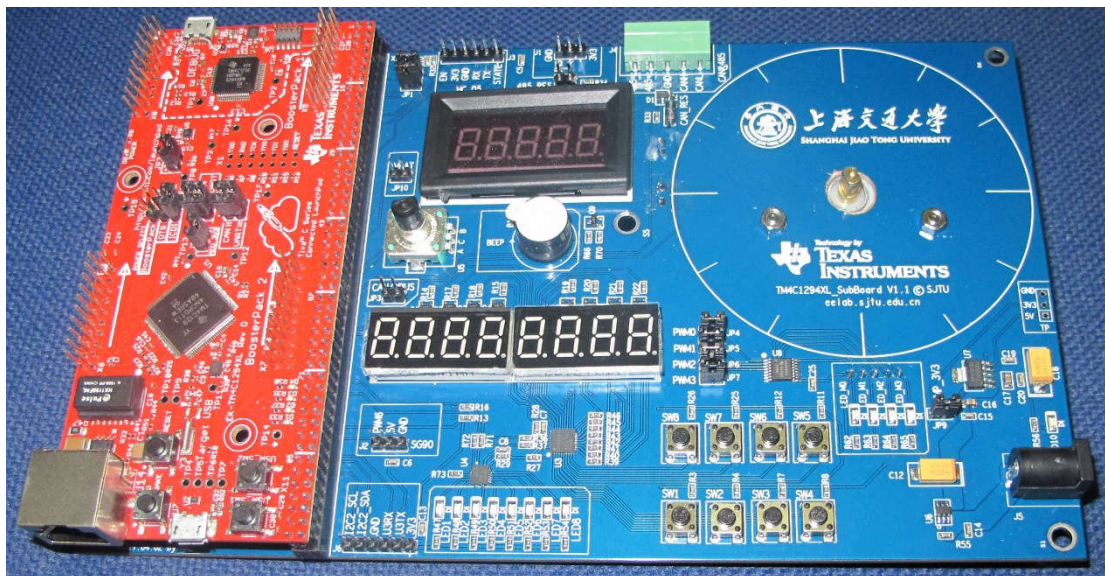


一、S800 板介绍



S800 板 = TM4C1294XL(红) + TM4C1294XL_SUB(蓝)

红色板为 TI 提供的基于 TM4C1294NCPDT (ARM CORTEX-M4F 内核, 128PIN) CPU 的评估板;蓝色板为扩展功能板,提供实验的各种对象及执行部件.

扩展板提供的功能如下:

- 整板静电 ESD 保护及过电流负载保护

- 整板电流显示

- 多路可选电源输入,包括

- DC5V

- MICRO-USB5V 输入

- MICRO-USB OTG 5V 输入

- I2C 扩展 GPIO

- 8 位共阴数码管

- 8 位输入按键

- 8 位共阴 LED

- USART- RS485 总线串行接口

- CAN 总线串行接口

- 5V 直流有刷电机或步进电机接口

- PWM 输出

- DAC 输出

- 外部模拟量输入

- 可调电位器模拟量输入

- 蜂鸣器

- SD 卡接口

- QEI 数字电位器接口

- 蓝牙模块接口

- 舵机控制接口

· RTC 备用电池

二、板上芯片简介

红板芯片

序号	芯片型号	说明
1	TM4C1294NCPDT	主 CPU, ARM-CORTEX-M4F
2	TM4C123GH6PMI	ICDI 板载仿真器 CPU, ARM-CORTEX-M4F
3	TPD4S012	4 通道 ESD 保护用于 USB 接口保护
4	TPS2052B	过流保护及负载分配开关
5	TPS73733	3.3VLDO 稳压器

蓝板芯片

序号	芯片型号	说明
1	PCA9557	I2C 扩展 8 位 GPIO
2	TCA6424	I2C 扩展 24 位 GPIO
3	TPD3S044	USB 接口限流保护及 ESD 保护
4	TPD4E001	4 通道 ESD 保护
5	SN65HVD08	RS-485 收发器
6	SN65HVD1050	CAN 总线收发器
7	ULN2003A	7 位达林顿管阵列
8	S8050D	40V0.5A0.3W NPN 管
9	EC12	数字电位器
10	CL3641AH	4 位共阴数码管

