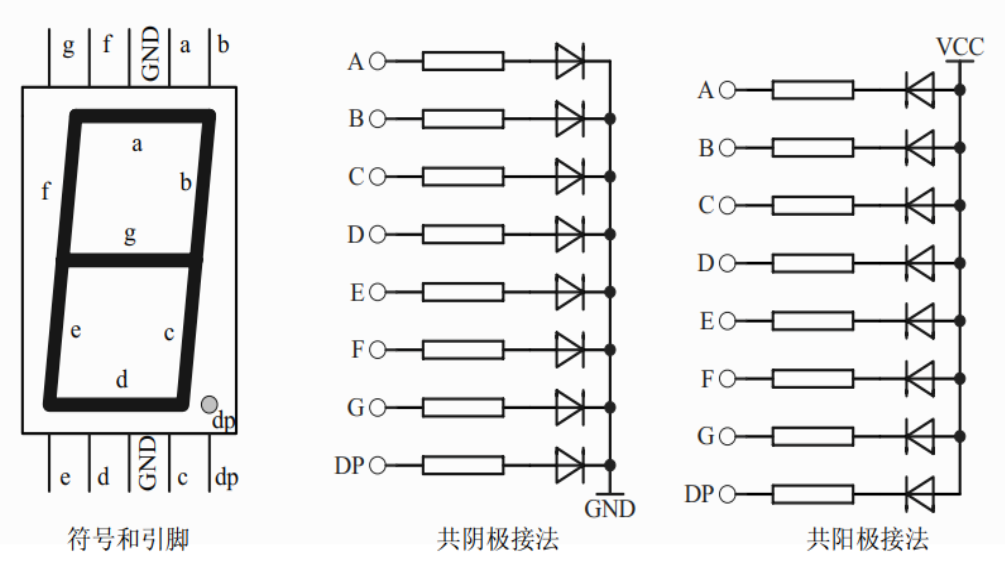
# TM1637数码管

## 数码管介绍

数码管结构如下图所示



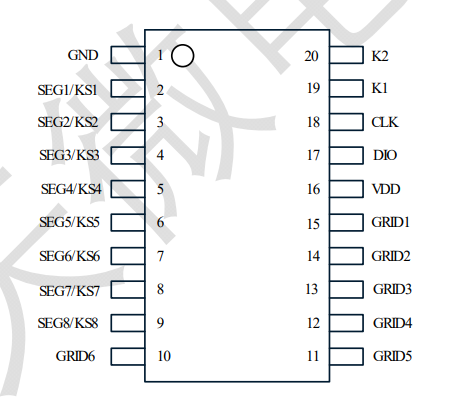
数码管内部8个发光二极管，其中A、B、C、D、E、F、G是条形发光二极管，组成数字，DP是圆点发光二极管。

数码管分为共阴极数码管和共阳极数码管。共阴极数码管公共端接地，共阳极数码管公共端接电源，A-DP接控制引脚。实际上数码管A-DP可以接MCU的8个GPIO来控制。以共阳极数码管为例，A脚低电平时，发光二极管导通，a处发光，以同样的方式让f、b、e、c、d处发光，数码管就显示“0”,此时低电平以0表示，高电平以1表示，数码管的“0”编码就是11000000（顺序DP-A）,也就是0xC0。以同样的方式显示其他数字，共阳极数码管的0-F的字形码为：0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e，共阴极数码管则反之。

