

LoRa w praktyce (1)

Komunikacja bezprzewodowa jest obecnie jedną z najpopularniejszych metod budowania sieci czujników. Na rynku dostępna jest cała gama interfejsów, które można wykorzystać w zależności od wymagań projektu. Jednym z ich, opisywanym w artykule, jest stworzony przez firmę Semtech standard komunikacji LoRa.

LoRa jest interfejsem radiowym działającym m.in. w pasmach ISM 434 MHz i 868 MHz umożliwiającym komunikację o dużym zasięgu przy niewielkim poborze mocy. Dzięki temu świetnie nadaje się między innymi do budowy sieci zasilanych bateryjnie czujników rozproszonych na dużych obszarach. Odległości pomiędzy komunikującymi się ze sobą urządzeniami mogą być liczone w kilometrach – w zależności od ukształtowania i typu terenu. Osiągnięcie takiego zasięgu wiąże się jednak z ceną, którą jest maksymalny transfer danych, który może wahać się, w zależności od konfiguracji modemu od kilobajtów, do pojedynczych bajtów na sekundę.

Modulacja

Przepustowość kanału informacyjnego, pasmo oraz stosunek sygnału do szumu są ze sobą związane zależnością zwaną twierdzeniem Shannona-Hartleya. Wygląda ona następująco:

$$C = B \cdot \log_2 \left(1 + \frac{S}{N} \right), \text{ gdzie:}$$

C – przepustowość kanału w bitach na sekundę,

B – pasmo sygnału w hercach

S/N – stosunek mocy sygnału do mocy szumów.

Z twierdzenia tego można wyciągnąć następujące wnioski:

- Zwiększenie pasma sygnału umożliwia zwiększenie transferu przy stałym stosunku mocy sygnału do szumu.
- Spadek stosunku mocy sygnału do szumu (np. na skutek zwiększenia odległości między nadajnikiem, a odbiornikiem) wymaga zwiększenia pasma przy zachowanej przepustowości łącza.

Na powyższych założeniach oparta jest m.in. modulacja CSS (*Chirp Spread Spectrum*), w której strumień danych modulowany jest sygnałem o liniowo wzrastającej częstotliwości. Na tej technice bazuje modulacja LoRa, której zarejestrowany spektrogram został przedstawiony na **rysunku 1**. Z lewej strony pokazana została tzw. preambuła, niezawierająca danych, na której widać w jaki sposób wygląda sygnał modulujący, którego zakres częstotliwości definiuje szerokość pasma. Zmodulowany sygnał został przedstawiony na przebiegu po prawej stronie.

Sposób, w jaki przebiega modulacja sygnału jest zależny od trzech głównych parametrów:

- BW (*modulation bandwidth*) – opisuje w jakim zakresie zmienia się częstotliwość modulująca,
- SF (*spread factor*) – określa jak szybko zmienia się częstotliwość modulująca,
- CR (*code rate*) – wprowadza redundancję zapewniając jednocześnie korekcję błędów powstałych podczas transmisji.

Parametry te wpływają na maksymalny zasięg oraz przepustowość łącza, która jest wyrażona zależnością:

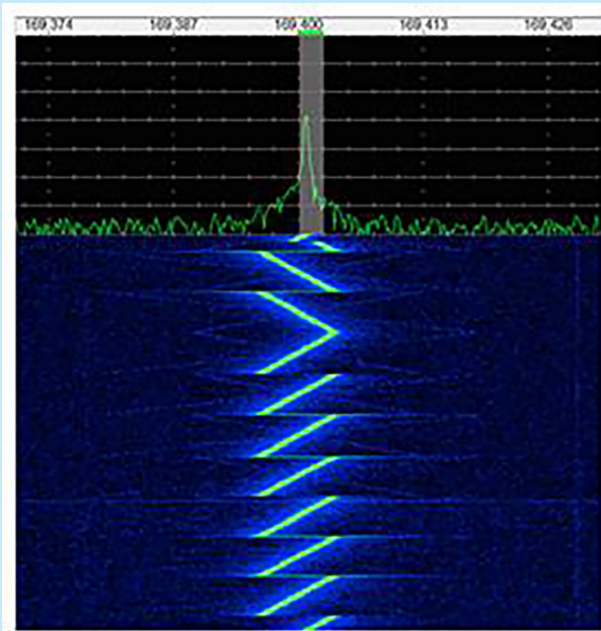
$$R_b \left[\frac{\text{b}}{\text{s}} \right] = SF \cdot \frac{CR \cdot BW [\text{Hz}]}{2^{SF}}$$

