嵌入式技术基础与实践（第6版）
教学资源分享

苏州大学 王宜怀，yihuaiw@suda.edu.cn

2021年8月17日

内容简介

嵌入式计算机系统简称为嵌入式系统，其概念最初源于传统测控系统对计算机的需求，嵌入式系统主要分为微控制器（MCU）与应用处理器（MAP）两大类，MCU面向电子系统智能化，是嵌入式系统的最基础的知识。本书以嵌入式硬件构件及底层软件构件设计为主线，以通用嵌入式计算机（GEC）为基础，基于嵌入式软件工程的思想，按照“通用知识—驱动构件使用方法—测试实例—构件制作过程”的线条，逐步阐述MCU应用与实践。

目录

一、MCU课程教学脉络

1.1 MCU课程教学目标

（1）掌握硬件最小系统与软件最小系统框架。
（2）掌握常用基本输出的概念、知识要素、构件使用方法及构件设计方法。如通用I/O（GPIO）、模数转换ADC、数模转换DAC、定时器模块等。
（3）掌握若干嵌入式通信的概念、知识要素、构件使用方法及构件设计方法。如串行通信接口UART、串行外设接口SPI、集成电路互联总线I2C，CAN、USB、嵌入式以太网、无线射频通信等。
（4）掌握常用应用模块的构件设计方法、使用方法及数据处理方法。如显示模块（LED、LCD、触摸屏等）、控制模块（控制各种设备，包括PWM等控制技术）等。数据处理如图形、图像、语音、视频等处理或识别等。
（5）掌握一门实时操作系统的基本用法与基本原理。作为软件辅助开发工具的实时操作系统，也可以作为一个知识要素。（本书不涉及，作为独立课程）
（6） 掌握嵌入式软硬件的基本调试方法。如断点调试、打桩调试、printf调试方法等。在嵌入式调试过程中，特别要注意确保在正确硬件环境下调试未知软件，在正确软件环境下调试未知硬件。

1.2 MCU的选型问题

本书本书以意法半导体（ST）的ARM Cortex-M4内核的STM32L431微控制器为蓝本、以知识要素为核心、以构件化为基础阐述嵌入式技术基础与实践。满足基本教学需要并跟随时代发展。

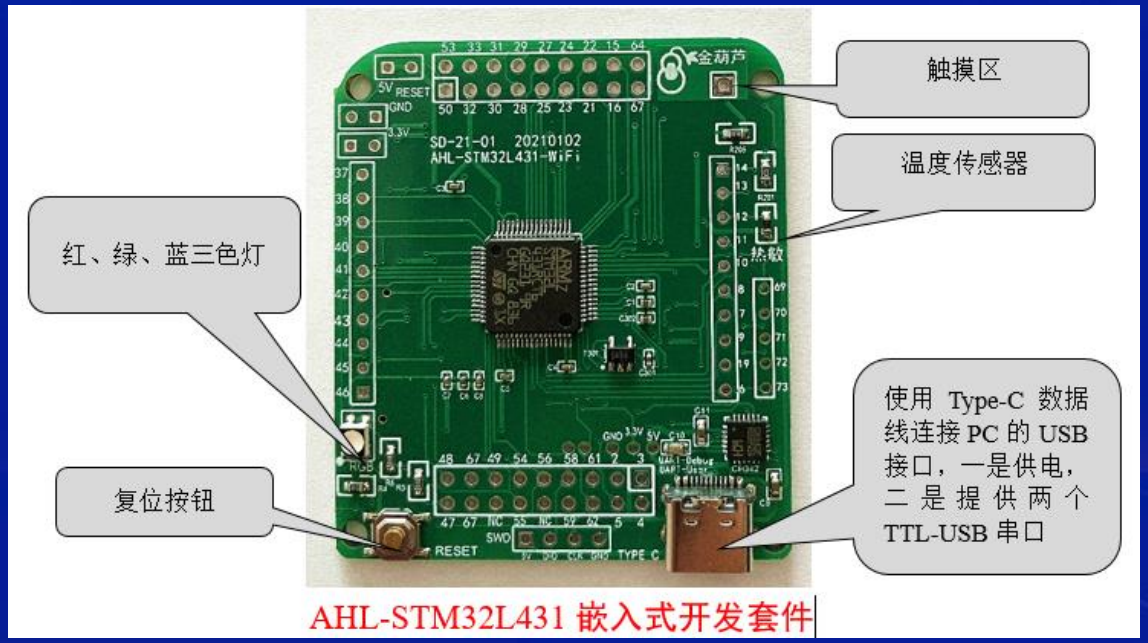
需要特别说明的是，虽然书籍撰写与教学必须以某一特定芯片为蓝本，但作为嵌入式技术基础，我们试图阐述嵌入式通用知识要素。因此，本书以知识要素为基本立足点设计芯片底层驱动，使得应用程序与芯片无关，具有通用嵌入式计算机（GEC）性质。

1.3 实践问题

学习MCU，必须在基本理解基本概念的基础上，进行软硬件相结合的实践，必须具有良好的实践载体。

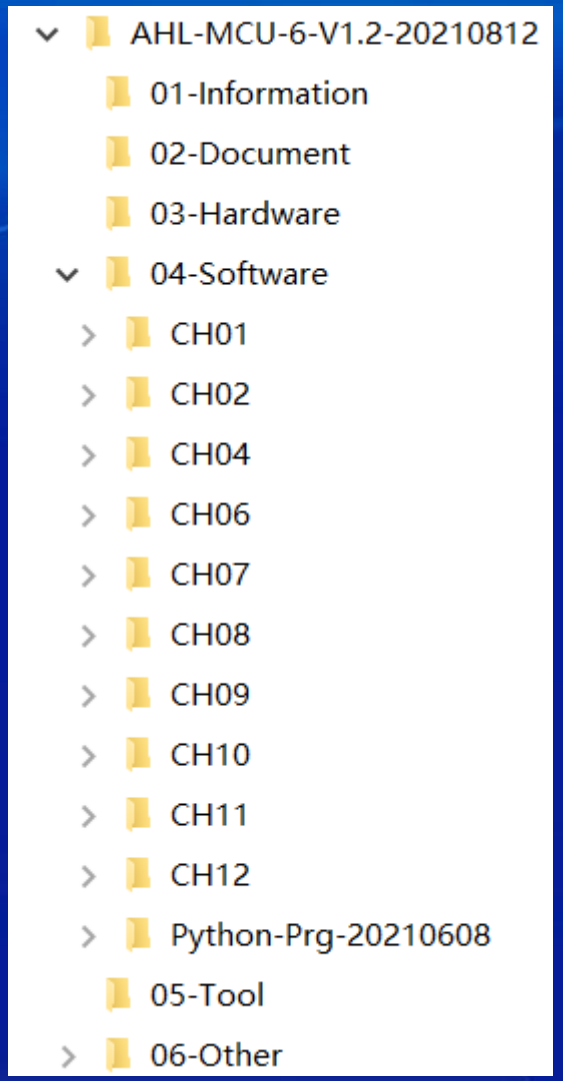
硬件系统：

对实践器材的基本要求是满足基本知识要素实践训练，价格低廉，使用方便，稳定可靠。AHL-STM32L431开发套件分迷你型、扩展型型二种型号，更详细的介绍见配套电子资源。迷你型可以完成基本知识要素实践，扩展型附加了九个外设部件，可用于实践创新训练。



