Science, Technology and Innovation
ISSN 2544-9125 ▪ 2025; 22 (3): 16-23
doi: 10.55225/sti.700

STI • 2025; 22 (3): 16-23

Radiosonda Meteomodem M20 jako radiowy tracker z amatorskim oprogramowaniem

Meteomodem M20 radiosonde as a radio tracker with amateur software

Jacek Jasielski[1] A,E,F, Maciej Witek[1] A,G

[1]Akademia Tarnowska, Wydział Nauk Technicznych, Katedra Katedra Elektroniki i Technologii Inteligentnych, ul. Mickiewicza 8, 33-100 Tarnów, Polska

Nota aplikacyjna

Abstrakt

W artykule opisano oprogramowanie do fabrycznych radiosond Meteomodem M20 pracujące w standardzie Horus V2. Opracowana przez działające w Akademii Tarnowskiej Studenckie Koło Naukowe Elektroników AMPER wersja programu działa na odzyskanej sondzie meteorologicznej i nie wymaga dodatkowych przeróbek (np. wymiany mikrokontrolera). Ze względu na mocno ograniczone zasoby (oryginalny mikrokontroler posiada jedynie 32kB pamięci Flash) do kompilacji użyto pakietu Keil. Jest to oprogramowanie komercyjne, lecz dla procesorów serii L0 pakiet jest darmowe jego użycie wymaga jedynie wcześniejszej rejestracji. Oprogramowanie zostało przetestowane w trzech kolejnych misjach balonowych. W misji BEM 1 została wysłana sonda z oprogramowaniem nadającym w standardzie RTTY natomiast w misji BEM 2 i BEM 3 testowano opisane w artykule oprogramowanie pracujące w standardzie Horus V2.

Abstract

This article describes the software for factory-installed Meteomodem M20 radiosondes operating in the Horus V2 standard. Developed by the AMPER Student Research Group for Electronics at the University of Tarnów, this version of the program runs on a recovered meteorological probe and requires no additional modifications (e.g., microcontroller replacement). Due to severely limited resources (the original microcontroller has only 32kB of Flash memory), the Keil package was used for compilation. This commercial software is available, but for L0 series processors, the package is free; its use requires only prior registration. The software was tested in three consecutive balloon missions. In the BEM 1 mission, a probe was sent with software transmitting in the RTTY standard, while in the BEM 2 and BEM 3 missions, the software described in the article was tested operating in the Horus V2 standard.

Słowa kluczowe

- balon stratosferyczny
- radiosonda
- Meteomodem
- M20
- ADF7012
- Horus
- 4FSK

Udziały autorów

A – przygotowanie badań
B – gromadzenie danych
C – analiza statystyczna uzyskanych wyników
D – interpretacja uzyskanych wyników
E – przygotowanie pierwotnej wersji tekstu
F – przegląd literatury
G – korekta i rewizja tekstu

Korespondencja

Jacek Jasielski
e-mail: j_jasielski@atar.edu.pl
Akademia Tarnowska
Wydział Nauk Technicznych
Katedra Elektroniki i Technologii Inteligentnych
ul. Mickiewicza 8
33-100 Tarnów, Poland

Informacje o artykule

Historia artykułu (Article history)

- Otrzymano (Received): 2025-12-15
- Zaakceptowano (Accepted): 2025-12-29
- Opublikowano (Published): 2025-12-30

Wydawca (Publisher)

Akademia Tarnowska
University of Applied Sciences in Tarnow
ul. Mickiewicza 8, 33-100 Tarnow, Poland

Finansowanie (Financing)

Badania nie zostały sfinansowane z grantów pochodzących ze środków publicznych, organizacji komercyjnych lub non-profit

Konflikt interesów (Conflict of interest)

Nie zadeklarowano konfliktu interesów.