Parametry mogą przyjmować następujące wartości:

BW {7,8 kHz, 10,4 kHz, 15,6 kHz, 20,8kHz, 31,25 kHz, 41,7 kHz, 62,5 kHz, 125 kHz, 250 kHz, 500 kHz},

SF {6, 7, 8, 9, 10, 11, 12},

CR {4/5, 4/6, 4/7, 4/8}.



Rysunek 1. Modulacja LoRa (źródło: digikey.com)

Łatwo więc policzyć, że maksymalny transfer wynosi 37,5 kb/s, natomiast minimalny zaledwie ok. 11,5 b/s. Należy jednak pamiętać, że w pakiecie danych radiowych znajdują się także preambuła oraz opcjonalny nagłówek, które dodatkowo zmniejszają transfer danych użytecznych.

Modulację LoRa pokazano na **rysunku 1**.

Modemy LoRa

Rodzinę dostępnych na rynku modemów LoRa można podzielić na dwie grupy. Pierwsza z nich obejmuje układy SX1272 i SX1273, które są przystosowane do pracy z częstotliwościami z zakresu 860..1020 MHz. Druga grupa to układy SX1276, SX1277, SX1278 i SX1279 mogące pracować z częstotliwościami od 137 do 1020 MHz, zależnie od modułu. W porównaniu do układów z pierwszej grupy mają one większą czułość odbiorników

```

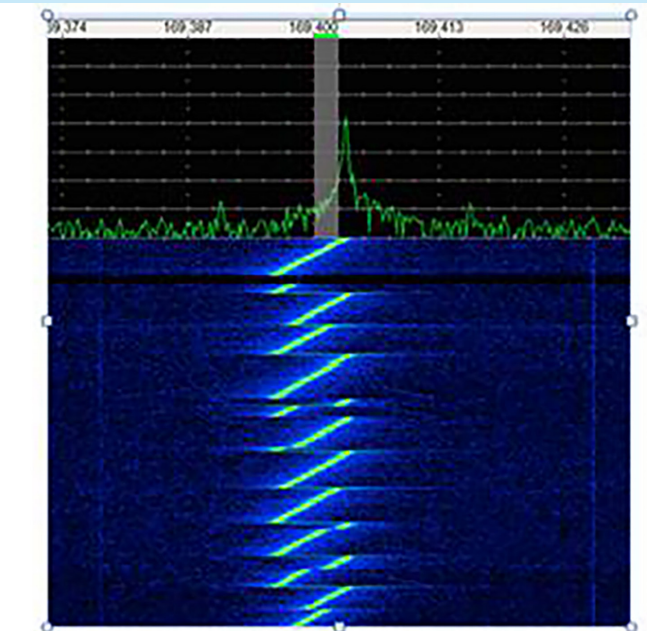
Listing 1. Funkcja main
#include <stdbool.h>
#include „stm3210xx.h”
#include „hw_gpio.h”
#include „hw_msp.h”
#include „radio.h”
#include „vcom.h”
#include „timeServer.h”

static TimerEvent_t ledTimer;
static uint8_t* buffer = „Hello LoRa”;

#define LORA_FREQUENCY 868000000
#define LORA_TX_POWER 14
#define LORA_BANDWIDTH 2//0
#define LORA_DATARATE 7//10
#define LORA_CODERATE 1
#define LORA_PREAMBLE_LEN 8

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    HW_Init();
    RadioEvents_t radioEvents;
    radioEvents.TxDone = txDoneEventCallback;
    radioEvents.RxDone = rxDoneEventCallback;
    Radio_Init(&radioEvents);
    Radio.SetChannel(LORA_FREQUENCY);
    Radio.SetTxConfig(MODEM_LORA, LORA_TX_POWER, 0, LORA_BANDWIDTH, LORA_DATARATE, LORA_CODERATE, LORA_PREAMBLE_LEN, false, true, false, 0, 0, 3000000);
    Radio.SetRxConfig(MODEM_LORA, LORA_BANDWIDTH, LORA_DATARATE, LORA_CODERATE, 0, LORA_PREAMBLE_LEN, 1000, false, 0, true, false, 0, false, true);
    Radio.Rx(0);
    TimerInit(&ledTimer, ledTimerCallback);
    TimerSetValue(&ledTimer, 500);
    HW_GPIO_SetIrq(GPIOB, GPIO_PIN_2, 0, gpioCallback);
    BSP_PB_Init(BUTTON_USER, BUTTON_MODE_EXTI);
    for(;;);
}