三、按键及 LED 资源列表

红板

名称	对应管脚	说明
RESET	RESET	TM4C1294NCPDT 芯片复位按键低有效
WAKE	WAKE	从睡眠模式唤醒按键低有效
USR_SW1	PJ0	用户输入按键低有效
USR_SW2	PJ1	用户输入按键低有效
D0		3.3V 电源指示, 绿 LED 高有效
D1	PN1	用户控制绿 LED 高有效
D2	PN0	用户控制绿 LED 高有效
D3	PF4	用户控制绿 LED 高有效
D4	PF0	用户控制绿 LED 高有效

蓝板

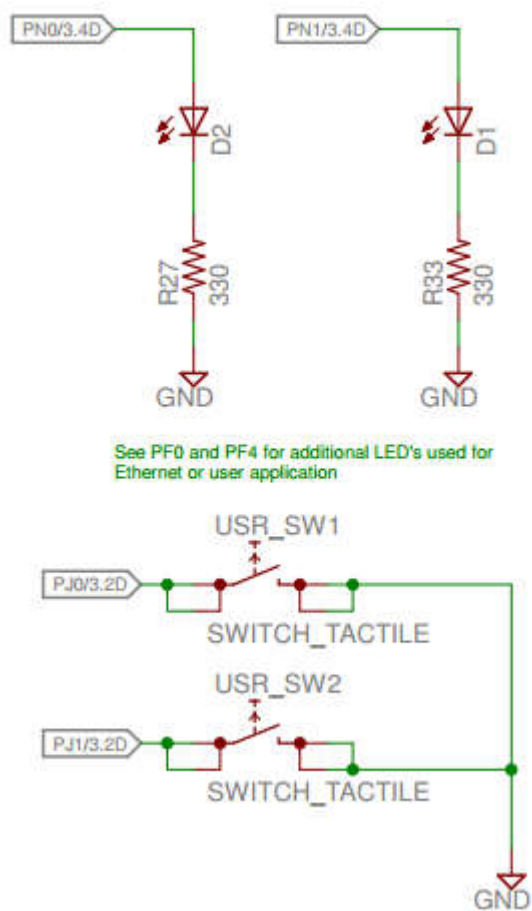
名称	对应管脚	说明
SW1-SW8	TCA6424-P01~P08	TCA6424 I2C 展 GPIO 芯片 P0 口低有效
LED1-LED8	PCA9557-P0-P7	PCA9557 I2C 展 GPIO 芯片 P0 口低有效

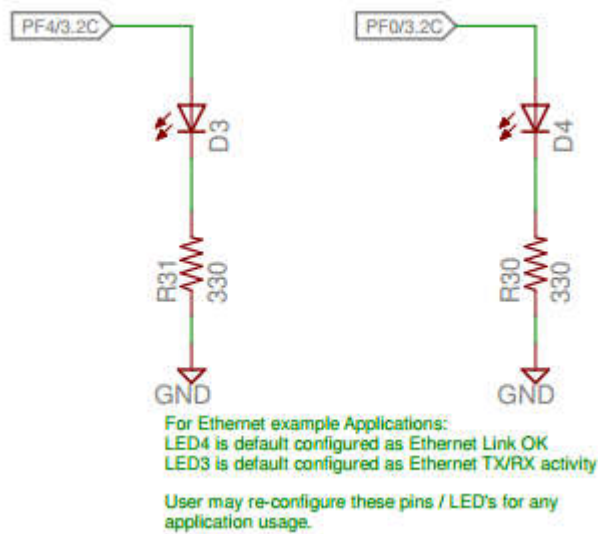
LED_M0	PF0	用户控制 LED 高有效
LED_M1	PF1	用户控制 LED 高有效
LED_M2	PF2	用户控制 LED 高有效
LED_M3	PF3	用户控制 LED 高有效
D10		3.3V 电源指示，红 LED 高有效

