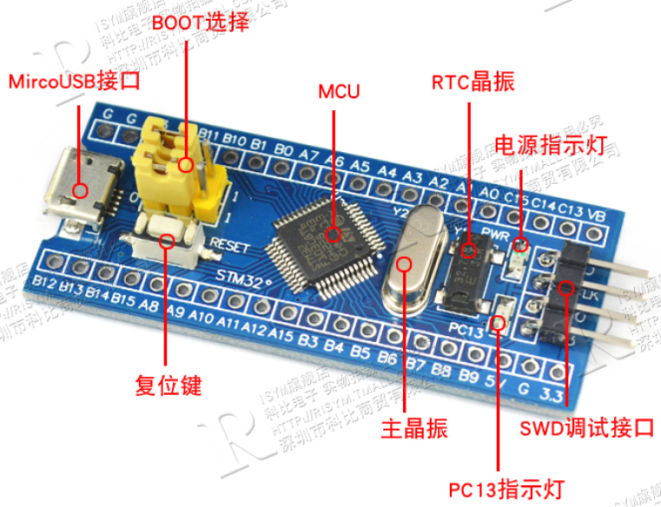
STM32F103C8T6核心板+L298N直流电机调速PWM调速

1. **功能说明**

用L298N驱动一台直流减速电机，引脚IN1，IN2可用于PWM控制。仅用STM32的两个端口给出PWM信号控制IN1，IN2即可实现正反转、加减速等动作。

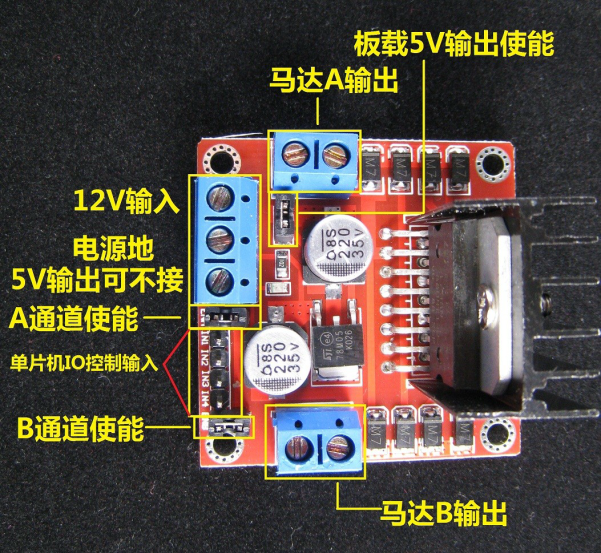
1. **硬件说明：**

**控制板：**STM32F03C8T6最小系统；



|  |  |
| --- | --- |
| 控制板说明 | |
| 型号： | STM32F103C8T6 |
| 内核： | ARM 32位的CortexTM-M3 CPU |
| 调试方式： | SWD |
| 尺寸： | 22.86mm×53.34mm |

直流电机驱动板：L298N是SGS公司的产品，比较常见的是15脚MulTIwatt封装的L298N，内部同样包含4通道逻辑驱动电路。可以方便的驱动两个直流电机，或一个两相步进电机。



1. **资源配置：**

|  |  |  |
| --- | --- | --- |
| STM32F03C8T6使用端口定义 | | |
| 端口 | 功能说明 | 配置 |
| PA2 | 连接L298N芯片的IN1 | 通用定时器的PWM功能，  TIM2\_CH3 |
| PA3 | 连接L298N芯片的IN2 | 通用定时器的PWM功能，  TIM2\_CH4 |
| PC13 | 控制板带的LED指示灯，周期性的变化，每变化一次，电机的转运方向变化一次。  由TIM3中断定时控制。 | 通用定时器TIM3定时中断 |

|  |  |  |  |
| --- | --- | --- | --- |
| L298N端口定义 | | | |
| IN1 | IN2 | ENA | 电机状态 |
| PA2 | 0 | 始终是1 | 顺时针运转，调整PA2的频率，可高速。 |
| 0 | PA3 | 始终是1 | 逆时针运转，调整PA2的频率，可高速。 |
| 0 | 0 | 始终是1 | 停止 |

