```
#include "motor.h"

/*                                  Note
 * The PWM is handled by TIM4.
 * In case of DC motor configuration:
 * - the master clock for TIM4 is 1MHz
 * - the counter counts up to 2000, result in 2ms of PWM period (500Hz)
 * - the PWM pulse width data can to 0~1999, coresponding to 0~100% duty cycle
*/

extern TIM_HandleTypeDef htim4;

/*
 * Setup the driving power for 4 motors. p1~p4 data range is 0~1999, which equals
 * to 0~100% duty cycle (for DC motor configuration)
 */
void set_motor_pwm(MotorControlTypeDef *motor_pwm)
{
  if (motor_pwm->motor1_pwm >= MOTOR_MAX_PWM_VALUE)
    htim4.Instance->CCR1 = MOTOR_MAX_PWM_VALUE;
  else if (motor_pwm->motor1_pwm <= MOTOR_MIN_PWM_VALUE)
    htim4.Instance->CCR1 = MOTOR_MIN_PWM_VALUE;
  else
    htim4.Instance->CCR1 = motor_pwm->motor1_pwm;

  if (motor_pwm->motor2_pwm >= MOTOR_MAX_PWM_VALUE)
    htim4.Instance->CCR2 = MOTOR_MAX_PWM_VALUE;
  else if (motor_pwm->motor2_pwm <= MOTOR_MIN_PWM_VALUE)
    htim4.Instance->CCR2 = MOTOR_MIN_PWM_VALUE;
  else
    htim4.Instance->CCR2 = motor_pwm->motor2_pwm;

  if (motor_pwm->motor3_pwm >= MOTOR_MAX_PWM_VALUE)
    htim4.Instance->CCR3 = MOTOR_MAX_PWM_VALUE;
  else if (motor_pwm->motor3_pwm <= MOTOR_MIN_PWM_VALUE)
    htim4.Instance->CCR3 = MOTOR_MIN_PWM_VALUE;
  else
    htim4.Instance->CCR3 = motor_pwm->motor3_pwm;

  if (motor_pwm->motor4_pwm >= MOTOR_MAX_PWM_VALUE)
    htim4.Instance->CCR4 = MOTOR_MAX_PWM_VALUE;
  else if (motor_pwm->motor4_pwm <= MOTOR_MIN_PWM_VALUE)
    htim4.Instance->CCR4 = MOTOR_MIN_PWM_VALUE;
  else
    htim4.Instance->CCR4 = motor_pwm->motor4_pwm;
}


void set_motor_pwm_zero(MotorControlTypeDef *motor_pwm)
{
```

4つのモータへPWM信号を出力する関数

モータNo.1:レジスタに値を書き込む。上下限値を超える場合は飽和させる。

モータNo.2

モータNo.3

モータNo.4

4つ全てのモータを止めるためのPWMレジスタ値を設定する関数

```
51   motor_pwm->motor1_pwm = 0;
52   motor_pwm->motor2_pwm = 0;
53   motor_pwm->motor3_pwm = 0;
54   motor_pwm->motor4_pwm = 0;
55 }
56
57
```