

Ćwiczenie 4

PROGRAMOWANIE SYSTEMÓW WBUDOWANYCH

Obsługa wyświetlacza i interfejsu SPI mikrokontrolera Cortex-M4 STM32F303RE lub STM32F411RE.

Celem ćwiczenia jest zapoznanie studenta z metodyką programowania, projektowania i tworzenia aplikacji na mikrokontrolery rodziny ARM Cortex-M4,. W trakcie ćwiczenia student nabędzie podstawowe informacje dotyczące środowiska i umiejętności posługiwania się nim oraz programowania wyświetlacza i interfejsu SPI mikrokontrolera.



**Zakład Systemów Informacyjno-
Pomiarowych**



IETiSIP, Wydział Elektryczny, PW

Ćwiczenie 4 polegać będzie na konfiguracji komunikacji i wykorzystania modułu wyświetlacza. Niech składa się z kilku podzadań:

Interfejs SPI

SPI - Serial Peripheral Interface, synchroniczny interfejs szeregowy, wykorzystywany w lokalnej transmisji danych. SPI wykorzystuje 3 linie transmisyjne co umożliwia transmisję full duplex (I2C jedynie half duplex). Jego właściwości predysponują do zastosowań w komunikacji z kartami pamięci (prędkość większa niż np.: I2C) oraz wyświetlaczami alfanumerycznymi i graficznymi (możliwość współpracy w konfiguracji równoległej i szeregowej)

Oznaczenie i zastosowanie poszczególnych linii danych jest następujące:

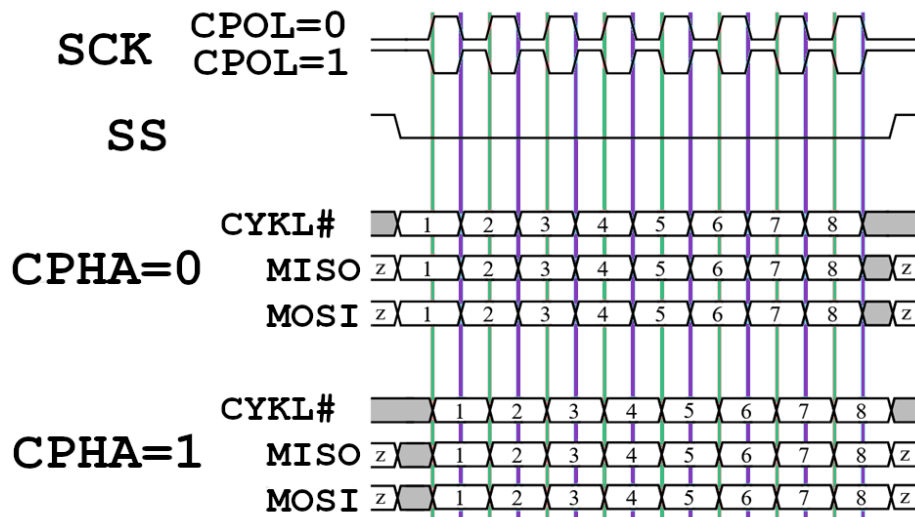
- **SCLK: Serial Clock** (wyjście z mastera – układu nadrzędnego). W protokole SPI za generację sygnału zegarowego odpowiedzialny jest układ master. Układ slave odbiera tylko sygnał zegarowy i nie ma nad nim kontroli.
- **MOSI: Master Output Slave Input** lub **Master Out Slave In** (wyjście danych z mastera) - pin danych. Ten pin służy do przesyłania danych w trybie master i odbierania danych w trybie slave.
- **MISO: Master Input Slave Output** lub **Master In Slave Out** (wyjście danych z urządzenia slave) - pin danych. Ten pin służy do przesyłania danych w trybie slave i odbierania danych w trybie master.
- **SS: Slave Select** (często aktywny niski, wyjście z układu mastera) W zależności od ustawień SPI i slave, pin SS używany do wyboru aktywnego urządzenia podrzędnego slave'a. Gdy jest jedno urządzenie główne i jedno podrzędne, pin SS nie jest wymagany. Pin wyboru slave'a wymagany jest wtedy, gdy master komunikuje się z kilkoma slave'ami. Służy do wyboru układu slave'a, do którego master chce przesłać dane.

