolenie na nadawanie po odebraniu żetonu (buławy) od czynnego poprzednio. Po zakończeniu transmisji żeton jest przekazywany dalej.
- sieć z wykorzystaniem szczelin czasowych (ang. *slot*), w której każde urządzenie ma przydzielony czas nadawania. Przykładami mogą być sieci GSM, WiMAX (802.16).

Topologia sieci oznacza jej fizyczną konstrukcję i sposób połączenia poszczególnych urządzeń.

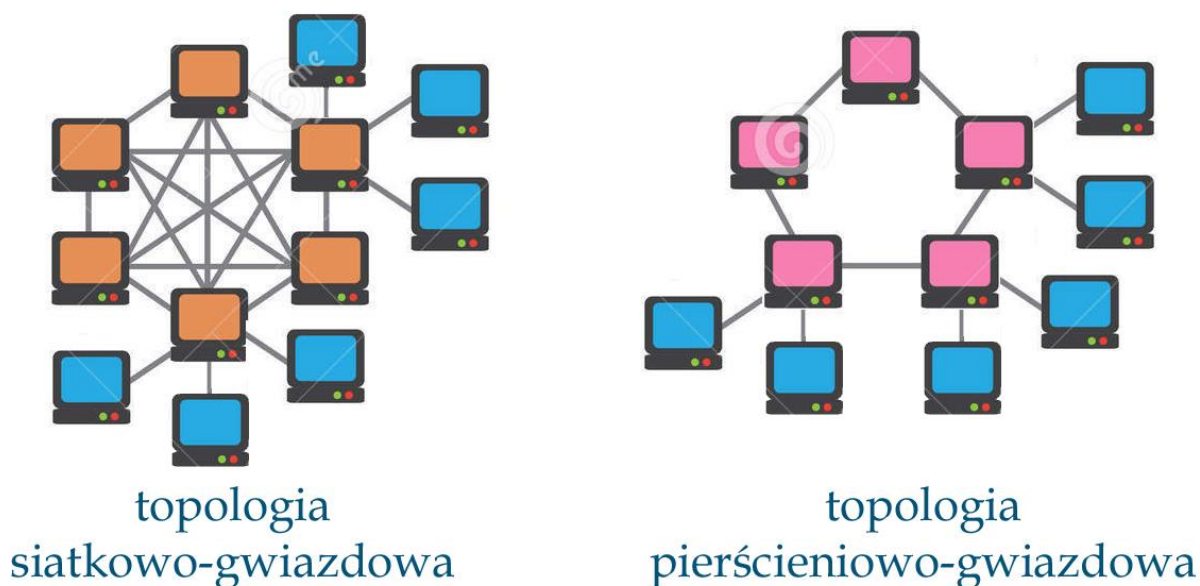


Rys. C.1. Podstawowe topologie sieci komputerowych. W topologii gwiazdy rozgałęzionej węzły końcowe gwiazdy zwykłej są środkami gwiazd podporządkowanych lub węzłem początkowym sieci hierarchicznej, węzły w topologii siatki lub pierścienia mogą być również środkami podporządkowanych gwiazd lub sieci hierarchicznych. Można więc łatwo wyobrazić sobie szereg topologii kombinowanych (źródło: *fr.dreanstime.com*)

Rozróżniane są następujące topologie:

- magistrala, w której wszystkie urządzenia są podłączone do jednego wspólnego kanału fizycznego, zwykle kabla;

