1171 TA的帖子 | 2 TA的资源

版主

发消息 加好友

发帖 回复 返回列表

查看: 2009 | 回复: 15

[原创] WS2812灯珠的STM32驱动方式(二)——DMA+PWM [复制链接]

发表于 2018-5-6 22:48:55 | 只看该作者 | 只看大图 | 倒序浏览 | 阅读模式 楼主 电梯直达

本帖最后由 通宵敲代码 于 2018-5-6 23:34 编辑

上节我们重点介绍了一下WS2812B，
这种内部集成驱动芯片的RGB灯珠，
并通过对驱动信号通信速率的详细计算，
重点说明了STM32等一众单片机通过I/O翻转方式，
控制这种灯珠的困难之处，这篇我们就介绍一下，
如何用最常见的方式来实现STM32对WS2812的控制。

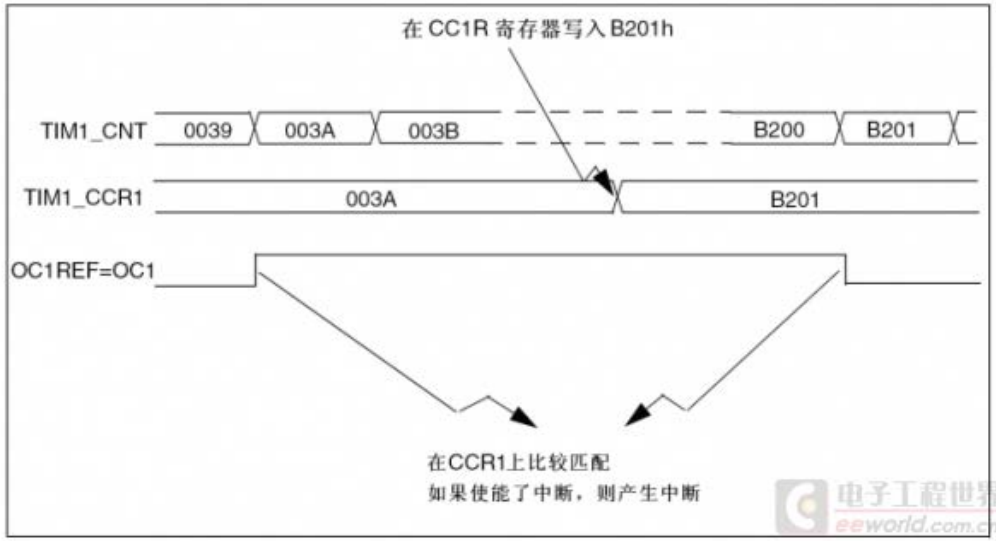
常用STM32的都知道，STM32有8个定时器，
其中TIM1跟TIM8是高级定时器，TIM2345是通用定时器，
TIM67是基本定时器，当然还有个滴答定时器，
其中TIM123458都可以独立控制输出PWM波形，
TIM1跟TIM8还可以输出两组互补PWM波形，
这个功能在电机驱动中十分常用。

表4　定时器功能比较

定时器	计数器分辨率	计数器类型	预分频系数	产生DMA请求	捕获/比较通道	互补输出
TIM1 TIM8	16位	向上，向下，向上/下	1~65536之间的任意整数	可以	4	有
TIM2 TIM3 TIM4 TIM5	16位	向上，向下，向上/下	1~65536之间的任意整数	可以	4	没有
TIM6 TIM7	16位	向上	1~65536之间的任意整数	可以	0	没有

