STM32 入门系列教程

GPIO 的编程

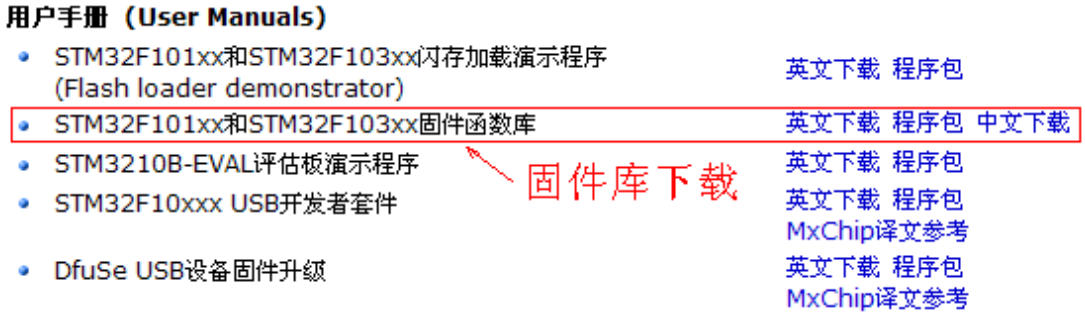
Revision 0.01

(2010-04-19)

本教程目的：帮助刚刚接触，甚至都没看过 STM32 的用户手册的同学，进行第一次接触 STM32 编程。如果您是高手，请绕道。

学过单片机的人都知道，要想入门，第一步就是要在开发板上试试 GPIO。大家看 STM32F103VET6 的手册时，会发现，该 CPU 共 100 个引脚，GPIO 引脚居然占了 80 个。可见，STM32 本身就是一个高级单片机，不可怕。闲话少说，进入主题。

要进行 STM32 的编程，与单片机不同，它有一个固件库。所谓固件库，实际上就是一大堆标准的函数（接口），我们写程序的时候只要去调用它即可。所以，在写程序之前，希望大家对固件库有一个了解。最新的固件库代码，可以从网址：http://www.stmicroelectronics.com.cn/stonline/mcu/MCU_Pages.htm 获取。该网址中还包括了如何使用固件库——文档 0427，截图如下所示：



实际上，固件库就是一个模板，我们下载这个模板，就可以利用这个模板作为一个平台来开发 stm32。本文使用的是网友上传的一个固件库，该固件库模板可以从 ARM 技术交流网下载到：

http://www.arm79.com/read.php?tid=2016

下载固件库，解压，里面的目录结构如下：

进入 Project 目录，即可看到一个 Uv2 或者 Uv3 的图标，双击打开 MDK 工程（请事先安装好 MDK 软件）：

在打开的界面左边栏，有该工程的所有文件，我们关注的是 User 文件夹，里面有一个 main.c 文件，我们以后大部分的例程都会在这里面编写。其他的文件夹暂且不需要去理它。

打开 main.c，找到：

```
int main(void)
{
#ifdef DEBUG
  debug();
#endif


  /* Infinite loop */
  while (1)
  {
  …..（循环代码）
  }
}
```

这个循环代码就是需要我们填写的代码部分。想做 GPIO 的练习？OK，直接把 GPIO 的初始化部分代码，写到这里，并且在 while(1)后面加入控制 LED 的程序就 ok！很简单吧。。。嘿嘿。那就开始吧。

进行 STM32 开发之前，请务必确保具备两个文档：

1、《STM32F10xxx_参考手册.pdf 》

2、《STM32 固件库.pdf》，即固件库的用户文档 UM0427 的中文翻译版本。

以上两个文档均可以在芯达开发板光盘的配套芯片手册目录中找到。

OK，打开《STM32 固件库.pdf》，该文档也可以在如下网址下载到：

http://www.arm79.com/read.php?tid=1785。

找到 P34 页，2.3 小节 外设的初始化和设置，GPIO 端口在 STM32 中，是作为一个外设存在的。现在我们按照这里的步骤，一步一步写程序。实际上就是在调用固件库里写好的函数而已。

步骤一 声明 GPIO 的结构：

```
GPIO_InitTypeDef  GPIO_InitStructure;
```

步骤二 为变量 GPIO_InitStructure 的成员赋值，如果只设置其中的一部分成员，我们需要如下代码：

```
/**
```

```
 *   LED1 -> PB8 , LED2 -> PB9 , LED3 -> PE0 , LED4 -> PE1
 */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 |GPIO_Pin_9;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 |GPIO_Pin_1;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOE, &GPIO_InitStructure);
```