1. **软件实现**

1：通用定时器TIM3的中断定时实现步骤

① 使能定时器时钟。

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM3, ENABLE);

② 初始化定时器，配置【ARR自动装载值,PSC预分频系数。】

TIM\_TimeBaseInit(TIM3, &TIM\_TimeBaseStructure);

③ 开启定时器中断，配置NVIC。

TIM\_ITConfig(TIM3,TIM\_IT\_Update,ENABLE);

NVIC\_Init(&NVIC\_InitStructure);

④ 使能定时器。

TIM\_Cmd(TIM3, ENABLE); //实际是配置 控制寄存器1 (TIM3\_CR1)

⑥ 编写中断服务函数。

TIMx IRQHandler();

2：通用定时器TIM2的Channel1、Channel2的PWM实现步骤

① 使能定时器2和相关IO口时钟。

使能定时器2时钟：RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM2, ENABLE);

使能GPIOB时钟：RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA, ENABLE);

② 初始化IO口为复用功能输出。函数：GPIO\_Init();

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

③ 初始化定时器：ARR,PSC等：TIM\_TimeBaseInit(TIM2, &TIM\_TimeBaseStructure);

④ 初始化输出比较参数:

TIM\_OC3Init(TIM2, &TIM\_OCInitStructure);

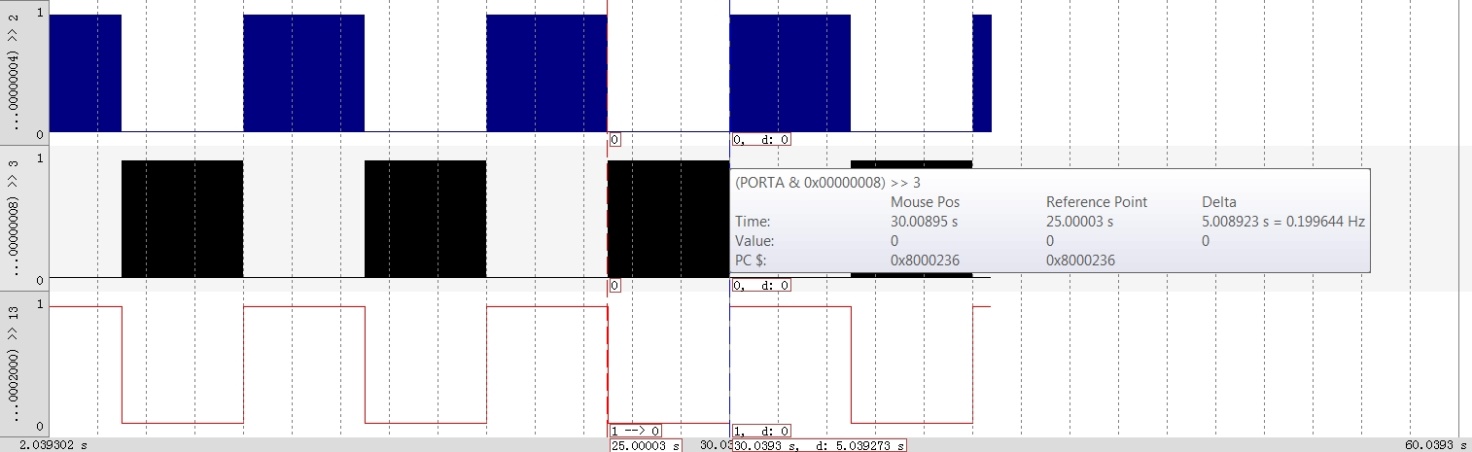
TIM\_OC4Init(TIM2, &TIM\_OCInitStructure);

⑤ 使能预装载寄存器： TIM\_OC2PreloadConfig(TIM2, TIM\_OCPreload\_Enable);

⑥ 使能定时器。TIM\_Cmd();

⑦ 不断改变比较值CCRx，达到不同的占空比效果:TIM\_SetCompare3() ; TIM\_SetCompare4() ;