开发环境：苏州大学嵌入式人工智能与物联网实验经过十多年努力研发了AHL-GEC-IDE（http://sumcu.suda.edu.cn/AHLwGECwIDE/main.htm），可以方便地用于RTOS实践。该IDE以通用嵌入式计算机GEC为目标对象，开发了AHL-GEC-IDE集成开发环境。它具有编辑、编译、链接等功能，特别是在配合金葫芦硬件后，可直接运行调试程序，兼容常用嵌入式集成开发环境。该环境摒弃了许多烦琐的设置，配合软件最小系统模板，使读者易于入门，简捷实用；特别是配合程序模板中提供的printf输出调试功能，使得打桩调试及运行跟踪显示得以方便实现。

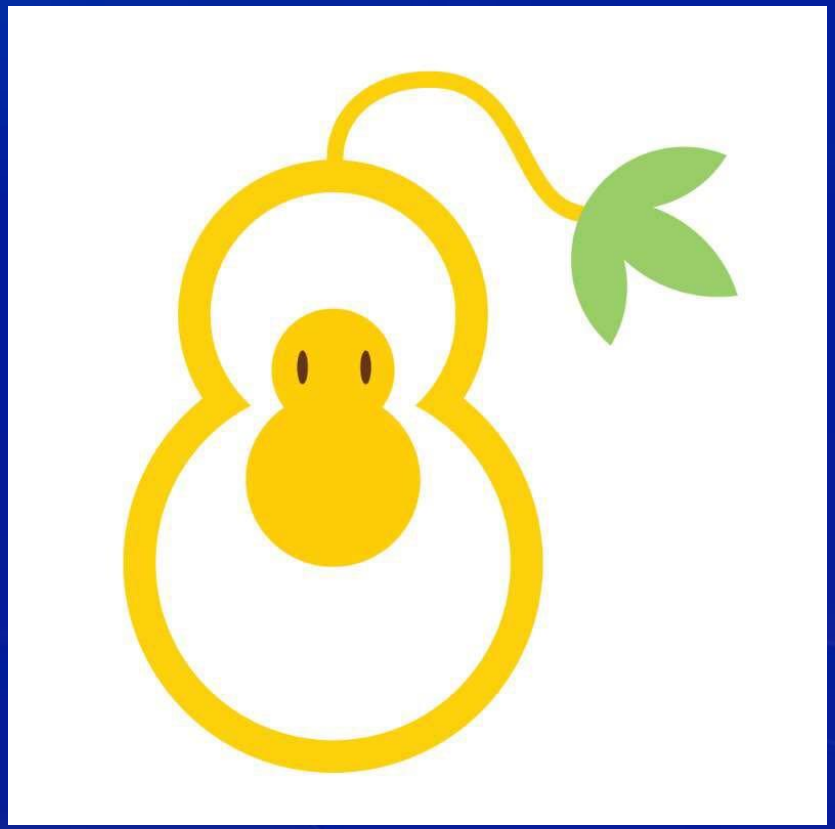
样例程序：规范的体现基本知识要素的样例程序是学习MCU的基础。样例设计要有统一的考虑，要能够体现对应的知识点。



