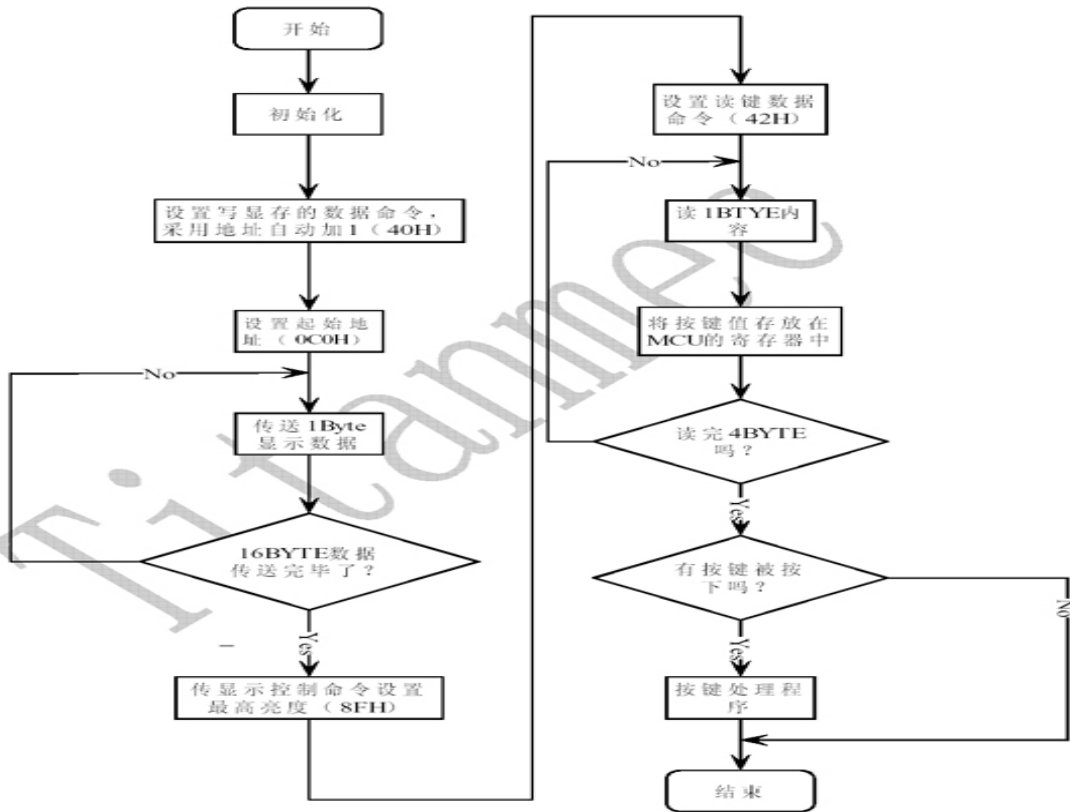


# S3F9454 C for TM1629B driver

早期作数码管的驱动设计时，多采用美信的驱动 IC，近期使用了天威的 TM1629B，不禁眼前一亮，好用，而且价廉物美，值得向工程师朋友推荐。

MCU 使用三星 S3F9454，接口只需要三个 IO 口，如其 Data\_sheet 所示。下面的程序流程图可与程序相印证，仅供参考。



```
//=====
//
// MCU:      S3F9454@内部 3.2Mhz
//=====

#include "ioS3C9454.h"
#include "intrinsic.h"
#include "app01.h"

#define uchar unsigned char
#define uint  unsigned int
#define ulong unsigned long

#define nop (__no_operation())
#define di  (__disable_interrupt())
#define ei  (__enable_interrupt())

// data for TM6219b, refrent 采用自动加一地址程序流程
#define  Cnt_CWdata  0
#define  Cnt_CA      1
#define  Cnt_CD      18
#define  Cnt_CRdata  19
```

```

#define Data_CWdata 0X40
#define Data_CA 0XC0
#define Data_CD 0X8F
#define Data_CD_OFF 0X87
#define Data_CRdata 0X42

#define Tm6219CntMax 192
#define Tm6219CntRead 160

//-----          const code / 表          -----

__code const volatile uchar SmartOption[4] @0x003c = {0x0,0x0,0x67,0x00}; // 不允许低电压复位;外部 8M 晶振

__code const volatile uchar DigitCode[16] @0x0040 =      ///数码管显示代码定义(在 ROM=0X40)
{
    0x3f,  // '0'
    0x06,  // '1'
    0x5b,  // '2'
    0x4f,  // '3'
    0x66,  // '4'
    0x6d,  // '5'
    0x7d,  // '6'
    0x27,  // '7'
    0x7f,  // '8'
    0x6f,  // '9'
    0x77,  // 'A'
    0x7c,  // 'b'
    0x39,  // 'C'
    //0x5E, // 'd'
    0x00,  // 'd' void
    //0x79, // 'E'
    0x38,  // 'E' L
    //0x71, // 'F'
    0x76,  // 'F' H
};
//-----          interrupt / 中断          -----
#pragma vector=0x00
__interrupt void int_9454() //中断函数
{
}

//-----          option_reload_pro 配置重载          -----
void option_reload_pro( void )
{
// TOCON = TOCON_D;
CLKCON = CLKCON_D;
BTCN = BTCN_D;
}