```



radiowych, a ich pasmo zaczyna się już od 7,8 kHz, w przeciwieństwie do pierwszej grupy modemów, gdzie dopuszczalne pasmo zaczyna się od 125 kHz. Na **rysunku 2** można zobaczyć dokładne parametry poszczególnych modułów zaczerpnięte z oficjalnej dokumentacji firmy Semtech.

Zestawy ewaluacyjne

Dostępne na rynku zestawy deweloperskie z modemami Semtech LoRa można podzielić na dwie grupy. Pierwszą z nich stanowią płytki zawierające jeden z wyżej wymienionych modemów z możliwością podłączenia dowolnego zewnętrznego mikrokontrolera za pośrednictwem SPI. Przykładami tego typu zestawów są pokazane na **rysunku 3**:

- Adafruit RFM95W,
- Semtech SX1276MB1LAS,
- Semtech SX1272MB2DAS.

W drugiej grupie zestawów ewaluacyjnych znajdują się moduły, które oprócz modemów LoRa zawierają mikrokontrolery, dzięki czemu nie wymagają dołączania zewnętrznych układów przed rozpoczęciem pracy. Przykładami takich układów są (**rysunek 4**):

- Adafruit Feather M0 RFM95 LoRa Radio z mikrokontrolerem ATSAM21G18 (Cortex M0),
- STMicroelectronics B-L072Z-LRWAN1 z modułem CMWX1ZZABZ-091 (mikrokontroler STM32L072CZ z modemem SX1276 w jednej obudowie).

Przykładowy program

Przedstawimy teraz krótki program umożliwiający komunikację radiową pomiędzy dwoma urządzeniami. Przykład został przygotowany dla zestawu deweloperskiego B-L072Z-LRWAN1 od STMicroelectronics i wykorzystuje bibliotekę STM32Cube z rozszerzeniem LRWAN. Biblioteka podstawowa STM32Cube

Part Number	Frequency Range	LoRa™ Parameters			
		Spreading Factor	Bandwidth	Effective Bitrate	Sensitivity
SX1272	860 - 1020 MHz	6 - 12	125 - 500 kHz	0.24 - 37.5 kbps	-117 to -137 dBm
SX1273	860 - 1020 MHz	6 - 9	125 - 500 kHz	1.7 - 37.5 kbps	-117 to -130 dBm
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

Rysunek 2. Parametry dostępnych na rynku modemów LoRa (źródło: semtech.com)

