**步进电机综合控制系统**

**第一步**：硬件电路连接

【定时器】

内部定时器输出映射到PA15[CN\_7\_17]，定义10号下降沿触发外部中断PC10[CN\_7\_1]

将定时器输出信号连接到10号中断即连接CN\_7\_17与CN\_7\_1引脚

【LCM12864驱动】

LCM12864\_date->GPIOA[7:0] LCM12864\_contr->GPIOB[4:0]

【五线四相步进电机驱动】

右下角接5V电压 GND接地

P1.7→GPIOD0[CN\_7\_9] P1.6→GPIOD1[CN\_7\_10]

P2.2→GPIOD2[CN\_7\_4] P2.1→GPIOD3[CN\_7\_11]

【外部中断按键13控制转速】

外部中断按键->PC13 调整转动方向正反

外部中断按键->PC8[CN\_10\_2]速度增加，极限间隔4

外部中断按键->PC9[CN\_10\_1]速度降低，极限间隔16

【霍尔编码器反馈信号】

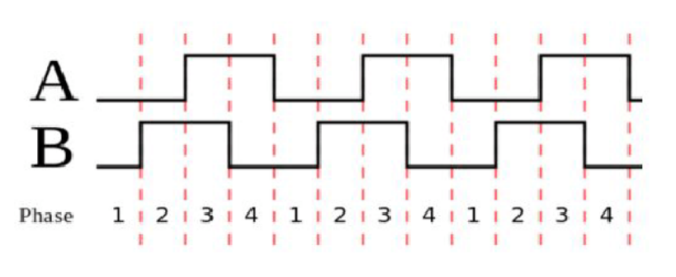
反馈信号A相→PC11[CN\_7\_2]作为通用输入

反馈信号B相→PC12[CN\_7\_3]作为中断

电源控制线悬空

**第二步**：四相编码器原理以及实验思路

四相编码器AB相信号，如下图：



外部中断12连接在B相信号线，且其为下降沿触发。GPIOC11连接在A相信号线，且为通用输入模式。①当步进电机向前旋转（上图中信号方向“→”），中断12触发即B相信号线的下降沿时，A相信号线为高电平，此时正向中断触发次数（number）加一。②当步进电机向后旋转（上图中信号方向“⬅”），中断12触发即B相信号线的下降沿时，A相信号线为低电平，此时反向中断触发次数（number1）加一。

**第三步**：定时器相关设置原理以及实验思路

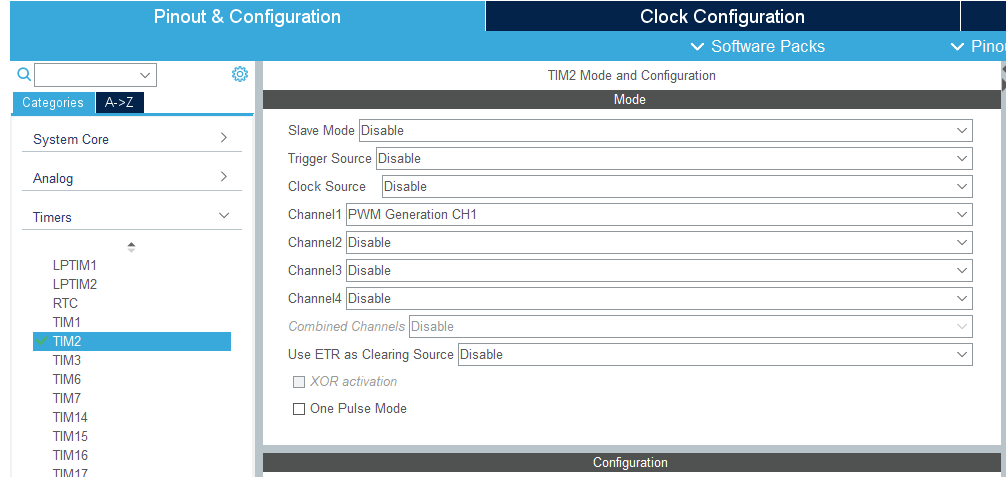
此实验用的是五线四相步进电机，其转速特别低。故用四相编码器测试其转动速度的间隔要比较长才可以。故我们需要周期为10s即频率为0.1Hz占空比为50%的信号。定时器5s高电平，5s低电平。如果用直流电机，其转速就要快很多，检测间隔就可以大幅减小。

