

# STM32

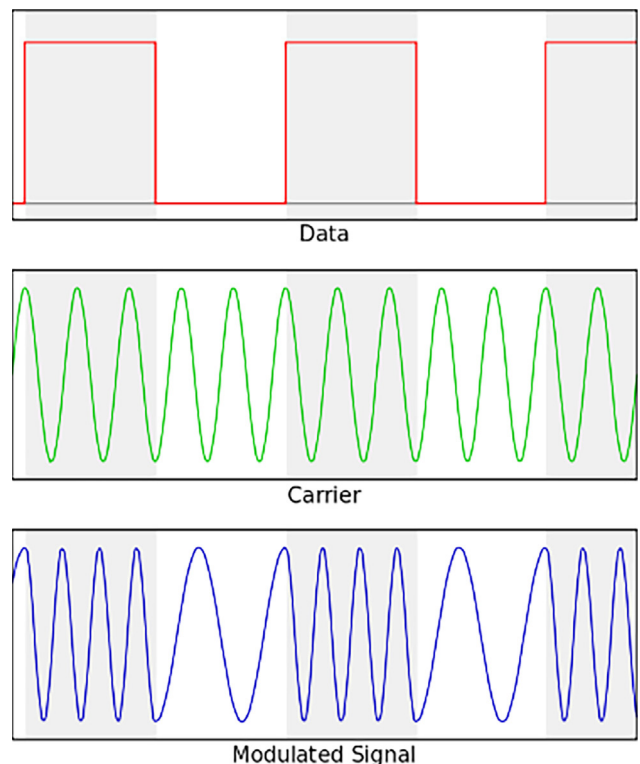
## FSK i LoRa w zestawach startowych STMicroelectronics

Przedstawiona w poprzednich wydaniach EP modulacja LoRa stanowi bardzo dobre rozwiązanie dla systemów wymagających komunikacji bezprzewodowej na duże odległości. Jednak jak w porównaniu z nią wypadają standardowe i powszechnie stosowane techniki modulacji? W artykule przedstawiamy modulację FSK realizowaną na modemach SX1276 firmy Semtech oraz S2-LP od STMicroelectronics.

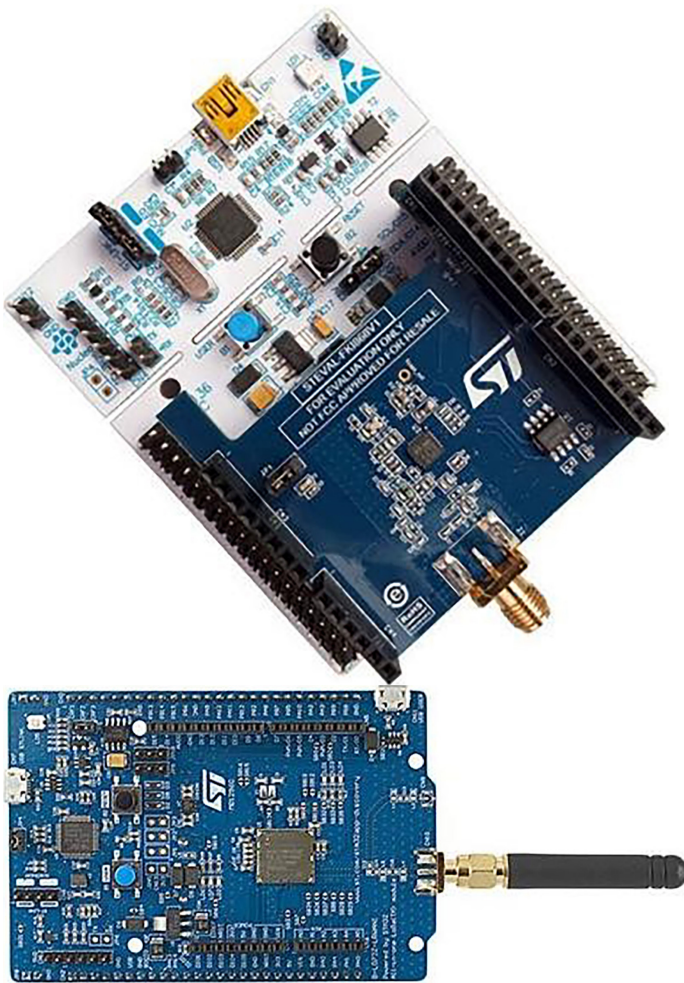
FSK jest rodzajem modulacji częstotliwości, w której informacja cyfrowa jest przekazywana za pomocą dyskretnych zmian częstotliwości nośnej. W przypadku sygnału binarnego mamy do czynienia z dwiema częstotliwościami, z których jedna odpowiada stanowi logicznemu 0, a druga – stanowi 1. Modulacja FSK może być opisana za pomocą trzech parametrów:

1. Częstotliwość nośna (*carrier frequency, base frequency*).
2. Dewiacja częstotliwości (*frequency deviation*).
3. Prędkość transmisji danych (*datarate*).

Podczas modulacji sygnału binarnego, różnica między częstotliwością nośną, a częstotliwościami odpowiadającymi stanom niskiemu i wysokiemu jest określona przez dewiację częstotliwości. Zmiana częstotliwości może odbywać się poprzez kluczowanie pomiędzy dwoma generatorami, lub przestrajanie jednego generatora. Drugi przypadek został przedstawiony na rysunku 1.



Rysunek 1. Modulacja FSK z ciągłą fazą (źródło: elprocus.com)



**Rysunek 2. Zestawy STEVAL-FKI868V1 z płytką NUCLEO-L152RE (z lewej) i B-L072Z-LRWAN1 (z prawej) (źródło: digikey.com)**

Modyfikacją modulacji FSK jest modulacja GFSK, w której sygnał przed modulacją jest przepuszczany przez filtr Gaussa. W rezultacie zmiana częstotliwości sygnału po modulacji przebiega płynnie, co z kolei umożliwia ograniczenie szerokości widma. Modulacja ta jest używana m. in. w komunikacji Bluetooth.

### Moduły radiowe

FSK jest modulacją powszechnie używaną, dlatego na rynku dostępnych jest wiele modułów radiowych, które można wykorzystać do tego rodzaju

komunikacji. W przykładach użyte zostały modemy SX1276 (umożliwiający także komunikację z modulacją LoRa) oraz S2-LP. Są one dostępne w zestawach ewaluacyjnych pokazanych na **rysunku 2**:

- **B-L072Z-LRWAN1** z układem CMWX1ZZABZ-091 zawierającym modem SX1276 i mikrokontroler STM32L082 w jednej obudowie,
- **STEVAL-FKI868V1** zawierający modem S2-LP i stanowiący rozszerzenie dla płytki NUCLEO-L152RE z mikrokontrolerem STM32L152RE.

Na zasięg transmisji danych wpływa m.in. czułość odbiornika. Jest ona zależna od konfiguracji modemu, czyli od takich parametrów, jak: częstotliwość nośna, dewiacja częstotliwości i prędkość transmisji. W **tabeli 1** i **tabeli 2** przedstawiono wartości czułości odbiorników, wskazywane przez dokumentacje modułów, dla częstotliwości nośnej 868 MHz w zależności od pozostałych parametrów. Z tabel można łatwo wywnioskować, że największą czułość, a przez to największy zasięg można osiągnąć przez maksymalne ograniczenie prędkości transmisji i dewiacji częstotliwości.