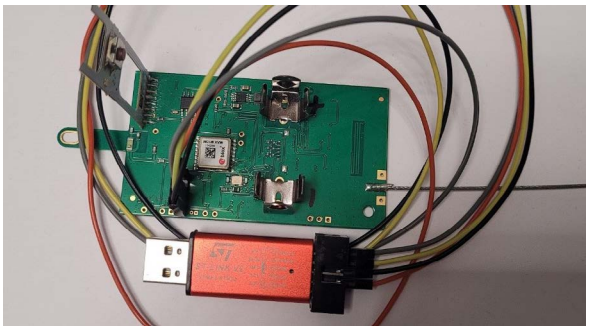
Wprowadzenie

W ostatnich latach wiele ośrodków naukowych i dydaktycznych wypuszcza badawcze misje balonowe podczas których przeprowadzane są różne badania. Także Akademia Tarnowska od dwóch lat wysyła takie misje (Szczepanik1/2/3 [1], Bem1/2/3). Jednym z podstawowych elementów każdej misji jest tracker (np. przeprogramowana radiosonda meteorologiczna), który na bieżąco przesyła lokalizację balonu, pozwalając na śledzenie trasy jego przelotu oraz później – po wylądowaniu – na zlokalizowanie miejsca upadku oraz podjęcie ładunku. Dodatkowo może ona zostać wyposażona w czujniki mierzące podczas lotu temperaturę, wilgotność oraz ciśnienie. W Internecie można znaleźć wiele rozwiązań radiosond do samodzielnego wykonania (np. seria sond PecanPico [2], czy rozwiązania wykorzystujące modemy LoRa SX1278 oraz bibliotekę Radiolib [3]). Bardzo atrakcyjne może okazać się wykorzystanie radiosond meteorologicznych. Radiosondy meteorologiczne to profesjonalne urządzenia wyposażone w dokładne czujniki umożliwiające pomiar temperatury, wilgotności czy ciśnienia, odbiornik GNSS (ang. Global Navigation Satellite System). Dane te następnie są wysyłane drogą radiową do stacji naziemnej, która zbiera je i wylicza położenie i prędkość wiatru na danej wysokości sondowania. Radiosondaż atmosfery prowadzony jest na całym świecie, w tym z czterech ośrodków w Polsce (Wrocław, Tarnów, Legionowo, Łeba). Gdy sonda zostanie wypuszczona, balon na pewnej wysokości pęknie i spadnie na ziemię, sondy uważa się za zużyte i nie odzyskuje się ich w celu ponownego profesjonalnego wykorzystania (nie dotyczy sond ozonowych!). Umożliwia to hobbystom ich poszukiwanie i ponowne wykorzystanie w sposób amatorski jako systemy do nauki programowania lub trackery radiowe, na których autorzy skupią się w tym artykule.

Szczególnie popularne są rozwiązania wykorzystujące radiosondy RS41 firmy Vaisala do których można znaleźć kilka wersji amatorskiego oprogramowania [4, 5], pozwalające na transmisję pozycji i telemetrii w standardach APRS, RTTY czy Horus. Ostatnio pojawiło się również amatorskie oprogramowanie do radiosond Meteomodem M20. Niestety pierwsze dostępne wersje wymagały wymiany mikrokontrolera z oryginalnego STM32L051 na STM serii F4. W rozwiązaniu opracowanym przez pracowników AT i Studenckiego Koła Naukowego Elektroników „AMPER" użyto sondy bez konieczności wymiany elementów (rozważane jest jedynie użycie TCXO do generowania zegara dla układu radiowego, gdyż układ radiowy ADF7012 jest sterowany zegarem mikrokontrolera (czyli sonda nie posiada temperaturowej stabilizacji częstotliwości). Trzy wersje z testowym oprogramowaniem zostały wysłane w ramach ostatnich misji stratosferycznych. Pierwsza wersja pracowała w standardzie RTTY 7N2, natomiast dwie kolejne w standardzie 4FSK Horus V2.

Budowa radiosondy meteorologicznej M20

Sonda meteorologiczna M20 zbudowana jest z czterech podstawowych bloków: zasilania, sterowania, czujników i transmisji danych.



Rysunek 1. Zdjęcie radiosondy M20 z podłączonym programatorem ST-Link V2

Na rysunku 1 przestawiono widok płytki radiosondy meteorologicznej Meteomodem M20. Na płytce dostępne są piny umożliwiające zmianę oprogramowania (niezbędne jest jedynie wlutowanie goldpinów). Na zdjęciu widać również dołączony programator używany do programowania. Jej schemat blokowy przedstawiono na rysunku 2.