虽然直接用GPIO也能控制数码管，但是一个数码管就要占用8个GPIO口，很耗费资源，因此就需要使用专门的驱动芯片来节省资源。

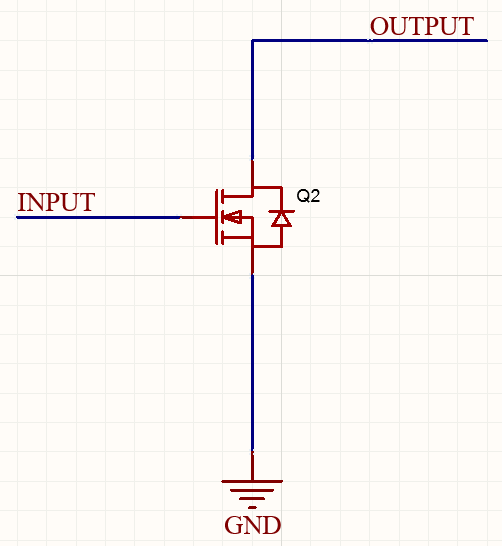
## TM1637驱动芯片

TM1637 是一种带键盘扫描接口的LED（发光二极管显示器）驱动控制专用电路，支持6个8段共阳极数码管，引脚信息如下：



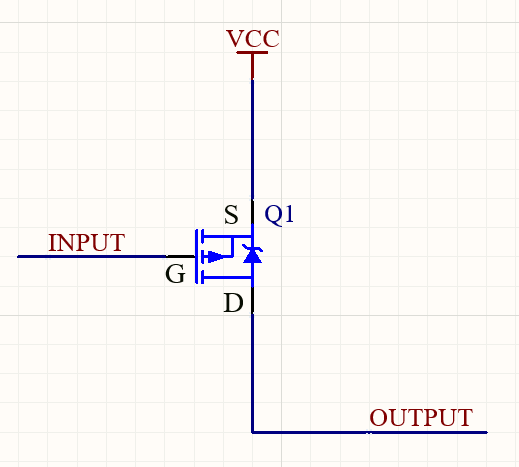


这里的SG1-SG8引脚是N管开漏输出，GRID6-GRID1是P管开漏输出，N管开漏如下所示：(手册说需要5V供电，实测3.3V供电也是可以)



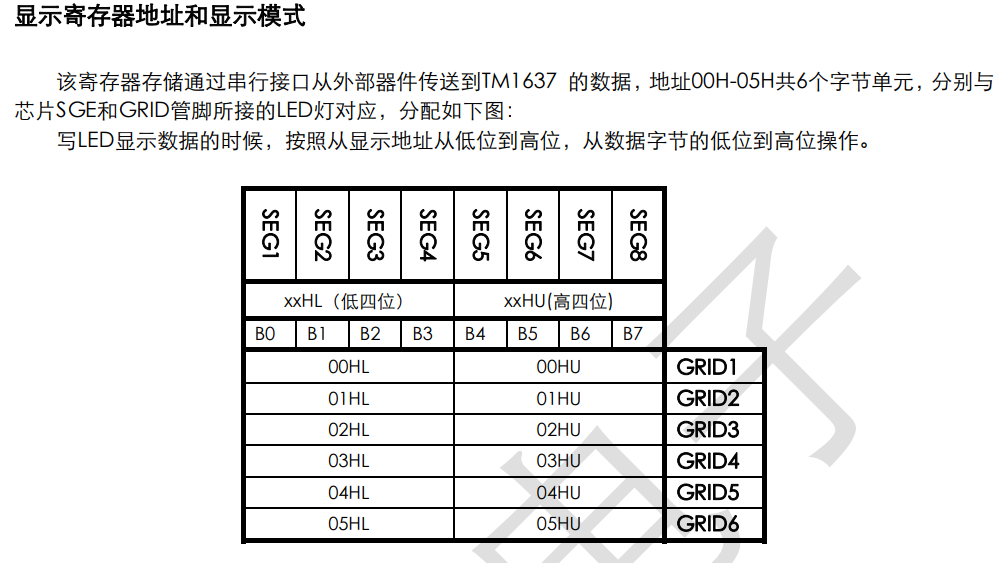
当INPUT输入高电平时，Q2导通，OUTPUT输出低电平。N管开漏无法直接输出高电平，但可以外接上拉电阻输出高电平。