- 红板与蓝板均有 PF0 对应控制 LED，两个 LED 一绿（高有效）一红（低有效），显示效果互补。

四、板上器件编程与使用

4.1 红板上资源编程





红板上共有两个外部晶体振荡器供选择，一个为外部 25M 高精度无源晶振 HSE，为正常使用时提供外部时钟信号；一个为 32.768K 无源晶振 LSE，主要用于低功耗或电池供电时提供外部时钟信号。

同时在 CPU 内部也提供一个低精度的 16M 振荡器 HSI（正负 50%误差），可以用于不需要精确定时的场合。不能用于同步通讯如 USART，CAN，ETHERNET 等。

如果不使用 PLL 倍频电路，则只能使用 25M 的 HSE 或 16M 的 HSI 作为时钟。

如果使用 PLL 倍频电路，则可以升频到 480M 后再分频作为 CPU 时钟使用，最大可接受频率为 120M 时钟。

频率越高，则速度越快，功耗越大。

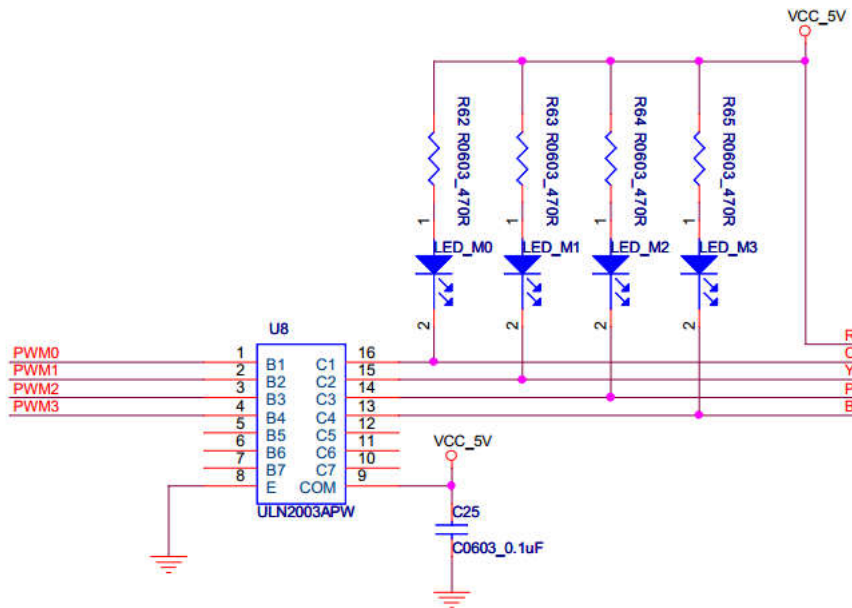
红板上共有四个用户可控 LED，如上图所示。分别为 PF0,PF4,PN0,PN1 管脚控制。只有当管脚输出高电平时，LED 才会亮。

红板上共有两个用户可控按键输入，如上图所示。分别为 USR_SW1, USR_SW2 接到 PJ0 与 PJ1，当按键按下时，与地相通。这两个引脚并没有外接上拉电阻，因此作为输入使用时，必须配置为内部弱上拉，才能清楚地分开未按下与按下状态。