### Przykładowy program

Testową komunikację nawiążemy za pomocą zestawów NUCLEO-L152RE z rozszerzeniami STEVAL-FKI868V1. Projekt zostanie przygotowany w środowisku SW4STM32. Przed przystąpieniem do tworzenia projektu trzeba pobrać sterowniki do modułu S2-LP (STSW-S2LP-DK) ze strony (<https://goo.gl/EqAFg1>). Po zainstalowaniu pakietu mamy do dyspozycji sterowniki, przykładowe projekty i aplikację graficzną umożliwiającą konfigurację modemu. Najważniejsze są oczywiście sterowniki, które zostaną wykorzystane w prezentowanej aplikacji. Projekt w środowisku SW4STM32 należy utworzyć korzystając z opcji *File → New → C Project*, wybierając konfigurację dla płytki NUCLEO-L152RE i biblioteki Cube HAL (**rysunek 3**). Następnie należy skopiować do projektu następujące źródła z katalogu w którym zostały wcześniej zainstalowane biblioteki:

- Do katalogu głównego projektu:
  - *Projects/Drivers/BSP/Components/S2LP – sterowniki do modemu radiowego S2-LP,*
  - *Projects/Drivers/BSP/STM32\_Nucleo – biblioteki obsługi peryferiów płytki NUCLEO-L152RE.*
- Do katalogu inc/:

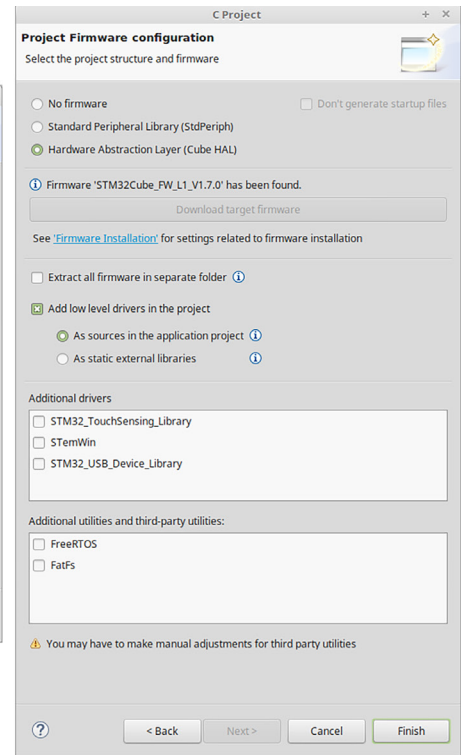
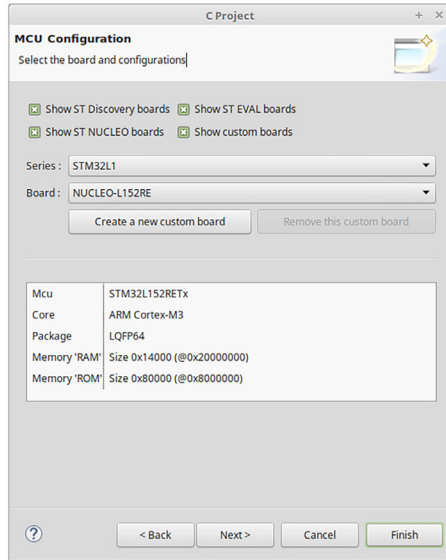
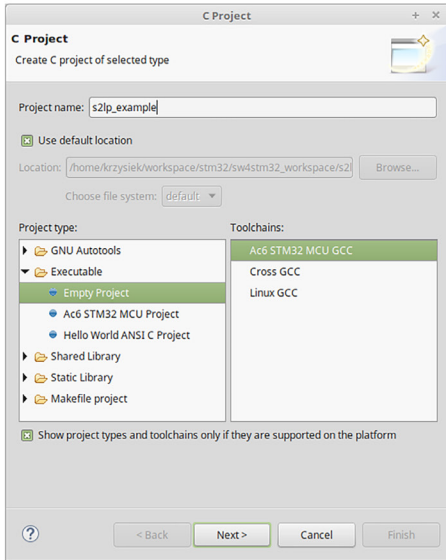
**Tabela 1. Czuość odbiornika modułu S2-LP (źródło: st.com)**