P管开漏如下图所示：

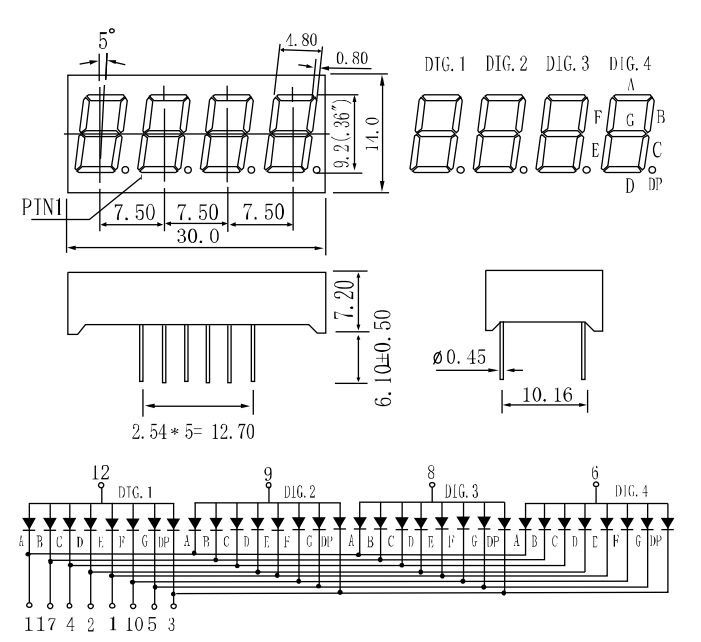


当INPUT输入低电平时，Q1导通，OUTPUT输出高电平。P管开漏无法直接输出低电平，但可以外接下拉电阻输出低电平。

显示寄存器地址和显示模式如下图所示：



这里实际上就是GRID1-6对应6个数码管，SEG1-8对应每位数码管的8个发光二极管。本次我们使用的数码管模块是一个4位共阳极数码管，如下图所示

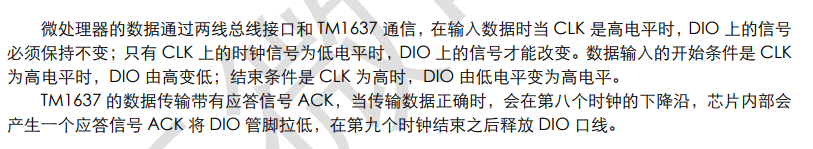


a-dp依次连接SEG1-SEG8,4个公共脚DIG.1-DIG.4依次连接GRID1-GRID4。以地址00H为例，将寄存器的SEG1位置1，芯片上SEG1引脚处的Nmos管打开，输出低电平，对应的数码管DIG.1的a管导通发亮。打开或关闭其它发光二极管也是这样处理。

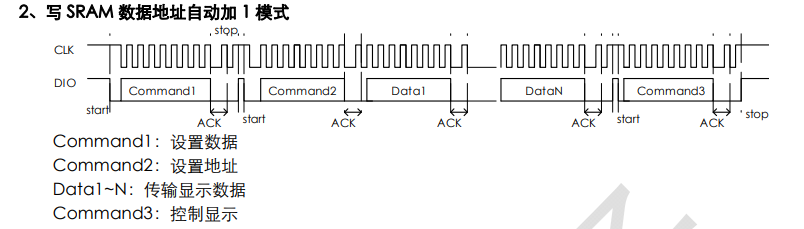
这里可以注意到，数码管的原理和一般的小灯没什么不同，但这个4位数码管并不是一个个单独控制，a-dp都是并联的，假如DIG.1的二极管都导通，DIG.2一上电也会跟着都导通，那要怎样才能让数字同时出现？这就需要利用视觉暂留现象，人眼看到的影像消失后，人眼会保留0.1-0.4s左右的图像，所以只需要以很快的速度轮流向各位数码管送出字形码和开关数码管，看起来就好像各位数码管同时都在显示。

还有一点需要注意的是，TM1637依照这样的连接方式控制共阳极数码管，显示寄存器的值应当是阳极数码管的字形码取反。例如要显示“0”字形码值是11000000，显示寄存器值就是00111111。

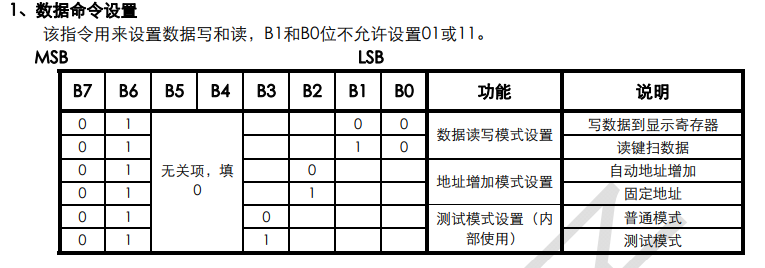
## 通讯协议



TM1637的通讯方式很类似于IIC，但跟IIC不同。IIC协议传输数据是高位在前，低位在后，TM1637是低位在前，高位在后。数码管显示的时序如下（使用地址自动加1的模式。）

