

Podstawy techniki mikroprocesorowej 2

Lab 3 - Silniki krokowe

Imię i Nazwisko:	Remigiusz Mielcarz, Grzegorz Salzburg
Nr indeksu:	252887, 252912

Termin zajęć: dzień tygodnia, godzina:	Środa 14:10-17:10 TP
Numer grupy ćwiczeniowej:	Y03-45f
Data wykonania ćwiczenia:	10.11.2021
Termin do oddania sprawozdania:	24.11.2021
Prowadzący kurs:	Dr inż. Krzysztof Halawa

Spis treści

1	Cel projektu	2
2	Zadania do wykonania	2
3	Schemat podłączenia	2
4	Schemat i konfiguracja pinów mikrokontrolera	3
5	Kod programu	3
6	Podsumowanie	6

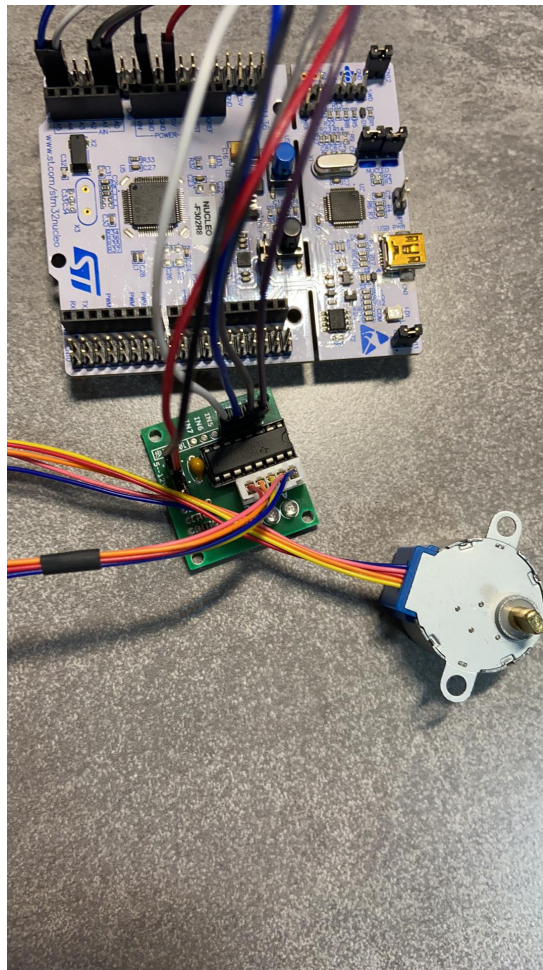
1 Cel projektu

Celem projektu jest zapoznanie się z działaniem i szczegółową budową modułu silnik krokowego z użyciem mikrokontrolera STM32. Implementacja różnych rodzajów sterowania, użycia timera i przerwań do sterowania silnikiem.

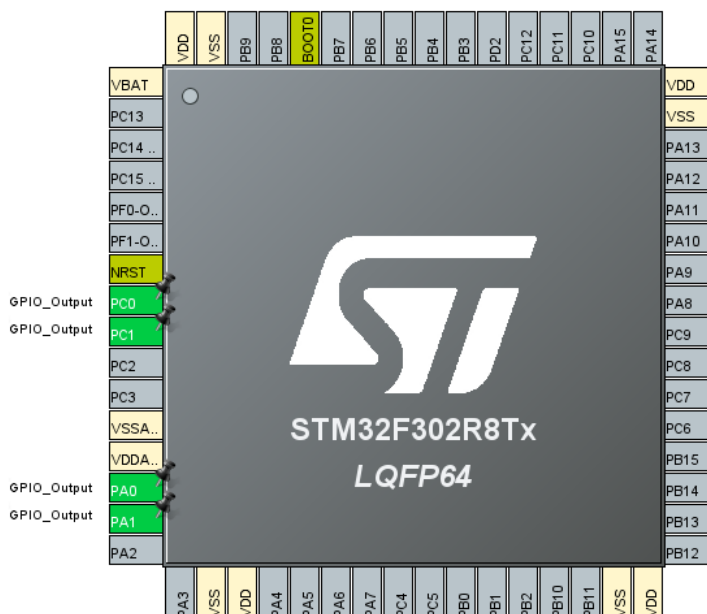
2 Zadania do wykonania

- Zaimplementowanie sterowania falowego (ang. wave drive),
- Zmiana kierunku obrotów z zadania powyżej,
- Zaimplementowanie sterowania pełnokrokowego (ang. full step) z wykorzystaniem dwóch uzwojeń w każdym kroku,
- Zaimplementowanie sterowania półkrokowego (ang. half step),

3 Schemat podłączenia



4 Schemat i konfiguracja pinów mikrokontrolera



Rysunek 1: Konfiguracja pinów mikrokontrolera w środowisku CubeIDE.

