### TM1637数码管-按键修改时间

项目要求:

1. 开机显示时间: 12:30
2. 每隔10秒,切换"小时:分钟"与"分钟:秒钟"
3. 按"按键1",则时间增加1小时,按"按键2",则时间增加1分钟

电路连接：

**电路连接**

开发板:PB6--TM1637:CLK

开发板:PB7--TM1637:DIO

开发板: V3 --TM1637:VCC

开发板: G--TM1637:GND

PB9——“按键1”—— “接地”

PA0——“按键2”—— “接地”

1. 如何表示时间？

struct Time

{

 int hour;

 int min;

 int sec;

};

初始化时间为: 12 :30:00

extern struct Time time;

void time\_init(void)

{

 time.hour=12;

 time.min=30;

 time.sec=0;

}

1. 如何走时？

每经过1秒时间，则sec 增加1，并检查时间的合法性

void time\_add1S()

{

 time.sec++;

 time\_check();

}

1. 如何检查时间的合法性？

void time\_check(void)

{

 if(time.sec>59)

 {

 time.min+=1; //分钟数加1

 time.sec -= 60; //秒数减60

 }

 if(time.min>59)

 {

 time.hour+=1;

 time.min -= 60;

 }

 if(time.hour>23)

 {

 time.hour -= 24;

 }

}

1. 按"小时:分钟"格式显示时间

void time\_display\_hour\_min()

{

 if(time.sec%2 == 0)

 Display10(time.hour,time.min,1); //冒号亮

 else

 Display10(time.hour,time.min,0); //冒号灭

}

1. 按"分钟:秒钟"格式显示时间

void time\_display\_min\_sec()

{

 Display10(time.min,time.sec,1); //冒号常亮

}

1. 实现每隔10秒,切换"小时:分钟"与"分钟:秒钟"

struct Time time;

int main(void)

 {

 while(1)

 {

 if((time.sec / 10)%2==0) //每隔10秒切换显示方式

 {

 time\_display\_hour\_min();

 }

 else

 {

 time\_display\_min\_sec();

 }

 Delay\_ms(1000); //延时1ms

 time\_add1S(); //走时

 }

 }

1. 初始化“按键1”与“按键2”为上拉输入模式
2. 初始化“按键1”与“按键2”引脚所对应的外部中断9、0
3. 编写外部中断9、0所对应的中断函数

/\*\*

 \* 功能：外部中断9\_5中断服务函数,

 \*/

void EXTI9\_5\_IRQHandler(void)

{

 if(EXTI\_GetITStatus(EXTI\_Line9) == SET) //EXTI\_Line9中断已发生

 {

 EXTI\_ClearITPendingBit(EXTI\_Line9); //清除 EXTI\_Line9上的中断标志位

 Delay\_ms(10);

 if(KEY\_B9 ==0 )

 {

 time\_add1H();

 time\_display\_hour\_min();

 }

 }

}

void EXTI0\_IRQHandler(void)

{

 if(EXTI\_GetITStatus(EXTI\_Line0) == SET) //EXTI\_Line9中断已发生

 {

 EXTI\_ClearITPendingBit(EXTI\_Line0); //清除 EXTI\_Line9上的中断标志位

 Delay\_ms(10);

 if(KEY\_A0 == 0)

 {

 time\_add1M();

 time\_display\_hour\_min();

 }

 }

}

1. 设置中断抢占优先级

void setNVIC(void)

{

 NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2); //设置中断优先级分组

 NVIC\_InitTypeDef NVIC\_InitStructure; //定义NVIC初始化结构体

 NVIC\_InitStructure.NVIC\_IRQChannel = EXTI9\_5\_IRQn; //设置中断号

 NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 2; //设置抢占优先级

 NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 2; //设置子优先级

 NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //开启中断

 NVIC\_Init(&NVIC\_InitStructure); //设置生效

 NVIC\_InitStructure.NVIC\_IRQChannel = EXTI0\_IRQn; //设置中断号

 NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 1; //设置抢占优先级

 NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 1; //设置子优先级

 NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //开启中断

 NVIC\_Init(&NVIC\_InitStructure); //设置生效

}