- pierścień, w którym wszystkie urządzenia są połączone z dwoma sąsiadami i całość tworzy zamknięty krąg; w transmisji węzły przekazują sobie kolejno żeton (buławę) i każdy z nich pełni też funkcję regeneratora sygnału;
- podwójny pierścień, różni się od rozwiązania poprzedniego jedynie tym, że urządzenia są połączone podwójnymi łączami; awaria jednego urządzenia nie przerywa działania sieci; topologia jest stosowana m.in. w budowie sieci szkieletowych i miejskich;
- gwiazda, jest topologia podstawową w sieciach komputerowych i polega na tym, że wszystkie urządzenia są połączone ze sobą w jednym wspólnym punkcie, którym może być serwer nadrzędny; jest ona spotykana m.in. w krajowych sieciach DMR gdzie stacje przemiennikowe lub użytkownik są podłączone do krajowego serwera (ang. *master*) BM albo IPSC2, dla prywatnych mikroprzemienników (ang. *hotspot*) zainstalowane są oddzielne serwery; topologia gwiazdowa jest stosowana również w sieci Hamnetu dla stacji skupionych wokół lokalnego węzła (przemiennika; ang. *node*); awaria punktu centralnego (koncentratora) może sparaliżować całą sieć; awaria każdego z pozostałych węzłów nie wpływa na resztę sieci;
- rozszerzona gwiazda, rozgałęziona gwiazda, topologia rozszerzonej gwiazdy polega na połączeniu poszczególnych sieci gwiazdowych ze wspólnym punktem centralnym w gwiazdę nadrzędną.
- hierarchiczna, topologia podobna do rozszerzonej gwiazdy, ewentualnie z większą liczbą stopni hierarchii; każdy z punktów centralnych steruje dostępem do sieci dla urządzeń podległych;
- siatka, w typowej dla sieci miejskich lub rozległych topologii siatki każde z urządzeń jest połączone z więcej niż jednym urządzeniem, w przypadkach szczególnych nawet ze wszystkimi pozostałymi, zapewnia to redundancję połączeń i wysoką odporność na awarie łącz i urządzeń; topologia spotykana m.in. w połączeniach między serwerami sieci DMR albo w Hamnecie na obszarach pozostających poza zasięgiem stacji węzłowych.



Rys. C.2. Przykłady topologii kombinowanych

Wymianę danych w sieciach regulują protokoły transmisji. Są to zbiory reguł definiujących procesy komunikacji urządzeń zarówno pomiędzy warstwami równoległymi w modelu komunikacji jak i pomiędzy warstwami sąsiadującymi. Są to więc swego rodzaju kodeksy drogowe dla danych. Określają one budowę sieci fizycznej, sposoby łączenia komputerów z siecią, sposoby formatowania danych do transmisji, sposoby ich wysyłania i sposoby reakcji na błędy i przekłamania. W związku z rozpowszechnieniem Internetu do najczęściej stosowanych należy grupa protokołów TCP/IP.



## C1. Protokoły TCP/IP

Grupa protokołów TCP/IP jest podzielona na warstwy podobnie jak w przypadku modelu komunikacyjnego OSI (ISO). Mimo opracowania jej niezależnie od wspomnianego modelu możliwe jest łatwe przyporządkowanie poszczególnych protokołów odpowiednim warstwom OSI (ISO) lub ich grupom (tabela C.1.1) Organizacja warstwowa oznacza, że poszczególne warstwy (usługi, moduły programów) wyższe komunikują się z warstwami podległymi przekazując rozkazy i odbierając dane, a każda z warstw jest odpowiedzialna za pewien zespół zadań, jak przekodowywanie i grupowanie danych, wybór trasy połączenia (trasowanie), transmisja danych na łączu fizycznym lub sprawdzanie jej poprawności. Każda z warstw, to jest każdy z protokołów otrzymuje od warstwy niższej informacje we właściwym dla niej formacie i porządku, a wszelkie akcje podejmowane przez warstwy niższe są dla niej niewidoczne (ukryte). Umożliwia to wymiennosc protokołów w ramach systemu i udoskonalanie ich działania.

Warstwowy model rodziny protokołów TCP/IP ogranicza się do (wyróżnionych w tabeli kolorami) warstw: zastosowań (aplikacji) – odpowiadającej warstwom 5–7, warstwy transportowej odpowiadającej warstwie 4, warstwy internetowej odpowiadającej warstwie 3 i warstwy dostępu do sieci równoważnej warstwom 1–2. Model ten jest obecnie podstawowym modelem komunikacji w sieciach lokalnych i w Internecie.

W sieciach TCP/IP dane są transmitowane w postaci datagramów, czyli bloków zawierających oprócz informacji użytecznej także wiadomości służbowe pozwalające na prawidłową transmisję bloku przez sieć do adresata i ponowne złożenie ich w całość. W zależności od sposobów transmisji datagramy mogą być dzielone na mniejsze jednostki – pakiety AX.25 lub ethernetowe – i uzupełniane o dodatkowe informacje niezbędne dla ich prawidłowego przekazania. Informacje te, zawarte w nagłówkach pakietów są następnie usuwane po stronie odbiorczej, a pakiety są składane w datagramy. Informacje użytkowe i administracyjne przekazane przez warstwy wyższe nie ulegają zmianom i są dostępne dla odpowiednich warstw po stronie przeciwnej.