使用Keil5 模拟软件的Logic Analyzer功能，PA2，PA3,PC13的波形图如下所示：



1. **附录：程序源码**

1 #include "delay.h"

2 #include "sys.h"

3 #include "timer.h"

4

5 void GPIO\_Config(void) //配置PA2, PA3

6 {

7 GPIO\_InitTypeDef GPIO\_InitStructure;

8

9 /\* GPIOA clock enable \*/

10 RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA |

RCC\_APB2Periph\_GPIOC,ENABLE); //

11

12 /\*GPIOA Configuration: TIM3 channel 1 and 2 as alternate function

push-pull \*/

13 GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_2 | GPIO\_Pin\_3;

14 GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF\_PP; //

复用推挽输出

15 GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

16 GPIO\_Init(GPIOA, &GPIO\_InitStructure);

17

18 GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_13; //LED0-->PC.13

端口配置

19 GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; //推挽输出

20 GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; //IO口速度为50MHz

21 GPIO\_Init(GPIOC, &GPIO\_InitStructure);

//根据设定参数初始化GPIOC.13

22 }

23

24 int main(void)

25 {

26 delay\_init(); //延时函数初始化

27 NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

//设置NVIC中断分组2:2位抢占优先级，2位响应优先级

28

29 //每隔5s的周期，变换电机的转向。

30 //定时中断，最小单位钟频率=72MHz/36KHz=2KHz

31 TIM3\_Int\_Init(9999,35999) ;

32

33 GPIO\_Config(); //LED,PWM端口初始化,配置PC13,PA2,PA3

34 TIM2\_PWM\_Init(47999,2 ); //分频。最小单位钟频率=72000000/(2+1)=24Mhz

35

36 while(1)

37 {

38

39 }

40 }

1 #include "timer.h"

2

3 //通用定时器3中断初始化

4 //这里时钟选择为APB1的2倍，而APB1为36M

5 //arr：自动重装值。

6 //psc：时钟预分频数

7 //这里使用的是定时器3!

8

9 #define PC13 PCout(13)// PC13

10

11 void TIM3\_Int\_Init(u16 arr,u16 psc)

12 {

13 TIM\_TimeBaseInitTypeDef TIM\_TimeBaseStructure;

14 NVIC\_InitTypeDef NVIC\_InitStructure;

15

16 RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM3, ENABLE); //时钟使能

17 // RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_AFIO ,ENABLE );

18 // GPIO\_PinRemapConfig(GPIO\_FullRemap\_TIM3 ,ENABLE);

19

20 TIM\_TimeBaseStructure.TIM\_Period = arr;

//设置在下一个更新事件装入活动的自动重装载寄存器周期的值

计数到5000为500ms

21 TIM\_TimeBaseStructure.TIM\_Prescaler = psc;

//设置用来作为TIMx时钟频率除数的预分频值 10Khz的计数频率

22 TIM\_TimeBaseStructure.TIM\_ClockDivision = TIM\_CKD\_DIV1;

//设置时钟分割:TDTS = Tck\_tim

23 TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up;

//TIM向上计数模式

24 TIM\_TimeBaseInit(TIM3, &TIM\_TimeBaseStructure);

//根据TIM\_TimeBaseInitStruct中指定的参数初始化TIMx的时间基数单位

25

26 TIM\_ITConfig(TIM3,TIM\_IT\_Update,ENABLE );

//使能指定的TIM3中断,允许更新中断

27

28 NVIC\_InitStructure.NVIC\_IRQChannel = TIM3\_IRQn; //TIM3中断

29 NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0;

//先占优先级0级

30 NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 3; //从优先级3级

31 NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //IRQ通道被使能

32 NVIC\_Init(&NVIC\_InitStructure);

//根据NVIC\_InitStruct中指定的参数初始化外设NVIC寄存器

33

34 TIM\_Cmd(TIM3, ENABLE); //使能TIMx外设

35 }

36 //定时器3中断服务程序

37 void TIM3\_IRQHandler(void) //TIM3中断

38 {

39 if (TIM\_GetITStatus(TIM3, TIM\_IT\_Update) != RESET)