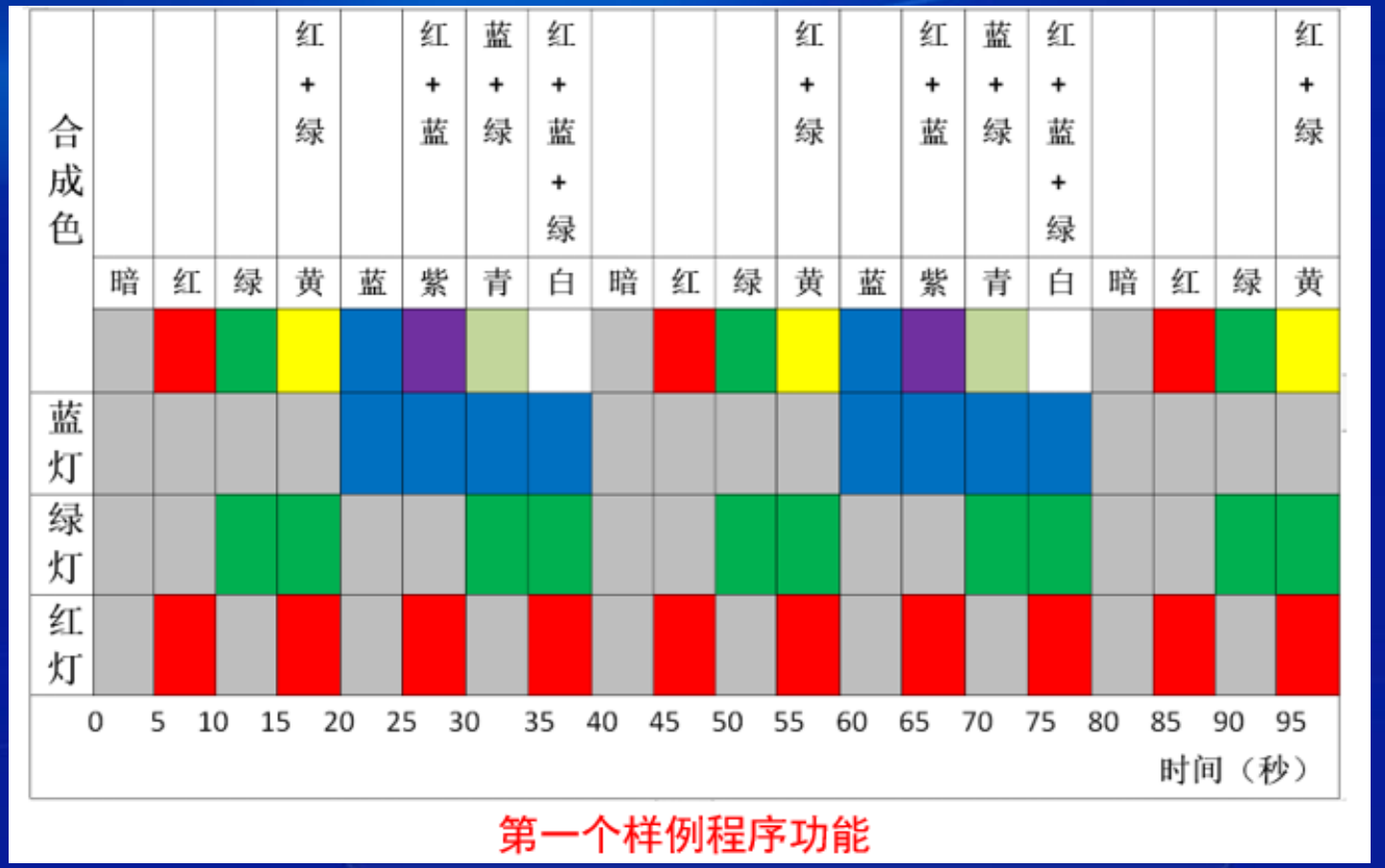
1.4 第一个实践样例

“照葫芦画瓢”基本理念

照葫芦画瓢：出自宋•魏泰《东轩笔录》第一卷，比喻照着样子模仿，表示简单易行之含义。而古希腊哲学家亚里士多德说过：“人从儿童时期起就有模仿本能，他们用模仿而获得了最初的知识，模仿就是学习”。借此，期望通过建立符合软件工程基本原理的“葫芦”，为“照葫芦画瓢”提供基础，达到降低学习难度之目的。



第一个样例功能



第一个样例程序功能

建立标准工程框架：做到“分门别类，各有归处”

01_Doc	文档文件夹：文档作为工程密切相关部分，是软件工程的基本要求
02_CPU	CPU 文件夹：存放 CPU 相关文件
03_MCU	MCU 文件夹：含有 linker_file、startup、MCU_drivers 下级文件夹
04_GEC	GEC 文件夹：引入通用嵌入式计算机（GEC）概念，预留该文件夹
05_UserBoard	用户板文件夹：含有硬件接线信息的 User.h 文件及应用驱动
06_SoftComponent	软件构件文件夹：含有与硬件无关的软件构件
07_AppPrg	应用程序文件夹：应用程序主要在此处编程

工程框架

1.5 为教学所做的准备工作总结

（1）实践硬件：AHL-STM32L431（含在MCU-6版书中）
（2）开发环境：AHL-GEC-IDE
（3）样例程序编制：针对各基本要素，按照统一模板编制了样例程序
（4）撰写书籍：嵌入式技术基础与实践（第6版）—基于STM32L431微控制器，清华大学出版社，2021年
（5）课件：需要时可以获取，正在进行其他教学资源建设
（6）视频：正在努力制作
（7）MOOC：国家级本科一流课程《嵌入式系统及应用》
（8）其他资源

国家级一流本科课程

证书

★ ★ ★ ★ ★

课程类别：线上一流课程

课程名称：嵌入式系统及应用

课程负责人：王宜怀

课程团队其他主要成员：张建、王林

主要建设单位：苏州大学

主要开课平台：爱课程（中国大学 MOOC）

中华人民共和国教育部

2020 年 11 月

证书编号：2020118275

二、MCU-6版教材脉络梳理

2.1 内容简介

本书以意法半导体（ST）的ARM Cortex-M4内核的STM32L431微控制器为蓝本、以知识要素为核心、以构件化为基础阐述嵌入式技术基础与实践，同时本书随附实践硬件系统。全书共12章，其中第1章在运行一个嵌入式系统实例基础上简要阐述嵌入式系统的知识体系、学习误区与学习建议；第2章给出处理器简介；第3章给出MCU存储映像、中断源与硬件最小系统。第4章以GPIO为例给出规范的工程组织框架，阐述底层驱动应用方法；第5章阐述嵌入式硬件构件与底层驱动构件基本规范。第6章给出串行通信接口UART及第一个带中断的实例。1～6章囊括了学习一个微控制器入门环节的完整要素。7～11分别给出了Systick、Timer、PWM、Flash在线编程、ADC、DAC、SPI、I2C、CAN、DMA、系统时钟、看门狗、复位模块及电源控制模块等。第12章给出了RTOS、嵌入式人工智能、NB-IoT、4G、Wi-Fi及WSN等应用案例。

2.2 一级目录

2.3 配套开发环境AHL-GEC-IDE

苏州大学嵌入式团队在早期研发集成开发环境的基础上，经过三年多的努力，以通用嵌入式计算机GEC为目标对象，开发了AHL-GEC-IDE集成开发环境。它具有编辑、编译、链接等功能，特别是在配合金葫芦硬件后，可直接运行调试程序，兼容常用嵌入式集成开发环境。该环境摒弃了许多烦琐的设置，配合软件最小系统模板，使读者易于入门，简捷实用；特别是配合程序模板中提供的printf输出调试功能，使得打桩调试及运行跟踪显示得以方便实现。

三、核心内容导引

3.1 关于嵌入式系统含义的理解

总地说来，可以从计算机本身的角度概括表述嵌入式系统。嵌入式系统，即嵌入式计算机系统，它是不以计算机面目出现的“计算机”，这个计算机系统隐含在各类具体的产品之中，这些产品中的计算机程序起到了重要作用。

计算机是因科学家需要一个高速的计算工具而产生的。可以说，是因为通信、测控与数据传输等领域对计算机技术的需求催生了嵌入式系统的产生。

3.2 关于嵌入式系统的分类

从嵌入式系统的学习角度来看，因为应用于不同领域的嵌入式系统，其知识要素与学习方法有所不同，所以可以按应用范围简单地把嵌入式系统分为电子系统智能化（微控制器类）和计算机应用延伸（应用处理器）这两大类。一般来说，微控制器与应用处理器的主要区别在于可靠性、数据处理量、工作频率等方面，相对应用处理器来说，微控制器的可靠性要求更高、数据处理量较小、工作频率较低。

微控制器类嵌入式系统产品，从形态上看，更类似于早期的电子系统，但内部计算程序起核心控制作用。

应用处理器类嵌入式系统产品，从形态上看，更接近通用计算机系统。从开发方式上看，也类似于通用计算机的软件开发方式。

可以使用通用嵌入式计算机（GEC）进行共性抽取，纳入统一体系研究。

3.3 关于嵌入式系统的特点

1．**嵌入式系统属于计算机系统，但不单独以通用计算机的面目出现**。嵌入式系统它不仅具有通用计算机的主要特点，又具有自身特点。嵌入式系统也必须要有软件才能运行，但其隐含在种类众多的具体产品中。同时，通用计算机种类屈指可数，而嵌入式系统不仅芯片种类繁多，而且由于应用对象大小各异，嵌入式系统作为控制核心，已经融入各个行业的产品之中。

