

/**

本例程为 9234+9793 (ROHM 88 段 LCDdrive) 例程。

采用 I2C 方式传送数据。

若使用其他厂家的 I2C 方式通讯的 LCDdrive 可能需要稍加改动，
ISP 通讯方式不可使用。

**/

```
#define uchar unsigned char
#define slave_addr 0x7c //slave 地址
#define ICSET 0xea //ICSET
```

```
#define BLKCTL 0xf0
//#define DISCTL 0xa0 //正常
#define DISCTL 0xbc //低功耗
#define ADSET 0x00
#define MODESET 0xc8
#define APCTL8 0xfe
#define APCTL 0xfc
#define EVRSET0 0xe0
#define EVRSET1 0xe5
#define EVRSET2 0xf8
```

```
#define SDA P6.1
#define MSDA PM6.1
#define SCL P6.2
#define MSCL PM6.2
```

```
unsigned char buff[24];
```

```
void start_i2c();
void stop_i2c();
void send_byte(uchar tempb);
void data_display(void);
```

```
void ini_display(void);
void on_display(void);
void main(void)
{
    ini_display();
    on_display();
    while(1)
        {data_display();
        }
}
```

```
/******
```

起动总线函数

函数原型: void start_i2c();

Function: start on the I2C bus

```
*****/
```

```
void start_i2c()
```

```
{
    SDA=1;
    nop();
    SCL=1;
    nop();nop();nop();nop();
    SDA=0;
    nop();nop();nop();nop();
    SCL=0;

}
```

```
/******
```

停止总线函数

函数原型: void stop_i2c();

Function: stop the I2C bus

```
*****/
```

```
void stop_i2c()
```

```
{
```

```
SDA=0;
nop();
SCL=1;
nop();nop();
SDA=1;
nop();
}
```

```
/******************************************************************
```

字节数据传送函数

函数原型: void send_byte(uchar tempb);

Function: 将数据 C 发送出去，可以是地址，也可以是数据，发完后等待回应，并对此状态位进行操作（不应答或非应答都使 ack=0 ），发送数据正常，ack=1;ack=0 表示被控器无应答或损坏。

```
******************************************************************/
```

```
void send_byte(uchar tempb)
```

```
{
  uchar bit_count;
  for (bit_count=0;bit_count<8;bit_count++)
  {
    if ((tempb<<bit_count)&0x80) {SDA=1;}
    else {SDA=0;}
    nop();
    SCL=1;
    nop();nop();nop();nop();nop();
    SCL=0;
  }
  MSDA=1;
  nop();nop();
//  SDA=1;
  nop();nop();

  SCL=1;
  if (SDA==1) ack=0;
```

```
else ack=1;
MSDA=0;
SCL=0;
nop();
}
```

```
void data_display(void)
{ register uchar i;
  flag_dis=0;
  stop_i2c();
  start_i2c();
  send_byte(slave_addr);
//  send_byte(DISCTL);
//  send_byte(BLKCTL);
//  send_byte(APCTL);
//  send_byte(MODESET);
//  if(LVIF)
//    send_byte(EVRSET0); //3V
//  else
//    send_byte(EVRSET1); //5V
//  send_byte(EVRSET2);
  send_byte(ADSET);
  clrwdt();
  for(i=0;i<25;i++)
    {send_byte(buff[i]);}

  stop_i2c();

}
```

```
/******初始化 LCD-DRIVER******/
```

```
void ini_display(void)
{register  uchar i;
  MSCL=0;
  MSDA=0;
  stop_i2c();
```

```
start_i2c();
send_byte(slave_addr);
send_byte(ICSET);
send_byte(DISCTL);
send_byte(ADSET);
for(i=0;i<25;i++)
    send_byte(0x0);
stop_i2c();
}
```

```
void on_display(void)
{start_i2c();
  send_byte(slave_addr);
  send_byte(DISCTL);
// send_byte(BLKCTL);
  send_byte(APCTL);
  send_byte(MODESET);
stop_i2c();
}
```