Tabela 1. Parametry konfiguracji transmisji i odbioru

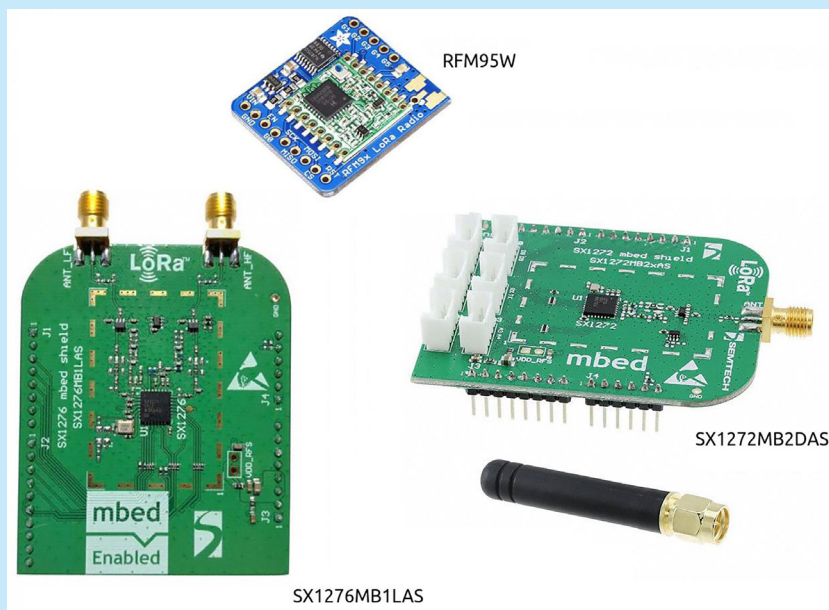
SetTxConfig		SetRxConfig	
Argument	Opis	Argument	Opis
modem	Typ modulacji (LoRa/FSK).	modem	Typ modulacji (LoRa/FSK)
power	Moc sygnału nadawanego [dBm].	bandwidth	Pasma (LoRa 0: 125 kHz, 1: 250 kHz, 2: 500 kHz, FSK <2,6kHz:250kHz>).
fdev	Dewiacja częstotliwości (tylko FSK).	datarate	Przepustowość (dla LoRa wyrażona jako 2^{SF} , gdzie SF = {6..12}, dla FSK jako bit/s).
bandwidth	Pasma (tylko LoRa, 0: 125 kHz, 1: 250 kHz, 2: 500 kHz).	coderate	Korekcja błędów (tylko LoRa: 1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8).
datarate	Przepustowość (dla LoRa wyrażona jako 2^{SF} , gdzie SF = {6..12}, dla FSK jako bit/s).	bandwidthAfc	Pasma automatycznej regulacji częstotliwości (tylko FSK <2,6kHz:250kHz>)
coderate	Korekcja błędów (tylko LoRa: 1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8).	preambleLen	Długość preambuły pakietu.
preambleLen	Długość preambuły pakietu.	symbTimeout	Maksymalny czas trwania trybu odbioru pojedynczego pakietu wyrażony w liczbie symboli (tylko LoRa).
fixLen	Stała/zmienna długość pakietu danych.	fixLen	Stała/zmienna długość pakietu danych.
crcOn	Włączona/wyłączona generacja CRC.	payloadLen	Długość pakietu danych, gdy fixLen = true.
freqHopOn	Włączona/wyłączona opcja rozpraszania widma Frequency-hopping (tylko LoRa).	crcOn	Włączona/wyłączona kontrola CRC.
hopPeriod	Okres zmiany częstotliwości jeśli freqHopOn = true (tylko LoRa).	freqHopOn	Włączona/wyłączona opcja rozpraszania widma Frequency-hopping (tylko LoRa).
iqInverted	Inwersja sygnałów I oraz Q (tylko LoRa).	hopPeriod	Okres zmiany częstotliwości jeśli freqHopOn = true (tylko LoRa).
timeout	Ograniczenie czasu transmisji w [ms].	iqInverted	Inwersja sygnałów I oraz Q (tylko LoRa).
-	-	rxContinuous	Włączony/wyłączony tryb ciągłego odbioru.

dostarcza sterowników do peryferiów mikrokontrolera STM32L072CZ i jest używana przez wyższe warstwy oprogramowania oraz rozszerzenie I-CUBE-LRWAN, które zawiera sterownik do modemu SX1276 oraz stos LoRaWAN. Biblioteka podstawowa jest dostępna na stronie http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-software/i-cube-lrwan.html.