定时器参数的计算如下：我们需要频率为0.1Hz占空比为50%的信号。假设定时器输入时钟（即TPCLK）设置为8MHz，如果我们将预分频器值设置为128，得到的计数器频率为62500Hz。要想获得0.1Hz的频率（即周期为10s），需要将计数器周期设置为625000。要想将占空比设置为50%，需要将脉冲设置为312500即625000的一半。

**第四步**：创建工程

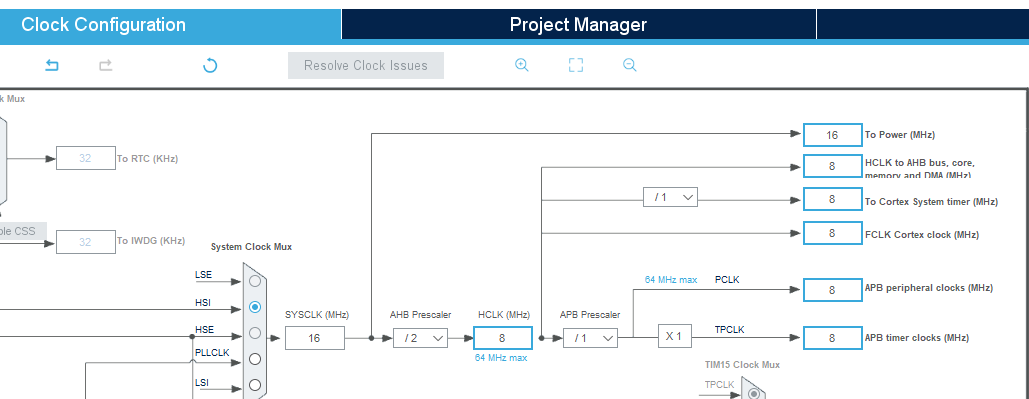
①使能定时器2通道1，这是连接到LD4的PA5的复用功能。配置如下：

在“引脚布局”选项卡，展开定时器部分,配置Channel1



②将定时器2通道1默认分配给PA0，我们要重新映射到PA15。方法如下：将光标停留在PA0引脚上，按住“Ctrl”键同时在PA0上点击鼠标左键，将鼠标指针从PA0拖放到PA15执行重新映射。

③将APB定时器时钟设置为8MHz：“时钟配置”选项卡，在“HCLK(MHz)”下输入8再按下回车，此时“TPCLK”下的“APB timer clocks”（即定时器输入时钟）也变为8MHz,符合我们的要求。设置完后的效果如下图：



④接下来我们要配置定时器2：切换到“引脚布局与配置”选项卡，参数设置中写入我们之前计算所得的值，在“预分频值”填入“127”，“计数器周期”填入“625000”，“脉宽值”设置为“312500”。