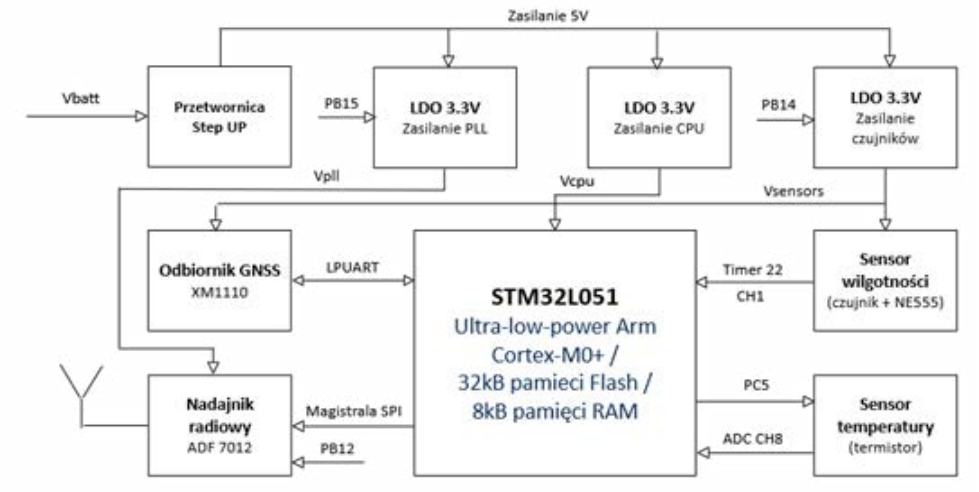
Blok zasilania

Ze względu na zasilanie baterijne, a zatem zmiany napięcia związane z rozładowaniem baterii, niezbędnym układem jest przetwornica DC-DC podwyższająca (i wstępnie stabilizująca) napięcie z baterii. To napięcie używanie jest później do zasilania lokalnych stabilizatorów LDO (Low Dropout) oraz również stopnia wyjściowego mocy RF. Pierwszy z układów LDO zasila mikrokontroler sterujący radiosondą i nie jest sterowany. Dwa pozostałe są sterowane przez porty mikrokontrolera i zasilają układ syntezy ADF7012 oraz czujniki (GNSS, wilgotności). Blok pomiaru temperatury (z termistorem) zasilany jest bezpośrednio z portu mikrokontrolera.

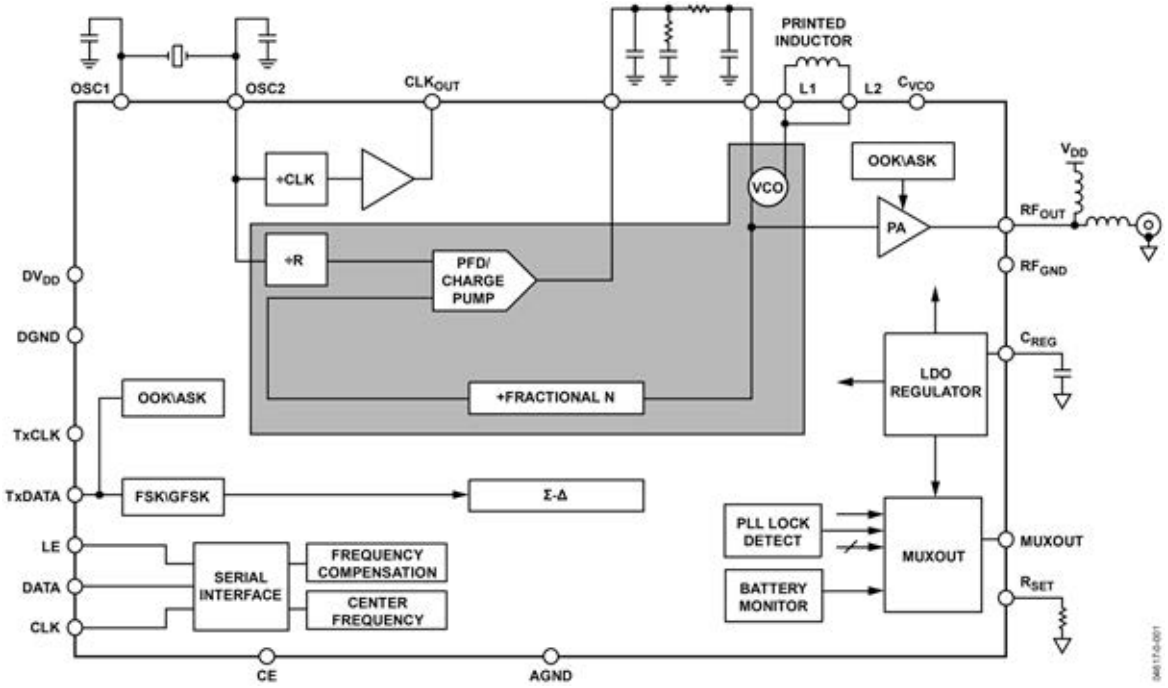
Blok sterowania

Do sterowania radiosondą wykorzystano energooszczędny mikrokontroler serii STM32L051 posiadający

wbudowaną pamięć Flash o pojemności 32 kB i pamięć RAM 8 kB. Do sterowania wykorzystane są piny GPIO oraz dwie magistrale – dwukierunkowa szeregowa LPUART (Low Power Universal Asynchronous Receiver-Transmitter) do komunikacji z układem GNSS oraz jednokierunkowa SPI (Serial Peripheral Interface) do programowania układu ADF7012. W celu kontroli napięcia zasilania do jednego z wejść ADC (Analog-to-Digital Converter) mikrokontrolera podłączone jest dodatni biegun baterii. Za pomocą przetwornika ADC można również dokonać pomiaru temperatury mikrokontrolera.



