Hello STM32F103

Leoncha凉茶

2024 年 3 月 23 日

生存与发展是第一要务

摘要

这是一本为大学生，高中毕业生，单片机初学者，对学习单片机感到无力的人所写的书，如果你不知道该怎么学习，对学习产生了消极情绪，对自己的能力期望过高，请耐心读下去。

我们知道，单片机是一块PCB板，有一块或多块主控芯片，一些必备的模块和一些外设，传感器组成。有时这些板子上还带有裸露的接口，这是我们就可以自由的把一些传感器连接到IO口上。

我这里有两个问题：什么是单片机？单片机最小系统的三要素是什么？标准答案是：单片机是一种集成电路芯片。单片机最小系统三要素包括电源电路、晶振电路、复位电路。这两个问题划清了单片机与普通电路的区别，圈定了单片机的范围。

对于初学者来说，STM32系列单片机内容繁杂，难以为继，尤其是51都没有搞清楚的小朋友。这是一个致命的问题。首先，要搞清楚STM32由哪些部分构成，都有什么作用，有哪些应用？对于软件、云服务、物联网、机器人，STM32在这些行业中处于怎样的生态位，起到什么作用。

STM32系列单片机内容繁杂，难以为继，尤其是51都没有搞清楚的小朋友。这是一个致命的问题。首先，要搞清楚STM32由哪些模块组成，可以发挥什么作用，可以搭建什么应用？对于软件、云服务、物联网、机器人以及家用电器领域，STM32在单片机行业中处于怎样的生态位？起到什么作用？

知识是信息与相互关系的集合。信息是指经验与总结，是符号；相互关系是指逻辑关系，因果关系。如果想通过前人的总结学会知识，只接收信息是不够的，当你开始询问为什么时，就是在请求逻辑关系与因果关系。

笔者才疏学浅，只能为迷路的初学者指明方向，剩下的一切都要大家自己学习。

下面是模块清单，初学者请跳过：STM32F103芯片、F103板上资源、ARM Cortex-M3架构相关知识（ARM系列处理器的特点，中断触发，寄存器，总线结构）、Keil编译器、STMCubeMX、Linux下开源开发库libopencm3、ST-Link/J-Link、DEBUG、UCOSII、RTOS、SRAM、FLASH、E2PROM、ARM架构采用哈弗架构、改进哈弗架构、冯诺伊曼架构、看门狗（软件开门狗和硬件看门狗）、RTC时钟、SD卡、PWM波（PWM电机、DAC）、LED/LCD、触摸板库、HAL库、CMSIS STM32内容：GPIO八种模式、时钟树（四个时钟源，两个高速时钟源，两个低速时钟源，一个PLL时钟源，芯片和板上各有一套高低速时钟源）、NVIC中断（四个特点）、Timer（三种定时器）、总线结构（AHB、APB1、APB2，其中APB时钟频率是AHB的二分之一）、通信协议（UART、USART、USMART、I2C、SPI、1-wire、RS232、RS485、USB、2.4GHz、bt、esp8266、zigbee、lora、nb-iot）、流水线

1 点灯

通过配置GPIO来点亮小灯 STM32开发环境搭建、GPIO精讲、原理图

1.1 内容概述

1.1.1 GPIO讲解

大家在学习、搜集STM32资料时会发现，有的文章中说GPIO有八种工作模式，四种输入、四种输出；有的说是三种输入、两种输出，这些描述无所谓对错，关注这些就走上岔路了。

单片机的硬件基础决定了单片机的工作方式，晶振电路会为芯片提供”心跳“，“心跳”的频率与MCU的处理能力决定了单片机的性能。晶振也决定了单片机处理信号的方式，模拟信号必须转换为数字信号处理，也必须借助数字信号进行输出。否则只能通过设计特殊的电路结构来对信号进行定向处理，但这也将导致该结构不再通用，不过好消息是现在的MCU“心跳”很快，我们暂时无需担心信号来不及采集的问题。

GPIO就是MCU发挥功能的”门户“，其特点鲜明：

1.GPIO只能输出数字信号，对于连续的信号，也只能定时采样，无法真正接受连续的信号。 2.GPIO的结构决定了GPIO的工作模式，输入与输出是将GPIO视为了一根从MCU中伸出的铜线。这根铜线存在两种状态，一种是MCU“监听”这根铜线上的“电平”，另一种是MCU为这根铜线设置一个电平，对外提供电压，我们无需担心电力不足或者电力过高导致的问，这两种状态就是输入与输出。 3.一组GPIO中，各个端口可以分别设置不同的模式。

上述的“输入”与“输出”状态均为概念性描述，接下来解释GPIO八种工作状态：

这里不得不提一句单片机的电源电路（最小系统三要素）了，

1.1.2 HAL库

STM32F103并不是空中楼阁，在STM32F103之下，有ARM Cortex-M3架构的支持，CMSIS（ARM Cortex微控制器软件接口标准）就是ARM公司推出维护的，ST公司推出的HAL库就建立在这个标准的基础上，很明显，这些标准之间存在立体的上下相互支持关系。

1.1.3 delay函数

1.2 流程概述

1.2.1 初始化

第一步：初始化hal库第二步：设置时钟第三步：初始化delay函数第四步：初始化LED(自行编写)

1.2.2 循环部分

使用HAL库设置两个引脚拉高或拉低，500ms后交换状态

1.2.3 初始化LED

1.创建GPIO结构体 2.开启时钟 3.选择GPIO端口，设置GPIO口状态

2 按键输入

通过配置GPIO捕获按键按下产生的电信号 GPIO，NVIC，按键去抖

2.1 内容概述

2.1.1 NVIC

ARM Cortex-M3架构 NVIC

2.2 流程概述

2.2.1 初始化

2.2.2 时钟树

时钟树的组成，可以和ARM架构来一起讲，还能再扩展一点，把ARM中断也讲一下。

2.3 内容讲解

ARM中断，现场保护，寄存器，时钟树

2.4 时钟树

AHB、APB1/APB2、sysclk、systick

3 流水灯

使用Timer来控制小灯

3.1 内容概述

点灯、NVIC、精讲时钟树、Timer(定时器) 精讲时钟树!

3.2 流程概述

4 PWM电机控制

使用PWM波控制电机

4.1 内容概述

4.2 流程概述

5 看门狗

5.1 模块介绍

IWDG、WWDG、timer

5.2 内容概述

5.2.1 看门狗讲解

看门狗就是闹钟，看门狗有一个按钮，你可以上设置10s之内必须按一下，否则，闹钟就会响，分为两种，一个是窗口看门狗，一个是独立看门狗
窗口看门狗，之所以称为窗口，是因为其喂狗时间是一个有上下限的范围内，你可以通过设定相关寄存器，设定其上限时间和下限时间：喂狗的时间不能过早也不能过晚。
独立看门狗由内部低速时钟，RC振荡器提供时钟信号。

5.3 流程概述