Interfejs może pracować w jednym z czterech trybów konfigurowanych za pomocą bitów **CPOL** (bit polaryzacji) oraz **CPHA** (bit fazy) – tabela 1 oraz rysunek 1.

Tabela 1. Tryby pracy interfejsu SPI.

Tryb pracy	CPOL	CPHA	Aktywny stan zegara	Zbocze zegara wykorzystywane do pobrania i przesuwania danych
0	0	0	wysoki	Dane pobierane na zboczu narastającym, dane transmitowane na zboczu opadającym
1	0	1	wysoki	Dane pobierane na zboczu opadającym, dane transmitowane na zboczu narastającym
2	1	0	niski	Dane pobierane na zboczu opadającym, dane transmitowane na zboczu narastającym
3	1	1	niski	Dane pobierane na zboczu narastającym, dane transmitowane na zboczu opadającym

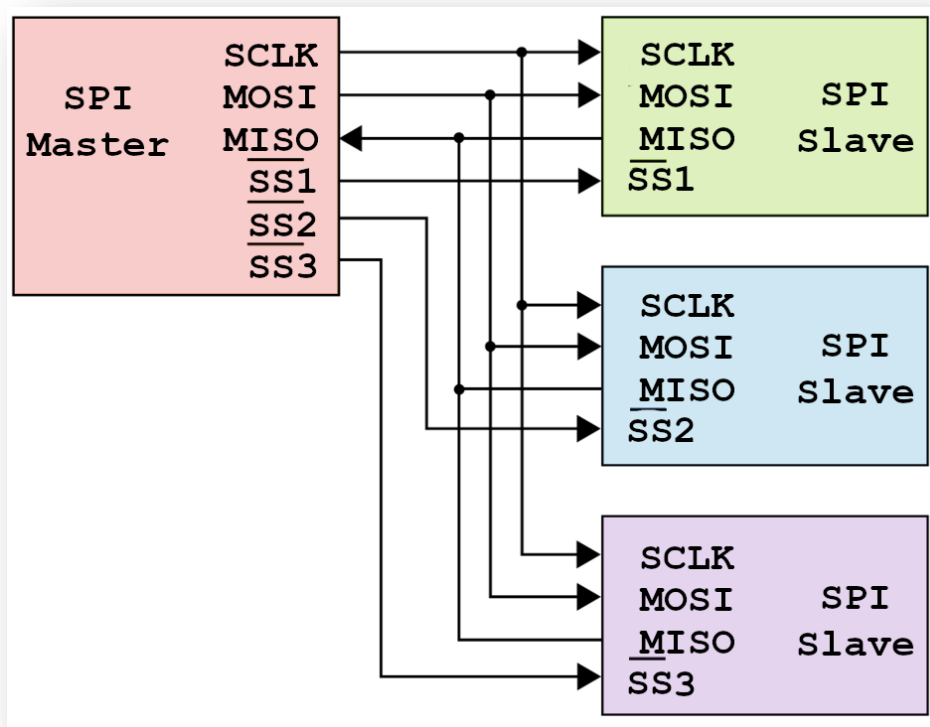




Rysunek 1. Tryby pracy interfejsu SPI.