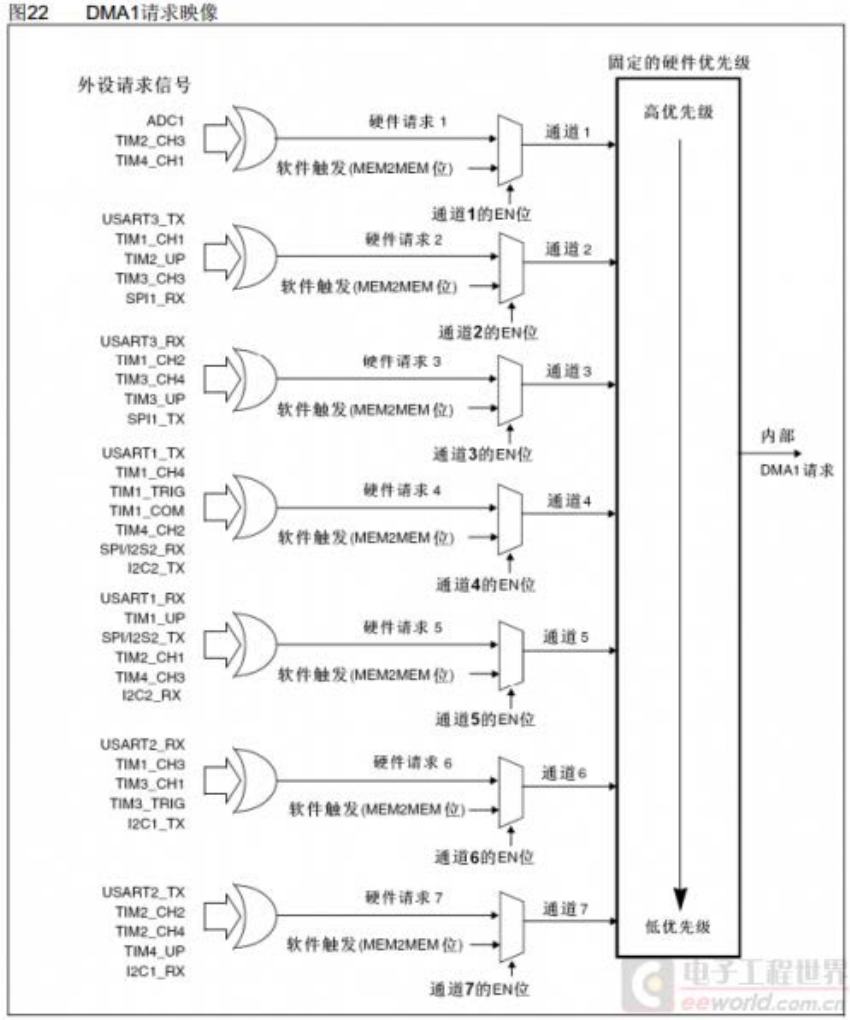
STM32定时器的PWM输出功能，
是通过CNT与CCR寄存器的数值比，
直接控制对应引脚输出高低电平的，
也就是说只要我们愿意，可以用定时器直接输出36MHz的方波，
当然波形可能就不那么好看了，
不过用来驱动我们800kbps的WS2812，可以说是绰绰有余了，

图127　输出比较模式，翻转OC1



但是还有一个问题，通常情况下，
我们的WS2812不会仅仅是一个或者几个LED串联，
如果串联的LED较多的话，又会面临驱动信号的稳定性问题，
毕竟我们要不断地改变定时器CCR的数值，
来控制对应的I/O发送信号0还是信号1，
而项目中我们的单片机又不可能只是用来控制LED，
这时候我们的DMA功能就派上用场了。
这也就是我们今天要介绍的最容易想到的一种方法DMA+PWM，

图22　DMA1请求映像



此帖出自DIY/开源硬件专区论坛

#在这里快速回复#
true

最新活动
- 【泰有聊】第二期：泰克CEO为你解读低...
- TE Connectivity 小有乾坤蕴藏无限可能

推荐资源
- 汽车电子嵌入式系统简介
- 运算放大器应用技术手册

相关帖子
- WS2812灯珠的STM32驱动方式(一)——了...
- 零死角玩转STM32

ADI 系统方案精选【用于多路抽头输出隔...
CAN总线技术在设备故障诊断系统中的...
Jeffrey携手飞猪超级新发线，疯玩日本
我是大侦探之：寻找遗落的TI电源拼图
开关电源变换：设计和分析
求购AM335X工控核心板
EEWORLD 示波器问卷有奖大调查
电动伺服电动机方程和时间常数
串口通信最远传输距离能达到多少