2．**嵌入式系统开发需要专用工具和特殊方法**。嵌入式系统不像通用计算机那样，有了计算机系统就可以进行应用软件的开发。一般情况下，微控制器或应用处理器的芯片本身不具备开发功能，必须要有一套与相应芯片配套的开发工具和开发环境。

3．**使用MCU设计嵌入式系统，数据与程序空间采用不同存储介质**。在通用计算机系统中，程序存储在硬盘上。实际运行时，通过操作系统将要运行的程序从硬盘调入内存（RAM），运行中的程序、常数、变量均在RAM中。一般情况下，以MCU为核心的嵌入式系统中，其程序被固化到非易失性存储器 中。变量及堆栈使用RAM存储器。

4．**开发嵌入式系统涉及软件、硬件及应用领域的知识**。嵌入式系统与硬件紧密相关，嵌入式系统的开发需要硬件和软件的协同设计、协同测试。

3.4 嵌入式系统的学习路径

嵌入式系统的初学者应该通过选择一个具体MCU作为蓝本，期望通过学习实践，获得嵌入式系统知识体系的通用知识，其基本原则是：入门时间较快、硬件成本较少、软硬件资料规范、知识要素较多、学习难度较低。

要想成为一名知识结构合理且比较全面的嵌入式系统工程师，应该选择一个较典型的微控制器作为入门芯片，且从不带操作系统（NOS）学起，由浅入深，逐步推进。

针对许多初学者选择“xxx嵌入式操作系统+xxx处理器”的嵌入式系统的入门学习模式，本书认为是不合适的。本书的建议是：首先把嵌入式系统软件与硬件基础打好，再根据实际应用需要，选择一种实时操作系统（RTOS）进行实践。

嵌入式系统设计是一个软件和硬件协同设计的工程，不能像通用计算机那样，软件和硬件完全分开来看，要在一个大的框架内协调工作。要想成为一名合格的嵌入式系统设计工程师，在初学阶段，必须重视打好嵌入式系统的硬件与软件基础。

嵌入式系统与硬件紧密相关，是软件与硬件的综合体，没有对硬件的理解就不可能写好嵌入式软件，同样没有对软件的理解也不可能设计好嵌入式硬件。

3.5 基础阶段的学习建议

对于看似庞大的嵌入式系统知识体系，可以使用“电子札记”的方式进行知识积累与补缺补漏，任何具有一定理工科基础的学生，通过一段稍长时间的静心学习与实践，都能学好嵌入式系统。