Parameter	Test conditions	SMPS on	Unit
Sensitivity 1% BER @ 2-GFSK BT = 0.5	DR = 0.3 kbps, FDEV = 0.25 kHz, CHF = 1 kHz	-129	dBm
	DR = 1.2 kbps, FDEV = 1.2 kHz, CHF = 4 kHz	-123	
	DR = 38.4 kbps, FDEV = 20 kHz, CHF = 100 kHz	-110	
	DR = 100 kbps, FDEV = 50 kHz, CHF = 200 kHz	-107	

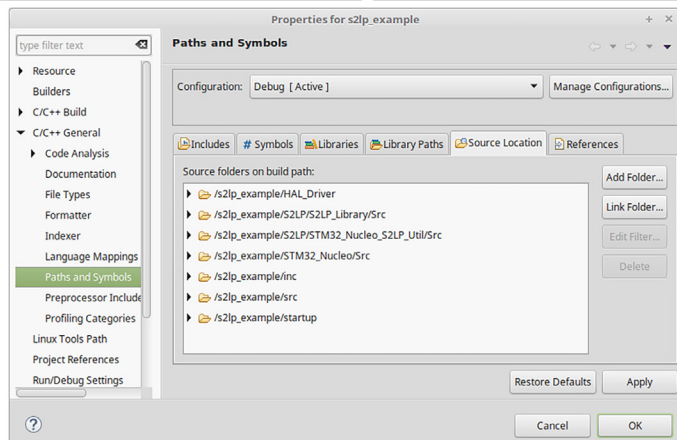
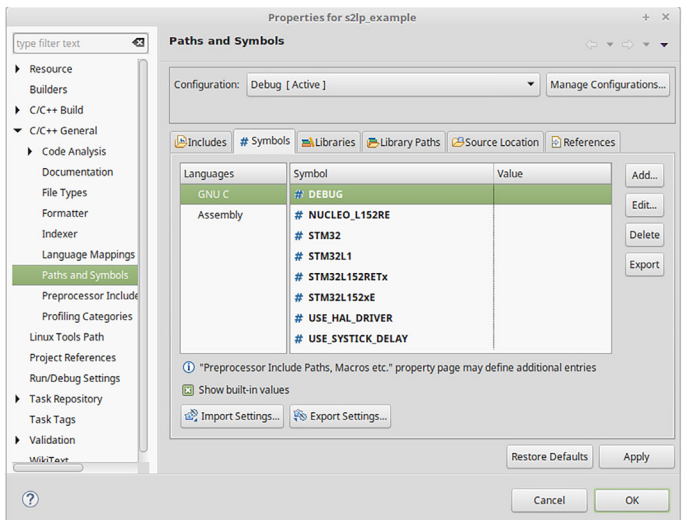
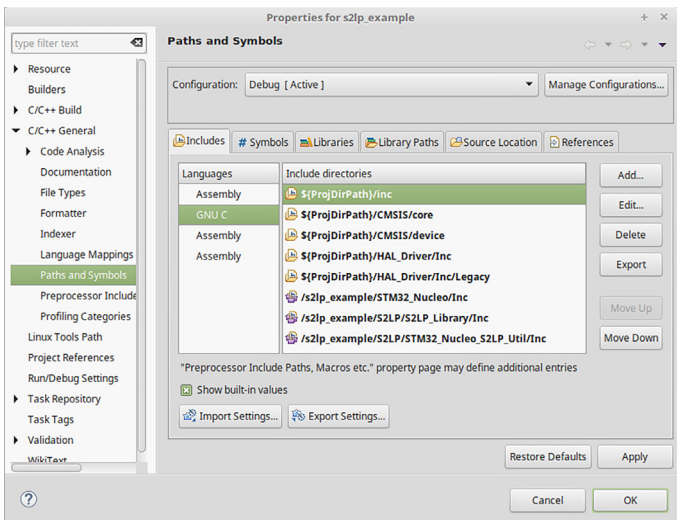
**Tabela 2. Czuość odbiornika modułu SX1276 (źródło: semtech.com)**