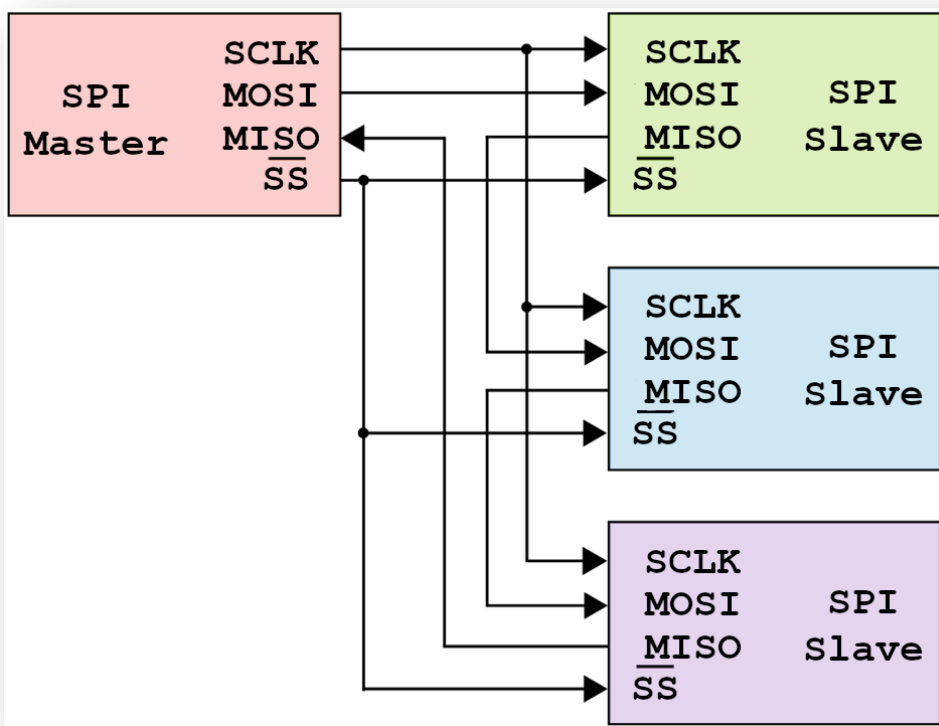
Odpowiednia konfiguracja jest istotna przy współpracy z konkretnym urządzeniem komunikującym się z mikrokontrolerem.

Typowe konfiguracje interfejsu SPI przedstawiają rysunki 2 oraz 3.



Rysunek 2. Konfiguracja równoległa. Wymagana jest wtedy niezależna linia SS dla każdego układu podrzędnego (slave).





Rysunek 3. Konfiguracja szeregowo (daisy chain). Wystarczy pojedyncza, wspólna linia SS dla wszystkich układów podrzędnych (slave). Warto zwrócić uwagę na „przeplot” linii MISO/MOSI w układach podrzędnych (slave).

Dysponując funkcjami obsługi wyświetlacza proszę stworzyć menu postaci poniższej struktury:

1. Konfiguracja
 - a. Częstotliwość pomiarów
 - b. Pomiar jednokrotny
 - c. Pomiar wielokrotny
 - i. Pojedyncza seria
 - ii. Pomiar ciągły
2. Start pomiarów
3. Stop pomiarów
4. Wyjście

Można menu opatrzyć graficznie. W pliku image.c znajduje się pewien obraz, który można podmienić. W tym celu potrzebny będzie konwerter bitmap na zapis tablicowy (R5G6B5).

Jako dane pomiarowe można wykorzystać sensor temperatury z poprzedniego ćwiczenia. Warunki pracy można ustalać np. za pomocą Timera.



W ćwiczeniu wykorzystywany będzie wyświetlacz: 1.3" LCD Module, wykorzystujący sterownik ST7789 (rozdzielczość 240x240pikseli)

Opis wyprowadzeń wyświetlacza:

- **VCC** - zasilanie układu. Wyświetlacz można zasilać z 3 lub 5V.
- **GND** - masa układu.
- **DIN** (*SPI data input*) - inna nazwa na *MOSI* (*Master Output Slave Input*). Wejście danych.
- **CLK** (*SPI Clock input*) – linia taktująca, zegar.
- **CS** (*Chip Select*) - alternatywna nazwa linii *SS* (*Slave Select*). Stan niski aktywuje nasłuchiwanie w urządzeniu, do którego mają być przesyłane dane.
- **DC** (*Data/Command selection*) - stan niski na tej linii powoduje, że odbierane dane odczytywane traktowane są jako komendy do sterownika. W przeciwnym wypadku, jako zwykłe dane.
- **RST** (*Reset*) - stan niski na tej linii powoduje zresetowanie wyświetlacza.
- **BL** (*Backlight*) podświetlenie (PWM), na początek można pozostawić odłączone.