收藏3 干货 评分 分享

AD

回复 使用道具 举报

通宵敲代码

1171 TA的帖子 | 2 TA的资源

版主

发消息 加好友

楼主 | 发表于 2018-5-6 22:49:12 | 只看该作者 沙发

本帖最后由 通宵敲代码 于 2018-5-6 23:28 编辑

上面介绍完原理，下面就是如何实现功能了，
我们直接来看程序，声明一下我们是用原子的例程移植的。
这里我们依然使用是STM32的HAL库，
没办法，用熟了，顺手，而且各种以前的资源不能浪费了不是。

首先是时钟配置，这个没什么好说的，能跑多快就多快吧。
就跟没人会希望自己媳妇丑一样。

```
//091209
void Stm32_Clock_Init(u8 PLL)
{
  unsigned char temp=0;
  MYRCC_DeInit();      //复位并配置向量表
  RCC->CR|=0x00010000;  //外部高速时钟使能HSEON
  while(!(RCC->CR>>17));//等待外部时钟就绪
  RCC->CFGR=0x00000400; //APB1=DIV2;APB2=DIV1;AHB=DIV1;
  PLL-=2;//抵消2个单位
  RCC->CFGR|=PLL<<18;   //设置PLL值9倍频 2~16
  RCC->CFGR|=1<<16;    //PLLSRC ON
  FLASH->ACR|=0x32;    //FLASH 2个延时周期

  RCC->CR|=0x01000000;  //PLLON
  while(!(RCC->CR>>25));//等待PLL锁定
  RCC->CFGR|=0x00000002;//PLL作为系统时钟
  while(temp!=0x02)     //等待PLL作为系统时钟设置成功
  {
    temp=RCC->CFGR>>2;
    temp&=0x03;
  }
}
```

(哦，不好意思这段是之前用寄存器写的，拿过来用了，见谅)

下面是今天的主角，DMA+PWM的配置方式。

```
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
  /* GPIOA Configuration: TIM2 Channel 1 as alternate function push-pull */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOA, &GPIO_InitStructure);
```

先配置下GPIO吧，我们使用的是TIM2的CH1通道，
对用的PWM输出引脚是PA0，注意要用复用输出功能哦。

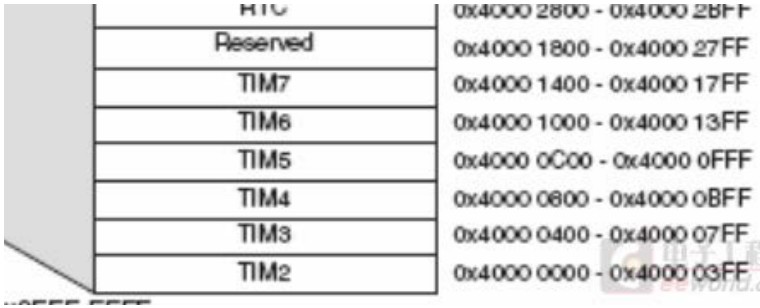
下面是TIM2+CH1通道的配置，注意要配置成800KHz的频率，
内核时钟72M / 90 = 800KHz，应该不难理解吧。
然后是CH1输出通道的，配置成PWM1输出模式即可，
注意电平的极性，不放心的可以用示波器抓抓波形。

```
36    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
37    /* Compute the prescaler value */
38    //PrescalerValue = (uint16_t) (SystemCoreClock / 24000000) - 1;
39    /* Time base configuration */
40    TIM_TimeBaseStructure.TIM_Period = 90-1; // 800kHz
41    TIM_TimeBaseStructure.TIM_Prescaler = 0;
42    TIM_TimeBaseStructure.TIM_ClockDivision = 0;
43    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
44    TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);
45
46    /* PWM1 Mode configuration: Channel1 */
47    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
48    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
49    TIM_OCInitStructure.TIM_Pulse = 0;
50    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
51    TIM_OC1Init(TIM2, &TIM_OCInitStructure);
```

