



**AKADEMIA GÓRNICZO-HUTNICZA**

Dokumentacja do projektu

# **Biblioteka do obsługi ENC28J60**

z przedmiotu

## **Języki Programowania Obiektowego**

Elektronika i Telekomunikacja, III rok

*Adrian Homa, Kamil Russ*

Piątek, godzina 11:30

prowadzący: Mgr Inż. Jakub Zimnol

03.01.2025

# 1. Cel projektu

Celem projektu było utworzenie uniwersalnego sterownika do modułu ENC28J60 wraz z obsługą stosowanego w module protokołu SPI. Komunikacja SPI zostaje zapewniona poprzez bezpośrednie wysyłanie bajtów oraz całego pakietu danych z wykorzystaniem DMA. Do projektu zostały wykorzystane przerwania DMA oraz EXTI w przypadku pinu INT sygnalizującego przerwanie modułu ENC28J60. Do celów debugowania, stworzona została osobna biblioteka implementująca protokół UART w sposób równie uniwersalny jak biblioteka SPI. Całość posłuży w przyszłości do zbudowania większego projektu konwertera ART-NET->DMX512 z wykorzystaniem wyżej wymienionych bibliotek, obsługującego protokół ARP oraz pakiety ART-NET (pakiety UDP). Cały projekt został wykonany na mikrokontroler STM32F446RETx.

## 2. Struktura projektu

Projekt został podzielony na trzy główne foldery, które organizują kod następująco:

- Src – kody źródłowe

- main.cpp – główna kod programu – zawiera program testujący enc28j60 – opis w pkt.6.
- interrupts.cpp – plik zawierający kod obsługi przerwań.
- uart\_dma.cpp – plik zawierający kod biblioteki do obsługi protokołu UART.
- spi\_dma.cpp – plik zawierający kod biblioteki do obsługi protokołu SPI.
- enc28j60.cpp – plik zawierający kod biblioteki do obsługi ENC28J60.

- Inc – pliki nagłówkowe z deklaracjami

- main.hpp – plik nagłówkowy zawierający deklarację wszystkich zmiennych globalnych.
- uart\_dma.hpp – plik nagłówkowy biblioteki UART\_DMA.
- spi\_dma.hpp – plik nagłówkowy biblioteki SPI\_DMA.
- enc28j60.hpp – plik nagłówkowy biblioteki do obsługi ENC28J60.
- enc28j60\_reg.hpp – plik nagłówkowy z definicjami wszystkich rejestrów oraz makra to ustawiania poszczególnych bitów w rejestrach ENC28J60.

- Device – pliki z nagłówkami/makrami od producenta mikrokontrolera + implementacja zegara

- cmsis\_\*.h/core\_cm4.h/mpu\_armv7.h – pliki zawierające makra od firmy ARM do obsługi procesora.
- stm32f4\*.h – pliki od firmy ST zawierające makra/struktury oraz mapę adresową rejestrów procesorów z rodziny stm32f4xx.
- RccConfig.h – plik definiujący zegary magistral oraz główny zegar taktujący rdzeń.

Każda biblioteka zawiera namespace będący inicjałami twórcy, w przypadku biblioteki do modułu ENC28J60, która została napisana wspólnie, przestrzeń nazw to połączone inicjały obu twórców. W folderze głównym znajduje się również lista CMake oraz pliki linkera procesora STM32F446RETx.

### 3. Biblioteka SPI

Biblioteka SPI została napisana tak aby możliwe było wykorzystanie dowolnej dozwolonej przez producenta konfiguracji GPIO z zastosowaniem walidacji, która w przypadku zdefiniowania błędnych parametrów wysyła komunikat przez SWD o tym, która zmienna została błędnie zdefiniowana. Każda funkcja została krótko opisana w pliku nagłówkowym w stylu doxygen. Użytkownik ma pełną swobodę w sposobie wysyłania/odbierania danych przez ten interfejs. Może tego dokonać z wykorzystaniem DMA oraz bezpośrednio wysyłając bajt po bajcie. Dołączona została możliwość uruchomienia przerwań SPI, aby operować na interfejsie w sposób bezkolizyjny. SPI może zostać stworzona z wykorzystaniem parametrów użytkownika oraz w sposób domyślny. Parametry użytkownika definiowane są w postaci struktury, a referencja do niej przekazywana do konstruktora. W przypadku konstruktora domyślnego wybrany interfejs to SPI1, DMA2, natomiast GPIO adekwatne dla portu A. Pin CS sterowany jest w sposób programowy. Częstotliwość zegara SPI to 11.5MHz.