Należy zatem stworzyć projekt (New STM32 Project) i z listy MCU wybrać kontroler STM32F303.

Podstawowe kroki są analogiczne do tych z poprzednich ćwiczeń. Konfiguracja specyficzna dla naszego układu jest następująca:

Konfiguracja kontrolera STM32F303:

piny (**Pinout**):

Tryb pracy GPIO_Output

F411PA5 -> **CLK (SPI: SCLK) F303** PB3(**SPI1**) PB13(**SPI2**)

F411PA7 -> **DIN (SPI: MOSI) F303** PB5(**SPI1**) PB15(**SPI2**)

PB6 -> CS

PA9 -> RST

PA8 -> DC

PB0 -> BL (NC) Jasność podświetlenia sterowana sygnałem PWM – opcja, domyślnie pozostaje odłączone.

Konfiguracja jest jednokierunkowa z kontrolera do wyświetlacza.

Clock configuration:

FCLK = 72MHz

W zakładce **Pinout&Configuration** należy wybrać **Connectivity->SPI2->Mode: Transmit Only Master**

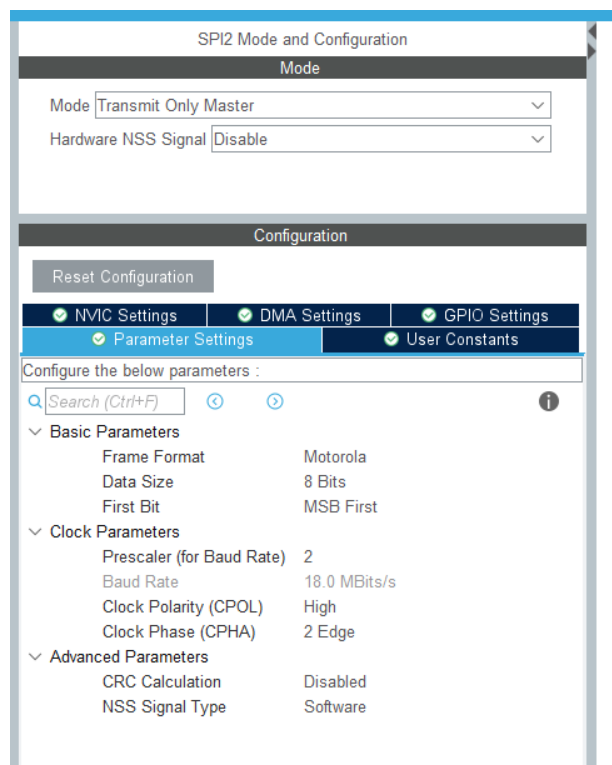


Zakład Systemów Informatycznych
Pomiarowych



IETiSIP, Wydział Elektryczny, PW

Pozostałe parametry należy ustawić zgodnie z rysunkiem 4:



Rysunek 4. Konfiguracja SPI

W katalogu projektu należy utworzyć podkatalog User i tam skopiować zawartość **LCD_ST7899_drivers.zip**

W pliku main.c należy umieścić w sekcji **USER CODE includes** poniższe dyrektywy:

```
#include "LCD_Driver.h"  
#include "GUI_Paint.h"  
#include "fonts.h"  
#include "image.h"
```