Tabela C.1.1

Układ warstw TCP/IP na tle modelu OSI

Model OSI (ISO)	Protokoły TCP/IP			
7. Warstwa zastosowań	Warstwa zastosowań			
6. Warstwa prezentacji	Telnet, RLOGIN, RSH, REXEC	DNS (ang. <i>Dynamic Name System</i> ), DHCP (ang. <i>Dynamic Host Configuration Protocol</i> )	HTTP, HTTPS	FTP (ang. <i>File Transfer Protocol</i> ), SMTP (ang. <i>Simple Mail Transfer Protocol</i> ), POP3 (ang. <i>Post Office Protocol</i> )
5. Warstwa posiedzenia (sesji)				
4. Warstwa transportu	Warstwa transportu			
	TCP	SCTP	UDP	UDP-Lite
3. Warstwa sieciowa3	Wartswwa internetu			
	IP, IPsec	ARP	ICMP	
2. Warstwa prze- sła (łącza danych)	Warstwa dostępu do sieci			
1. Warstwa fizyczna	AX.25	X.25, Ethernet (IEEE 802.3), IEEE802.11 (WLAN)	PPP, MAC (ang. <i>Media Access Protocol</i> )	SLIP, CSLIP

Uwagi:

Siedmiowarstwowy model OSI (ISO/IEC7498-1:1994) można uzupełnić o warstwę ósmą do której zalicza się użytkowników i urządzenia wykorzystywane bezpośrednio przez nich, lub też w innym ujęciu o warstwę finansową, a do warstwy 9 (politycznej) zalicza się m.in. ogólną sytuację polityczną – Warstwa zastosowań jest także nazywana warstwą aplikacji.

Transmitowany strumień danych – stanowiący zawartość wybranego pliku lub dokumentu i przekazany przez protokoły wyższych warstw – jest dzielony przez protokół TCP (ang. *Transmission Control Protocol*) na segmenty (datagramy) o określonej długości (np. 500 oktetów; często stosowane są długości 1500, 1000 lub 576 oktetów, długość ta może być również negocjowana w momencie nawiązania połączenia, przy czym każdy z datagramów jest uzupełniany o nagłówek zawierający numer kanału logicznego (ang. *port*) nadawcy i adresata, numer potwierdzenia i sumę kontrolną. Długość nagłówka wynosi dwadzieścia oktetów. Jednostką tutaj używaną jest oktet mający długość ośmiu bitów podobnie jak bajt, jednak dla uniknięcia nieporozumień we współpracy komputerów o różnej szerokości słowa i dla podkreślenia uniwersalności systemu nie stosuje się terminu bajt. Datagramy odebrane prawidłowo są kwitowane przez odbiorcę (a raczej przez protokół TCP po jego stronie), odebrane błędnie lub nieodebrane (niepokwitowane) są powtarzane po pewnym czasie. Dla przyspieszenia transmisji dopuszczalne jest nadawanie dalszych datagramów jeszcze przed otrzymaniem pokwitowania, pod warunkiem nieprzekroczenia dopuszczalnej liczby nie pokwitowanych datagramów. Liczba ta nie jest stała, lecz zależna od pojemności i zapelnienia buforów po stronie odbiorczej i jest każdorazowo sygnalizowana nadawcy. Wybór długości datagramu wpływa na efektywną szybkość transmisji, ponieważ krótsze datagramy zawierają procentowo więcej danych administracyjnych i wymagają częstszych pokwitowań, jednocześnie dłuższe datagramy są bardziej narażone na przekłamania w wyniku zakłóceń.

Tak utworzone datagramy protokół TCP przekazuje warstwie trzeciej – protokołowi IP (ang. *Internet Protocol*) z podaniem adresu numerycznego stacji docelowej. Zadaniem protokołu IP jest wybór odpowiedniej trasy transmisji i nie musi on analizować treści otrzymanego datagramu. Protokół IP umieszcza na początku każdego datagramu swój własny nagłówek zawierający adresy internetowe nadawcy i adresata, informację o protokole warstwy wyższej (nie musi być to jedynie protokół TCP) oraz własną sumę kontrolną dla sprawdzania poprawności odbieranego nagłówka. Nagłówek IP zawiera między innymi pole określające maksymalną liczbę retransmisji datagramu w sieci i zmniejszaną o jeden po każdej retransmisji. Zapobiega to nieograniczonemu krążeniu datagramu w przypadku zapętlenia się tras łączności, czego w bardziej rozbudowanych sieciach nie sposób uniknąć. Pole to nazwane jest TTL (ang. *Time To Live*). Datagramy o zerowym stanie licznika TTL są usuwane z sieci – tzn. nie są dalej retransmitowane.