Rysunek 2. Uproszczony schemat blokowy radiosondy M20



Rysunek 3. Schemat blokowy układu nadajnika ADF7012 [6]

Blok czujników

Jednym z ważniejszych układów radiosondy jest układ GNSS. W zależności od wersji może to być układ XM110 (który wymaga wcześniejszej wymiany oprogramowania układowego) lub w nowszych wersjach U-Blox NEO M10N. Oba układy wysyłają dane w standardzie NMEA z tą samą prędkością 9600 BAUD. Umożliwia to użycie oprogramowania zarówno w starszych jak i nowszych modelach. Moduł GNSS podłączony jest dwukierunkowo z portem LPUART mikrokontrolera sterującego. Dzięki temu możliwy jest zarówno odczyt danych jak i wysłanie komend sterujących powodujących włączenie właściwego trybu śledzenia satelitów, niezbędnego do pracy układu na dużych wysokościach.

Oryginalna sonda używa również czujników temperatury (termistor) i wilgotności (czujnik pojemnościowy podłączony do układu NE555 stanowiącego konwerter pojemności na częstotliwość, odczytywaną przez Timer kontrolera).

Ze względu na ograniczoną ilość pamięci, zrezygnowano z obsługi wbudowanych czujników. Dodatkowo wymagałyby one kalibracji, co skomplikowałoby korzystanie z nich i zwiększyło zapotrzebowanie na pamięć.

Blok transmisji danych

ADF7012 firmy Analog Devices jest układem nadajnika radiowego małej mocy pracującego w pasmach sub-GHz (poniżej 1 GHz) – układ może pracować z zakresie od 75 MHz do 1 GHz z szybkością transmisji danych do 179,2 kb/s. Wspiera takie modulacje jak FSK, GFSK, OOK, GOOK, ASK [6].

Do ustawienia częstotliwości nośnej pętli PLL należy wykorzystać zależność:

$$f_{OUT} = \frac{f_{MCO}}{R}\left(N_{INT} + \frac{N_{FRAC}}{2^{12}}\right)$$

Przy czym podzielniki: R, całkowity i ułamkowy muszą przyjmować wartości R < 16, N_{INT}< 256, N_{FRAC}< 4096.

Układ jest programowany za pomocą jednokierunkowej szeregowej magistrali SPI (sygnały: DATA, CLK, LE). Zmodulowany sygnał wyjściowy z wyjścia RFOUT układu ADF7012 jest dodatkowo wzmacniany przez tranzystorowy wyjściowy wzmacniacz mocy RF. Wzmacniacz ten jest załączany przez mikrokontroler na czas transmisji a następnie wyłączany w celu oszczędzania energii.

Zmiana oprogramowania (firmware) odbiornika GNSS

Radiosonda M20 w wersji V06 posiada odbiornik GNSS XM1110, który pracuje z oprogramowaniem wysyłającym dane binarnie. Wymagałoby to używania dwóch różnych wersji oprogramowania w zależności od modułu GNSS, jednak dostępny jest pakiet FlashTool [7] wraz z różnymi wersjami oprogramowania systemowego, umożliwiający zmianę na pracujące w standardzie NMEA, zatem zdecydowano się na zmianę do takiej wersji.

Konfiguracja programu do zmiany firmware jest bardzo prosta. Wymaga podania portu do którego podłączony jest odbiornik, pliku Download Agent (czyli właściwej dla konkretnego odbiornika wersji bootloadera) oraz żądanej wersji ROM. Po zaprogramowaniu odbiornik działa w standardzie NMEA z prędkością transmisji 9600 BAUD.

Na rysunku 4 przedstawiono przykładową sentencję odebraną z odbiornika GNSS po dokonanej zmianie oprogramowania.

```
$GNGGA,185456.000,5000.7410,N,02058.9198,E,1,8,1.76,177.9,M,41.6,M,,*4B
$GPGSA,A,3,08,16,27,10,,,,,,,,,2.01,1.76,0.97*04
$GLGSA,A,3,82,79,,,,,,,,,,,2.01,1.76,0.97*17
$GAGSA,A,3,11,25,,,,,,,,,,,2.01,1.76,0.97*19
$GPGSV,3,1,11,27,79,271,25,10,65,164,18,08,42,300,33,16,35,208,18*74
$GPGSV,3,2,11,15,16,041,,32,03,152,,07,02,302,13,13,02,013,*7E
$GPGSV,3,3,11,30,02,333,,36,,,,21,,,16*4A
$GLGSV,2,1,08,81,70,354,,80,34,296,12,88,34,082,,82,32,292,19*64
$GLGSV,2,2,08,79,23,235,22,73,09,347,,72,08,016,,70,04,013,*60
$GAGSV,1,1,04,11,63,270,25,25,48,260,19,16,43,223,07,02,22,318,18*69
$GNRMC,185456.000,A,5000.7410,N,02058.9198,E,0.03,69.01,230125,,,A*40
$GNVTG,69.01,T,,M,0.03,N,0.06,K,A*18
```

Rysunek 4. Przykładowa sekwencja danych z układu XM1110 z oprogramowaniem NMEA

Jak widać na rysunku urządzenie wysyła wszystkie ważniejsze ramki protokołu NMEA: GGA, GSA, GSV, RMC i VTG, zatem w prosty sposób można odczytać żądane dane takie jak aktualny czas, długość i szerokość geograficzna, wysokość czy liczbę używanych satelitów.