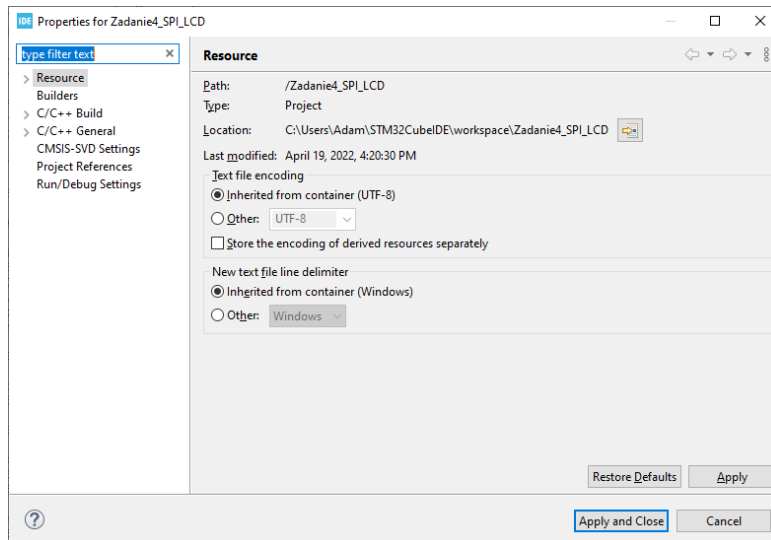
W tym momencie poprawne skompilowanie projektu nie jest możliwe gdyż musimy uzupełnić informacje w zakresie położenia plików nagłówkowych i źródłowych. Zaczniemy od nagłówków. W tym celu należy wybrać z menu **Project->Properties** (albo prawym klawiszem wybrać korzeń projektu zawierający jego nazwę). Otworzy się okno tak jak na rysunku 5:



Zakład Systemów Informatycznych-
Pomiarowych



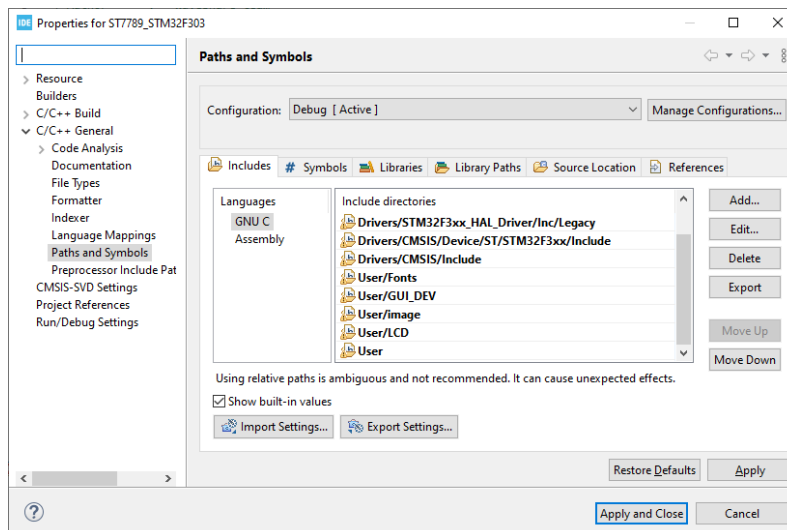
IETiSIP, Wydział Elektryczny, PW



Rysunek 5. Własności projektu

Z lewej strony wybrać należy zakładkę **C/C++ General** -> **Paths and Symbols**, następnie zakładka **Includes** i dodać (**Add...**) następujące podkatalogi zawierające nagłówki – rysunek 6.

- User/Fonts
- User/ GIU_DEV
- User/image
- User/LCD
- User



Rysunek 6. Uzupełnianie ścieżek z nagłówkami.

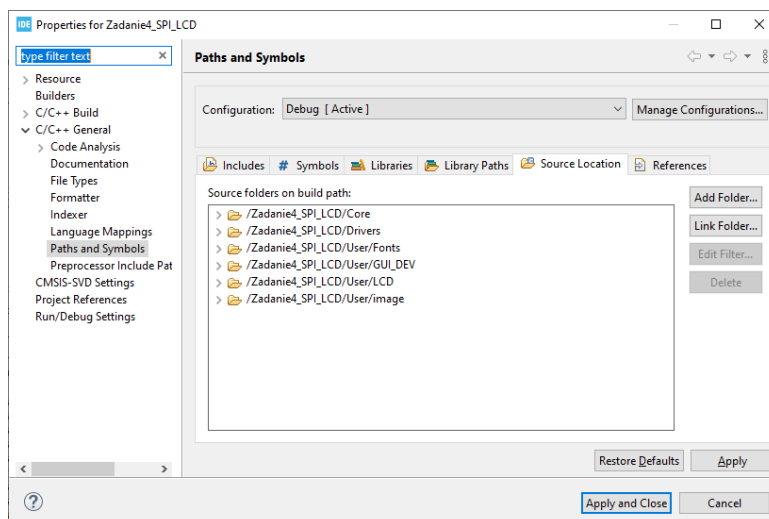
Podobnie należy postąpić z dodaniem katalogów zawierających kody źródłowe – rysunek 7.