4.2 蓝板上资源编程

4.2.1 GPIO 资源

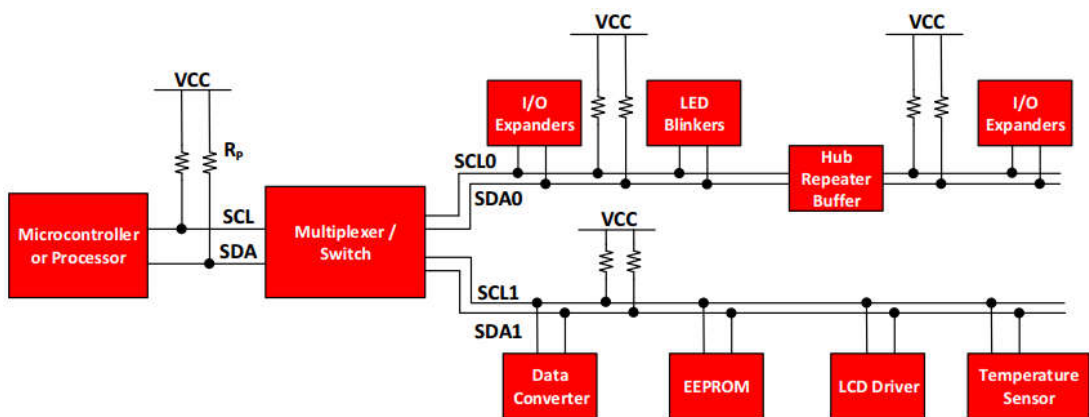
PWM0-3 对应 PF0-3，经过 ULN2003 达林顿管后，当 CPU 管脚输出为高，则输出 C1-C4 输出为低，点亮 LED；当 CPU 管脚输出为低，则输出 C1-C4 输出为高，熄灭 LED。



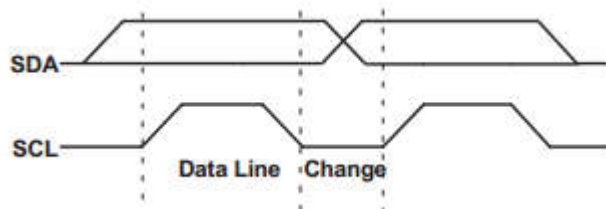
4.2.2 I2C 通讯简述

I2C 总线是一种简单的双向一线制串行通信总线。多个符合 I2C 总线标准的器件都可以通过同一条 I2C 总线进行通信，而不需要额外的地址译码器。

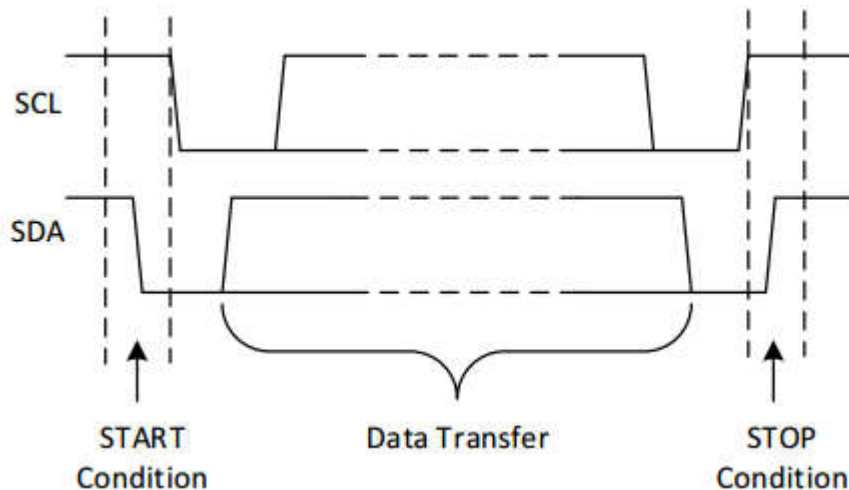
I2C 总线仅使用两个信号：SDA 和 SCL。SDA 是双向串行数据线，SCL 是双向串行时钟线。当 SDA 和 SCL 线为高电平时，总线为空闲状态。I2C 总线连接方式如下图所示。



在时钟 SCL 的高电平期间，SDA 线上的数据必须保持稳定。SDA 仅可在时钟 SCL 为低电平时改变，如下图所示。



I2C 总线的协议定义了两种状态：起始和停止。当 SCL 为高电平时，在 SDA 线上从高到低的跳变被定义为起始条件；而当 SCL 为高电平时，在 SDA 线上从低到高的跳变则被定义为停止条件。总线在起始条件之后被看作为忙状态。总线在停止条件之后被看作为空闲。