然后就是我们的DMA得配置了，要设置好数组地址跟寄存器地址
然后就是DMA传输的字节长度，传输模式等等了

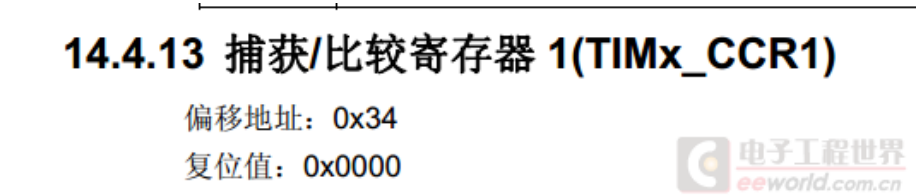
```
53  /* configure DMA */
54  /* DMA clock enable */
55  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);
56
57  /* DMA1 Channel6 Config */
58  DMA_DeInit(DMA1_Channel2);
59
60  DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)TIM2_CCR1_Address; // physical address of Timer 3 CCR1
61  DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)LED_BYTE_Buffer;   // this is the buffer memory
62  DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;          // data shifted from memory to peripheral
63  DMA_InitStructure.DMA_BufferSize = 42;
64  DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
65  DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;        // automatically increase buffer index
66  DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
67  DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
68  DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;             // stop DMA feed after buffer size is reached
69  DMA_InitStructure.DMA_Priority = DMA_Priority_High;
70  DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
71
72  DMA_Init(DMA1_Channel2, &DMA_InitStructure);
73
74    /* TIM3 CC1 DMA Request enable */
75  TIM_DMACmd(TIM2, TIM_DMA_Update, ENABLE);
76  }
```

有关寄存器地址大家应该会找吧，不会的看下边，
先从Datasheet找到我们的存储器映像图表，找到TIM寄存器租的起始地址，



可以看到TIM2寄存器组的起始地址是0x40000000，

然后找到Reference Manual里，我们需要使用的TIM2_CCR1寄存器，



可以看到CCR1的偏移地址是0x34，所以我们需要的寄存器地址就是，
0x40000000 + 0x34 = 0x40000034，不懂得童鞋回去好好啃啃计算机原理吧，

需要的外设资源配置好了，接下来就是万事俱备只欠东风了，
下面就是我们有关LLED控制信号的处理函数了，
look，不多解释。

```
void WS2812_send(uint8_t (*color)[3], uint16_t len)
{
  uint8_t i;
  uint16_t memaddr;
  uint16_t buffersize;
  buffersize = (len*24)+43; // number of bytes needed is #LEDs * 24 bytes + 42 trailing bytes
  memaddr = 0;          // reset buffer memory index

  while (len)
  {
    for(i=0; i<8; i++) // GREEN data
    {
      LED_BYTE_Buffer[memaddr] = ((color[0][1]<<i) & 0x0080) ? TIMING_ONE:TIMING_ZERO;
      memaddr++;
    }
    for(i=0; i<8; i++) // RED
    {
        LED_BYTE_Buffer[memaddr] = ((color[0][0]<<i) & 0x0080) ? TIMING_ONE:TIMING_ZERO;
        memaddr++;
    }
    for(i=0; i<8; i++) // BLUE
    {
        LED_BYTE_Buffer[memaddr] = ((color[0][2]<<i) & 0x0080) ? TIMING_ONE:TIMING_ZERO;
        memaddr++;
    }
    len--;
  }
//===========================================================//
//bug：最后一个周期波形不知道为什么全是高电平，故增加一个波形
    LED_BYTE_Buffer[memaddr] = ((color[0][2]<<8) & 0x0080) ? TIMING_ONE:TIMING_ZERO;
//===========================================================//
```

上图最后面那个bug，是之前原子的程序就带着的，
家里没有示波器目前没做验证，有空会控死抓抓波形看看。
程序能用，也就没深究下去。

注意一下，到这可没玩呢，说好了要用DMA方式传输数据的，
下面才是数据传输的主要内容呢。

```
    DMA_SetCurrDataCounter(DMA1_Channel2, buffersize);  // load number of bytes to be transferred
    DMA_Cmd(DMA1_Channel2, ENABLE);       // enable DMA channel 6
    TIM_Cmd(TIM2, ENABLE);              // enable Timer 3
    while(!DMA_GetFlagStatus(DMA1_FLAG_TC2)) ;  // wait until transfer complete
    TIM_Cmd(TIM2, DISABLE);   // disable Timer 3
    DMA_Cmd(DMA1_Channel2, DISABLE);      // disable DMA channel 6
    DMA_ClearFlag(DMA1_FLAG_TC2);         // clear DMA1 Channel 6 transfer complete flag
}
```