（1）遵循“先易后难，由浅入深”的原则，打好软硬件基础。

（2）充分理解知识要素、掌握底层驱动构件的使用方法。

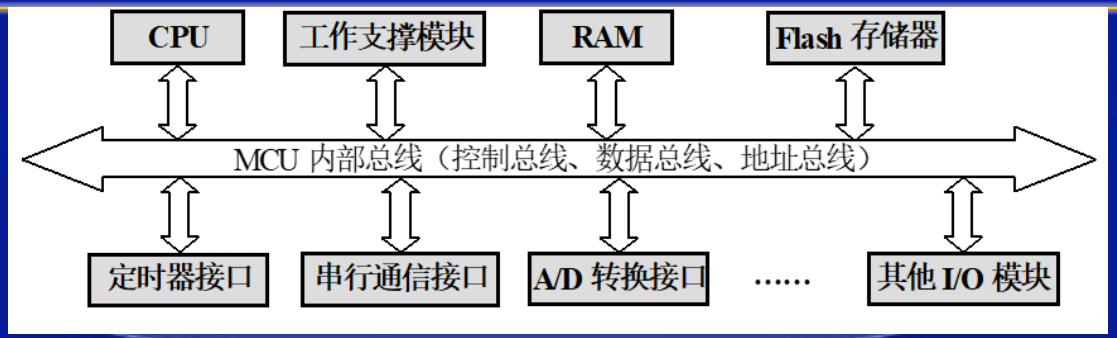
（3）基本掌握底层驱动构件的设计方法。

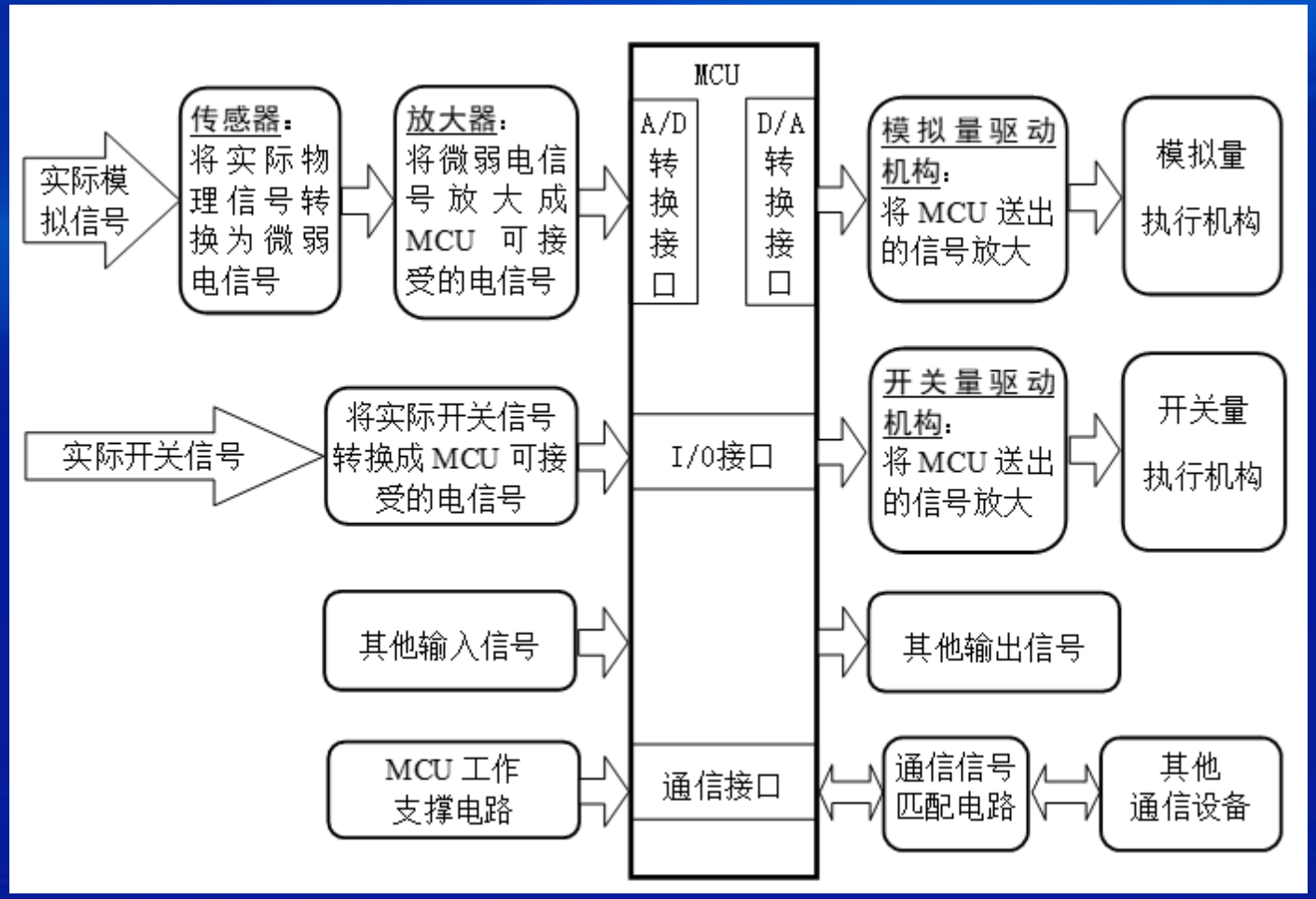
（4）掌握单步跟踪调试、打桩调试、printf输出调试等调试手段。

（5）日积月累，勤学好问，充分利用本书及相关资源。

3.6 MCU与MAP

MCU是在一块芯片内集成了CPU、存储器、定时器/计数器及多种输入输出（I/O）接口的比较完整的数字处理系统。以MCU为核心的系统是应用最广的嵌入式系统，是现代测控系统的核心。MCU出现之前，人们必须用纯硬件电路实现测控系统。MCU出现以后，测控系统中的大部分计算与控制功能由MCU的软件实现，输入、输出与执行动作等通过硬件实现，带来了设计上的本质变化。MAP是在低功耗CPU的基础上扩展音视频功能和专用接口的超大规模集成电路，其功能与开发方法接近PC。



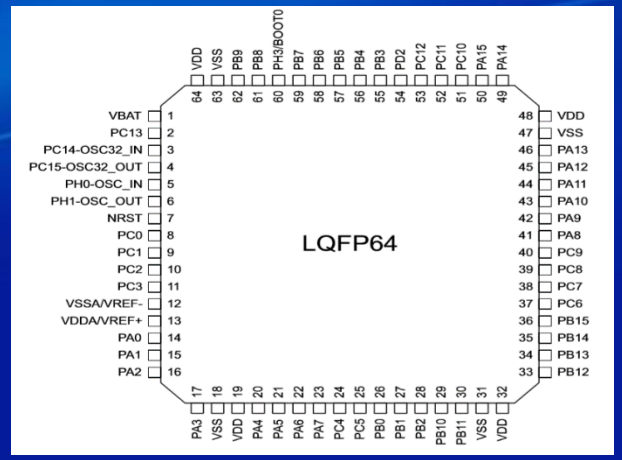


软件在其中起到了重要作用

3.7 初识一个MCU

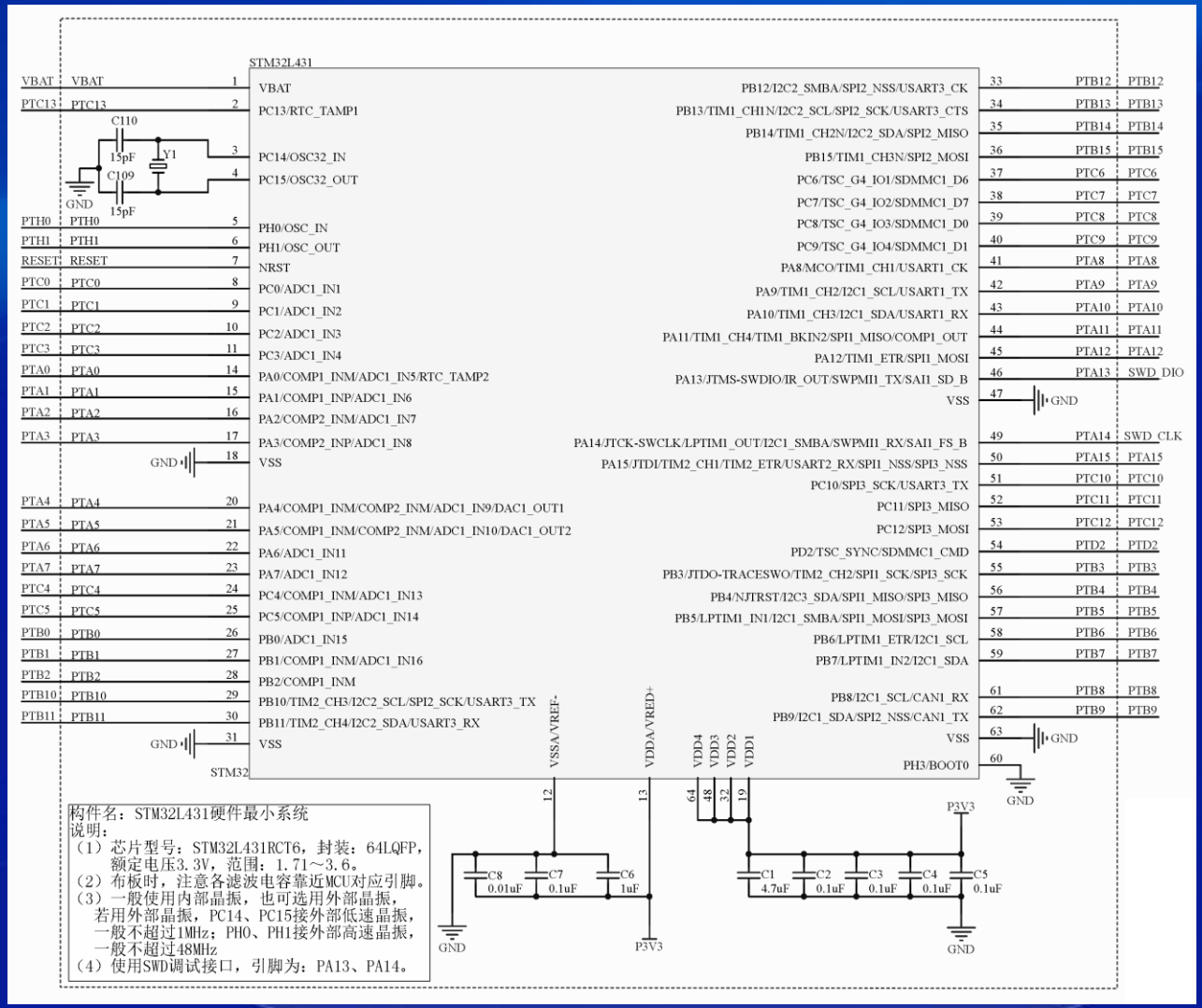
初识一个MCU首先要从认识型号标识开始，可以从型号标识中获得芯片家族、产品类型、具体特性、引脚数目、Flash大小、温度范围、封装类型等信息，这些信息是购买购买芯片的基本要求；其次了解内部RAM及Flash的大小、地址范围，以便设置链接文件，为程序编译及写入做好准备；再次了解中断源及中断向量号，为中断编程做准备。