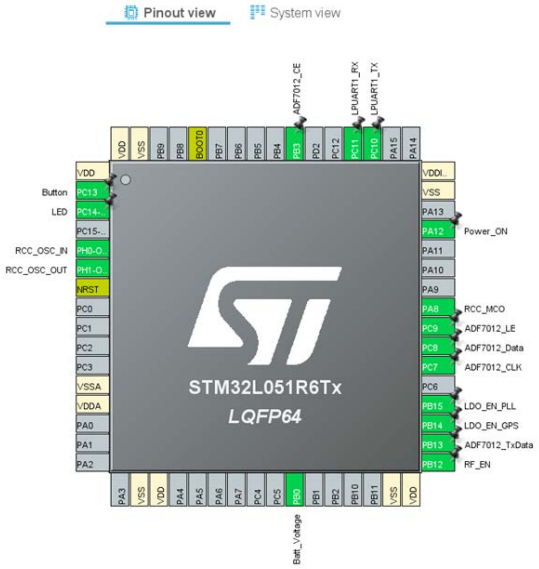
Oprogramowanie układowe

Działanie programu opiera się na trzech blokach:

1. Inicjalizacja – ustawienie właściwych poziomów portów, konfiguracja zegarów, LPUART, timera i przetwornika ADC, uruchomienie systemu przerwań od timera i LPUART.
2. Pętla główna programu:
   - odczyt danych z ADC;
   - parsowanie danych NMEA;
   - przygotowanie pakietu HorusV2 i uruchomienie transmisji.
3. Obsługa systemu przerwań:
   - obsługa przerwań LPUART - odbiór kolejnych bajtów protokołu NMEA, sprawdzanie skompletowania pełnej ramki, sprawdzenie jej poprawności;
   - obsługa przerwań Timera 2 – wysyłanie kolejnych symboli pakietu, znacznik czasu 10mS, sterowanie diodą LED.

Inicjalizacja

Do przygotowania projektu użyto aplikacji STM32CubeMX [8]. Pozwala ona na konfigurację mikrokontrolera. Można w nim w prosty sposób skonfigurować porty GPIO (kierunek, typ czy stan początkowy) ustawić parametry portów szeregowych, tryb pracy liczników czy przetwornika ADC. Służy również do konfiguracji zegarów wewnętrznych mikrokontrolera oraz zegara MCO (Microcontroller Clock Output) używanego do taktowania ADF7012. Przy użyciu programu skonfigurowano port szeregowy LPUART (9600baud, 8N1) , przetwornik ADC (tryb pracy, kanał 8) oraz Timer2 w trybie AutoReload (czas 10ms). W sekcji konfiguracji zegara ustawiono częstotliwość zegara systemowego na 32MHz i MCO (sterujący układem PLL) na 16MHz. Widok mikrokontrolera ze zdefiniowanymi wyjściami przedstawiono na rysunku 5.



Rysunek 5. Widok mikrokontrolera ze zdefiniowanymi w programie STM32CubeMX wyprowadzeniami

Na początku programu (kod 1) należało ustawić odpowiednie stany na portach wyjściowych GPIO kontrolera. W pierwszej kolejności ustawiono stan wysoki na porcie Power_ON (PA12) sterującym kluczem podłączającym zasilanie z baterii do trackera. Kolejno załączono napięcia do układu GNSS (LDO_EN_GPS / ustawiono stan wysoki) i ADF (PLL_EN_GPS / ustawiono stan wysoki) oraz (w celu sygnalizacji) zaświecono diodę LED (port PC14/ ustawiono stan wysoki). W celu oszczędności energii należało odłączyć zasilanie wzmacniacza mocy RF (RF_EN /ustawiono stan wysoki). W kolejnym kroku przeprowadzono inicjalizację układu radiowego. Dla założonych wartości f MCO = 16MHz i R = 7 otrzymano współczynniki NINT =171 NFRAC = 1863 oraz minimalny skok FSTEP = 139Hz. Ostatnią czynnością było uruchomienie obsługi systemu przerwań (LPUART, Timer2).

