

LED 灯带应用笔记

一、驱动原理

如图 1 所示，给 LED 灯带发送时序，点亮一个 LED 需要 3 个 Byte (24BIT)，其 3 个 Byte 分别表示一个灯的 R、G、B 三原色，数据传输的顺序为高位先发，按照 GRB 的顺序发送数据 (G7→G6……B0)。数据发送给灯带不会立刻显示，只有当检测复位信号信号时，LED 才会有反应。数据 0、1 和复位信号对应的的时序如图 2 所示。

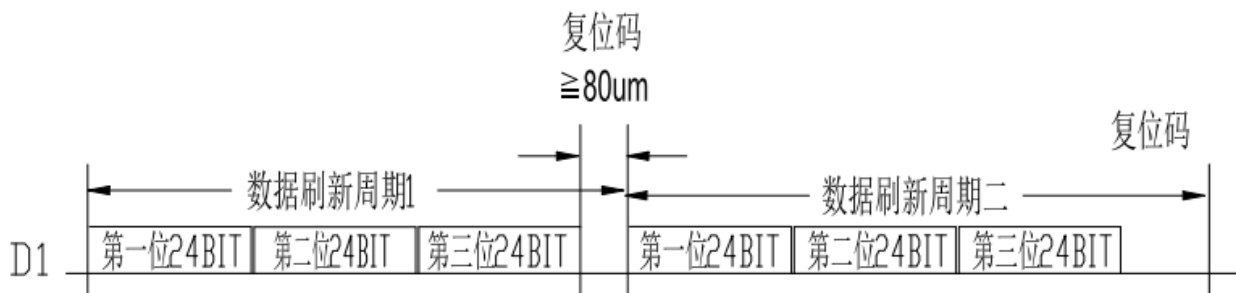


图 1 数据传输方式

输入码型:



图 2 时序波形图

二、技术难点

1、输出 0、1 的时序波形

解决办法: 设置 CPU 时钟为 16M, 用 NOP 指令延时。

指令周期=1/系统时钟=1/16M=62.5ns

300ns≈5 条 NOP 指令

900ns≈14 条 NOP 指令

2、连续输出一个 Byte 波形

解决办法：每个 Byte 有 8 个 bit，使用左移指令，对同一个位判断 8 次。将移位与判断指令放在时序波形中，代替部分 NOP 指令，根据移位与判断指令执行的时间调节 NOP 指令的数量。

实现代码：

```

PUT:
    MOVAI    0x08
    MOVRA    BIT_NUM
LOOP_8:
    BSET     IOP0, 3           ;P03D = 1
    JBCLR    INDF0, 7
    GOTO     L1
    NOP
    NOP                       ;0码
    BCLR     IOP0, 3
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    RLR     INDF0
    DJZR     BIT_NUM           ;为0 , 发了8个
    GOTO     LOOP_8
    RETURN

L1:
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    BCLR     IOP0, 3
    RLR     INDF0
    DJZR     BIT_NUM           ;为0 , 发了8个
    GOTO     LOOP_8
    RETURN
    
```

3、输出 R、G、B 三个 Byte 的波形（即点亮一个 LED 灯）

解决办法:

①采用 FSR0-INDF0 寻址的方式读取 R、G、B 的值。实现代码如下，0x08，0x09，0x0a 为存储 RGB 值的 RAM 地址，FSR0 为地址指针，INDF0 为 FSR0 所指地址中存储的值。改变 FSR0 指向的地址，FSR0 中的值就会发生改变，只需将 FSR0 分别指向 G、R、B 的地址，判断 INDF0 的 8 个 bit 位，就可以输出 R、G、B 三个 Byte 的波形。

SCAN:

```
MOVAI    0x08
MOVRA    FSR0
CALL     PUT
MOVAI    0x09
MOVRA    FSR0
CALL     PUT
MOVAI    0x0A
MOVRA    FSR0
CALL     PUT
RETURN
```

②采用传参的方式读取 R、G、B 的值。定义一个中间变量 COLOR，分别将 G、R、B 的值赋值给 COLOR，判断 COLOR 的 8 个 bit 位，可以输出 R、G、B 三个 Byte 的波形。实现代码如下。

SCAN:

```
MOVAR    GREEN
MOVRA    COLOR
CALL     PUT
MOVAR    RED
MOVRA    COLOR
CALL     PUT
MOVAR    BLUE
MOVRA    COLOR
CALL     PUT
RETURN
```

两种方式占用的 ROM 空间一样，方式②比方式①多占用一个 Byte RAM 空间，输出结果一样。