```
#include <reg52.h>      //中国欧软 ehfeo@126.com
#include <stdio.h>      //柳州市一职校 电气自动化专业部

#define MODE_IN1 0X28   //输入模式1地址
#define MODE_IN2 0X29   //输入模式2地址

sbit DO = P1^3;         //LC72144数据输出
sbit CL = P1^2;         //LC72144通讯同步时钟
sbit DI = P1^1;         //LC72144数据输入
sbit CE = P1^0;         //LC72144片选
sbit KEY1 = P3^0;       //换台按钮,向上换台
sbit KEY2 = P3^1;       //换台按钮,向下换台
unsigned char bdata IN1_byte3;
sbit SNS=IN1_byte3^0;   //0 当DVS位=0时， SNS=1则AM输入， 2-40MHZ；
                        //SNS=0则AM输入， 0.5-10MHZ
sbit DVS=IN1_byte3^1;   //1 当DVS位=1时,FM输入,10-160MHZ
sbit PDC0=IN1_byte3^2;  //1 子PLL鉴频器控制,设置都为1,做标准操作
sbit PDC1=IN1_byte3^3;  //1 子PLL鉴频器控制,设置都为1,做标准操作
sbit XR0=IN1_byte3^4;   //1 使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
sbit XR1=IN1_byte3^5;   //1 使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
sbit XR2=IN1_byte3^6;   //0 使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
sbit XR3=IN1_byte3^7;   //0 使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
unsigned char bdata IN1_byte4;
sbit DT0=IN1_byte4^0;   //1 end of adc
sbit DT1=IN1_byte4^1;   //0 end of adc
sbit ADS=IN1_byte4^2;   //1 start adc
sbit CTE=IN1_byte4^3;   //0 COUNT RESET
sbit CTS0=IN1_byte4^4;  //1 HCTR COUNT INPUT
sbit CTS1=IN1_byte4^5;  //1 HCTR COUNT INPUT
sbit GT0=IN1_byte4^6;   //1 64MS COUNT
sbit GT1=IN1_byte4^7;   //1 64MS COUNT
unsigned char bdata IN2_byte1;
sbit IO0=IN2_byte1^0;   //0 IN IF COUNTER
sbit IO1=IN2_byte1^1;   //1 OUT FM AM 1=AM 0=FM
sbit IO2=IN2_byte1^2;   //0 IN ST IND
sbit IO3=IN2_byte1^3;   //1 OUT ST/MONO 1=ST 0=MONO
sbit IO4=IN2_byte1^4;   //1 OUT FM AGC
sbit IO5=IN2_byte1^5;   //1 OUT AM AGC
sbit ADI0=IN2_byte1^6;  //1 ADC1 CONVERT S-METER SIGNAL
sbit ADI1=IN2_byte1^7;  //0 ADC1 CONVERT S-METER SIGNAL
unsigned char bdata IN2_byte2;
sbit OUT0=IN2_byte2^0;  //1 IN
sbit OUT1=IN2_byte2^1;  //0 FM
sbit OUT2=IN2_byte2^2;  //1 IN
sbit OUT3=IN2_byte2^3;  //1 STEREO
sbit OUT4=IN2_byte2^4;  //1 HIGH GAIN
sbit OUT5=IN2_byte2^5;  //1 HIGH GAIN
sbit CTP=IN2_byte2^6;   //0 PULL DOWN WHEN COUNT RESET(CTE=0)
sbit CTC=IN2_byte2^7;   //0 normal senstivity
unsigned char bdata IN2_byte3;
sbit HI6=IN2_byte3^0;   //1 USED AS HCTR INPUT
sbit LI7=IN2_byte3^1;   //1 USED AS LCTR INPUT
sbit IL0=IN2_byte3^2;   //0 open
sbit IL1=IN2_byte3^3;   //0 open
sbit ULD=IN2_byte3^4;   //0 OUT5 flag
sbit UL0=IN2_byte3^5;   //1 失锁检测,余量±1.0μs,延时1-2ms
sbit UL1=IN2_byte3^6;   //1 失锁检测,余量±1.0μs,延时1-2ms
sbit XS0=IN2_byte3^7;   //1 10.35MHz晶振
unsigned char bdata IN2_byte4;
sbit XS1=IN2_byte4^0;   //1 10.35MHz晶振
sbit XB=IN2_byte4^1;    //1 output 10.35MHz clk to driver la1787m
sbit DZ0=IN2_byte4^2;   //0 DZa DEAD ZONE
sbit DZ1=IN2_byte4^3;   //0 DZa DEAD ZONE
sbit TEST0=IN2_byte4^4; //0 MUST BE 0
sbit TEST1=IN2_byte4^5; //0 MUST BE 0
sbit TEST2=IN2_byte4^6; //0 MUST BE 0
sbit DLC=IN2_byte4^7;   //0 PLL NORMAL OPERATION
unsigned char PLL_High_8_Bit,PLL_Low_8_Bit,Current_station;
            //PLL高8字节，   低八字节，      当前电台指针
unsigned int  PLL_Temp;//PLL临时值
unsigned int code Freq_PLL[11]={0XEC6,0XFF8,0X101C,0X10E8,0X112C,
                                0X115C,0X1180,0X11C4,0X11DC,0X120C,0X123C};
                                //预存电台频率
                                //计算方法：（电台频率+10800）÷25
                                //83.75MHz,91.4MHz,92.3MHz....