⑤配置输入输出和中断引脚。

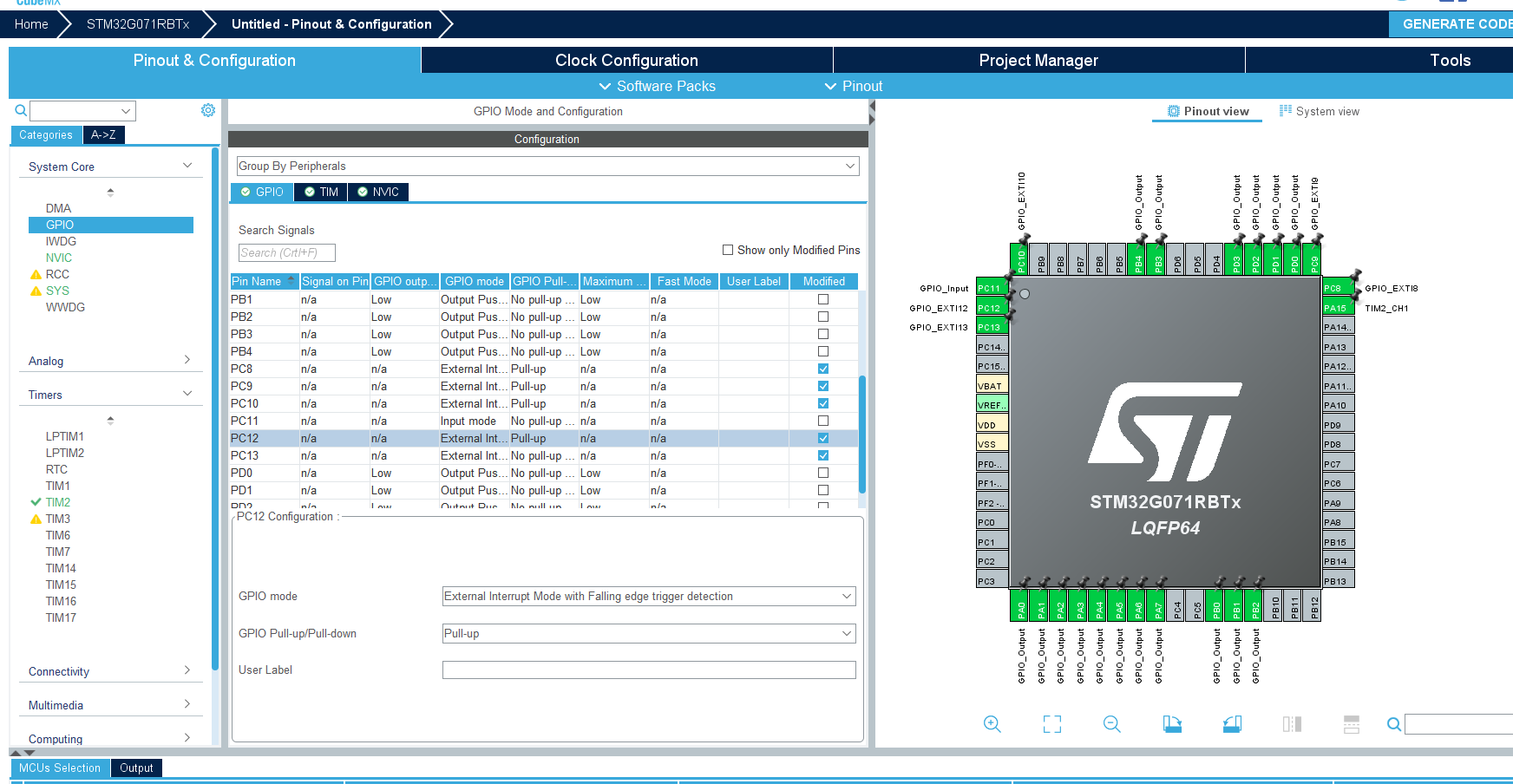
⑥使能4-15号外部中断，并将其设置为下降沿触发、上拉。

⑦总结如下：

TM2\_CH1：PA15 GPIO\_INPUT模式：GPIOC11

GPIO\_OUTPUT模式：GPIOA[7:0] GPIOB[4:0] GPIOD[3:0]

EXTI模式：PC13[按键]、PC12、PC10、PC8、PC9 （下降沿触发、使能）

****

**第五步**：生成代码

**第五步**：添加实验代码

①注释掉原有中断程序入口

②变量定义和函数声明

unsigned char Step\_table[]={0x08,0x04,0x02,0x01};

int time=4;

int mode=0;

int speed=0;

int number=0;//辅助计算speed

int number\_Q=0;

int number\_H=0;

unsigned char tstr[5];

void delay(void);

void lcdwritecmd(unsigned char cmd);

void lcdwritedata(unsigned char dat);

void lcdinit(void);

void lcdsetcursor(unsigned char x, unsigned char y);

void lcdshowstr(unsigned char x, unsigned char y, unsigned char \*str);

③中断引脚判断和释放中断标志位

ITStatus EXTI\_GetITStatus(uint32\_t EXTI\_Line)

{

ITStatus bitstatus = RESET; //chushi wei zhuangtai 0

uint32\_t enablestatus = 0; //chushi shineng zhuangtai 0

/\* Check the parameters \*/

assert\_param(IS\_GET\_EXTI\_LINE(EXTI\_Line));

enablestatus = EXTI->IMR1 & EXTI\_Line;

if (((EXTI->FPR1 & EXTI\_Line) != (uint32\_t)RESET) && (enablestatus != (uint32\_t)RESET))

bitstatus = SET;

else

bitstatus = RESET;

return bitstatus;

}

void EXTI\_ClearITPendingBit(uint32\_t EXTI\_Line)

{

/\* Check the parameters \*/

assert\_param(IS\_EXTI\_LINE(EXTI\_Line));

EXTI->FPR1 = EXTI\_Line;

}

④主函数非循环部分初始化LCM12864显示器并输出提示信息

HAL\_TIM\_PWM\_Start(&htim2,TIM\_CHANNEL\_1);

lcdinit();

lcdshowstr(0,0,"Direction");

lcdshowstr(6,0,"Qian");

lcdshowstr(0,1,"Distance\_Q");

lcdshowstr(6,1,"0");

lcdshowstr(8,0,"Distance\_H");

lcdshowstr(14,0,"0");

lcdshowstr(8,1,"Speed\_10s");

lcdshowstr(14,1,"0");