SDA 线上的每个字节必须为 8 位长。不限制每次传输的字节数。每个字节后面必须带有一个应答位。数据传输是 MSB 在前。当接收器不能接受另一个完整的字节时，它可以将时钟线 SCL 拉到低电平，以迫使发送器进入等待状态。当接收器释放时钟 SCL 时继续进行数据传输。

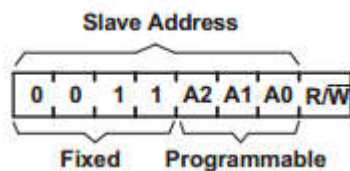
从机地址在起始条件之后发送。该地址为 7 位，后面跟的是第 8 位是数据方向位，这个数据方向位决定了下一个操作是接受（高电平）还是发送（低电平），0 表示传输（发送）；1 表示请求数据（接收）。数据传输始终由主机产生的停止条件来终止。然而，通过产生重复的起始条件和寻址另一个从机（而无需先产生停止条件），主机仍然可以再总线上通信。因此，在这种传输过程中可能会有接收/发送格式的不同组合。

首字节的前面 7 位组成了从机地址。第 8 位决定了消息的方向。首字节的 R/S 位为 0 表示主机将向所选择的从机发送信息。该位为 1 表示主机将接收来自从机的信息。

带有 I2C 总线的器件除了有从机地址 (Slave Address) 外，还有数据地址 (也称子地址)。从机地址是指该器件在 I2C 总线上被主机寻址的地址，而数据地址是指该器件内部不同部件和存储单元的编址。

4.2.3 I2C 芯片 PCA9557

地址定义



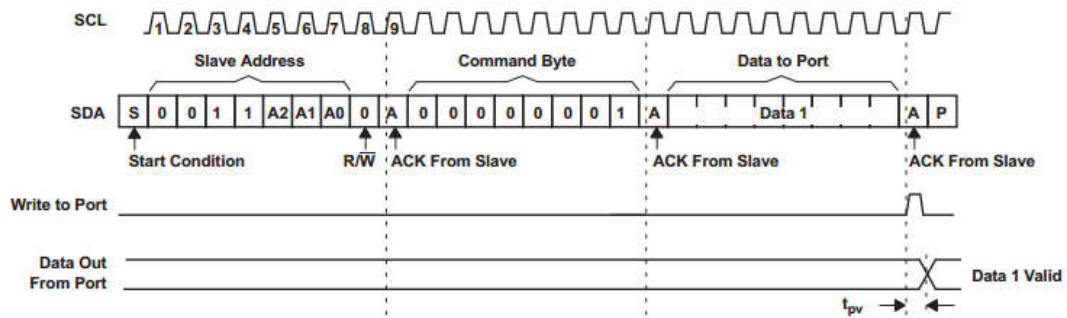


Figure 24. Write To Output Port Register

因为 PCA9557 为 I2C 转 8 位 GPIO 扩展芯片，在板上为 LED 驱动，低有效。即当管脚为低电平时，对应的 LED 亮。

PCA9557 的使用分为两步：

1. PCA9557 初始化，即将端口配置为输出。
2. 对 PCA9557 的输出端口赋值，例赋 0x0 时，则 8 个 LED 全亮，
当为 0x0ff 时，8 个 LED 全灭。

固件库文件驱动方式从上图的对输出端口写数据可见，首先给 I2C 总线输出从设备即 PCA9557 地址，然后给出两字节数据，第一个字节为命令，第二个字节为数据。命令即输出端口 01h，数据则为从 0x00-0x0ff。

因此固件库使用如下：其中 DevAddr 为设备地址即 0x18，RegAddr 为输出端口地址即 0x01，WriteData 为写给输出端口的数据。

```
uint8_t I2C0_WriteByte(uint8_t DevAddr, uint8_t RegAddr, uint8_t WriteData)
{
    uint8_t rop;
    while(I2CMasterBusy(I2C0_BASE)){};
    I2CMasterSlaveAddrSet(I2C0_BASE, DevAddr, false);
    I2CMasterDataPut(I2C0_BASE, RegAddr);
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_BURST_SEND_START);
    while(I2CMasterBusy(I2C0_BASE)){};
    rop = (uint8_t)I2CMasterErr(I2C0_BASE);

    I2CMasterDataPut(I2C0_BASE, WriteData);
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_BURST_SEND_FINISH);
    while(I2CMasterBusy(I2C0_BASE)){};
    rop = (uint8_t)I2CMasterErr(I2C0_BASE);
    return rop;
}
```