void delay(unsigned int time)
  {
  while(--time);
  }
void Init(void)
  {
  SNS=0;    //当DVS位=0时，SNS=1则AM输入，2-40MHZ；SNS=0则AM输入，0.5-10MHZ
  DVS=1;    //当DVS位=1时,FM输入,10-160MHZ
  PDC0=1;   //子PLL鉴频器控制,设置都为1,做标准操作
  PDC1=1;   //子PLL鉴频器控制,设置都为1,做标准操作
  XR0=1;    //使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
  XR1=1;    //使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
  XR2=0;    //使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
  XR3=0;    //使用25KHZ参考频率,R0=1,R1=1,R2=0,R3=0
  DT0=1;    //end of adc
  DT1=0;    //end of adc
  ADS=1;    //start adc
  CTE=0;    //COUNT RESET
  CTS0=1;   //HCTR COUNT INPUT
  CTS1=1;   //HCTR COUNT INPUT
  GT0=1;    //64MS COUNT
  GT1=1;    //64MS COUNT
  IO0=1;    //IN IF COUNTER
  IO1=1;    //OUT FM AM 1=AM 0=FM
  IO2=0;    //IN ST IND
  IO3=1;    //OUT ST/MONO 1=ST 0=MONO
  IO4=0;    //IN FM AGC
  IO5=0;    //IN AM AGC
  ADI0=1;   //ADC1 CONVERT S-METER SIGNAL
  ADI1=0;   //ADC1 CONVERT S-METER SIGNAL
  OUT0=0;   //IN
  OUT1=0;   //FM
  OUT2=1;   //IN
  OUT3=1;   //STEREO
  OUT4=1;   //HIGH GAIN
  OUT5=1;   //HIGH GAIN
  CTP=0;    //PULL DOWN WHEN COUNT RESET(CTE=0)
  CTC=0;    //normal senstivity
  HI6=1;    //USED AS HCTR INPUT
  LI7=1;    //USED AS LCTR INPUT
  IL0=0;    //open
  IL1=0;    //open
  ULD=0;    //OUT5 flag
  UL0=1;    //失锁检测,余量±1.0μs,延时1-2ms
  UL1=1;    //失锁检测,余量±1.0μs,延时1-2ms
  XS0=1;    //10.35MHz晶振
  XS1=1;    //10.35MHz晶振
  XB=1;     //output 10.35MHz clk to driver la1787m
  DZ0=0;    //DZA DEAD ZONE
  DZ1=0;    //DZA DEAD ZONE
  TEST0=0;  //MUST BE 0
  TEST1=0;  //MUST BE 0
  TEST2=0;  //MUST BE 0
  DLC=0;    //PLL NORMAL OPERATION
  }
```

```
void IN(unsigned char mode,dat1,dat2,dat3,dat4)//向PLL输入数据
  {
  unsigned char i;  //临时变量
  CE=0;
  CL=1;
  for(i=0;i<8;i++)  //发送模式地址
    {
    DI=mode&0x01;
    CL=0;
    CL=1;
    mode=mode>>1;
    }
    CE=1;
  for(i=0;i<8;i++)  //发送第一字节
    {
    DI=dat1&0x01;
    CL=0;
    CL=1;
    dat1=dat1>>1;
    }
  for(i=0;i<8;i++)  //发送第二字节
    {
    DI=dat2&0x01;
    CL=0;
    CL=1;
    dat2=dat2>>1;
    }
  for(i=0;i<8;i++)  //发送第四字节
    {
    DI=dat3&0x01;
    CL=0;
    CL=1;
    dat3=dat3>>1;
    }
  for(i=0;i<8;i++)  //发送第五字节
    {
    DI=dat4&0x01;
    CL=0;
    CL=1;
    dat4=dat4>>1;
    }
    CE=0;
  }

void Send_PLL(void)
  {
      PLL_Low_8_Bit=PLL_Temp&0x00ff;    //获得PLL值低八位
      PLL_High_8_Bit=PLL_Temp>>8;       //获得PLL值高八位
      IN(MODE_IN1,PLL_Low_8_Bit,PLL_High_8_Bit,IN1_byte3,IN1_byte4);//模式
  }
void main(void)
  {
  Init();       //初始化
  IN(MODE_IN2,IN2_byte1,IN2_byte2,IN2_byte3,IN2_byte4);//设定PLL参考频率等
  while(1)
    {
    if(!KEY1)   //如果按键被按下
      {
      while(!KEY1);         //等待按键弹起
      Current_station++;    //指向下一电台
      if(Current_station>10){Current_station=0;}    //越界检测
      PLL_Temp=Freq_PLL[Current_station];           //获得PLL值
      Send_PLL();           //发送PLL值,换台
      delay(50000);         //延时一段时间, 避免按键接触不良导致换台过快
      }
    if(!KEY2)                 //如果按键被按下
      {
      while(!KEY2);           //等待按键弹起
      Current_station--;      //指向上一电台
      if(Current_station>10){Current_station=0;}
                              //越界检测
      PLL_Temp=Freq_PLL[Current_station];
                              //获得PLL值
      Send_PLL();             //发送PLL值,换台
      delay(50000);           //延时一段时间, 避免按键接触不良导致换台过快
      }
    }
  }
```