Przygotowanie danych do pakietu

Przygotowanie danych z ADC

Oprócz danych odebranych z GNSS do utworzenia pakietu niezbędne są również pomiary z przetwornika ADC. Do jego wejść podłączone są dodatni biegun baterii (kanał 8 ADC) i wewnętrzny konwerter temperatura/napięcie pozwalający na pomiar temperatury układu (kanał 18 ADC). Wbudowany czujnik temperatury (kod 2) wymaga kalibracji za pomocą danych TEMP130_CAL i TEMP30_CAL zapisanych w pamięci kontrolera za pomocą zależności:

$$f_{OUT} = \frac{f_{MCO}}{R}\left(N_{INT} + \frac{N_{FRAC}}{2^{12}}\right)$$

Do obsługi użyto bibliotek HAL. W celu pomiaru (ADC mierzy trzy różne kanały) niezbędne było ustawienie właściwego kanału ADC. Po ustawieniu kanału uruchomiono pomiar i odczekano na jego zakończenie. Po zakończeniu odczytano wartość i obliczono wartość aktualnej temperatury kontrolera.

Standard NMEA nie wysyła parametru „prędkość wznoszenia/opadania" dlatego uśrednioną wartość należało obliczyć, wykorzystując dane z aktualnej i ostatnio wysyłanej ramki (wysokość NPM, czas).

Parsowanie danych ramek NMEA

Po skompletowaniu poprawnej ramki w programie głównym następuje proces parsowania danych z odbieranych ramek NMEA. Program używa trzech typów ramek: RMC (Recommended Minimum Specific GNSS Data), GGA (Global Positioning System Fix Data) i VTG (Course Over Ground and Ground Speed). Z ramki RMC odczytano czas, długość geograficzną, szerokość geograficzną. Z ramki GGA parsowane są wysokość NPM i liczba użytych satelitów. Ramka VTG pozwala na odczyt prędkość lotu radiosondy.

Przygotowanie pakietu do transmisji

W amatorskich radiosondach stratosferycznych najczęściej używanym standardem transmisji jest Horus 4FSK. Standard używany jest w dwóch wersjach V1 i V2. Różnią się one długością pakietu. W standardzie v2 (kod 3)zaimplementowano dodatkowe pola dla danych

```
HAL_Init();
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_LPUART1_UART_Init();
MX_TIM2_Init();
MX_ADC_Init();

HAL_GPIO_WritePin(Power_ON_GPIO_Port, Power_ON_Pin, GPIO_PIN_SET);
       // podtrzymanie zasilania radiosondy
HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_SET);
               // włączenie diody LED
HAL_GPIO_WritePin(LDO_EN_GPS_GPIO_Port, LDO_EN_GPS_Pin, GPIO_PIN_SET);             /
               / włączenie zasilania GNSS
HAL_GPIO_WritePin(LDO_EN_PLL_GPIO_Port, LDO_EN_PLL_Pin, GPIO_PIN_SET);
               // włączenie zasilania PLL

HAL_GPIO_WritePin(ADF7012_CS_GPIO_Port, ADF7012_CS_Pin, GPIO_PIN_SET);
               // aktywacja układu radiowego ADF7012
HAL_Delay(200);

// Konfiguracja ADF7012
adf_setup();                  // ustawienie rejestrów r0-r3
HAL_Delay(200);
adf_lock();                   // załączenie pętli PLL
HAL_Delay(200);
HAL_GPIO_WritePin(RF_EN_GPIO_Port, RF_EN_Pin, GPIO_PIN_SET);
                              // wyłączenie zasilania wzmacnacza mocy RF

HAL_TIM_Base_Start_IT(&htim2);
                      // Uruchomienie timera TIM2
HAL_UART_Receive_IT(&hlpuart1, &GPS_temp, 1);
                      // Uruchomienie odbioru NMEA przez lpurat
```

Kod 1. Kod inicjalizacji radiosondy

```
// Pomiar temperatury procesora
   ADC1->CHSELR  = 0x40000;   // Wybór kanału "Temperature Sensor"
   HAL_ADC_Start(&hadc);
   HAL_ADC_PollForConversion(&hadc, 100);
   ADCval = HAL_ADC_GetValue(&hadc);
   ADCtmp = (ADCval * STM_volt / VDD_CALIB - *TEMP30_CAL) * 100 /
*TEMP130_CAL - *TEMP30_CAL);
   STM_temp = ADCtmp + 30;
```

Kod 2. Kod obsługi ADC do pomiaru temperatury

telemetrycznych o długości 9 bajtów. Ich przeznaczenie może być zdefiniowane przez użytkownika. Dodatkowo zwiększono do 2 bajtów pole PayloadID, które definiuje znak krótkofalowy i typ danych telemetrycznych. Ramka w formie binarnej w standardzie V2 ma długość 32 bajty [9].