Zakład Systemów Informatycznych
Pomiarowych



IETiSIP, Wydział Elektryczny, PW



Rysunek 7. Uzupełnianie ścieżek z plikami źródłowymi.

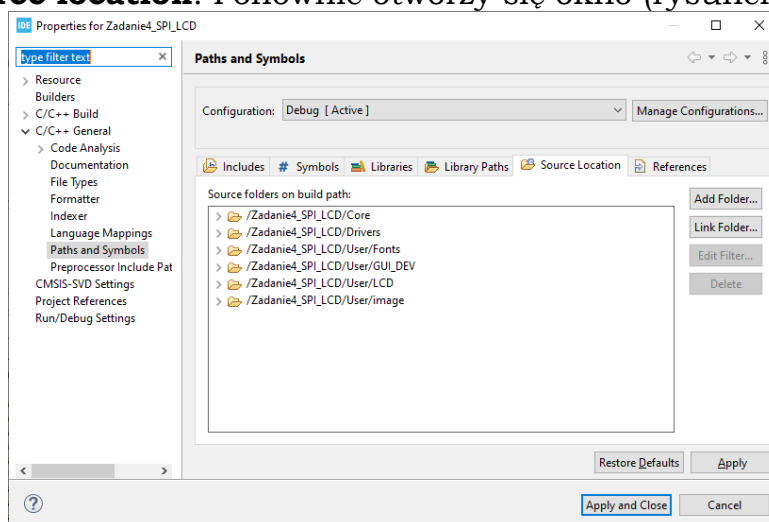
W oryginalnym pliku GUI_DEV/DEV_Config.h należy podmienić linię:

```
#include "stm32f4xx_hal.h"
```

na

```
#include "stm32f3xx_hal.h"
```

Należy jeszcze uzupełnić położenie plików źródłowych w projekcie. W tym celu należy ponownie wybrać z menu **Project->Properties** (albo kliknąć prawym klawiszem na korzeń projektu zawierający jego nazwę) i przejść do zakładki **Source location**. Ponownie otworzy się okno (rysunek 8):



Rysunek 8. Uzupełnianie ścieżek z plikami źródłowymi.

Tym razem należy wybrać **Source location-> Add Folder** i zaznaczyć nowy katalog User (widoczny już w oknie).

W pliku DEV_Config.c należy dodać jeszcze deklarację:

```
extern SPI_HandleTypeDef hspi1;
```