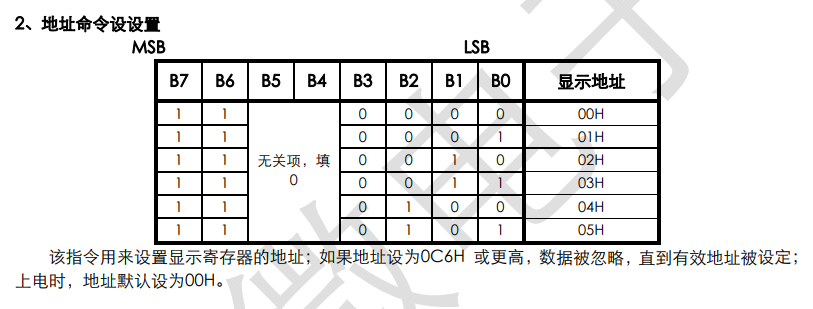


Command1数据命令如下：

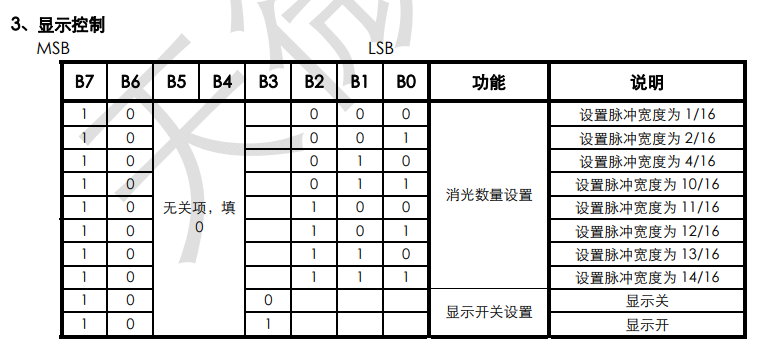


这里我们要写数据到显示寄存器，使用自动地址增加，普通模式，命令为：01000000即0x40

Command2地址命令如下：



Command3控制显示命令如下：



这个命令主要用来开关数码管和调整数码管亮度。

## 驱动程序

**TM1637.h**

#ifndef \_\_TM1637\_H     //防止重复定义（\_ADLX345\_H  开头)

#define \_\_TM1637\_H

#include "debug.h"

//=====================================================================

#define TM1637\_CKL\_PIN     GPIO\_Pin\_10

#define TM1637\_DIO\_PIN     GPIO\_Pin\_11

#define TM1637\_PORT        GPIOB

#define TM1637\_GPIO\_CLK    RCC\_APB2Periph\_GPIOB

//======================================================================

void TM1637\_Init(void);

//======================================================================

//函数名称：tm1637\_Display

//函数参数：a,b,c,d分别表示在第1,2,3,4位数码管上显示的数字

//函数返回：无

//功能概要：向TM1637写入传输的一个字节的数据

//======================================================================

void TM1637\_Display(uint8\_t a,uint8\_t a1,uint8\_t b,uint8\_t b1,uint8\_t c,uint8\_t c1,uint8\_t d,uint8\_t d1);

#endif

**TM1637.c**

#include "tm1637.h"    //包含TM1637头文件

//文件级全局变量

uint8\_t tab[22] =

{

        0x3F,    //0

        0x06,    //1

        0x5B,    //2

        0x4F,    //3

        0x66,    //4

        0x6D,    //5

        0x7D,    //6

        0x07,    //7

        0x7F,    //8

        0x6F,    //9

        0x77,    //10/A

        0x7C,    //11/B

        0x58,    //12/C

        0x5E,    //13/D

        0x79,    //14/E

        0x71,    //15/F

        0x76,    //16/H

        0x38,    //17/L

        0x54,    //18/U

        0x73,    //19/P

        0x3E,    //20/U

        0x00,    //21/空

};

uint16\_t CLK\_PIN;

uint16\_t DIO\_PIN;