Symbol	Description	Conditions	Min	Typ	Max	Unit
RFS_F_HF	Direct tie of RFI and RFO pins, shared Rx, Tx paths FSK sensitivity, highest LNA gain. Band 1	FDA = 5 kHz, BR = 1.2 kb/s	-	-119	-	dBm
		FDA = 5 kHz, BR = 4.8 kb/s	-	-115	-	dBm
		FDA = 40 kHz, BR = 38.4 kb/s*	-	-105	-	dBm
		FDA = 20 kHz, BR = 38.4 kb/s**	-	-105	-	dBm
		FDA = 62.5 kHz, BR = 250 kb/s***	-	-92	-	dBm
	Split RF paths, LnaBoost is turned on, the RF switch insertion loss is not accounted for. Band 1	FDA = 5 kHz, BR = 1.2 kb/s	-	-123	-	dBm
		FDA = 5 kHz, BR = 4.8 kb/s	-	-119	-	dBm
		FDA = 40 kHz, BR = 38.4 kb/s*	-	-109	-	dBm
		FDA = 20 kHz, BR = 38.4 kb/s**	-	-109	-	dBm
		FDA = 62.5 kHz, BR = 250 kb/s***	-	-96	-	dBm

- Projects/Projects\_Cube/S2LPLibrary\_Examples/Inc/SDK\_EVAL\_Config.h,
  - Projects/Projects\_Cube/S2LPLibrary\_Examples/Inc/cube\_hal.h,
  - Projects/Projects\_Cube/S2LPLibrary\_Examples/Inc/MCU\_Interface.h,
  - Projects/Projects\_Cube/S2LPLibrary\_Examples/Inc/stm32l1xx\_it.h.
- Do katalogu src/:
    - Projects/Projects\_Cube/S2LPLibrary\_Examples/Src/stm32l1xx\_it.c.
- Z uwagi na powtarzające się definicje funkcji obsługi UART i SPI, należy usunąć z projektu pliki:



Rysunek 3. Tworzenie nowego projektu



Rysunek 4. Ścieżki i symbole w ustawieniach projektu

- STM32\_Nucleo/Src/SDK\_EVAL\_Com\_UART\_DMA\_TX.c,
- STM32\_Nucleo/Src/SDK\_EVAL\_Spi\_Driver\_DMA.c.

Należy także usunąć katalog *Utilities*, który został automatycznie dodany do projektu, ponieważ zdefiniowane w znajdujących się w nim plikach są również w skopiowanych przed chwilą źródłach.

Po skopiowaniu plików należy dodać ścieżki do źródeł i nagłówków w ustawieniach projektu. Potrzebna będzie też definicja symbolu `USE_SYSTICK_DELAY`. Wszystkie ustawienia zostały przedstawione na **rysunku 4**.

Po przygotowaniu bibliotek i skonfigurowaniu projektu można przystąpić do implementacji prostej komunikacji pomiędzy dwoma urządzeniami. Program został napisany na podstawie przykładów znajdujących się w pakiecie STSW-S2LP-DK. Składa się on z dwóch funkcji `main()`, zawierającej konfigurację mikrokontrolera i modułu radiowego oraz `HAL_GPIO_EXTI_Callback()`, obsługującej przerwania od wszystkich linii GPIO.

Fragment funkcji `main()`, zawierający pełną konfigurację mikrokontrolera oraz moduły S2-LP pokazano na **listingu 1**. Rozpoczyna się on od inicjalizacji biblioteki Cube HAL, SPI i pinu `Shutdown` dla modułu radiowego, oraz GPIO dla przycisku i diody. W tej sekcji znajduje się również inicjalizacja znajdującej się na płycie pamięci EEPROM zawierającej m.in. fabrycznie zapisaną poprawkę do częstotliwości nośnej wykorzystywaną do korekcji wartości zmiennej `radioConfig.IFrequencyBase`.