3.8 芯片引脚与硬件最小系统



（书籍3.2节）

一个芯片的硬件最小系统是指可以使内部程序运行所必须的最低规模的外围电路，也可以包括写入器接口电路。使用一个芯片，必须完全理解其硬件最小系统。硬件最小系统引脚是我们必须为芯片提供服务的引脚，包括电源、晶振、复位、SWD接口等，做好这些服务之后，其他引脚就为用户提供服务了。硬件最小系统电路中着重掌握电容滤波原理及布板时靠近对应引脚的基本要求。


STM32L431
VBAT
PC13/RTC_TAMP1
PC14/OSC32_IN
PC15/OSC32_OUT
PH0/OSC_IN
PH1/OSC_OUT
NRST
PC0/ADC1_IN1
PC1/ADC1_IN2
PC2/ADC1_IN3
PC3/ADC1_IN4
PA0/COMP1_INM/ADC1_IN5/RTC_TAMP2
PA1/COMP1_INP/ADC1_IN6
PA2/COMP2_INM/ADC1_IN7
PA3/COMP2_INP/ADC1_IN8
VSS
PA4/COMP1_INM/COMP2_INM/ADC1_IN9/DAC1_OUT1
PA5/COMP1_INM/COMP2_INM/ADC1_IN10/DAC1_OUT2
PA6/ADC1_IN11
PA7/ADC1_IN12
PC4/COMP1_INM/ADC1_IN13
PC5/COMP1_INP/ADC1_IN14
PB0/ADC1_IN15
PB1/COMP1_INM/ADC1_IN16
PB2/COMP1_INM
PB10/TIM2_CH3/I2C2_SCL/SPI2_SCK/USART3_TX
PB11/TIM2_CH4/I2C2_SDA/USART3_RX
VSS
PB12/I2C2_SMBA/SPI2_NSS/USART3_CK
PB13/TIM1_CH1N/I2C2_SCL/SPI2_SCK/USART3_CTS
PB14/TIM1_CH2N/I2C2_SDA/SPI2_MISO
PB15/TIM1_CH3N/SPI2_MOSI
PC6/TSC_G4_IO1/SDMMC1_D6
PC7/TSC_G4_IO2/SDMMC1_D7
PC8/TSC_G4_IO3/SDMMC1_D0
PC9/TSC_G4_IO4/SDMMC1_D1
PA8/MCO/TIM1_CH1/USART1_CK
PA9/TIM1_CH2/I2C1_SCL/USART1_TX
PA10/TIM1_CH3/I2C1_SDA/USART1_RX
PA11/TIM1_CH4/TIM1_BKIN2/SPI1_MISO/COMP1_OUT
PA12/TIM1_ETR/SPI1_MOSI
PA13/JTMS-SWDIO/IR_OUT/SWPMI1_TX/SAI1_SD_B
VSS
PA14/JTCK-SWCLK/LPTIM1_OUT/I2C1_SMBA/SWPMI1_RX/SAI1_FS_B
PA15/JTDI/TIM2_CH1/TIM2_ETR/USART2_RX/SPI1_NSS/SPI3_NSS
PC10/SPI3_SCK/USART3_TX
PC11/SPI3_MISO
PC12/SPI3_MOSI
PD2/TSC_SYNC/SDMMC1_CMD
PB3/JTDO-TRACESWO/TIM2_CH2/SPI1_SCK/SPI3_SCK
PB4/NJTRST/I2C3_SDA/SPI1_MISO/SPI3_MISO
PB5/LPTIM1_IN1/I2C1_SMBA/SPI1_MOSI/SPI3_MOSI
PB6/LPTIM1_ETR/I2C1_SCL
PB7/LPTIM1_IN2/I2C1_SDA
PB8/I2C1_SCL/CAN1_RX
PB9/I2C1_SDA/SPI2_NSS/CAN1_TX
VSS
PH3/BOOT0
VBAT
PTC13
C110
15pF
Y1
C109
15pF
GND
PTH0
PTH1
RESET
PTC0
PTC1
PTC2
PTC3
PTA0
PTA1
PTA2
PTA3
PTA4
PTA5
PTA6
PTA7
PTC4
PTC5
PTB0
PTB1
PTB2
PTB10
PTB11
PTB12
PTB13
PTB14
PTB15
PTC6
PTC7
PTC8
PTC9
PTA8
PTA9
PTA10
PTA11
PTA12
PTA13
SWD_DIO
PTA14
SWD_CLK
PTA15
PTC10
PTC11
PTC12
PTD2
PTB3
PTB4
PTB5
PTB6
PTB7
PTB8
PTB9
VSSA/VREF-
VDDA/VREF+
VDD4
VDD3
VDD2
VDD1
P3V3
C8 0.01uF
C7 0.1uF
C6 1uF
C1 4.7uF
C2 0.1uF
C3 0.1uF
C4 0.1uF
C5 0.1uF
构件名：STM32L431硬件最小系统
说明：
（1）芯片型号：STM32L431RCT6，封装：64LQFP，额定电压3.3V，范围：1.71～3.6。
（2）布板时，注意各滤波电容靠近MCU对应引脚。
（3）一般使用内部晶振，也可选用外部晶振，若用外部晶振，PC14、PC15接外部低速晶振，一般不超过1MHz；PH0、PH1接外部高速晶振，一般不超过48MHz
（4）使用SWD调试接口，引脚为：PA13、PA14。