Pracę zaczynamy od utworzenia nowego projektu dla zestawu B-L072Z-LRWAN1 - kolejne kroki z wymaganymi opcjami zostały przedstawione na **rysunku 5**. W ostatnim z przedstawionych okien należy pobrać bibliotekę STM32Cube i dodać ją do źródeł projektu. W projekcie potrzebne też będą elementy rozszerzenia I-CUBE-LRWAN:

- Middlewares/Third_Party/Lora/Phy
- Middlewares/Third_Party/Lora/Utilities
- Drivers/BSP/MLM32L07X01
- Drivers/BSP/B-L072Z-LRWAN1

Powyższe moduły wymagają dostarczenia funkcji obsługi takich peryferiów jak GPIO, RTC i SPI. Znajdują się one w przykładach aplikacji dołączonych do zestawu B-L072Z-LRWAN1. Do opisywanego przykładu zostały dołączone pliki z aplikacji *Projects/Multi/Applications/LoRa/PingPong*: *debug.c*, *debug.h*, *hw.h*, *hw_conf.h*, *hw_msp.h*, *hw_gpio.c*, *hw_gpio.h*, *hw_rtc.c*, *hw_rtc.h*, *hw_spi.c*, *hw_spi.h*, *mlm32l0xx_hal_msp.c*, *mlm32l0xx_hw.c*, *mlm32l0xx_hw_conf.h*, *mlm32l0xx_it.c*, *vcom.c*, *vcom.h*.



Rysunek 3. Zestawy ewaluacyjne z modemem LoRa (źródło: digikey.com)

W pliku *mlm32l0xx_it.c* znajdują się definicje wszystkich potrzebnych funkcji obsługi przerwań, dlatego można automatycznie usunąć utworzony plik *stm32l0xx_it.c*. Wszystkie używane przerwania wymagają także umieszczenia ich w tablicy znajdującej się w pliku *startup_stm32.s*. Można dopisać brakujące elementy wraz z deklaracjami typu *.weak* lub zamienić istniejący plik na *Projects/Multi/Applications/LoRa/PingPong/SW4STM32/B-L072Z-LRWAN1/startup_stm32l072xx.s*. Ostateczną listę plików w projekcie można znaleźć na **rysunku 6**.

Po dodaniu wszystkich źródeł trzeba uzupełnić ścieżki w projekcie. Można to zrobić w ustawieniach projektu: *C/C++ General* → *Paths and Symbols*. W zakładce *Includes* należy dodać ścieżki