Kolejne wywołania to reset modułu S2-LP, synchronizacja zegara i konfiguracja linii przerwań służącej do powiadamiania mikrokontrolera. Następnie wysyłana jest komenda `SABORT`, która powoduje wyjście ze stanu transmisji, lub odbioru i przejście do stanu `READY`. W tym ostatnim stanie możliwa jest konfiguracja parametrów radia, przekazywanych do funkcji `S2LPRadioInit()` za pomocą struktury `SRadioInit`. W przykładzie radio zostało skonfigurowane następującymi parametrami:

```
SRadioInit radioConfig = {
    .IFrequencyBase = 868.0e6,
    .xModulationSelect = MOD_2FSK,
    .IDatarate = 300,
    .IFreqDev = 250,
    .IBandwidth = 5000};
```

Zgodnie z nimi częstotliwość nośna wynosi 868 MHz, prędkość transmisji 300 bps, dewiacja częstotliwości 250 Hz, a pasmo odbiornika 5000 Hz. Typ modulacji został ustawiony na 2FSK. Jest to zwykła modulacja FSK z użyciem dwóch częstotliwości, w przeciwieństwie do 4FSK, w której do transmisji używa się czterech częstotliwości, co umożliwia transfer dwóch bitów jednocześnie.

Kolejne wywołania to ustawienie mocy nadawczej na 14 dBm oraz parametrów filtracji RSSI. Jest to metoda detekcji pakietów przychodzących na podstawie zmian mocy sygnału. W przykładzie została ona skonfigurowana następująco:

```
SRssiInit rssiInit = {
    .cRssiFlt = 15,
    .cRssiThreshdBm = -130,
    .xRssiMode = RSSI_DYNAMIC_6DB_STEP_MODE,
};
```

Pierwszym parametrem jest wzmocnienie filtra RSSI w dBm, drugim próg RSSI po przekroczeniu którego pakiet danych zostanie odebrany, a ostatnim tryb filtracji. Tryb statyczny oznacza odbiór danych bezpośrednio po przekroczeniu progu, natomiast w trybie dynamicznym oprócz poziomu sprawdzany jest również przyrost RSSI. W kolejnej sekcji konfigurowane są pakiet danych za pomocą struktury:

```
PktBasicInit xBasicInit={
    .xPreambleLength = 16,
    .xSyncLength = 32,
    .lSyncWords = 0x88888888,
    .xFixVarLength = S_ENABLE,
    .cExtendedPktLenField = S_DISABLE,
```

```
.xCrcMode = PKT_CRC_MODE_8BITS,
.xAddressField = S_DISABLE,
.xFec = S_DISABLE,
.xDataWhitening = S_ENABLE,
};
```

Struktura ta zawiera następujące informacje:

- długość preambuły,
- długość i wartość słowa synchronizującego,
- stała lub zmienna długość pakietu,
- rozmiar pola z długością danych (jeden lub dwa bajty),
- typ sumy kontrolnej,
- włączenie lub wyłączenie pola z adresem odbiornika,
- włączenie lub wyłączenie autokorekty danych,
- *data whitening*, czyli zapobieganie transmisji ciągów zer i jedynek.

Następnie jest ustawiana długość danych w pakiecie. Jeśli jest włączona opcja zmiennej długości danych (`xFixVarLength = S_ENABLE`), ustawiona długość dotyczy wyłącznie nadajnika, ponieważ odbiornik otrzyma tą wartość w pakiecie. W przeciwnym razie obie strony muszą mieć jawnie ustawioną długość danych za pomocą funkcji `S2LPPktBasicSetPayloadLength()`.

Na koniec pozostały jeszcze do wybrania źródła przerwań. Z uwagi na fakt, że S2-LP używa tylko jednej linii przerwań, należy wybrać zdarzenia, które będą to przerwanie wywoływać. W przykładzie używane są tylko dwa źródła przerwań – koniec nadawania pakietu i jego udany odbiór.