//内部函数声明

void TM1637\_start(void);

uint8\_t TM1637\_ack(void);

void TM1637\_stop(void);

void TM1637\_Write(uint8\_t Data);

//====================接口函数声明=========================================

//======================================================================

//函数名称：TM1637\_Init

//函数参数：tm1637用到的两个引脚

//函数返回：无

//功能概要：向TM1637初始化

//======================================================================

void TM1637\_Init()

{

    GPIO\_InitTypeDef GPIO\_InitStructure = {0};

    RCC\_APB2PeriphClockCmd(TM1637\_GPIO\_CLK, ENABLE);

    GPIO\_InitStructure.GPIO\_Pin = TM1637\_CKL\_PIN|TM1637\_DIO\_PIN;

    GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_OD;

    GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

    GPIO\_Init(TM1637\_PORT, &GPIO\_InitStructure);

    GPIO\_WriteBit(TM1637\_PORT, TM1637\_CKL\_PIN, SET);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

}

//======================================================================

//函数名称：tm1637\_Display

//函数参数：a,b,c,d分别表示在第1,2,3,4位数码管上显示的数字

//       a1,b1,c1,d1分别表示是否显示数字之后的点。1表示显示，0表示不显示

//函数返回：无

//功能概要：向TM1637写入传输的一个字节的数据

//======================================================================

void TM1637\_Display(uint8\_t a,uint8\_t a1,uint8\_t b,uint8\_t b1,uint8\_t c,

        uint8\_t c1,uint8\_t d,uint8\_t d1)

{

    TM1637\_start();

    TM1637\_Write(0x40);    //写数据到显示寄存器+自动地址加1+普通模式

    TM1637\_ack();

    TM1637\_stop();

    TM1637\_start();

    TM1637\_Write(0xc0);    //设置显示首地址即第一个LED

    TM1637\_ack();

    TM1637\_Write(tab[a]|a1<<7);

    TM1637\_ack();

    TM1637\_Write(tab[b]|b1<<7);    //h为1时显示时钟中间的两点

    TM1637\_ack();

    TM1637\_Write(tab[c]|c1<<7);

    TM1637\_ack();

    TM1637\_Write(tab[d]|d1<<7);

    TM1637\_ack();

    TM1637\_stop();

    TM1637\_start();

    TM1637\_Write(0x89);    //显示控制：开显示，2/16亮度

    TM1637\_ack();

    TM1637\_stop();

}

//==========================内部函数声明===================================

//======================================================================

//函数名称：TM1637\_start

//函数参数：无

//函数返回：无

//功能概要：TM1637开始传输数据。CLK为高，DIO由高变低，数据输入开始

//======================================================================

void TM1637\_start(void)

{

     GPIO\_WriteBit(TM1637\_PORT, TM1637\_CKL\_PIN, SET);

     GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

     Delay\_Ms(2);

     GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);//CLK为高，DIO由高变低，数据输入开始

}

//======================================================================

//函数名称：TM1637\_ack

//函数参数：无

//函数返回：

//功能概要：TM1637应答数据

//======================================================================

uint8\_t TM1637\_ack(void)

{

    uint8\_t ackBit;

    //GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

    ackBit = GPIO\_ReadInputDataBit(TM1637\_PORT, TM1637\_DIO\_PIN);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

    Delay\_Ms(2);

    return ackBit;

}

//======================================================================

//函数名称：TM1637\_stop

//函数参数：无

//函数返回：无

//功能概要：TM1637传输一位数据完成。CLK为高时，DIO由低变高

//======================================================================

void TM1637\_stop(void)

{

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

    Delay\_Ms(2);

}

//======================================================================

//函数名称：TM1637\_Write

//函数参数：Data:要显示的一个字节的数据

//函数返回：无

//功能概要：向TM1637写入传输的一个字节的数据

//======================================================================

void TM1637\_Write(uint8\_t Data)

{

    uint8\_t i;

    for(i=0;i<8;i++)