Wybór tras retransmisji jest dokonywany na bieżąco, tzn. każda ze stacji węzłowych na trasie rozstrzyga po otrzymaniu datagramu i po przeanalizowaniu adresu docelowego o tym, do której stacji sąsiadujących (logicznie) należy go skierować. Decyzja jest podejmowana częściowo w oparciu o wymieniane automatycznie informacje o jakości i obciążeniu łączy. Oznacza to, że trasa połączenia może ulegać wielokrotnym i niezauważalnym przez użytkownika zmianom w trakcie trwania sesji łączności. Jedną z konsekwencji tego faktu jest przemieszczanie kolejności datagramów u odbiorcy (przykładowo datagramy nadane później docierają korzystniejszą trasą przed niektórymi nadanymi wcześniej). Uporządkowanie kolejności datagramów należy do zadań protokołu TCP.

Warstwa IP przekazuje datagramy uzupełnione o swoje nagłówki warstwie drugiej. W zależności od stosowanego w sieci sposobu transmisji danych warstwa druga może odpowiadać protokołowi ethernetowemu (X.25), AX.25 lub innym. Warstwa ta stosuje własny system adresów różny od adresów internetowych. W przypadku Ethernetu są to 48-bitowe adresy znormalizowane w skali światowej, a w przypadku protokołu AX.25 są to również jednoznaczne znaki wywoławcze stacji. Protokół warstwy drugiej uzupełnia nadawane datagramy o swój własny nagłówek zawierający specyficzne adresy sprzętowe (np. znaki wywoławcze stacji) i kod warstwy wyższej oznaczający używany przez nią protokół oraz ewentualnie typ pakietu. Na końcu pakietu dołączana jest suma kontrolna. W zależności od wybranych parametrów datagramy są dzielone po stronie nadawczej na pakiety o mniejszej długości i składane z tych pakietów u adresata.

Po stronie adresata pakiety, a następnie złożone z ich zawartości datagramy są przekazywane kolejnym warstwom wyższym, przy czym każda z warstw analizuje swój nagłówek w celu wykrycia ewentualnych przekłamań, a następnie usuwa go. Każda z warstw otrzymuje więc datagram w postaci identycznej z nadaną przez odpowiednią warstwę nadawczą.

Podsumowując przebieg komunikacji: protokoły warstwy 5 – 7 przekazują w dół hierarchii dane, z których na poziomie warstwy 4 tworzone są segmenty lub datagramy, na poziomie warstwy 3 pakiety, na poziomie drugiej ramki, transmisja w warstwie fizycznej odbywa się bitowo, a po stronie odbiorczej wszystko przebiega odwrotnie.

Warstwa druga przekazuje pakiety do transmisji przez łącze radiowe lub kablowe korzystając przykładowo z protokołu SLIP (ang. *Serial Line Interface Protocol*), PPP (ang. *Point to Point Protocol*) lub innych. W komunikacji z modemami TNC w połączeniach AX.25 może być to przykładowo protokół KISS (ang. *Keep It Simply Stupid*)<sup>1</sup>.

Grupa protokołów TCP/IP pozwala na jednoczesne nawiązanie kontaktu w większej liczbie kanałów logicznych (np. jednoczesną transmisję plików przy użyciu protokołu FTP i wymianę poczty elektronicznej). Każdy z tych protokołów jest nadrzędny w stosunku do TCP, odpowiada więc grupie warstw 5 – 7 modelu ISO-OSI. Jednoznaczne przyporządkowanie datagramów do poszczególnych zadań (usług) wymaga zaopatrzenia ich w numer wykorzystywanego kanału. Jest to swego rodzaju adres usługi na fizycznym sewerze, który może równolegle oferować więcej z nich (można by wyobrazić sobie tę sytuację przez analogię do urzędu, w którym w zależności od rodzaju sprawy petent musi się udać do właściwego okienka lub pokoju, ale w tym samym gmachu). Datagramy zawierają wprawdzie numery kanałów nadawcy i adresata, dla stacji docelowej znaczenie ma jedynie numer kanału u adresata.