Po skompletowaniu danych program generuje 32-bitowy pakiet binarny. Jego integralność jest chroniona za pomocą 16-bitowego kodu CRC16-CCITT. Ponieważ telemetria nie została zaimplementowana w programie pola definiowalne nie są modyfikowane. Kodowanie zwiększa wymiar ramki do 65 bajtów. Dodatkowo przed nadawaniem do pakietu dodano 25 bajtów o wartości 0 x 0 (służące do synchronizacji fali nośnej) oraz 8 bajtów preambuły o wartości 0x1b (czyli kolejno symboli 00, 01, 10, 11) służących do dostrojenia czterech podnośnych poszczególnych symboli. Po utworzeniu kompletnego pakietu zmienna tx_on przyjmuje wartość 2 (zezwolenie na start transmisji pakietu w obsłudze przerwania od Timera 2)

Obsługa przerwania LPUART (odczyt danych z układu GNSS)

Kolejne bajty danych z układu GNSS odbierane są przez port szeregowy LPUART. Odbiór wykorzystuje system

```
struct HorusBinaryPacketV2
{
  uint16_t    PayloadID;    // Definiuje znak i rolę bajtów użytkownika
  uint16_t    Counter;      // Licznik
  uint8_t     Hours;        // Godziny Zulu
  uint8_t     Minutes;      // Minuty
  uint8_t     Seconds;      // Sekundy
  float       Latitude;     // Szerokość geogr. w formacie xx.yyyy
  float       Longitude;    // Długość geogr. w formacie xx.yyyy
  uint16_t    Altitude;     // Wysokość w metrach
  uint8_t     Speed;        // Prędkość w km/h
  uint8_t     Sats;         // Ilość satelitów
  int8_t      Temp;         // Temperatura kontrolera
  uint8_t     BattVoltage;  // Napięcie baterii 0 = 0.5v, 255 = 5.0V,
  // Dane telemetryczne definiowane przez użytkownika, dla ID 4FSKTEST-V2
  int16_t     dummy1;       // Prędkość wznoszenia / 100
  int16_t     dummy2;       // Temperatura zewnętrzna / 10
  uint8_t     dummy3;       // Wilgotność zewnętrzna
  uint16_t    dummy4;       // Ciśnienie zewnętrzne / 10
  uint16_t    unused;       // 2 bajty nie używane
  uint16_t    Checksum;     // Suma kontrolna CRC16-CCITT
}  __attribute__ ((packed));
```

Kod 3. Struktura pakietu Horus V2

przerwań. Po odebraniu symbolu $ (oznaczającego początek ramki) rozpoczyna się kompletowanie kolejnej ramki. Odbiór znaku 0x10 (LF) kończy ramkę. Po skompletowaniu pełnej ramki następuje sprawdzenie poprawności sumy kontrolnej (porównanie CRC wyliczonej na podstawie odebranej ramki z przesłaną w sentencji NMEA). Gdy jest ona poprawna, zmienna new_GPS_msg przyjmuje wartość 1.

Obsługa przerwania Timer 2 (obsługa wysyłania symboli 4FSK, pomiar czasu)

Transmisja rozpoczyna się gdy zmienna tx_on ma wartość 2. Przesyłane kolejne bajty pakietu HorusV2 dzielone są na dwubitowe symbole, które transmitowane są drogą radiową. Głównym problemem do rozwiązania była obsługa modulacji 4FSK (transmisja z użyciem 4 różnych nośnych). Nie można tego osiągnąć przez bezpośrednie programowanie częstotliwości nośnej ADF7012 ze względu na długi czas synchronizacji pętli fazowej po modyfikacji rejestru r1. We wcześniejszej wersji programu obsługującego transmisję RTTY używano sprzętowej modulacji FSK, co naprowadziło na skuteczne rozwiązanie. Wymagany odstęp częstotliwości modulacji FSK był definiowany w rejestrze r2 przez wartość parametru modulation_deviation , a sama transmisja realizowana sprzętowo przez pin TxData układu ADF. W wersji 4FSK W modulacji 4FSK ten parametr modyfikowany jest przed każdym wysłaniem symbolu, tak aby uzyskać właściwy odstęp $n \cdot F_{STEP}$ (n = 0,1,2,3).

```
void adf_send_symbol(char symbol)
{
        adf_config.r2.modulation_deviation =
2 * symbol;
        adf_write_register_two();
  ADF_TxD_High;        // Sterowanie
sprzętowe
}
```

Kod 4. Kod procedury wysłania symbolu 4FSK

Wartość symbolu (kod 4) przekazywana jest do procedury, gdzie wyliczany jesz żądana wartość parametru modulation_deviation w rejestrze r2, a zawartość rejestru następie przesyłana do układu ADF7012 ustawiając właściwą częstotliwość transmitowanego symbolu. Ponieważ FSTEP = 139Hz otrzymany odstęp częstotliwości wynosi 278Hz. Wartość ta różni się od teoretycznej dla standardu Horus wynoszącej 270Hz, ale nie uniemożliwia poprawnego odbioru.