I2C 芯片 TCA6424

地址定义

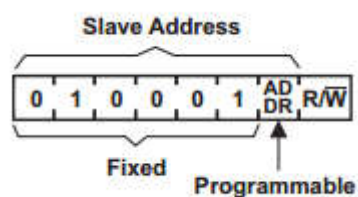
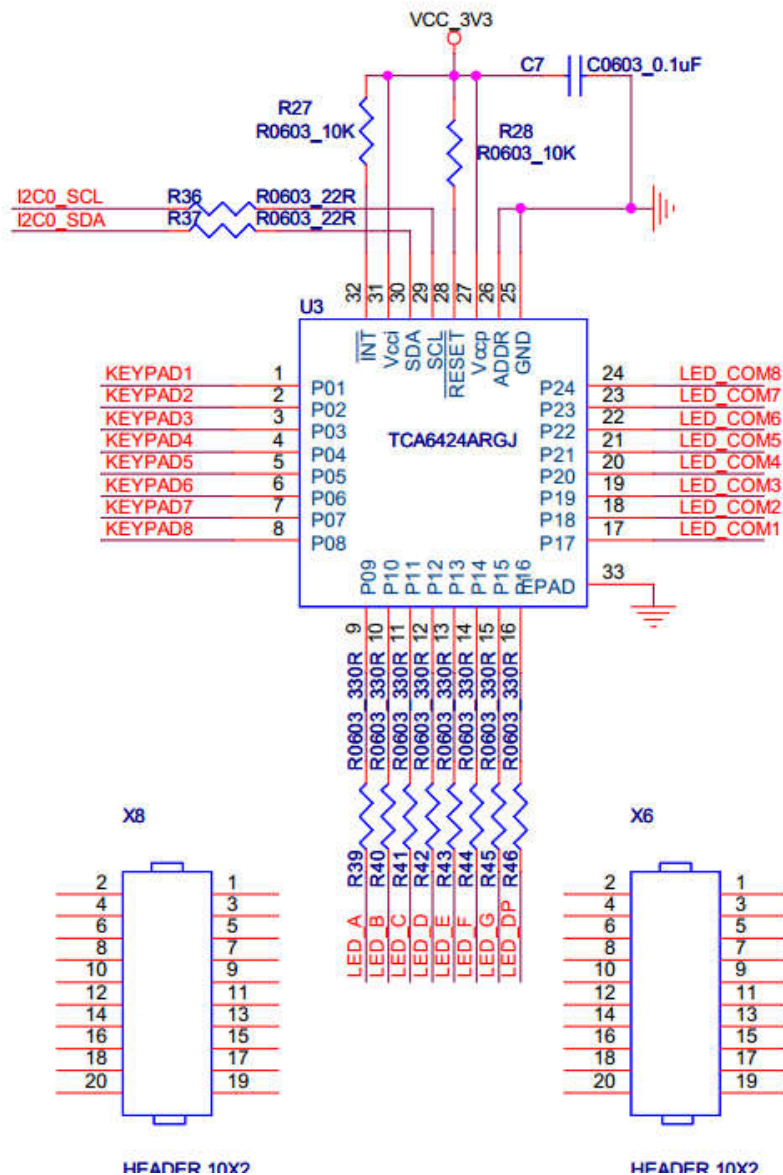


Table 3. Address Reference

ADDR	I ² C BUS SLAVE ADDRESS
L	34 (decimal), 22 (hexadecimal)
H	35 (decimal), 23 (hexadecimal)