Tabela C.1.2

Funkcje najważniejszych protokołów z rodziny TCP/IP

Protokół	Znaczenie
HTTP	Udostępnianie stron www
HTTPS	Zabezpieczony dostęp do stron www
TELNET (RLOGIN)	Zdalny dostęp do systemów w sieci
RSH	Zdalne wywoływanie programów na komputerach w sieci
REXEC	
SSH	Zabezpieczony zdalny dostęp do systemów
SMTP	Transmisja poczty elektronicznej
POP3	
FTP	Transmisja plików, uwzględnia prawa dostępu
TFTP	Transmisja plików, nie wyposażony w mechanizmy kontroli dostępu
TCP	Podział na datagramy i ich transmisja zabezpieczona przed przekłamaniami w ramach połączenia między korespondentami
UDP	Niezabezpieczona (bezpolegająca) transmisja datagramów, rozgłaszanie, transmisja strumieni danych
UDP-Lite	
SCTP	Protokół stosowany w transmisji strumieni danych
IP	Wybór tras transmisji w sieci
IPsec	Protokół IP z zabezpieczeniami
ICMP (ang. <i>Internet Control Message Protocol</i> )	Sprawdzenie „drożności” połączenia z dowolnym uczestnikiem w sieci, polecenie <i>ping</i>
SLIP (ang. <i>Serial Line Internet Protocol</i> )	Protokoły warstwy 2 OSI; warstwy dostępu do sieci w modelu TP/IP.
CSLIP (ang. <i>Compressed Serial Line Internet Protocol</i> )	
PPP (ang. <i>Point to Point Protocol</i> )	
X.25	
AX.25	

Transmisja danych dźwiękowych lub wizyjnych, zwłaszcza do wielu użytkowników naraz nie może odbywać się przy użyciu protokołu TCP ponieważ powstające w wyniku oczekiwania na pokwitowanie lub w wyniku powtórzeń pakietów opóźnienia byłyby nie do przyjęcia dla widzów lub słuchaczy. Nie mówiąc już o transmisji do wielu adresatów naraz. Powódz napływających od nich pokwitowań i żądań powtórzenia uniemożliwiłaby jakąkolwiek transmisję. W tych przypadkach stosowana jest nie zabez-

<sup>1</sup> Być może jakaś odmiana takiego protokołu znalazłaby zastosowanie w rozmowach z różnymi politykami

pieczona przed przekłamaniami transmisja w protokole UDP (ang. *User Datagram Protocol*). W ten sposób transmitowane są m.in. pakiety głosowe pochodzące z sieci Echolinku, D-Stara, DMR-u, C4FM itd. Protokół UDP używany jest także przy zapytaniach kierowanych do serwerów DNS czy DHCP albo w połączeniach VPN. Ponieważ protokół UDP nie posiada żadnych zabezpieczeń ewentualne wykrywanie i korektę przekłamań pozostawiono korzystającym z niego zastosowaniom. Może być to przykładowo korekcja wyprzedzająca FEC. Dane FEC stanowią część danych użytkowych i nie zajmują oddzielnego pola w pakiecie UDP (nie są one widzialne dla protokołu UDP). Jest ona powszechnie stosowana w transmisji strumieni danych wizyjnych i fonicznych. Wersją uproszczoną, zapewniającą zmniejszenie opóźnień w transmisji jest protokół UDP-Lite. W transmisji strumieni danych stosowany jest także protokół SCTP (*Stream Control Transmission Protocol*).

## C.2. Adresy internetowe

Do jednoznacznej identyfikacji systemów komputerowych w sieciach TCP/IP służy specjalny adres numeryczny tzw. adres internetowy. W standardzie Ipv4 ma on długość 4 bajtów czyli 32 bitów i jest zapisywany w postaci czterech liczb dziesiętnych oddzielonych kropkami. Liczba dostępnych adresów już od dłuższego czasu jest niewystarczająca, dlatego też stopniowo będzie wchodził w użycie standard Ipv6 o adresach 128-bitowych, a więc o tyle dłuższych, że nawet przy uwzględnieniu potrzeb Internetu przedmiotów powinno ich nie zabraknąć przez dłuższy czas.