Po przesłaniu pełnej ramki zmienna tx_on ustawiana jest na 0. W kolejnych wywołaniach przerwania odmierzany jest żądany czas oczekiwania pomiędzy kolejnymi ramkami. Po jego zakończeniu zmienna ta jest ustawiana na 1 co wymusza w segmencie głównym generację kolejnej ramki.

Transmisja 4FSK jest transmisją wąskopasmową, podczas której jednym z kluczowych aspektów jest stabilność zegara. Układ ADF7012 taktowany jest zegarem MCU, który przy dużych zmianach temperatury (od -70 do +20°C) zmienia generowaną częstotliwość. Nie jest

to problemem przy ręcznej zmianie okna np. podczas odbioru w programie SDRSharp, natomiast automatyczne amatorskie stacje mogą mieć problem z poprawnym odbiorem sygnału przy większym odchyleniu częstotliwości nośnej. Dlatego też rozważane jest zastąpienie sygnału MCO przez zewnętrzny generator TCXO (Temperature Compensated Crystal Oscillator) o znacznie mniejszych zmianach częstotliwości powodowanych zmianami temperatury.

Podsumowanie

Celem projektu było wykorzystanie użytych przez IMGW do badania atmosfery (w pełni sprawnych) radiosond meteorologicznych w amatorskich balonowych misjach stratosferycznych jako trackery radiowe. Aby to zapewnić, wykorzystano popularny amatorski standard Horus V2 wykorzystujący modulację 4-FSK. Ze względu na małą ilość dostępnej pamięci w mikrokontrolerze, celowo pominięto wykorzystanie oryginalnych czujników sondy i skupiono się na transmisji pozycji sondy. Przy ograniczonej pojemności pamięci, udało się zaimplementować pomiar temperatury kontrolera i napięcia baterii zasilającej, odpowiednią modulację i kodowanie ramki danych, bez ingerowania w komponenty na PCB. Oprogramowanie jest publicznie dostępne przez Koło Naukowe „Amper".

Podziękowania

Autorzy składają serdeczne podziękowania Panu Waldemarowi Krauze (znak SP5GFN) członkowi Zarządu Tarnowskiego Klubu Krótkofalowców i Modelarzy za udostępnienie znaku krótkofalarskiego na potrzeby testów oprogramowania i misji stratosferycznych KNE AMPER Akademii Tarnowskiej.

Składamy również podziękowania dla byłych studentów AT (członków KNE Amper) którzy uczestniczyli w przygotowaniach i realizacji misji balonowych.

Bibliografia

[1] Antosz J, Arabik R, Jasielski J, Witek M, Ciężadło Ł. Misje stratosferyczne Akademii Tarnowskiej: część 1: sprzęt i oprogramowanie. Science, Technology and Innovation. 2023;18(3–4):30-45. https://doi.org/10.55225/sti.565.

[2] Krahn T. PecanPico4 [Internet]. GitHub; c2025 [cited 2025 Nov 30]. Available from: https://github.com/tkrahn/pecanpico4.

[3] Kroes R. HorusBinary Tracker [Internet]. GitHub; c2025 [cited 2025 Nov 30]. Available from: https://github.com/RoelKroes/HorusBinary_Tracker

[4] Nousiainen M. RS41ng [Internet]. GitHub; c2025 [cited 2025 Nov 30]. Available from: https://github.com/mikaelnousiainen/RS41ng

[5] Versatile, feature-rich and user-friendly custom firmware for ALL revisions of Vaisala RS41 radiosondes [Internet]. GitHub; c2025 [cited 2025 Nov 30]. Available from: https://github.com/Nevvman18/rs41-nfw

[6] Analog Devices. ADF7012 [Internet]. c2025 [cited 2025 Nov 30]. Available from: https://www.analog.com/en/products/adf7012.html

[7] Blitzortung.org. GlobalTop FlashTool [Internet]. c2025 [cited 2025 Nov 30]. Available from: https://www.blitzortung.org/Compendium/Hardware/GlobalTop/GlobalTop%20FlashTool%20lightningmaps_org/

[8] STMicroelectronics. STM32Cube initialization code generator (STM32CubeMX) [Internet]. c2025 [cited 2025 Nov 30]. Available from: https://www.st.com/en/development-tools/stm32cubemx.html

[9] Project Horus Team. Project Horus [Internet]. c2025 [cited 2025 Nov 30]. Available from: http://www.projecthorus.org/