TIM截图20180506230138.png (52.84 KB, 下载次数: 0)

```
/* configure DMA */
/* DMA clock enable */
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);

/* DMA1 Channel6 Config */
DMA_DeInit(DMA1_Channel2);

DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)TIM2_CCR1_Address; // physical address of Timer 3 CCR1
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)LED_BYTE_Buffer;  // this is the buffer memory
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;              // data shifted from memory to peripheral
DMA_InitStructure.DMA_BufferSize = 42;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;         // automatically increase buffer index
DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;               // stop DMA feed after buffer size is reached
DMA_InitStructure.DMA_Priority = DMA_Priority_High;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;

DMA_Init(DMA1_Channel2, &DMA_InitStructure);

  /* TIM3 CC1 DMA Request enable */
TIM_DMACmd(TIM2, TIM_DMA_Update, ENABLE);
}
```

点评

naga568 謝謝分享。 详情 回复 发表于 2018-5-7 07:51

AD 开源代码共享 开源设计共享 开源硬件

回复 评分 举报

楼主 | 发表于 2018-5-6 22:49:25 | 只看该作者 板凳

本帖最后由 通宵敲代码 于 2018-5-6 23:34 编辑

定义了三个数组，主函数里写个三色循环呼吸灯看看效果。

```
uint8_t rgb0[][3] = {0,0,0};
uint8_t rgb1[53][3] = {{0,0,0},{10,0,0},{20,0,0},{30,0,0},{40,0,0},{50,0,0},{60,0,0},{70,0,0},{80,0,0},{90,0,0},
                       {100,0,0},{110,0,0},{120,0,0},{130,0,0},{140,0,0},{150,0,0},{160,0,0},{170,0,0},{180,0,0},{190,0,0},
                       {200,0,0},{210,0,0},{220,0,0},{230,0,0},{240,0,0},{250,0,0},{255,0,0},{250,0,0},{240,0,0},{230,0,0},
                       {220,0,0},{210,0,0},{200,0,0},{190,0,0},{180,0,0},{170,0,0},{160,0,0},{150,0,0},{140,0,0},{130,0,0},
                       {120,0,0},{110,0,0},{100,0,0},{90,0,0},{80,0,0},{70,0,0},{60,0,0},{50,0,0},{40,0,0},{30,0,0},
                       {20,0,0},{10,0,0},{0,0,0}};
uint8_t rgb2[53][3] = {{0,0,0},{0,10,0},{0,20,0},{0,30,0},{0,40,0},{0,50,0},{0,60,0},{0,70,0},{0,80,0},{0,90,0},
                       {0,100,0},{0,110,0},{0,120,0},{0,130,0},{0,140,0},{0,150,0},{0,160,0},{0,170,0},{0,180,0},{0,190,0},
                       {0,200,0},{0,210,0},{0,220,0},{0,230,0},{0,240,0},{0,250,0},{0,255,0},{0,250,0},{0,240,0},{0,230,0},
                       {0,220,0},{0,210,0},{0,200,0},{0,190,0},{0,180,0},{0,170,0},{0,160,0},{0,150,0},{0,140,0},{0,130,0},
                       {0,120,0},{0,110,0},{0,100,0},{0,90,0},{0,80,0},{0,70,0},{0,60,0},{0,50,0},{0,40,0},{0,30,0},
                       {0,20,0},{0,10,0},{0,0,0}};
uint8_t rgb3[53][3] = {{0,0,0},{0,0,10},{0,0,20},{0,0,30},{0,0,40},{0,0,50},{0,0,60},{0,0,70},{0,0,80},{0,0,90},
                       {0,0,100},{0,0,110},{0,0,120},{0,0,130},{0,0,140},{0,0,150},{0,0,160},{0,0,170},{0,0,180},{0,0,190},
                       {0,0,200},{0,0,210},{0,0,220},{0,0,230},{0,0,240},{0,0,250},{0,0,255},{0,0,250},{0,0,240},{0,0,230},
                       {0,0,220},{0,0,210},{0,0,200},{0,0,190},{0,0,180},{0,0,170},{0,0,160},{0,0,150},{0,0,140},{0,0,130},
                       {0,0,120},{0,0,110},{0,0,100},{0,0,90},{0,0,80},{0,0,70},{0,0,60},{0,0,50},{0,0,40},{0,0,30},
                       {0,0,20},{0,0,10},{0,0,0}};
```

```
  while(1)
  {
    WS2812_send(rgb0,8);
    for(i=0;i<53;i++)
    {
      WS2812_send(&rgb1[i],8);
      delay_ms(50);
    }
    for(i=0;i<53;i++)
    {
      WS2812_send(&rgb2[i],8);
      delay_ms(50);
    }
    for(i=0;i<53;i++)
    {
      WS2812_send(&rgb3[i],8);
      delay_ms(50);
    }
  }
}
```