### 4. Biblioteka UART

Biblioteka UART w sposób analogiczny implementuje dowolne dozwolone przez producenta definiowanie GPIO. Posiada również pełną walidację danych z informacją zwrotną przez SWD w przypadku błędnej definicji. Różni się natomiast podejściem z ustawianiem Baud Rate. W przypadku protokołu UART na procesorach z serii STM32F4xx obliczanie samego wartości definiujących rejestr BRR było by kłopotliwe, zatem użytkownik definiuje wartość oczekiwanego Baud Rate'u, a biblioteka sama oblicza odpowiednią wartość tego rejestru podczas procesu inicjalizacji. Analogicznie jak w przypadku SPI istnieją dwa konstruktory. Pierwszy wykorzystuje referencję do struktury, która szczegółowo jest opisana w pliku nagłówkowym. Konstruktor domyślny natomiast stosuje interfejs UART2, DMA1, podstawowe ustawienie GPIO dla portu A oraz Baud Rate wynoszący 9600. Wybrany interfejs domyślnie w płytce rozwojowej NUCLEO podłączony jest do debuggera oraz programatora ST-LINK dzięki czemu, możliwe jest połączenie go jako portu szeregowego w komputerze.

### 5. Biblioteka ENC28J60

Biblioteka do obsługi ENC28J60 posiada wszystkie niezbędne funkcje do obsługi tego sterownika warstwy fizycznej sieci ethernet. Do jej celów został napisany dodatkowy plik nagłówkowy `enc28j60_reg.hpp` zawierający wszystkie możliwe adresy rejestrów oraz makra do ustawiania bitów w rejestrach tego urządzenia. Inicjalizacja w wyniku wielu niedokładności konstrukcyjnych wymagała zastosowania oczekiwania. Nie zostało tutaj zastosowane oczekiwanie na timerach i przerwaniach, ponieważ inicjalizacja wykonuje się tylko raz podczas całego kodu. Możliwe jest ustawienie na jakich zakresach pamięci drivera mają znajdować się bufor odbiorczy oraz nadawczy. Możliwym jest również ustawienie adresu mac aby urządzenie filtrowało pakiety tylko z określonym adresem lub z wybraną grupą adresów (np. multicast, broadcast). Zaimplementowano funkcję zapisu oraz odczytu z rejestrów ENC28J60 oraz zapisu i odczytu z bufora. W przypadku zapisu i odczytu z bufora użytkownik ma możliwość wysyłania danych z wykorzystaniem DMA oraz bezpośrednio bajt po bajcie z wykorzystaniem funkcji biblioteki SPI (pkt. 3). Istnieje również funkcja zapisu oraz odczytu rejestrów 16 bitowych, które są rozłożone na dwa rejestry 8 bitowe w przestrzeni pamięci ENC28J60. Instancja biblioteki SPI została zdefiniowana jako prywatna, aby użytkownik po skonstruowaniu obiektu (utworzeniu obiektu klasy ENC28J60) nie mógł zmienić parametrów interfejsu komunikacyjnego. Użytkownik ma możliwość załączenia oraz wyłączenia transmisji/odbioru pakietów z sieci Ethernet. Dodatkowo do celów weryfikacyjnych użytkownik ma możliwość wykorzystania Gettera portu oraz pinu sygnału INT, który może wykorzystać do implementacji przerwania EXTI. Odpowiednie przerwanie zostaje załączone w zależności od wyboru GPIO przez użytkownika. Użytkownik natomiast musi zaimplementować obsługę tego przerwania, bo ta zależy od sposobu wykorzystania linii INT przez użytkownika. Przykładowa obsługa tego przerwania oraz przerwań DMA została załączona w pliku `interrupts.cpp`.

## 6. Main.cpp – opis działania

W pliku main.cpp znajduje się kod służący do testowania modułu ENC28J60. Program posiada dwie główne funkcjonalności

- Po wysłaniu znaku 0x41 (litera A w kodzie ASCII) na port UART2, urządzenie inicjuje wysyłanie ramki z zapytaniem ARP na określony adres IP zdefiniowany w programie.
- Wszystkie pakiety przychodzące, które zawierają adres MAC zgodny z tym skonfigurowanym w programie, są wyświetlane na terminalu portu szeregowego.

Kod korzysta z portu UART2, aby umożliwić podłączenie terminalu portu szeregowego przez USB wprost do ST-Linka płytki rozwojowej NUCLEO.