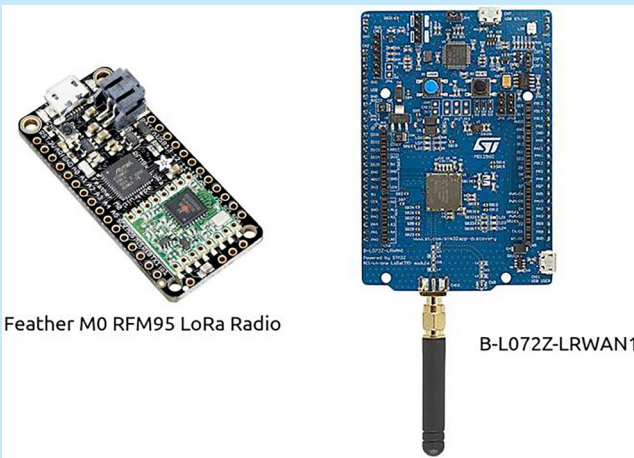
```

Listing 2. Funkcje obsługujące zdarzenia radia oraz przerwania
licznika i przycisku
static void gpioCallback()
{
    BSP_LED_On(LED3);
    Radio.Send(buffer, 10);
}

static void ledTimerCallback()
{
    BSP_LED_Off(LED2);
}

void txDoneEventCallback()
{
    BSP_LED_Off(LED3);
    Radio.Rx(0);
}

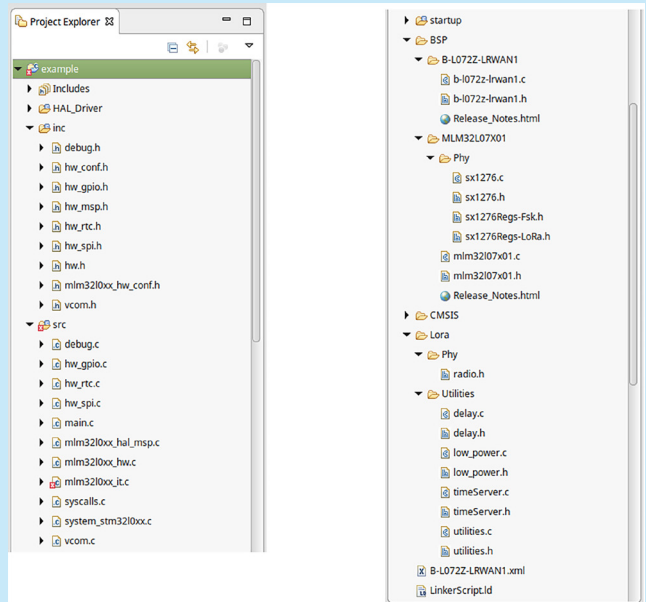
void rxDoneEventCallback(uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr)
{
    BSP_LED_On(LED2);
    TimerStart(&ledTimer);
    PRINTF(„->”);
    for(int i=0; i<size; i++) PRINTF(„%c”, payload[i]);
    PRINTF(„\n”);
}
    
```



Rysunek 4. Zestawy ewaluacyjne z modemem LoRa oraz mikrokontrolerem (źródło: digikey.com)

do nagłówków, a w zakładce *Source Location* ścieżki do plików źródłowych. Przy dodawaniu ścieżek warto zaznaczyć opcje dodania ich do wszystkich konfiguracji. Rozszerzenie I-CUBE-LRWAN potrzebuje do poprawnego działania także odpowiednich symboli - podobnie jak w przypadku ścieżek warto dodać je do wszystkich języków i konfiguracji. Wszystkie ustawienia zostały przedstawione na **rysunku 7**.

Próba kompilacji tak skonfigurowanego projektu zakończy się błędem z powodu wielokrotnej definicji funkcji *HAL_MspInit*.



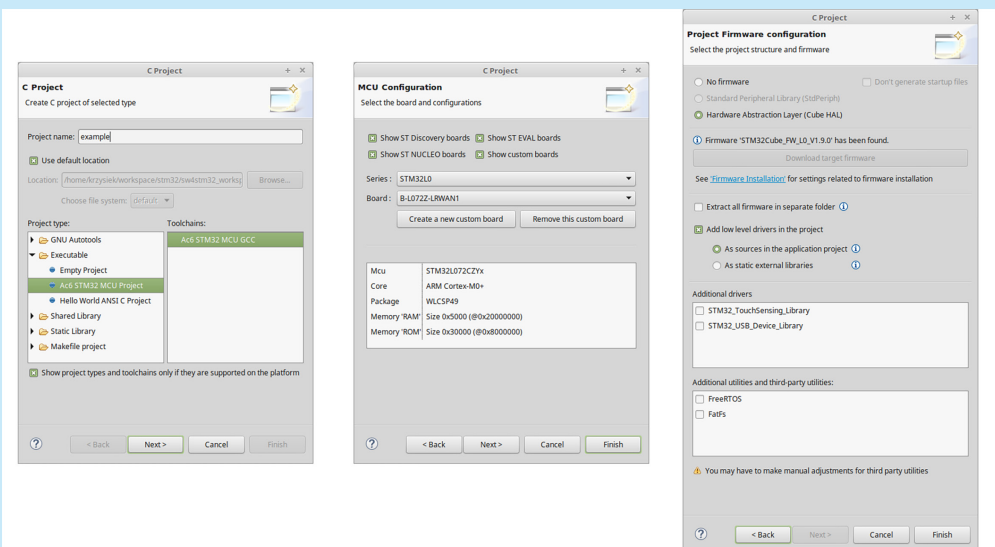
Rysunek 6. Lista plików projektu

Problem ten można rozwiązać usuwając plik *HAL_Driver/Src/stm32l0xx_hal_msp_template.c* dostarczający pustych definicji funkcji konfiguracyjnych.