Adresy numeryczne są niezbyt praktyczne do zapamiętania dlatego też w sieciach TCP/IP stosowane są adresy symboliczne – nazwy mówiące więcej użytkownikom. Powiązanie adresów symbolicznych z numerycznymi jest zadaniem serwerów DNS (ang. *Domain Name Server*).

Adresy IP wyczerpałyby się już dawno gdyby w sieciach lokalnych nie stosowano adresów należących do grup nie koordynowanych światowo. Adresy należące do tych grup mogą powtarzać się dowolnie w sieciach lokalnych pod warunkiem, że nie są widoczne na zewnątrz. Należą do nich serie 10.x.x.x i 192.168.168.x.x. Sieci lokalne widoczne są na zewnątrz pod wspólnym adresem. Dla sieci domowych wystarczy jeden adres. Zadaniem modemu internetowego (trasownika, ang. *router*) jest odpowiednie kierowanie komunikatów i poleceń do Internetu od każdego z wchodzących w skład sieci lokalnej komputerów lub innych urządzeń i kierowanie nadchodzących z zewnątrz odpowiedzi do właściwego urządzenia. Sieci lokalne występują na zewnątrz pod wspólnym adresem przyznanym przez operatora internetowego z jego puli adresów. Mogą być to adresy koordynowane globalnie albo też analogicznie do sieci lokalnych adresy z grupy niekoordynowanej i wówczas operator internetowy jest dopiero widoczny w sieci światowej przez grupę adresów globalnie koordynowanych (globalnych).

W regularnych odstępach czasu (przykładowo raz na dobę) operator przerywa połączenie z klientami i jeżeli ponowne jego nawiązanie jest możliwe (modem u klienta jest włączony i funkcjonuje) klienci otrzymują od nowa adresy IP i są one z reguły różne od poprzednich (są to więc adresy dynamiczne). Pozwala to operatorom na efektywniejsze zarządzanie pulą adresów IP i nie blokowanie niepotrzebnie nieużywanych w danym momencie. Oznacza to jednak, że użytkownicy nie mogą być dostępni pod stałym adresem (statycznym) od strony Internetu, i że prywatne serwery albo urządzenia do zdalnego sterowania radiostacjami nie mogłyby wypełniać swoich zadań. Lekarstwem na tą sytuację są usługi w rodzaju *dyndns*, *noip* i podobnych. Część z nich jest bezpłatna, przynajmniej w ograniczonym, ale wystarczającym dla zastosowań prywatnych, zakresie. Usługi te oferują klientom stały adres symboliczny (lub kilka) i regularnie odpytują adresy IP modemów u klientów, tak że w krótkim czasie po zmianie adresu IP znane jest już powiązanie aktualnego adresu dynamicznego IP klienta z jego stałym adresem symbolicznym zarejestrowanym u usługodawcy. Oznacza to, że adresy takie jak *telemetry.oe1kda.ddns.net* albo *ic705.oe1kda.ddns.net* są zawsze osiągalne z zewnątrz i pozwalają w tym przykładzie na odczyt danych pomiarowych czy meteorologicznych z własnego serwera http albo na zdalne sterowanie radiostacją. Modemy internetowe mają w swoich konfiguracjach przeważnie możliwość włączenia jednej z takich usług. Dla przedsiębiorstw usługi takie o szerszym zakresie możliwości są dostępne odpłatnie.

Jeżeli na komputerze użytkownika uruchomionych jest więcej serwerów równolegle i występują one dzięki temu od jednym wspólnym adresem IP wewnątrz sieci lokalnej do ich rozróżnienia stosuje się kanały logiczne. Przykładowo więc adres *oe1kda.ddns.net:8080* oznaczałby skorzystanie z serwera HTTP u OE1KDA, a adres *oe1kda.ddns.net:1234* z serwera zdalnego sterowania radiostacją.

Adresy globalne przeważnie nie są potrzebne użytkownikom prywatnym, ale w szczególnych sytuacjach konieczne jest skontaktowanie się z operatorem. Jednym z takich szczególnych przypadków jest, sądząc z opisu w instrukcjach Icoma, korzystanie z funkcji punktu dostępowego („Access Point”) albo terminalowej („Terminal”) nowszych modeli radiostacji D-Starowych.