//检查指定的TIM中断发生与否:TIM 中断源

40 {

41 TIM\_ClearITPendingBit(TIM3, TIM\_IT\_Update );

//清除TIMx的中断待处理位:TIM 中断源

42 PC13=!PC13;

43

44 if (PC13 ==1)

45 {

46 TIM\_SetCompare3(TIM2,24000);

47 TIM\_SetCompare4(TIM2,0);

C:\STM32F103C8T6核心板+L298N\_PWM直流电机调速\HARDWARE\TIMER\timer.c

48 }

49 else

50 {

51 TIM\_SetCompare3(TIM2,0);

52 TIM\_SetCompare4(TIM2,24000);

53 }

54 }

55 }

56

57 //TIM2 PWM部分初始化两个PWM波形，分别控制电机的正转与反转

58 //PWM输出初始化

59 //arr：自动重装值

60 //psc：时钟预分频数

61 void TIM2\_PWM\_Init(u16 arr,u16 psc)

62 {

63

64 TIM\_TimeBaseInitTypeDef TIM\_TimeBaseStructure;

65 TIM\_OCInitTypeDef TIM\_OCInitStructure;

66

67 RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM2, ENABLE); //使能定时器3时钟

68

69 //初始化TIM2

70 TIM\_TimeBaseStructure.TIM\_Period = arr;

//设置在下一个更新事件装入活动的自动重装载寄存器周期的值

71 TIM\_TimeBaseStructure.TIM\_Prescaler =psc;

//设置用来作为TIMx时钟频率除数的预分频值

72 TIM\_TimeBaseStructure.TIM\_ClockDivision = 0; //设置时钟分割:TDTS =

Tck\_tim

73 TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up;

//TIM向上计数模式

74 TIM\_TimeBaseInit(TIM2, &TIM\_TimeBaseStructure);

//根据TIM\_TimeBaseInitStruct中指定的参数初始化TIMx的时间基数单位

75

76 //初始化TIM2 Channel1 PWM模式

77 TIM\_OCInitStructure.TIM\_OCMode = TIM\_OCMode\_PWM1;

//选择定时器模式:TIM脉冲宽度调制模式2

78 TIM\_OCInitStructure.TIM\_OutputState = TIM\_OutputState\_Enable;

//比较输出使能

79 TIM\_OCInitStructure.TIM\_Pulse = 13500 ;

80 TIM\_OCInitStructure.TIM\_OCPolarity = TIM\_OCPolarity\_High;

//输出极性:TIM输出比较极性高

81 TIM\_OC3Init(TIM2, &TIM\_OCInitStructure);

//根据T指定的参数初始化外设TIM2 OC4

82 TIM\_OC3PreloadConfig(TIM2, TIM\_OCPreload\_Enable);

//使能TIM2在CCR2上的预装载寄存器

83

84 //初始化TIM2 Channel2 PWM模式

85 TIM\_OCInitStructure.TIM\_OCMode = TIM\_OCMode\_PWM1;

//选择定时器模式:TIM脉冲宽度调制模式2

86 TIM\_OCInitStructure.TIM\_OutputState = TIM\_OutputState\_Enable;

//比较输出使能

87 TIM\_OCInitStructure.TIM\_Pulse = 0;

88 TIM\_OCInitStructure.TIM\_OCPolarity = TIM\_OCPolarity\_High;

//输出极性:TIM输出比较极性高

89 TIM\_OC4Init(TIM2, &TIM\_OCInitStructure);

//根据T指定的参数初始化外设TIM2 OC4

90 TIM\_OC4PreloadConfig(TIM2, TIM\_OCPreload\_Enable);

//使能TIM2在CCR2上的预装载寄存器

C:\STM32F103C8T6核心板+L298N\_PWM直流电机调速\HARDWARE\TIMER\timer.c

91

92 TIM\_ARRPreloadConfig(TIM2, ENABLE);

93 TIM\_Cmd(TIM2, ENABLE); //使能TIM2

94

95 }

96