Po pomyślnym skonfigurowaniu projektu z biblioteką I-CUBE-LRWAN można przystąpić do implementacji prostej komunikacji radiowej pomiędzy dwoma węzłami, których rolę pełnią dwa zestawy B-L072Z-LRWAN1. Na **listingu 1** została umieszczona funkcja *main*, w której znajduje się pełna konfiguracja radia. Pierwsze trzy funkcje są odpowiedzialne za inicjalizację biblioteki STM32Cube (*HAL_Init*), zegara systemowego (*SystemClock_Config*, wygenerowana przy tworzeniu projektu), sterownika radia i biblioteki I-CUBE-LRWAN (*HW_Init*).

Następna sekcja to inicjalizacja modułu radiowego i konfiguracja parametrów transmisji. W pierwszej kolejności rejestrowane są callbacki wołane po skończonej transmisji i po odbiorze pakietu. Dla uproszczenia zostały pominięte pozostałe, odpowiadające m. in. za błędy transmisji i detekcję zajętości kanału radiowego. Użyte funkcje zostaną omówione w dalszej części tekstu. Kolejne linie kodu konfiguruje parametry modemu radiowego. Funkcja *SetChannel* ustawia częstotliwość sygnału, natomiast pozostałe dwie funkcje konfiguruje parametry transmisji i odbioru. Listy argumentów obu funkcji zostały opisane w **tabeli 1**. Istotne w tym przypadku jest, aby

zarówno odbiornik, jak i nadajnik były skonfigurowane w ten sam sposób. W tym momencie radio gotowe jest już do komunikacji, więc zostaje wprowadzone w tryb ciągłego nasłuchu (argument 0 funkcji Rx oznacza brak limitu czasu na odbiór pakietu). W dalszej części programu konfigurowany jest licznik sterujący czasem świecenia diody sygnalizującej odbiór pakietu. Korzysta on z interfejsu dostarczanego przez moduł *Lora/Utilities/timeServer*. Dzięki niemu można w łatwy sposób zarządzać licznikami opartymi na sprzętowym RTC. Jest on używany przez



Rysunek 5. Tworzenie nowego projektu dla zestawu B-L072Z-LRWAN1 w środowisku AC6

sterownik SX1276, ale można go także użyć w swoim kodzie. Ostatnią czynnością wykonywaną przez funkcję *main* jest skonfigurowanie przycisku służącego do nadawania pakietu. Przerwanie jest włączane za pomocą funkcji dostarczanej przez moduł *hw_gpio*, natomiast konfiguracja GPIO znajduje się w BSP płytki B-L072Z-LRWAN1.