### C.3 Protokół AX.25

W protokole AX.25 (ang. *packet-radio*) pochodzącym od profesjonalnego protokołu X.25 występują również dwa rodzaje transmisji danych. W połączeniach między dwoma korespondentami nadawane są pakiety numerowane typu I. Zawierają one sumę kontrolną CRC i są po odebraniu kwitowane pozytywnie w przypadku nie wystąpienia przekłamań na trasie lub negatywnie, przez zażądanie powtórzenia w przypadku stwierdzenia przekłamań. W przypadku nie otrzymania powtórzenia w zadanym czasie nadawca automatycznie powtarza pakiet. Mechanizm zabezpieczeń tego typu nosi oznaczenie ARQ (ang. *Automatic Repeat Request*). Dla zwiększenia przepustowości łącza możliwe jest nadawanie kilku pakietów przed otrzymaniem pokwitowania. Ich maksymalna liczba wynosi 7 i jest ustalana za pomocą parametru MAXFRAME w konfiguracji TNC. Połączenia tego typu występują w komunikacji dwóch stacji przez sieć *packet-radio*, co w tej chwili należy do rzadkości.

W drugim przypadku, gdy dane są transmitowane do wielu odbiorców nadawane są nienumerowane, bezpołączeniowe pakiety typu UI. Nie są one kwitowane ani wyposażone w inne mechanizmy kontroli przekłamań albo ich korekty. W razie potrzeby dane użytkowe muszą być uzupełnione o dane korekcyjne (FEC). Transmisja przy użyciu pakietów nienumerowanych występuje m.in. w systemie APRS.

W protokole AX.25 jako adresy służą znaki wywoławcze stacji z ewentualnym dodatkiem rozszerzenia, np. OE1KDA-11. W systemie APRS rozszerzeniom znaku są przypisane konkretne znaczenia informujące odbiorców o rodzaju stacji: domowa, samochodowa, piesza, meteorologiczna itd. W komunikatach APRS znajdują się jednak odniesienia do tabel informujących o tym dokładniej. Szczegółowe omówienie formatów komunikatów APRS zawiera tom 8 „Biblioteki polskiego krótkofalowca”.



## Literatura i adresy internetowe

- Immler, Christian, Bernauer, Hannah, „Raspberry Pi Serverbuch“, Franzisverlag, Monachium 2014, ISBN 978-3-645—60330-0
- Kainka, Burkhard, DK7JD, „Software Defined Radio nutzen. Das SDR-Praxisbuch“, wydawnictwo *Elektor*, Aachen [Aachen] 2019, ISBN 978-3-89576-338-0, [www.elektor.de](http://www.elektor.de)
- Klotz, Leigh L. jr., WA5ZNU, „Ham Radio for Arduino and PICAXE“, wydawnictwo ARRL 2013, ISBN 978-0-87259-324-4
- Kofler, Kühnast, Scherbeck, „Raspberry Pi. Das umfassende Handbuch“, wydawnictwo Rheinwerk, wydanie 3, Bonn 2016, ISBN 978-3-8362-4220-2
- Popiel, Glen, KW5GP, „Arduino Projects for Ham Radio“, wydawnictwo ARRL 2017, ISBN 978-1-62595-070-3
- Purdum, Jack, W8TEE, Peter, Albert, AC8GY, „Microcontroller Projects for Amateur Radio“, ARRL 2020, ISBN 978-1-62595-128-1
- Richards, Mike, G4WNC, „Raspberry Pi Explained for Radio Amateurs“, wydawnictwo RSGB 2020, ISBN 9781 9101 9384 6

Roczniki czasopism wymienionych w tekście.