GPIO EXPAND



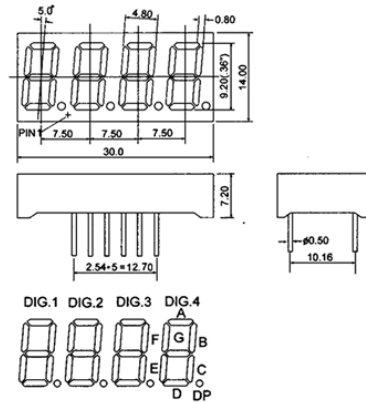
从上图可以看出，蓝板使用的 TCA6424 芯片的地址为 0x22

因为 TCA6424 为 I2C 转 24 位 GPIO 扩展芯片，分为 3 组，每组 8 位。在板上分别为：
P0 为按键 SW1-SW8；

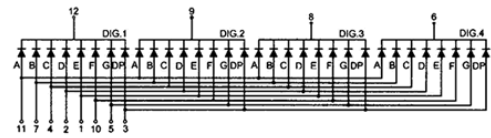
P1 为动态共阴数码管的脚位信号，高电平时点亮相应的笔划；

P2 为动态共阴数码管的片选信号，当为高电平时，驱动对应的 8050 三极管导通，从而选通对应的位。

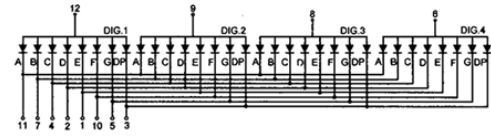
LG3641



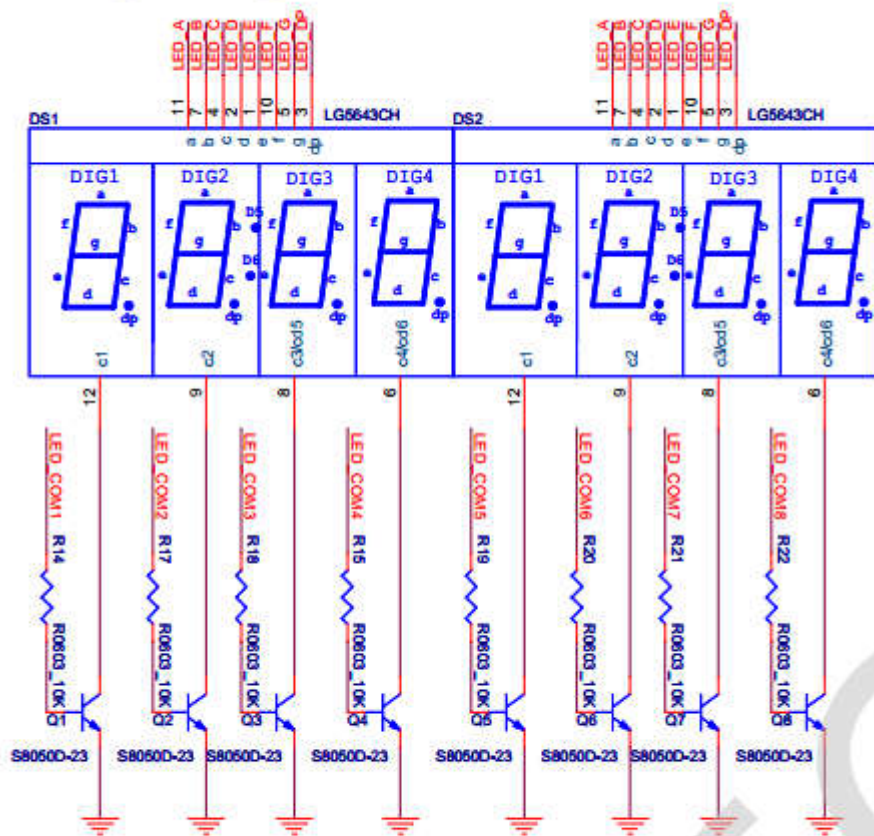
PCB:D3641A/B



PCB:D3641A/B



DIGITRON



TCA6424 的使用分为两步：

3. TCA6424 初始化，即将端口 P0 配置为输入，P1, P2 配置为输出。
4. 对 PCA9557 的输出端口赋值。从而点亮动态数码管。