3.9 引入通用嵌入式计算机（GEC)概念

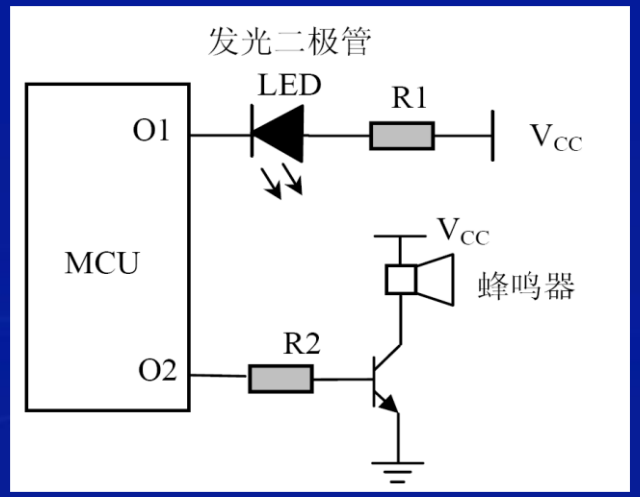
（书籍3.3节）

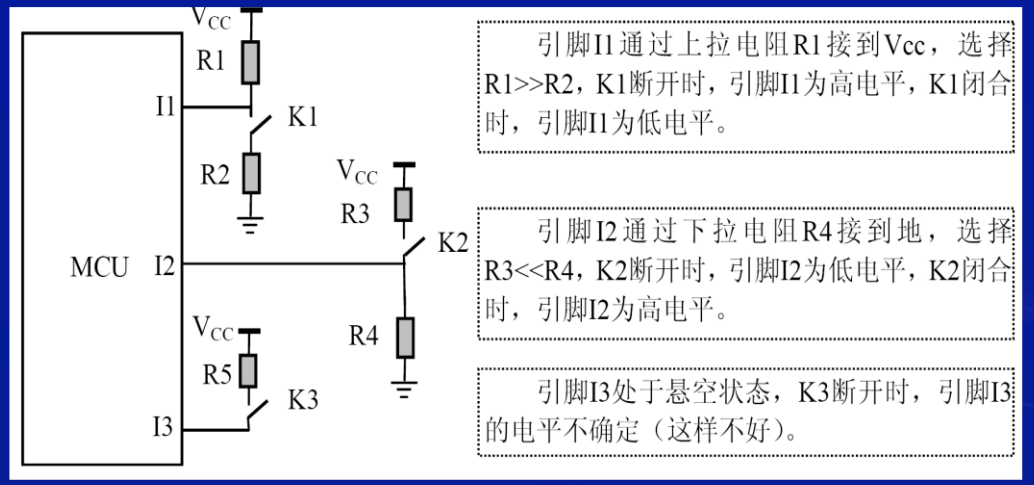
引入通用嵌入式计算机概念的目的不仅仅是为了降低硬件设计复杂度，更重要目的是降低软件开发难度。硬件上，使其只要供电就可工作，关键是其内部有BIOS。BIOS不仅可以驻留构件，还可以驻留实时操作系统，提供方便灵活的动态命令 等等。在最小的硬件系统基础上，辅以Wi-Fi、NB-IoT、5G等，可以形成不同应用的GEC系列，为嵌入式人工智能与物联网的应用提供技术基础。

3.10 GPIO概念

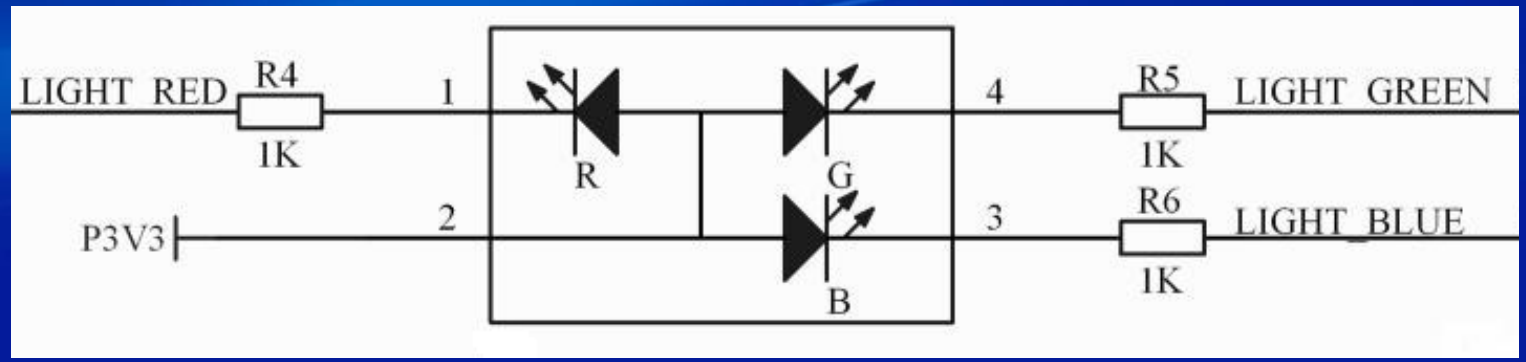
（书籍4.1节）

GPIO是输入/输出的最基本形式，MCU的引脚若作为GPIO输入引脚，即开关量输入，其含义就是MCU内部程序可以获取该引脚的状态，是高电平1，或是低电平0。若作为输出引脚，即开关量输出，其含义就是MCU内部程序可以控制该引脚的状态，是高电平1，或是低电平0。希望掌握开关量输入/输出电路的基本连接方法。





3.11 软件如何干预硬件



（书籍4.2节）

看看软件是如何干预硬件的。例如想点亮图3-4中的蓝色LED小灯，由该电路可以看出，只要使得标识LIGHT_BLUE的引脚为低电平，蓝色LED就可以亮起来。为了能够做到软件干预硬件，必须将该引脚与MCU的一个具有GPIO功能的引脚连接起来，通过编程使得MCU的该引脚电平为电平（逻辑0），蓝色LED就亮起来了，这就是软件干预硬件的基本过程。

（直接运行程序）

构件API（书籍第82页）

```
//=========================================================================
//函数名称：gpio_init
//函数返回：无
//参数说明：port_pin：(端口号)|(引脚号)（如：(PTB_NUM)|(9) 表示为 B 口 9 号脚）
//          dir：引脚方向（0=输入，1=输出,可用引脚方向宏定义）
//          state：端口引脚初始状态（0=低电平，1=高电平）
//功能概要：初始化指定端口引脚作为 GPIO 引脚功能，并定义为输入或输出，若是输出，
//          还指定初始状态是低电平或高电平
//=========================================================================
void gpio_init(uint16_t port_pin, uint8_t dir, uint8_t state);
```

GPIO构件的组成：GPIO.h、GPIO.c，不可多不可少，许多工程师没有做到

3.12 基于构件的程序框架

表4-2 工程文件夹内的基本内容

名称	文件夹		简明功能及特点
文档文件夹	01_Doc		工程改动时，及时记录。
CPU 文件夹	02_CPU		与内核相关的文件。
MCU 文件夹	03_MCU	linker_File	链接文件夹，存放链接文件。
		MCU_drivers	MCU 底层构件文件夹，存放芯片级硬件驱动。
		startup	启动文件夹，存放芯片头文件及芯片初始化文件。
GEC 相关文件夹	04_GEC		GEC 芯片相关文件夹，存放引脚头文件。
用户板文件夹	05_UserBoard		用户板文件夹，存放应用级硬件驱动，即应用构件。
软件构件文件夹	06_SoftComponent		抽象软件构件文件夹，存放硬件不直接相关的软件构件。
无操作系统源程序文件夹	07_AppPrg	include.h	总头文件，包含各类宏定义。
		isr.c	中断服务程序文件，存放各中断服务程序子函数。
		main.c	主程序文件，存放芯片启动的入口函数 main。