⑤主函数循环部分控制5线4相步进电机转动

if(mode==0)

for(int i=0;i<4;i++)

{

GPIOD->ODR=Step\_table[i];

HAL\_Delay(time);

}

else

for(int i=0;i<4;i++)

{

GPIOD->ODR=Step\_table[3-i];

HAL\_Delay(time);

}

⑥添加LCM12864相关驱动程序

void delay ()

{

for(int i=0;i<99;i++)

for(int j=0;j<99;j++)

{}

}

void lcdwritecmd(unsigned char cmd)

{

delay(); // 使能 写 指令

GPIOB->ODR=0x18; //RST=1\PSB=1\E=0\RW=0\RS=0

GPIOA->ODR=cmd;

GPIOB->ODR=0x1C; //RST=1\PSB=1\E=1\RW=0\RS=0

delay();

GPIOB->ODR=0x18; //RST=1\PSB=1\E=0\RW=0\RS=0

}

void lcdwritedata(unsigned char dat)

{

delay(); // 使能 写 数据

GPIOB->ODR=0x19; //RST=1\PSB=1\E=0\RW=0\RS=1

GPIOA->ODR=dat;

GPIOB->ODR=0x1D; //RST=1\PSB=1\E=1\RW=0\RS=1

delay();

GPIOB->ODR=0x19; //RST=1\PSB=1\E=0\RW=0\RS=1

}

void lcdinit()

{

lcdwritecmd(0x34); //E 使能 RS1数据0指令 RW1读0写

lcdwritecmd(0x30);

lcdwritecmd(0x0C);

lcdwritecmd(0x01);

}

void lcdsetcursor(unsigned char x, unsigned char y) //字符定位

{

unsigned char address;

if(y==0)

address=0x80+x;

else if(y==1)

address=0x90+x;

else if(y==2)

address=0x88+x;

else

address=0x98+x;

lcdwritecmd(address|0x80);

}

void lcdshowstr(unsigned char x, unsigned char y, unsigned char \*str) //字符输出

{

lcdsetcursor(x,y);

while((\*str)!='\0')

{

lcdwritedata(\*str);

str++;

}

}

⑦添加中断服务程序

void EXTI4\_15\_IRQHandler(void)

{

if (EXTI\_GetITStatus(0x0100) != 0x00) // 0x0100==EXTI\_PIN\_8 加速

{

EXTI\_ClearITPendingBit(0x0100);

if(time>=6)

time=time-2;

else ;

}

else if (EXTI\_GetITStatus(0x0200) != 0x00) // 0x0200==EXTI\_PIN\_9 减速

{

EXTI\_ClearITPendingBit(0x0200);

if(time<=14)

time=time+2;

else ;

}

else if (EXTI\_GetITStatus(0x2000) != 0x00) // 0x2000==EXTI\_PIN\_13 换方向

{

EXTI\_ClearITPendingBit(0x2000); //Clear interrupt flag bit

if(mode==0)

{

mode++;

lcdshowstr(6,0,"Hou ");

}

else

{

mode=0;

lcdshowstr(6,0,"Qian");

}

}

else if (EXTI\_GetITStatus(0x1000) != 0x00) // 0x1000==EXTI\_PIN\_12 四相编码器检测中断到达

{

EXTI\_ClearITPendingBit(0x1000);

if(HAL\_GPIO\_ReadPin(GPIOC,GPIO\_PIN\_11)==1)

{

number\_Q++;

number++;

sprintf(tstr,"%d",number\_Q);

lcdshowstr(6,1,tstr);

}

else if(HAL\_GPIO\_ReadPin(GPIOC,GPIO\_PIN\_11)==0)

{

number\_H++;

number++;

sprintf(tstr,"%d",number\_H);

lcdshowstr(14,0,tstr);

}

else ;

}

else if (EXTI\_GetITStatus(0x0400) != 0x00) // 0x0400==EXTI\_PIN\_10 测速中断到达

{

EXTI\_ClearITPendingBit(0x0400);

speed=number;

number=0;

sprintf(tstr,"%d",speed);

lcdshowstr(14,1," ");

lcdshowstr(14,1,tstr);

}

}

**第六步**：效果阐述

外部中断13号按键改变步进电机转动方向。

外部中断按键8号9号调节步进电机转动速度。

四相编码器检测步进电机运行状态。

定时器以10秒为周期触发中断，测量步进电机转速。

LCM12864显示步进电机转动方向、距离、速度等。