电子工程世界 eeworld.com.cn

附上相关程序

这个是用TIM2_CH1实现三色呼吸灯的。

STM32F1_DMA_PWM_WS2813E_2018_05_06-Bingo.zip (3.28 MB, 下载次数: 203)

这个是用TIM2_CH123实现三路LED控制的。

STM32F1_DMA_PWM_WS2813E_2018_05_06-TIM2_CH123.zip (3.29 MB, 下载次数: 154)

这个是用TIM2345_CH1实现四路LED控制的。

STM32F1_DMA_PWM_WS2813E_2018_05_06-TIM2345.zip (3.29 MB, 下载次数: 149)

点评

bioger 谢谢分享 详情 回复 发表于 2018-5-7 08:31

回复 评分 举报

naga568

368 | 0

发表于 2018-5-7 07:51:59 | 只看该作者 4楼

> 通宵敲代码 发表于 2018-5-6 22:49
> 上面介绍完原理，下面就是如何实现功能了，
> 我们直接来看程序，声明一下我们是用原子的例程移植的。
> 这 ...

TA的帖子 | TA的资源

纯净的硅（初级）

发消息 加好友

謝謝分享。

回复 评分 举报

bioger

294 TA的帖子 | 0 TA的资源

一粒金砂（高级）

发消息 加好友

发表于 2018-5-7 08:31:00 | 只看该作者 5楼

> 通宵敲代码 发表于 2018-5-6 22:49
> 定义了三个数组，主函数里写个三色循环呼吸灯看看效果。
>
> 附上相关程序
> 这个是用TIM2_CH1实 ...

谢谢分享

回复 评分 举报

通宵敲代码

1171 | 2

楼主 | 发表于 2018-5-7 11:28:47 | 只看该作者 6楼

在开启DMA数据传输，定时器开启之前，
增加一条计数寄存器清空指令，
防止寄存器非零造成第一次脉冲信号不准确。

同时在DMA数据传输完成后，应先关闭DMA通道，再关闭定时器，
防止DMA传输完成，但PWM最后一次信号未发送完影响最后一个灯的显示。

TA的帖子 | TA的资源

版主

发消息 加好友

回复 评分 举报

eleven@

发表于 2018-5-19 21:33:52 | 只看该作者 7楼

楼主，为什么我的PA0没有输出呢？用万用表也测不出来，我按照你的代码，原封不动，可就是没有输出，还请大佬指导一波

3 TA的帖子 | 0 TA的资源

一粒金砂（初级）

发消息 加好友

回复 评分 举报

eleven@

发表于 2018-5-19 21:37:26 | 只看该作者 8楼

还有，LED_Init();这个来初始化LED貌似没什么用吧，程序里好像没用上吧，输出是PA0吧？

3 TA的帖子 | 0 TA的资源

点评

通宵敲代码 嗯，这个确实没用，以前的函数留下的。 输出引脚确实是PA0，程序都是验证过的，你好好检查一下硬件。 详情 回复 发表于

一粒金砂（初级）

发消息 加好友

2018-5-25 17:12

回复 评分 举报

eleven@

发表于 2018-5-20 12:33:27 | 只看该作者 9楼

我仿真出来PA0一直是低电平

3 TA的帖子 | 0 TA的资源

一粒金砂（初级）

发消息 加好友

回复 评分 举报

通宵敲代码

楼主 | 发表于 2018-5-25 17:12:02 | 只看该作者 10楼

> eleven@ 发表于 2018-5-19 21:37
> 还有，LED_Init();这个来初始化LED貌似没什么用吧，程序里好像没用上吧，输出是PA0吧？