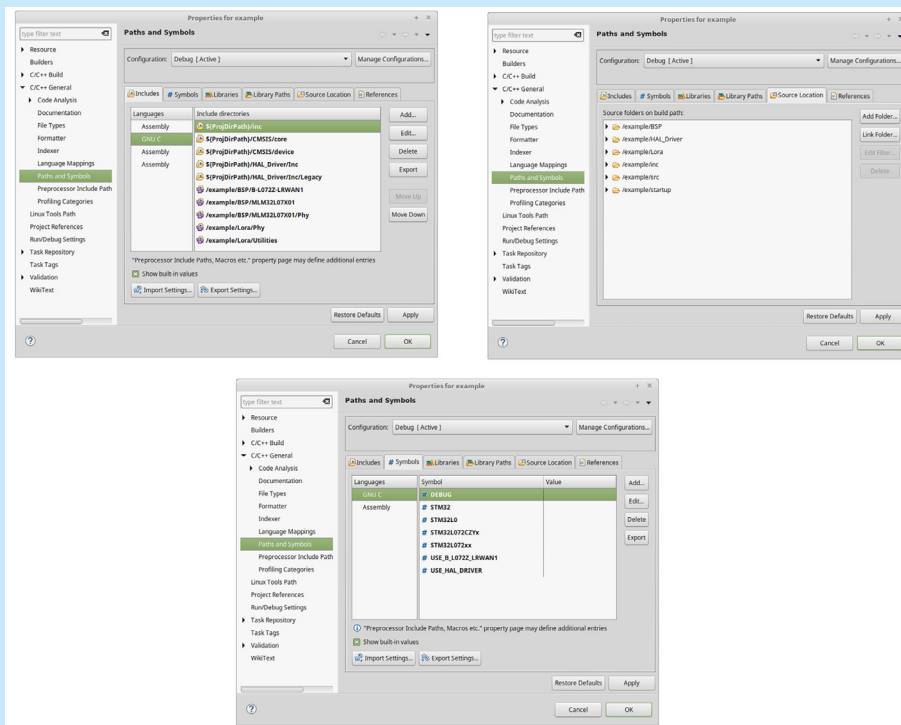
Do omówienia pozostały funkcje obsługi przerwań przedstawione na liście 2:

- **gpioCallback** przerwanie od przycisku, które przełącza radio w tryb nadawania pakietu, zaświeca diodę sygnalizującą nadawanie pakietu,
- **txDoneEventCallback** przerwanie po skończonej transmisji pakietu, przełącza radio z powrotem w tryb nasłuchu, gasi diodę nadawczą,
- **rxDoneEventCallback** przerwanie po odebraniu pakietu, zapala diodę sygnalizującą odbiór, uruchamia licznik kontrolujący czas świecenia diody, wypisuje treść otrzymanej wiadomości,
- **ledTimerCallback** przerwanie licznika kontrolującego diodę.

Komentarza wymaga jeszcze funkcja *PRINTF*. W rzeczywistości jest to makro dostarczane przez moduł *vcom*, który pozwala na łatwe wypisywanie komunikatów na port szeregowy USART2 (TX: PA2, RX: PA3). Domyślnie jest on skonfigurowany w następujący sposób:

- BaudRate 115200 bps,
- WordLength 8 bitów,
- StopBits 1,
- Parity None,
- HardwareFlowControl None.

Z makra *PRINTF* korzysta także driver SX1276, o czym można się przekonać przeglądając plik źródłowy *BSP/MLM32L07X01/Phy/*



Rysunek 7. Konfiguracja ścieżek i symboli projektu

sx1276.c. Można je wyłączyć modyfikując odpowiednio zawartość pliku *vcom.h*.

Po przyłączeniu płytki do komputera należy kliknąć prawym przyciskiem myszy na nazwę projektu w drzewie plików po lewej stronie i wybrać opcję *Debug As → Ac6 STM32 C/C++ Application* aby zaprogramować pamięć mikrokontrolera.

Działanie przykładu jest następujące. Po przyciśnięciu przycisku urządzenie wysyła zdefiniowany w kodzie pakiet danych, co sygnalizuje niebieską diodą. Drugie urządzenie odbiera pakiet, co sygnalizuje zieloną diodą oraz wypisuje zawartość pakietu na port USART2. Jest to bardzo prosty przykład komunikacji za pomocą modemu LoRa SX1276, który pozwala na zapoznanie się z bibliotekami dostarczonymi przez ST. W ramach eksperymentów warto zmodyfikować parametry transmisji, jednocześnie obserwując czas nadawania sygnalizowany przez diodę LED.

Krzysztof Chojnowski

Na niemal 200 stronach przystępnym językiem przybliżamy zagadnienia warsztatowe i zasady kompozycji, podpowiadamy, jak najlepiej wykorzystać swój sprzęt i czego nie może zabraknąć w torbie ambitnego fotografa krajoobrazu!

przejrzysz i kupisz na

www.ulubionykiosk.pl