**实验报告**

（STM32小车运行控制原理学习，以小车前进功能为例）

学号：2024100192

姓名：庞晓宇

小组所有成员：欧阳铖，潘梓琛

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **学号** | **姓名** |
| 2023117131 | 欧阳铖 | 2023117101 | 潘梓琛 |

**一、小车运动相关函数介绍**

小车运动的相关函数在motor.c中定义，函数的功能描述及对应的函数签名如下：

void MotorGPIO\_Configuration(void)

功能：配置电机控制所需的 GPIO 引脚，设置为推挽输出模式，速度为 2MHz。

void CarMove(void)

功能：根据占空比（\*\_speed\_duty）和当前计数值（speed\_count）控制电机的启停状态，实现 PWM 控制电机转动。

void CarGo(void)

功能：设置小车向前运动，将对应电机的速度设为正向最大值（SPEED\_DUTY），目前左前、左后电机被注释或未启用。

void CarBack(void)

功能：设置小车后退，所有电机的速度设为负的最大值（-SPEED\_DUTY）。

void CarLeft(void)

功能：设置小车向左转，左侧电机设为反向低速（-20），右侧电机设为高速，并对右后轮进行额外补偿。

void CarRight(void)

功能：设置小车向右转，右侧电机设为反向低速（-20），左侧电机设为高速，并对左后轮进行额外补偿。

void CarStop(void)

功能：设置所有电机速度为 0，使小车停止。

void MotorInit(void)

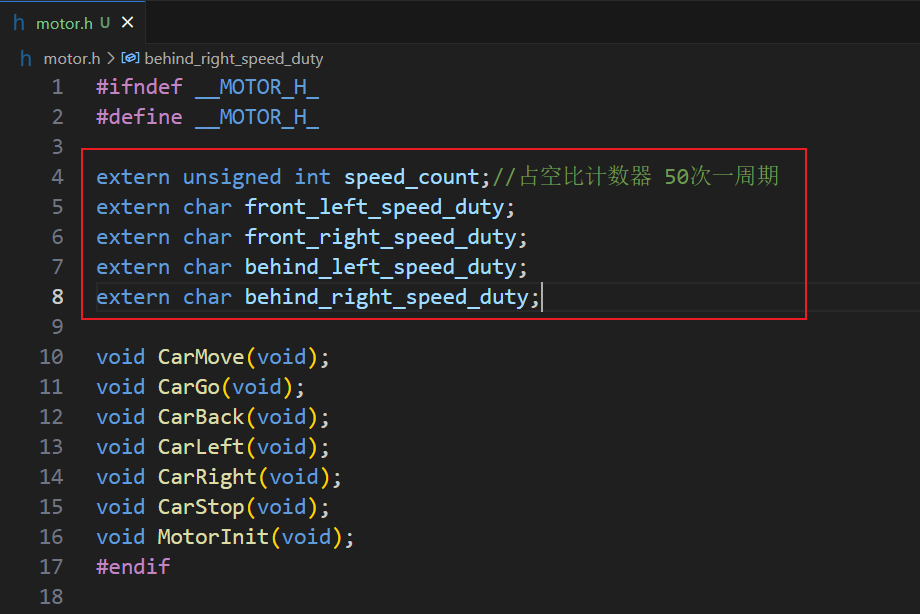
功能：初始化电机模块，调用 GPIO 配置函数并让小车初始状态为停止。

**二、程序运行过程介绍**

介绍“小车前进功能”中所用到的函数和变量，以及是如何运行使小车前进的。

要求300字以上，图文并茂（可以是手工画图并拍照）

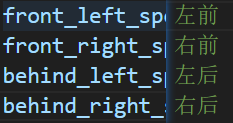
在motor.h中有如下定义：



结合motor.c可以得知：

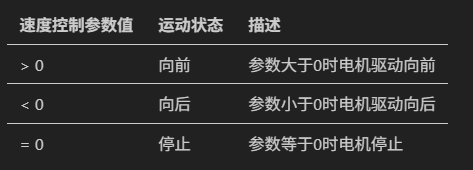
-函数：CarMove（void）

总体来说是用于根据各个轮子的速度控制参数（front\_left\_speed\_duty、front\_right\_speed\_duty、behind\_left\_speed\_duty、behind\_right\_speed\_duty）以及当前的速度计数 speed\_count 来控制每个轮子的运动状态。

速度的逻辑判断：针对每个轮子（小车的四个轮子对应为）

代码会根据对应的控制参数判断并控制相应轮子的状态。

若速度控制参数大于0则向前运动，小于0向后运动，0停止运动。列表如下：





-PWM 调速逻辑：借助 speed\_count 与速度控制参数对比，实现简单的 PWM（脉冲宽度调制）调速。当 speed\_count 小于速度控制参数时，轮子前进或后退；反之则停止。

-函数：void Cargo(void)

//front\_left\_speed\_duty=SPEED\_DUTY;

front\_right\_speed\_duty=SPEED\_DUTY;

behind\_left\_speed\_duty=SPEED\_DUTY;

//behind\_right\_speed\_duty=SPEED\_DUTY;

通过把四个轮子的速度都设为一样（正速），则其同时前进，实现了前进功能。

-函数：void CarBack(void)

front\_left\_speed\_duty=-SPEED\_DUTY;

front\_right\_speed\_duty=-SPEED\_DUTY;

behind\_left\_speed\_duty=-SPEED\_DUTY;

behind\_right\_speed\_duty=-SPEED\_DUTY;

通过把四个轮子的速度都设为负数，从而实现了后退功能。

-函数：CarLeft(void)

front\_left\_speed\_duty=-20;

front\_right\_speed\_duty=SPEED\_DUTY;

behind\_left\_speed\_duty=-20;

behind\_right\_speed\_duty=SPEED\_DUTY+10;

首先是左前轮开始减速，这里就直接把速度设置为负数了。然后是右前轮速度正常（正速），然后左后侧也减速（这里也设为负数了），最后右后轮加速，从逻辑上就实现了左转的功能。

-函数：void CarRight(void)

front\_left\_speed\_duty=SPEED\_DUTY;

front\_right\_speed\_duty=-20;

behind\_left\_speed\_duty=SPEED\_DUTY+10;//նݓ۳ÖȽ֯f

behind\_right\_speed\_duty=-20;

和左转逻辑相似，首先是右前轮开始减速，这里就直接把速度设置为负数了。然后是左前轮速度正常（正速），然后右后侧也减速（这里也设为负数了），最后左后轮加速，从逻辑上就实现了右转的功能。

-函数：void CarStop(void)

front\_left\_speed\_duty=0;

front\_right\_speed\_duty=0;

behind\_left\_speed\_duty=0;

behind\_right\_speed\_duty=0;

直接把四个轮子速度都设成零了，从而实现停止功能。