albo w zależności od konfiguracji

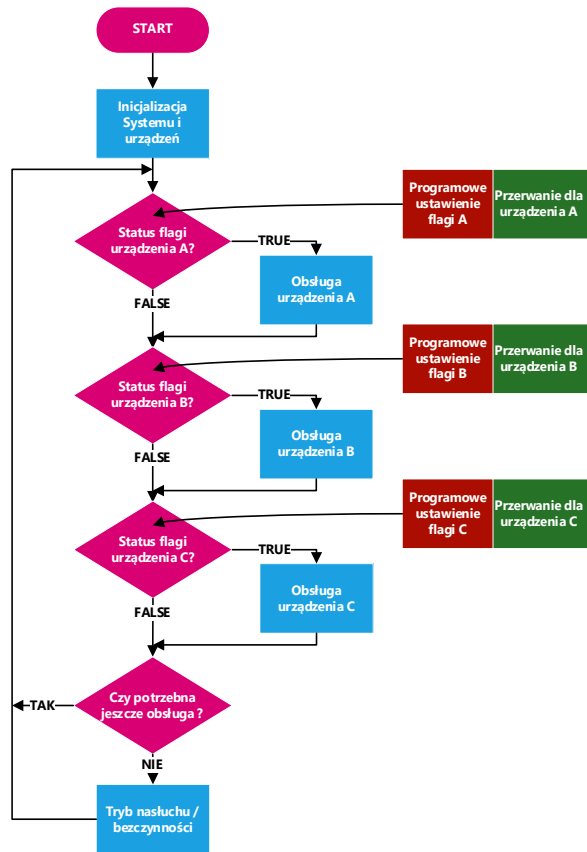
```
extern SPI_HandleTypeDef hspi2;
```



Zakład Systemów Informatycznych
Pomiarowych

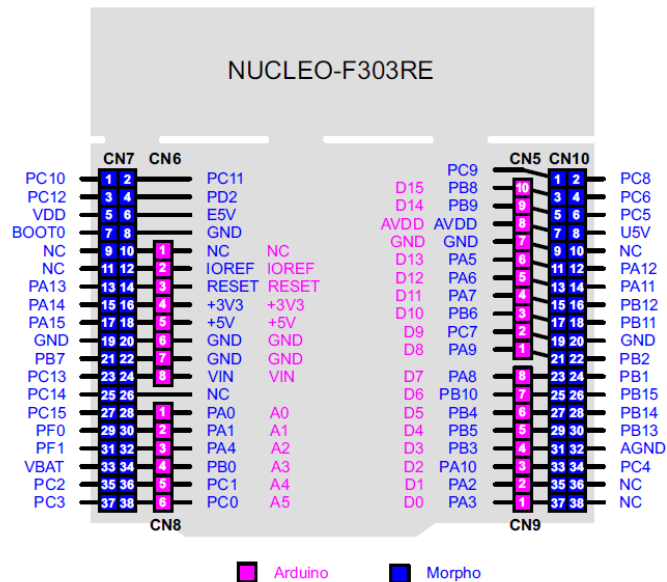


IETiSIP, Wydział Elektryczny, PW



Rysunek 10. Schemat szablonu obsługi wyświetlacza.

Rysunki 10 i 11 przedstawiają wyprowadzenia dla płytek NUCLEO dla dwóch mikrokontrolerów STM23F411RE i STM23F4303RE.



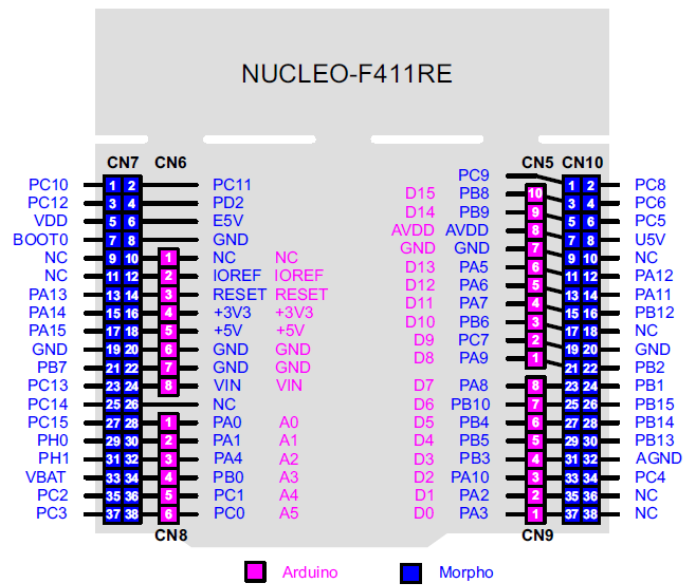
Rysunek 10. Wyprowadzenia na płytce NUCLEO-F303RE. Źródło: STM32 Nucleo-64 boards (MB1136.pdf).



Zakład Systemów Informatycznych
Pomiarowych



IETiSIP, Wydział Elektryczny, PW



Rysunek 11. Wyprowadzenia na płytce NUCLEO-F411RE. Źródło: STM32 Nucleo-64 boards (MB1136.pdf).



**Zakład Systemów Informatycznych-
Pomiarowych**



IETiSIP, Wydział Elektryczny, PW