### Literatura i adresy internetowe do poszczególnych podrozdziałów

- [1.2.1] [www.pa7lim.nl/penaut](http://www.pa7lim.nl/penaut) – witryna autora PA7LIM
- [1.2.2] [www.pa7lim.nl/penaut-request](http://www.pa7lim.nl/penaut-request) – strona rejestracji użytkowników
- [1.2.3] <http://penaut.pa7lim.nl:5678> – pulpit, wykaz aktywności
- [1.2.4] <https://play.google.com/apps/testing/peanut.peanut> – dostęp do programu w witrynie Google Play
- [1.3.1] [www.pa7lim.nl](http://www.pa7lim.nl) – witryna PA7LIM, programy *Peanut* i *BlueDV*
- [1.3.2] „D-STAR komputerowo”, Krzysztof Dąbrowski, OE1KDA, Świat Radio 5/2019, str. 59; 6/2019 str. 58
- [1.3.3] [www.combitronics.nl](http://www.combitronics.nl)
- [1.3.4] [nwdigitalradio.com](http://nwdigitalradio.com)
- [1.3.5] <http://ambeboard.zumradio.com/>
- [1.3.6] <https://reflectorloversclub.jimdofree.com/shop/>
- [1.3.7] <https://reflectorloversclub.jimdofree.com/>
- [1.3.8] [www.dvsinc.com/products/a300x.shtml](http://www.dvsinc.com/products/a300x.shtml) – informacje o wokoderach firmy DVSI
- [3.1.1.1] „APRS-Internet-Service-Client”, Franz G. Aletsee, DL6FCD, CQDL 7/2019, str. 26
- [3.1.1.2] [www.arduino.cc](http://www.arduino.cc) – informacje o Arduino, dokumentacja, przykładowe programy
- [3.1.1.3] [www.aprs2.net/serverstats.php](http://www.aprs2.net/serverstats.php) – spis serwerów APRS-IS
- [3.1.1.4] [www.aprs-is.net/javAPRSFilter.aspx](http://www.aprs-is.net/javAPRSFilter.aspx) –
- [3.1.1.5] [www.github.com/fgaletsee/ArduPRS](https://github.com/fgaletsee/ArduPRS)
- [3.1.1.6] [ham.zmailer.org/oh2mqk/aprx/PROTOCOLS](http://ham.zmailer.org/oh2mqk/aprx/PROTOCOLS)
- [3.1.1.7] [www.github.com/fgaletsee/APRSEthernet](https://github.com/fgaletsee/APRSEthernet)
- [3.1.1.8] [www.aprs.org/doc/APRS101.pdf](http://www.aprs.org/doc/APRS101.pdf)
- [3.1.1.9] <https://apps.magicbug.co.uk/passcode>
- [3.1.1.10] „Biblioteka polskiego krótkofalowca”, tom 33 „Telemetria”, [www.swiatradio.com.pl](http://www.swiatradio.com.pl), strona „Biblioteka Radioamatora”
- [3.1.2.1] [http://wiki.oevsv.at/index.php?title=APRS\\_Arduino-Modem&oldid=14810](http://wiki.oevsv.at/index.php?title=APRS_Arduino-Modem&oldid=14810) – opis w witrynie OeVSV
- [3.1.2.2] <http://unsigned.io/projects/microaprs> – witryna duńskiego konstruktora
- [3.1.2.3] <https://github.com/oe7mbt/APRS-Micromodem> – projekt płyki drukowanej w formacie KiCAD
- [3.1.2.4] <https://github.com/markqvist/LibAPRS> – biblioteka APRS dla Arduino

- [6.1] „Build Your own D-STAR Hotspot”, Bob Wilton, KF5TPQ, QST 2/2020, str. 30
- [6.2] [www.pistar.uk](http://www.pistar.uk) – oprogramowanie i informacje praktyczne
- [6.3] [https://amateurstationnotes.com/images/1-Playing\\_with\\_Pi-Star.pdf](https://amateurstationnotes.com/images/1-Playing_with_Pi-Star.pdf) – instrukcja do programu „Pi-Star” po angielsku
- [6.4] <https://www.balena.io/etcher/> – program kopiujący na moduły SD
- [6.5] <https://sourceforge.net/projects/win32diskimager/> – program kopiujący „Disk Imager”
- [6.6] [https://www.chip.de/downloads/Win32-Disk-Imager\\_46121030.html](https://www.chip.de/downloads/Win32-Disk-Imager_46121030.html)
- [6.7] [www.advanced-ip-scanner.com](http://www.advanced-ip-scanner.com) – program wyświetlający adresy IP w sieci domowej

**W serii „Biblioteka polskiego krótkofalowca” dotychczas ukazały się:**

- Nr 1 – „Poradnik D-STAR”, wydanie 1 (2011), 2 (2015) i 3 (2019)
- 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
- Nr 10 – „Poczta elektroniczna na falach krótkich” Tom 2
- 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”
- 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”
- Nr 34 – „Poradnik systemu C4FM”, wydanie 1 (2017) i 2 (2019)
- 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) i 2 (2020)
- 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
- Nr 50 – „Anteny ultrakrótkofalowe” Tom 1
- Nr 51 – „Anteny krótkofalowe” Tom 2
- Nr 52 – „Anteny ultrakrótkofalowe” Tom 2
- 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