```

```

if( TBuzzer )
    POCONH = POCONH_BB_D;
else
    POCONH = POCONH_D;

POCONL &= POCONL_A_D;
POPND = POPND_D;

// P1CON = P1CON_D;

P2CONH = P2CONH_D;
P2CONL = P2CONL_D;

PWMDATA = PWMDATA_D;
PWMCON = PWMCON_D;

STOPCON = STOPCON_D;

ei; // SAM88RCRI interrupt pending
}
//----- initial_pro / 初始化 -----
void initial_pro( void )
{
di;
option_reload_pro();
ei; // SAM88RCRI interrupt pending

// for constant data be

Buff0 = SmartOption[0];
}
//----- display_drive_pro / 显示驱动 -----
void display_drive_pro( void )
{
if( Buff0 >= Tm6219CntMax )
    Tm6219Cnt = 0;

if( ( Tm6219Cnt & 0X7 ) == 0 )
{
// detect data or command transf
// and insert STB set
Buff0 = Tm6219Cnt; // and get new data buff
Buff0 >>= 3;

if( Buff0 == Cnt_CWdata ) 0_STB_PORT |= (1<<0_STB_B);
else if( Buff0 == Cnt_CA ) 0_STB_PORT |= (1<<0_STB_B);
}
}

```

```

else if( Buff0 == Cnt_CD      )  O_STB_PORT |= (1<<O_STB_B);
else if( Buff0 == Cnt_CRdata )  O_STB_PORT |= (1<<O_STB_B);

if( Buff0 <= Cnt_CRdata )
    Tm6219DataBuff = Tm6218Data[Buff0];
}

if( Tm6219Cnt < Tm6219CntRead )
{
    // normal transf

    POCONL = POCONL_O_D;

    O_STB_PORT &= ~(1<<O_STB_B);

    if( Tm6219DataBuff & 0X1 )
    {
        O_DIO_PORT |= (1<<O_DIO_B);
    }
    else
    {
        O_DIO_PORT &= ~(1<<O_DIO_B);
    }

    O_CLK_PORT &= ~(1<<O_CLK_B);

    Tm6219DataBuff >>= 1;

    Tm6219Cnt++;

    O_CLK_PORT |= (1<<O_CLK_B);
}
else
{
    // Key read

    POCONL = POCONL_R_D;

    O_STB_PORT &= ~(1<<O_STB_B);

    O_CLK_PORT &= ~(1<<O_CLK_B);

    Tm6219DataBuff >>= 1;

    if( I_DIO_PORT & (1<<I_DIO_B) )
    {
        Tm6219DataBuff |= 0X80;
    }
    else
    {

```

```

        Tm6219DataBuff &= 0X7F;
    }

    Tm6219Cnt++;

    O_CLK_PORT |= (1<<O_CLK_B);

                                                    // data load

    Buff0 = Tm6219Cnt;

    if( ( Buff0&0x7 ) == 0 )
    {
        Buff0 >>= 3;

        Buff0 -= (Cnt_CRdata+1);
        Buff0 &= 0x3;

        Key[Buff0] = Tm6219DataBuff;
    }
}

//----- display_pro / 显示 -----
void display_pro( void )
{
    Tm6218Data[ Cnt_CWdata ] = Data_CWdata ;
    Tm6218Data[ Cnt_CA      ] = Data_CA      ;

    if( Fsys&Lamp ) Tm6218Data[ Cnt_CD      ] = Data_CD_OFF ;
    else            Tm6218Data[ Cnt_CD      ] = Data_CD      ;

    Tm6218Data[ Cnt_CRdata ] = Data_CRdata ;

                                                    // display data write to buff Tm6218Data[x]
}

//----- key_pro / 键输入 -----
void key_pro( void )
{
}

//----- time_base_pro / 时间基准 -----
void time_base_pro( void )
{
}

//----- output_pro / 输出 -----
void output_pro( void )
{
}

```

```

//----- sensor_pro / 传感器 -----
void sensor_pro( void )
{
}
//----- main_program / 主程序 -----
void main()
{
    initial_pro();
    while ( 1 )
    {
        option_reload_pro();

                                                                    // timer_base: 0.512ms

        if( FTimeBase )
        {
            FTimeBase = 0;

            display_drive_pro();

            TimeBase0 ++;
            if( TimeBase0 == 0 )
                TimeBase1 ++;

                                                                    // loop_time_base: 2ms

            switch( TimeBase0 & 0x3 )
            {
                case 0:
                    display_pro();
                    break;
                case 1:
                    key_pro();
                    break;
                case 2:
                    time_base_pro();
                    output_pro();
                    break;
                case 3:
                default:
                    sensor_pro();
                    break;
            }
        }
    }
}

```

#### Note:

- 1) 文中提及的名称和商标为相关所有者所有
- 2) [SPM专用编程器](#)