Po skończonej konfiguracji można przełączyć moduł w stan odbioru pakietów. Czas oczekiwania na dane można skonfigurować funkcją `S2LPTimerSetRxTimerCounter()`. Argument 0 oznacza tryb ciągłego nasłuchu.

Druga z funkcji jest odpowiedzialna za obsługę przerwań GPIO, zarówno od modułu radiowego, jak i od przycisku użytkownika. Została ona przedstawiona na **listingu 2**. Z racji tego, że funkcja ta jest wywoływana przez bibliotekę Cube HAL do obsługi wszystkich zarejestrowanych przerwań GPIO, jest w niej wykonywane źródło przerwania na podstawie argumentu. Jeżeli przerwanie pochodzi od modułu S2-LP, należy najpierw pobrać stany wszystkich możliwych zdarzeń wywołując funkcję `S2LPGpioIrqGetStatus()`. Dla uproszczenia programu w przykładzie obsługiwane są tylko dwa zdarzenia, skonfigurowane wcześniej jako bezpośrednie źródła przerwań: zakończenie transmisji i odbiór pakietu. Pierwsze z nich powoduje wyłączenie diody sygnalizującej stan nadawania i przejście w tryb nasłuchu. Drugie ze zdarzeń powoduje odczyt danych ze sprzętowej kolejki odbiorczej, opróżnienie jej, ponowne przejście w stan nasłuchu oraz ustawienie flagi `rxReady`. Flaga ta jest sprawdzana w pętli `while()` na końcu funkcji `main()`. Jeżeli zostanie ona ustawiona, wówczas na 100 ms jest zaświecana dioda sygnalizująca odbiór danych.

Obsługa drugiej linii przerwań, sygnalizującej wciśnięcie przycisku, sprowadza się do wysłania komendy `ABORT`, która przełącza moduł radiowy w stan `READY`, opróżnienia sprzętowej kolejki nadawczej, nadania pakietu danych i zapalenia diody sygnalizującej stan transmisji danych.

**Listing 1.** Konfiguracja radia w funkcji `main()`

```
SRadioInit radioConfig = {
    .IFrequencyBase = 868.0e6,
    .xModulationSelect = MOD_2FSK,
    .IDatarate = 300,
    .IFreqDev = 250,
    .IBandwidth = 5000};

PktBasicInit xBasicInit={
    .xPreambleLength = 16,
    .xSyncLength = 32,
    .lSyncWords = 0x88888888,
    .xFixVarLength = S_ENABLE,
    .cExtendedPktLenField = S_DISABLE,
    .xCrcMode = PKT_CRC_MODE_8BITS,
    .xAddressField = S_DISABLE,
    .xFec = S_DISABLE,
    .xDataWhitening = S_ENABLE,
};
```

```
Listing 2. Obsługa przerw GPIO w funkcji HAL_GPIO_EXTI_Callback
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == M2S_GPIO_PIN_IRQ)
    {
        S2LPirqs xIrqStatus;
        S2LPGpioIrqGetStatus(&xIrqStatus);
        if(xIrqStatus.IRQ_TX_DATA_SENT)
        {
            SdkEvalLedOff(LED1);
            S2LPCmdStrobeRx();
        }
        if(xIrqStatus.IRQ_RX_DATA_READY)
        {
            uint8_t cRxData = S2LPFifoReadNumberBytesRxFifo();
            uint8_t vectcRxBuff[16];
            S2LPSpiReadFifo(cRxData, vectcRxBuff);
            S2LPCmdStrobeFlushRxFifo();
            S2LPCmdStrobeRx();
            rxReady = 1;
        }
    }
    else if(GPIO_Pin == BUTTON1_PIN)
    {
        SdkEvalLedOn(LED1);
        uint8_t buffer[5] = {1, 2, 3, 4, 5};
        S2LPCmdStrobeSabort();
        S2LPSpiWriteFifo(5, buffer);
        S2LPCmdStrobeTx();
    }
}
```