5 Kod programu

```
1 #include "main.h"
2 #include "gpio.h"
3
4
5 void SystemClock_Config(void);
6
7 int main(void)
8 {
9     /* Configure the system clock */
10    SystemClock_Config();
11
12    /* Initialize all configured peripherals */
13    MX_GPIO_Init();
14
15    /* Infinite loop */
16    /* USER CODE BEGIN WHILE */
17
18
19    while (1)
20    {
21        // 2 1 4 3
22        // a + to 2 pin
```

```

23 // a - to 4 pin
24 // b + to 3 pin
25 // b - to 1 pin
26
27 // POCZ ZAD 1
28     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
29     HAL_Delay(1);
30     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
31
32     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
33     HAL_Delay(1);
34     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
35
36     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
37     HAL_Delay(1);
38     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
39
40     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
41     HAL_Delay(1);
42     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
43 // KONIEC ZAD 1
44
45 // POCZ ZAD 2
46     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
47     HAL_Delay(1);
48     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
49
50     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
51     HAL_Delay(1);
52     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
53
54     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
55     HAL_Delay(1);
56     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
57
58     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
59     HAL_Delay(1);
60     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
61 // KONIEC ZAD 2
62
63 // POCZ ZAD 3
64     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
65     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
66     HAL_Delay(1);
67     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
68     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
69

```

```

70     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
71     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
72     HAL_Delay(1);
73     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
74     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
75
76     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
77     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
78     HAL_Delay(1);
79     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
80     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
81
82     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
83     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
84     HAL_Delay(1);
85     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
86     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
87
88     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
89     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
90     HAL_Delay(1);
91     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
92     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
93 // KONIEC ZAD 3
94
95 // POCZ ZAD 4
96     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET); // Krok a
97     HAL_Delay(1);
98     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
99
100    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET); // Krok b
101    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
102    HAL_Delay(1);
103    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
104    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
105
106    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET); // Krok c
107    HAL_Delay(1);
108    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
109
110    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // Krok d
111    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
112    HAL_Delay(1);
113    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
114    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_RESET);
115
116    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // Krok e

```

```

117     HAL_Delay(1);
118     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
119
120     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET); // Krok f
121     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
122     HAL_Delay(1);
123     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
124     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
125
126     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET); // Krok g
127     HAL_Delay(1);
128     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
129
130     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET); // Krok h
131     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);
132     HAL_Delay(1);
133     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
134     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
135 // KONIEC ZAD 4
136
137 }
138 /* USER CODE END 3 */
139 }

```

6 Podsumowanie

- Zadanie 1 polegało na implementacji sterowania falowego. W tym przypadku wykorzystujemy jedno uzwojenie silnika, dlatego inną nazwą takiego sterowania jest sterowanie jednofazowe. Implementacja polegała na zmianie stanu kolejnych pinów na SET i RESET, w odpowiedniej kolejności. Ta kolejność jest równa A+, B-, A-, B+ czyli odpowiednia piny numer 2, 1, 4, 3.
- Zadanie 2 jest zadaniem analogicznym. W tym zadaniu silnik kręci się w drugą stronę. Implementacja polegała na zmianie stanów pinów w odwrotnej kolejności niż w 1, czyli: B+, A-, B-, A+ (piny 3, 4, 1, 2)
- Zadanie 3 polegało na implementacji sterowania pełnokrokowego, z wykorzystaniem dwóch uzwojeń w każdym kroku. Główny zamysł implementacji jest taki sam jak w poprzednich zadaniach, zmienia i rozbudowuje się algorytm zmiany stanu na kolejnych pinach. Algorytm ma 5 kroków. W każdym z nich ustawiamy stan SET na 2 kolejnych pinach, czekamy chwilę i następnie ustawiamy stan RESET na obydwu w danej konfiguracji: A-B+, A+ B+, A+ B-, A- B-, A- B+.
- Zadanie 4 polegało na implementacji sterowania półkrokowego. Konfiguracja będzie następująca: A-, A- B+, B+, A+ B+, A+, A+ B-, B-, A- B-.
- Z obserwacji poczynionych na zajęciach mogliśmy zauważyć różnicę, pomiędzy sterowaniami. Różnicą tą był opór jaki stawiał działał silnik krokowy, czyli różnił się moment obrotowy, jeden z nich był jakby silniejszy”.