

stepmotor6.c

```

1#include "stepmotor6.h"
2#include "tim.h"
3#include "gpio.h"
4#include "main.h"
5
6
7
8/* inicjalizacja biblioteki */
9void stepper_init_motor6(void)
10{
11    HAL_TIM_Base_Start_IT(&PULSE_TIM_MOTOR6);
12    HAL_Delay(2);
13}
14
15/* ustawia predkosc obrotowa silnika krokowego w rpm */
16void stepper_speed_motor6(uint16_t rpm)
17{
18    if(rpm > 0)
19    {
20        uint16_t arr_val = XT_TIM_CLK_MOTOR6 / (((XT_TIM_PSC_MOTOR6+1) * rpm *
        STEPS_PER_REV_MOTOR6 * MICROSTEP_NUM_MOTOR6) / 60) - 1;
21        uint16_t pulse_val = arr_val / 2;
22        __HAL_TIM_SET_AUTORELOAD(&PULSE_TIM_MOTOR6, arr_val);
23        __HAL_TIM_SET_COMPARE(&PULSE_TIM_MOTOR6, PULSE_TIM_CH_MOTOR6, pulse_val);
24    }
25    else
26    {
27        __HAL_TIM_SET_COMPARE(&PULSE_TIM_MOTOR6, PULSE_TIM_CH_MOTOR6, 0);
28    }
29}
30
31/* rozpoczyna ruch silnika z kierunkiem obrotu dir */
32void stepper_run_motor6(uint8_t dir)
33{
34    if(dir == STEPPER_CW_MOTOR6)
35    {
36        HAL_GPIO_WritePin(DIR_AXIS6_GPIO_Port, DIR_AXIS6_Pin, GPIO_PIN_SET);
37    }
38    else if (dir == STEPPER_CCW_MOTOR6)
39    {
40        HAL_GPIO_WritePin(DIR_AXIS6_GPIO_Port, DIR_AXIS6_Pin, GPIO_PIN_RESET);
41    }
42    HAL_Delay(2);
43    HAL_TIM_PWM_Start(&PULSE_TIM_MOTOR6, PULSE_TIM_CH_MOTOR6);
44}
45
46/* zatrzymuje silnik - wyłącza wyjścia kontrolera silnika */
47void stepper_stop_motor6(void)
48{
49    HAL_TIM_PWM_Stop(&PULSE_TIM_MOTOR6, PULSE_TIM_CH_MOTOR6);
50}
51
52/* ustawia zadana liczbe krokow do wykonania */
53void stepper_steps_motor6(uint16_t steps, volatile uint16_t *steplimit6)
54{
55    *steplimit6 = steps;
56}
57
58/* obraca wal silnika o zadny kat z zadana predkoscia w kierunku dir */
59void stepper_rot_motor6(uint16_t ang, uint16_t rpm, uint8_t dir, volatile uint16_t
    *steplimit6, volatile int* rotationCounter6, volatile uint16_t* isStop6)
60{

```

stepmotor6.c

```
61     if(dir == STEPPER_CW_MOTOR6)
62     {
63         *isStop6 = 0;
64         *rotationCounter6 += ang;
65     }
66     else if(dir == STEPPER_CCW_MOTOR6)
67     {
68         *isStop6 = 0;
69         *rotationCounter6 -= ang;
70     }
71     stepper_steps_motor6((((STEPS_PER_REV_MOTOR6 * MICROSTEP_NUM_MOTOR6)/40)*ang)/90),
    stepLimit6);
72     stepper_speed_motor6(rpm);
73     stepper_run_motor6(dir);
74 }
75
76 /* obraca wal silnika do pozycji domowej o zadany kat z zadana predkoscia w kierunku dir */
77 void stepper_rot_home_motor6(uint16_t rpm, uint8_t dir, volatile uint16_t *stepLimit6,
    volatile int* rotationCounter6, volatile uint16_t* isStop6)
78 {
79     if(*rotationCounter6 < 0)
80     {
81         *isStop6 = 0;
82         uint16_t ang = *rotationCounter6 * (-1);
83         stepper_rot_motor6(ang, rpm, STEPPER_CW_MOTOR6, stepLimit6, rotationCounter6,
    isStop6);
84     }
85     else
86     {
87         *isStop6 = 0;
88         uint16_t ang = *rotationCounter6;
89         stepper_rot_motor6(ang, rpm, STEPPER_CCW_MOTOR6, stepLimit6, rotationCounter6,
    isStop6);
90     }
91 }
92
93
94
```