以上四行代码，做一个说明：

实际上这里省略掉一个函数：GPIO_StructInit，它是用来初始化变量 GPIO_InitStructure 的，经过实验，发现不用也可以。大家可以尝试一下：）然后修改该变量中的成员，有三个成员。在芯达 STM32 开发板上，GPIO 端口接的是 PE0 引脚与 PE1 引脚、PB8 引脚与 PB9 引脚。因此，我们在 GPIO_Pin 成员这里赋值 GPIO_Pin_0 | GPIO_Pin_1，GPIO_Pin_8 |GPIO_Pin_9。在 GPIO_Speed 成员上赋值 GPIO_Speed_50MHz，GPIO_Mode 成员则是设置为 GPIO_Mode_Out_PP，表示推挽输出模式。

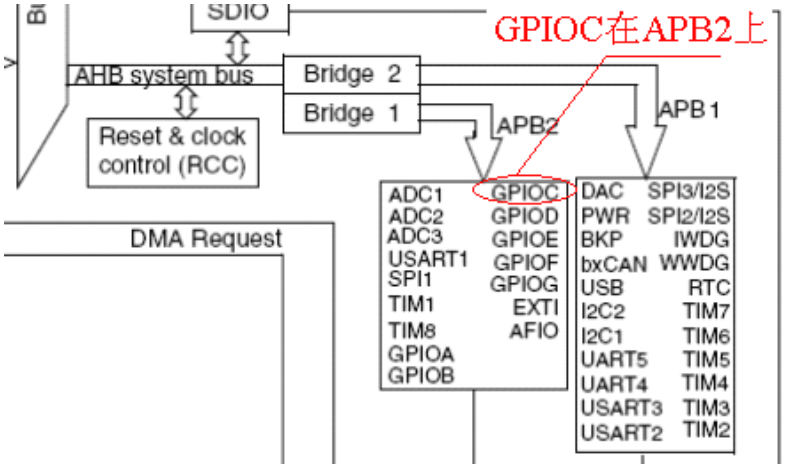
步骤三 调用函数 GPIO_Init()来初始化外设 GPIO，代码如下：

```
GPIO_Init(GPIOC , &GPIO_InitStructure);
```

步骤四 使能。注意，在固件库中，GPIO 没有 GPIO_Cmd 的函数，因此这个步骤省略。

通过以上四个步骤，我们已经对 GPIO 进行了设置。还有一个问题不能忽略：在设置外设前，我们必须给它调用一个时钟函数来使能外设时钟。

在 CPU 的用户手册中，我们知道，stm32 有好几个时钟的，现在我们用哪个时钟呢？OK，打开《STM32F10xxx 参考手册》中文版的 P17 页，截图如下页所示。我们使用的是 GPIOC 端口，因此，使用的是 APB2。

