**霍尔编码器**

**第一步**：硬件电路连接

【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

【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

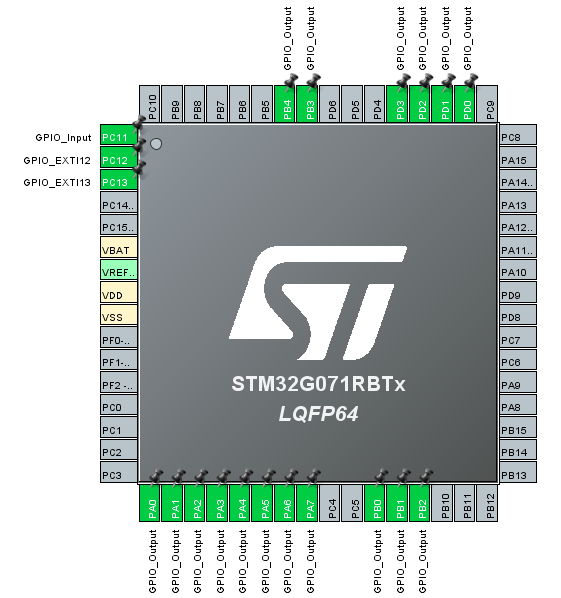
电源控制线悬空

**第二步**：创建工程

GPIO\_INPUT模式：GPIOC11

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

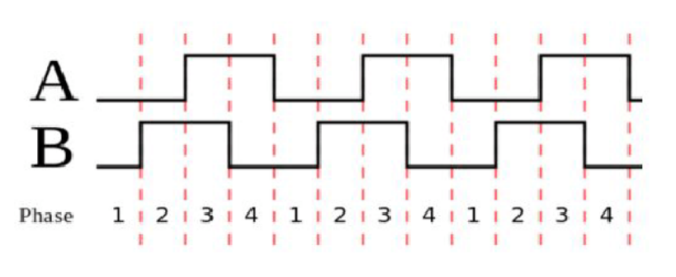
EXTI模式：PC13[按键]、PC12 （下降沿触发、使能）

****

**第三步**：生成代码

**第四步**：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 speed=4;

int mode=0;

int number=0;

int number1=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

lcdinit();

lcdshowstr(0,0,"Qian");

lcdshowstr(5,0,"Hou");

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(speed);

}

else

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

{

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

HAL\_Delay(speed);

}

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(0x2000) != 0x00) // 0x2000==EXTI\_PIN\_13

{

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

if(mode==0)

mode++;

else

mode=0;

}

if (EXTI\_GetITStatus(0x1000) != 0x00) // 0x1000==EXTI\_PIN\_12

{

EXTI\_ClearITPendingBit(0x1000);

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

{ number++;

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

lcdshowstr(0,1,tstr); }

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

{ number1++;

sprintf(tstr,"%d",number1);

lcdshowstr(5,1,tstr); }

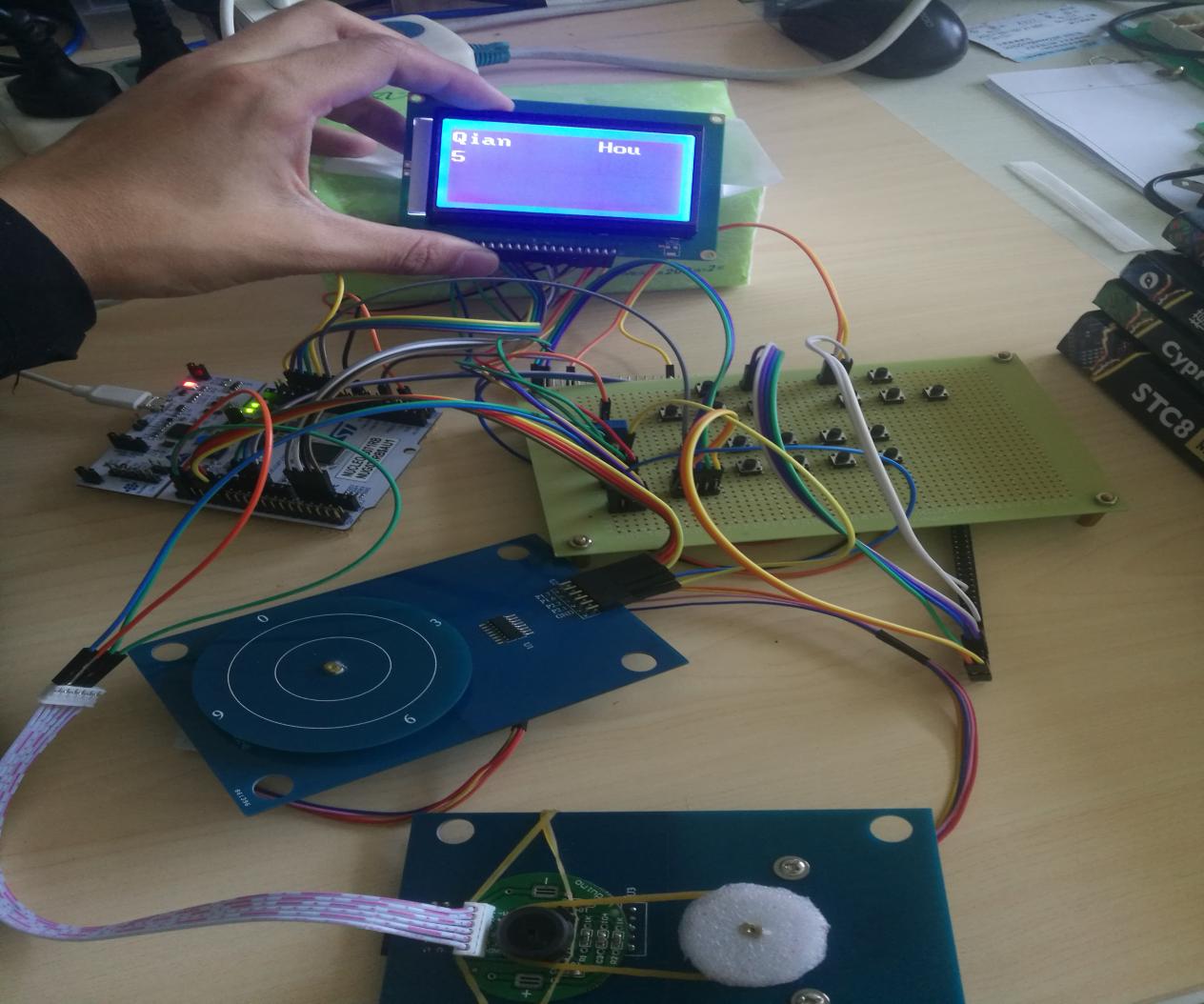
else ;

}

}

**第六步**：效果演示

按下13号外部中断按键，改变步进电机转动方向。4相编码器判断转动状态，并显示在LCD12864屏幕上。左侧显示向前转动时的中断触发次数，右侧显示向后转动时的中断触发次数。



**第七步**：示波器测试数据

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（蓝色）的下降沿时，A相信号线（黄色）为高电平，此时正向中断触发次数（number）加一。

向前

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（蓝色）的下降沿时，A相信号线（黄色）为低电平，此时反向中断触发次数（number1）加一。

向后

**第八步**：总结展望

结合定时器测定转速。