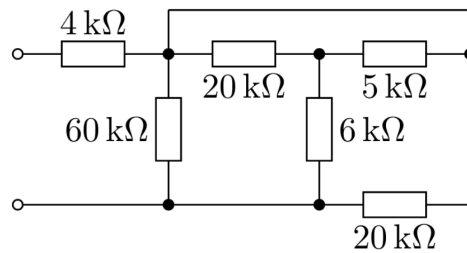


Mikrokontrolery STM32

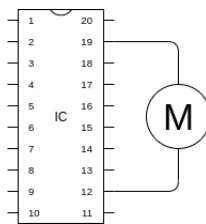
April 12, 2021

Zadanie 1

Wyznacz opór zastępczy następującego układu:



Zadanie 2



Chcemy sterować silnikiem dużej mocy za pomocą mikrokontrolera, podłączając go bezpośrednio do wyjść układu.

- Jak widzisz problemy wynikające z tego rozwiązania?
- Jak można by to rozwiązanie naprawić?
- Co twoim zdaniem się stanie po uruchomieniu takiego układu? (coś wybuchnie / coś się spali / nic się nie stanie)

Zadanie 3

Spróbuj zainstalować środowisko STM32CubeID oraz stworzyć prosty program migający dioda led. W tym celu:

1. W STM32CubeIDE stwórz nowy projekt o typie "STM32 Project".
2. Następnie odnajdź na liście mikrokontroler "STM32F103C8" (na tą chwilę plan jest taki aby korzystać z tzw. blue / black pill czyli tanich chińskich płytek z tym układem).
3. Nazwij projekt i przeklikaj się przez resztę ustawień (te domyślne opcje powinny działać)
4. Powinien ukazać się konfigurator mikrokontrolera jak na rysunku 1
5. Klikając na pin ustaw tryb "GPIO_Output", dioda led na tych płytkach znajduje się na pinie "PC13" (GPIOC pin numer 13)
6. W głównym pliku projektu *Core/Src/main.c* znajdź miejsce widoczne na listingu 1 i dodaj kilka instrukcji aby dioda led migiała np. co 1 sekundę. Skorzystaj w tym celu z bibliotek HAL z części do obsługi pinów (patrz "Funkcje z HAL" poniżej)
7. Zbuduj projekt i uruchom pod `qemu_stm32`
8. Powinieneś uzyskać następujący jak na listing 2

Przydatne rzeczy

Funkcje z HAL

Proszę nie czytać całej *dokumentacji*, chyba że naprawdę się nudzisz. Zalecam wyszukanie tylko następujących funkcji:

- *HAL_GPIO_ReadPin*
- *HAL_GPIO_WritePin*
- *HAL_GPIO_TogglePin*
- *HAL_Delay*

`qemu_stm32`

Instrukcja kompilacji:

- `git clone https://github.com/beckus/qemu_stm32.git`
- `cd qemu_stm32/`

- `./configure --enable-debug --target-list="arm-softmmu" --python=python2 --disable-werror`
- `make`

Uruchomienie: `./arm-softmmu/qemu-system-arm -M stm32-f103c8 -semihosting -kernel` (ścieżka do pliku `.bin` w katalogu `Debug` w projekcie)

Przydatne linki

- <https://forbot.pl/blog/kurs-stm32-f1-hal-sterowanie-gpio-w-praktyce-id22704>

Przesyłanie rozwiązań

Rozwiązania do zadania 1 i 2 zamieść w pdf'ie zawierającym skany, zdjęcia lub inną wygodną dla Ciebie formę. Do zadania 3 przeslij archiwum np. zip z kodem projektu.

W razie pytań pisać na michalszak@gmail.com

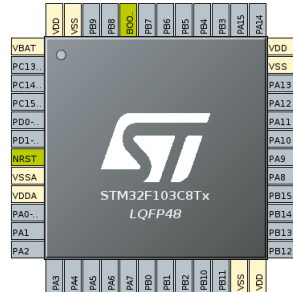


Figure 1: Konfigurator mikrokontrolera

Listing 1: Fragment *Core/Src/main.c*

```
[...]
MX_GPIO_Init();
/* USER CODE BEGIN 2 */

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    // tu wstaw kod

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}
[...]
```

Listing 2: Rezultat emulacji w terminalu

```
LED Off
LED On
LED Off
LED On
LED Off
LED On
LED Off
LED On
```