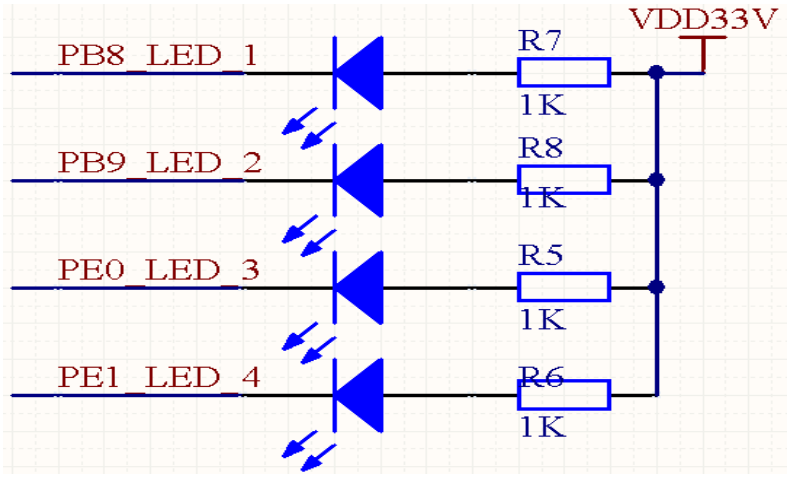


定下这个 APB2 对应的函数后，我们就调用它：

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB |
                       RCC_APB2Periph_GPIOB ENABLE);
```

注意，这个时钟的使能函数，应该放在最前面。必须先有时钟，才能做后续的操作。下面来控制 LED 吧！！

我们先看原理图里，LED 如何连接的：



从原理图，我们可以看到，要使得四个 LED 都亮起来，必须把对应的引脚清零。置 1 会让 LED 灭。现在打开 STM32 固件库文档，找到 10.2 小节，GPIO 库函数。GPIO 设置的所有函数，都在这里。我们看下要使用 GPIO 库的哪个函数。找到：GPIO_SetBits();和 GPIO_ResetBits();

这两个函数，根据说明，分别是设置某个引脚为高电平和低电平。GPIO_SetBits 函数是设置高电平，GPIO_ResetBits 函数则是清零操作。根据我们获得的信息，写出如下代码：

```
    while (1){
        /*====LED1-ON=======*/
        GPIO_ResetBits(GPIOB , GPIO_Pin_8);
        GPIO_SetBits(GPIOB , GPIO_Pin_9);
        GPIO_SetBits(GPIOE , GPIO_Pin_0);
        GPIO_SetBits(GPIOE , GPIO_Pin_1);
        Delay(0xfffff);
        Delay(0xfffff);
        Delay(0x5ffff);

        /*====LED12-ON=======*/
        GPIO_ResetBits(GPIOB , GPIO_Pin_8);
        GPIO_ResetBits(GPIOB , GPIO_Pin_9);
        GPIO_SetBits(GPIOE , GPIO_Pin_0);
        GPIO_SetBits(GPIOE , GPIO_Pin_1);
        Delay(0xfffff);
        Delay(0xfffff);
        Delay(0x5ffff);

        /*====LED123-ON=======*/
        GPIO_ResetBits(GPIOB , GPIO_Pin_8);
        GPIO_ResetBits(GPIOB , GPIO_Pin_9);
        GPIO_ResetBits(GPIOE , GPIO_Pin_0);
        GPIO_SetBits(GPIOE , GPIO_Pin_1);
```

```
        Delay(0xfffff);
        Delay(0xfffff);
        Delay(0x5ffff);

        /*====LED1234-ON=======*/
        GPIO_ResetBits(GPIOB , GPIO_Pin_8);
        GPIO_ResetBits(GPIOB , GPIO_Pin_9);
        GPIO_ResetBits(GPIOE , GPIO_Pin_0);
        GPIO_ResetBits(GPIOE , GPIO_Pin_1);
        Delay(0xfffff);
        Delay(0xfffff);
        Delay(0x5ffff);

        /*====LED1234-OFF=======*/
        GPIO_SetBits(GPIOB , GPIO_Pin_8);
        GPIO_SetBits(GPIOB , GPIO_Pin_9);
        GPIO_SetBits(GPIOE , GPIO_Pin_0);
        GPIO_SetBits(GPIOE , GPIO_Pin_1);
        Delay(0xfffff);
        Delay(0xfffff);
        Delay(0x5ffff);
    }
```

我们的目的是让所有的 LED 有规律地闪烁，并且无限循环。

那么，现在就可以进行下载到开发板调试了吗？呵呵，还不行。

我们还需要调用 SystemInit();函数，来初始化整个系统，包括时钟设置到 72MHZ。以上配置结束后，您就可以根据 MDK + Jlink 的相关教程，下载 HEX 文件到板子里进行调试了。

在芯达 STM32 开发板上运行的 LED 闪烁的例程，也可以从 ARM 技术交流网下载：www.arm79.com。

附：

福州芯达工作室简介

福州芯达工作室成立于 2009 年 9 月，我们专注于嵌入式产品的研发与推广，目前芯达产品涉及 ARM9 系列、STM32 系列。

芯达团队成员均硕士研究生毕业，具有一定研发实力。我们的愿景在于把福州芯达打造成国内一流的嵌入式品牌。或许我们现在做的还不够，但是我们真的努力在做，希望通过我们的努力，能够在您学习和使用芯达产品的过程中带来或多或少的帮助。

这是芯达为了配合 STM32 开发板而推出的入门系列教程。如果您在看了我们的教程后，理清了思路，我们都会倍感欣慰！让我们一起学习，共同进步，在征服嵌入式领域的道路上风雨同行！

官方网站：http://www.arm79.com/

官方淘宝：http://shop36353570.taobao.com/