    {

            //CLK为低时，DIO上才可以从传数据

        GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

        if(Data & 0x01)

            GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

        else

            GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);

        Delay\_Ms(2);

        Data=Data>>1;

        GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

        Delay\_Ms(2);

    }

}

**程序分析**：

引脚初始化，需要两个普通GPIO，配置成开漏输出（前提是总线上有5V上拉），不过因为是GPIO模拟的关系，也可以3.3V供电，配置成推挽输出，就不需要外部上拉电阻。接着上拉两个GPIO，保持总线空闲。

void TM1637\_Init()

{

    GPIO\_InitTypeDef GPIO\_InitStructure = {0};

    RCC\_APB2PeriphClockCmd(TM1637\_GPIO\_CLK, ENABLE);

    GPIO\_InitStructure.GPIO\_Pin = TM1637\_CKL\_PIN|TM1637\_DIO\_PIN;

    GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_OD;

    GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

    GPIO\_Init(TM1637\_PORT, &GPIO\_InitStructure);

    GPIO\_WriteBit(TM1637\_PORT, TM1637\_CKL\_PIN, SET);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

}

开始信号，CLK高电平时，DIO由高变低，开始传数据

void TM1637\_start(void)

{

     GPIO\_WriteBit(TM1637\_PORT, TM1637\_CKL\_PIN, SET);

     GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

     Delay\_Ms(2);

     GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);//CLK为高，DIO由高变低，数据输入开始

}

结束信号，CLK高电平时，DIO由低变高，结束传数据。

void TM1637\_stop(void)

{

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

    Delay\_Ms(2);

}

接收应答数据，在第八个时钟下降沿，实际也就是在主机发送完一个字节数据后，获取DIO电平，低电平表示应答，高电平表示非应答。

uint8\_t TM1637\_ack(void)

{

    uint8\_t ackBit;

    //GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

    ackBit = GPIO\_ReadInputDataBit(TM1637\_PORT, TM1637\_DIO\_PIN);

    Delay\_Ms(2);

    GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

    Delay\_Ms(2);

    return ackBit;

}

发送一个字节数据，CLK拉低时，改变DIO电平，然后拉高CLK电平，TM1637获取数据。需要注意的是TM1637使用的协议传输数据是低位在前。

void TM1637\_Write(uint8\_t Data)

{

    uint8\_t i;

    for(i=0;i<8;i++)

    {

            //CLK为低时，DIO上才可以从传数据

        GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,RESET);

        if(Data & 0x01)

            GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,SET);

        else

            GPIO\_WriteBit(TM1637\_PORT,TM1637\_DIO\_PIN,RESET);

        Delay\_Ms(2);

        Data=Data>>1;

        GPIO\_WriteBit(TM1637\_PORT,TM1637\_CKL\_PIN,SET);

        Delay\_Ms(2);

    }

}

修改寄存器，依照TM1637的通信协议，先发送数据命令0x40，设定写数据到寄存器+自动地址+1+普通模式，再发送显示首地址0xC0，然后依次发送显示数据调整数码管发光二极管，最后发送显示控制命令，调整亮度。

void TM1637\_Display(uint8\_t a,uint8\_t a1,uint8\_t b,uint8\_t b1,uint8\_t c,

        uint8\_t c1,uint8\_t d,uint8\_t d1)

{

    TM1637\_start();

    TM1637\_Write(0x40);    //写数据到显示寄存器+自动地址加1+普通模式

    TM1637\_ack();

    TM1637\_stop();

    TM1637\_start();

    TM1637\_Write(0xc0);    //设置显示首地址即第一个LED

    TM1637\_ack();

    TM1637\_Write(tab[a]|a1<<7);

    TM1637\_ack();

    TM1637\_Write(tab[b]|b1<<7);    //h为1时显示时钟中间的两点

    TM1637\_ack();

    TM1637\_Write(tab[c]|c1<<7);

    TM1637\_ack();

    TM1637\_Write(tab[d]|d1<<7);

    TM1637\_ack();

    TM1637\_stop();

    TM1637\_start();

    TM1637\_Write(0x89);    //显示控制：开显示，2/16亮度

    TM1637\_ack();

    TM1637\_stop();

}