**Test zasięgu**

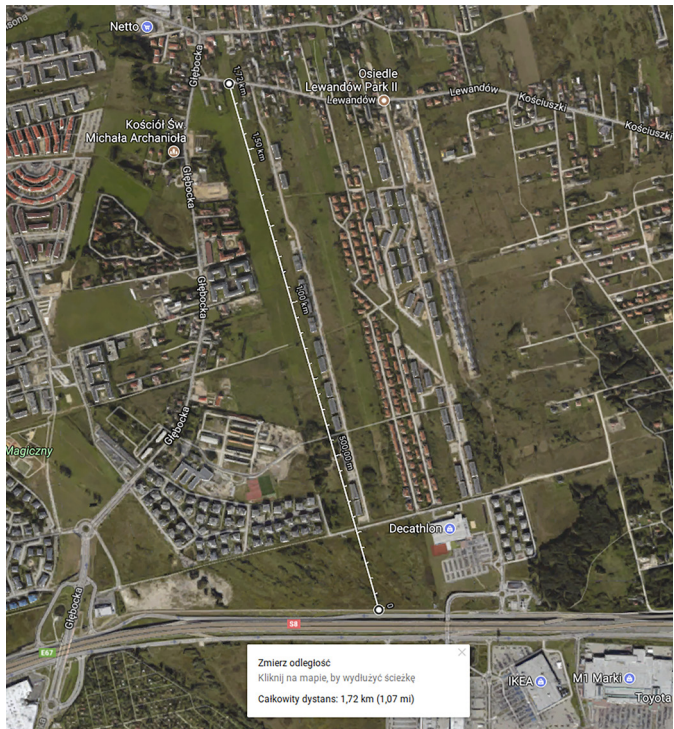
Głównym przeznaczeniem modułów S2-LP jest komunikacja na dużą odległość, dlatego w tym rozdziale przedstawione zostaną testy zasięgu. Podobnie jak w przypadku innym modułów radiowych, w tym także przedstawionych poprzednio SX1276, maksymalny zasięg jest uzależniony od konfiguracji modułu, położenia anten, ukształtowania terenu, przeszkód na drodze sygnału radiowego oraz innych czynników zewnętrznych. W teście zostały wykorzystane zarówno opisywane w artykule moduły S2-LP, jak i SX1276 w trybie modulacji FSK. Konfiguracja modemów została przedstawiona w tabeli 3.

W obu przypadkach anteny były umieszczone na wysokości ok. 1,5 m nad ziemią, a na drodze sygnału nie znajdowały się żadne przeszkody, co można zobaczyć na rysunku 5. Maksymalny zasięg jaki udało się osiągnąć to ok. 1720 m. Zwiększenie odległości w przypadku obu modułów powodowało utratę pakietów danych.

**Podsumowanie**

Moduł radiowy S2-LP oparty na modulacji FSK jest układem dobrze spisującym się w komunikacji na dużą odległość. W zależności od wymagań projektu może być on skonfigurowany do różnych prędkości transmisji danych, co ma jednak wpływ na maksymalny zasięg

Moduł	S2-LP	SX1276
Modulacja	FSK	FSK
Częstotliwość nośna	868 MHz	868 MHz
Dewiacja częstotliwości	600 Hz	600Hz
Prędkość transmisji danych	1200 bps	1200 bps
Pasma odbiornika	5000 Hz	2600 Hz
Moc nadajnika	14 dBm	14 dBm



**Rysunek 5. Punkty pomiarowe z zaznaczoną odległością (źródło maps.google.pl)**

sygnału radiowego. Dodatkowo, dostępne biblioteki i przykłady aplikacji sprawiają, że rozpoczęcie pracy z S2-LP jest stosunkowo proste i nie wymaga dużego nakładu pracy.

Krzysztof Chojnowski