嗯，这个确实没用，以前的函数留下的。
输出引脚确实是PA0，程序都是验证过的，你好好检查一下硬件。

1171 TA的帖子 | 2 TA的资源

版主

发消息 加好友

回复 评分 举报

DSCX05

发表于 2018-7-20 10:01:28 | 只看该作者 11楼

楼主你的硬件不是级联？程序上看起来是每一个都用一个DMA的通道？

点评

通宵敲代码 做的四路独立控制， 需要级联你直接延长灯带就行了， 我试过288颗灯级联的，毫无压力， 官方给的数据可以级联1000颗灯 详情 回复 发表于 2018-7-30 08:58

回复 评分 举报

32 TA的帖子 | 0 TA的资源

一粒金砂（中级）

发消息 加好友

mochou678

发表于 2018-7-29 16:08:17 | 只看该作者 12楼

优酷视频打不开啊，能直接发个图片我看下或者其他视频，看下显示效果，不知道我这边显示的效果对不对。

点评

通宵敲代码 百度一搜很多这个灯珠的视频， 我这边公司的网络也上不了优酷， 我记不清这个是呼吸灯还是流水灯了， 你看一下，只要颜色渐变流畅就没问题。 详情 回复 发表于 2018-7-30 09:01

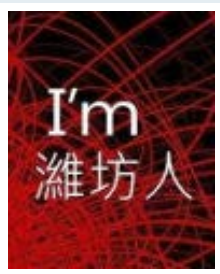
回复 评分 举报

1 TA的帖子 | 0 TA的资源

一粒金砂（初级）

发消息 加好友

通宵敲代码



1171 TA的帖子 | 2 TA的资源

版主

发消息 加好友

楼主 | 发表于 2018-7-30 08:58:59 | 只看该作者 13楼

> DSCX05 发表于 2018-7-20 10:01
> 楼主你的硬件不是级联？程序上看起来是每一个都用一个DMA的通道？

做的四路独立控制，
需要级联你直接延长灯带就行了，
我试过288颗灯级联的，毫无压力，
官方给的数据可以级联1000颗灯

回复 评分 举报

通宵敲代码

1171 TA的帖子 | 2 TA的资源

版主

发消息 加好友

楼主 | 发表于 2018-7-30 09:01:08 | 只看该作者 14楼

> mochou678 发表于 2018-7-29 16:08
> 优酷视频打不开啊，能直接发个图片我看下或者其他视频，看下显示效果，不知道我这边显示的效果对不对。

百度一搜很多这个灯珠的视频，
我这边公司的网络也上不了优酷，
我记不清这个是呼吸灯还是流水灯了，
你看一下，只要颜色渐变流畅就没问题。

回复 评分 举报

mk0007

发表于 2018-7-30 09:09:21 | 只看该作者 15楼

69 TA的帖子 | 0 TA的资源

一粒金砂（中级）

发消息 加好友

回复 评分 举报

soqy

发表于 2018-8-24 16:10:57 | 只看该作者 16楼

怎么输出来的脉冲没有24个？

1 TA的帖子 | 0 TA的资源

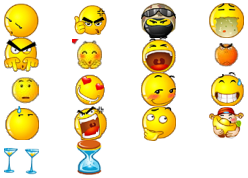
一粒金砂（初级）

发消息 加好友

回复 评分 举报

发帖 回复 返回列表

点此添加图片或附件



replysubmit 回帖后跳转到最后一页

本版积分规则

扫码关注
EEWORLD 官方微信

扫码关注
EE福利 唾手可得