3.13 嵌入式硬件构件与底层驱动构件基本规范

机械、建筑等传统产业的运作模式是先生产符合标准的构件（零部件），然后将标准构件按照规则组装成实际产品。其中，构件（Component）是核心和基础，复用是必需的手段。传统产业的成功充分证明了这种模式的可行性和正确性。软件产业的发展借鉴了这种模式，为标准软件构件的生产和复用确立了举足轻重的地位。

嵌入式硬件构件是指将一个或多个硬件功能模块、支撑电路及其功能描述封装成一个可复用的硬件实体，并提供一系列规范的输入/输出接口。嵌入式硬件构件根据接口之间的生产消费关系，接口可分为供给接口和需求接口两类。根据所拥有接口类型的不同，硬件构件分为核心构件、中间构件和终端构件三种类型。核心构件只有供给接口，没有需求接口，它只为其他硬件构件提供服务，而不接受服务。中间构件既有需求接口又有供给接口，它不仅能够接受其他构件提供的服务，而且也能够为其他构件提供服务。终端构件只有需求接口，它只接受其他构件提供的服务。设计核心构件时，需考虑的问题是：“核心构件能为其他构件提供哪些信号？”。设计中间构件时，需考虑的问题是：“中间构件需要接受哪些信号，以及提供哪些信号？”。设计终端构件时，需考虑的问题是：“终端构件需要什么信号才能工作？”。

嵌入式底层驱动构件是直接面向硬件操作的程序代码及使用说明。规范的底层驱动构件由头文件（.h）及源程序文件（.c）文件构成。头文件（.h）是底层驱动构件简明且完备的使用说明，即在不查看源程序文件情况下，就能够完全使用该构件进行上一层程序的开发，这也是设计底层驱动构件最值得遵循的原则。

在设计实现驱动构件的源程序文件时，需要合理设计外接口函数与内部函数。外接口函数，供上层应用程序调用，其头注释需完整表述函数名、函数功能、入口参数、函数返回值、使用说明、函数适用范围等信息，以增强程序的可读性。在具体代码实现时，严格禁止使用全局变量。

在嵌入式硬件原理图设计中，要充分利用嵌入式硬件进行复用设计；在嵌入式软件编程时，涉及与硬件直接打交道时，应尽可能复用底层驱动构件。若无可复用的底层驱动构件，应该按照基本规范设计驱动构件，然后再进行应用程序开发。

3.14 利用串行通信的接收中断阐述完整程序要素

MCU的串口通信模块UART，在硬件上，一般只需要三根线，分别称为发送线（TxD）、接收线（RxD）和地线（GND），在通信表现形式上，属于单字节通信，是嵌入式开发中重要的打桩调试手段。串行通信数据格式可简要表述为：发送器通过发送一个“0”表示一个字节传输的开始，随后一般是一个字节的8位数据，最后，发送器停止位“1”，表示一个字节传送结束。若继续发送下一字节，则重新发送开始位，开始一个新的字节传送。若不发送新的字节，则维持“1”的状态，使发送数据线处于空闲。从开始位到停止位结束的时间间隔称为一字节帧。串行通信的速度用波特率表征，其含义是每秒内传送的位数，单位是：位/秒，记为bps。

图6-1 串行通信数据格式

（书籍第112页）

首先应该掌握使用UART构件进行串口通信的编程，正确理解与使用初始化（uart_init）、发送单个字节（uart_send1）、发送N个字节（uart_sendN）、发送字符串（uart_send_string）、接收单个字节（uart_re1）、使能串口接收中断（uart_enable_re_int）等函数。对于UART构件的制作，有一定难度，可以根据自己的学习情况确定掌握深度，基本要求是在了解寄存器基础上，理解利用直接地址操作的串口发送打通程序，后续进行构件制作。这里可以看出，使用构件与制作构件的难度差异，这是软件编程的社会分工的重要分界点，利用GEC概念，把这两个过程分割开来，做构件与用构件属于不同工作范畴。

任何一个计算机程序原则上可以理解为两条运行线，一条为无限循环线，另一条为中断线。要对一个中断进行编程，要求掌握以下几个环节：（1）中断源、中断IRQ号、中断向量号；（2）产生中断的条件；（3）中断初始化；（4）中断处理程序的存放位置及编写中断处理程序。读者可通过串口通信接收中断体会这个过程。

书籍1～6章囊括了学习一个微控制器入门环节的完整要素。

3.15 第7章

1. 关于基本定时功能。从编程角度，基本定时功能的编程步骤主要有：第一步，是给出定时中断的时间间隔，一般以毫秒为单位，在主程序外设初始化阶段给出；第二步，是确认对应的中断处理程序名，与中断向量号相对应，为了增强可移植性，一般需在user.h对其重新宏定义；第三步，使用user.h中重新宏定义的中断处理程序名在isr.h中进行中断处理程序功能的编程实现。

从构件设计角度，基本定时功能的要点有：时钟源、计数周期、溢出时间、溢出中断。ARM Cortex-M4处理器内核中的Systick定时器是一个24位计数器，RTC模块是具有日历功能的16位计数器，Timer模块内还有几个仅做为基本计时的16位计数器。

2. 关于PWM、输入捕捉与输出比较功能。目前大部分MCU内部均有PWM、输入捕捉与输出比较功能，因其需要定时器配合工作，所以这些电路包含在定时器中。PWM信号是一个高/低电平重复交替的输出信号，其分辨率由时钟源周期决定，编程可以改变其周期、占空比、极性、对齐方式等技术指标，主要用于电机控制。输入捕捉是用来监测外部开关量输入信号变化的时刻，这个时刻是定时器工作基础上的更精细时刻，主要用于测量脉冲信号的周期与波形。输出比较是用程序的方法在规定的较精确时刻输出需要的电平，实现对外部电路的控制，主要用于产生一定间隔的脉冲。

3.16 第8章

1．关于Flash存储器在线编程。Flash存储器可在线编程可以基本取代电可擦除可编程只读存储器，用于保存运行过程中希望失电后不丢失的数据。STM32L431芯片内部有256KB的Flash存储器，其起始地址为0x0800_0000，按照扇区进行组织，每个扇区大小为2KB，以扇区为基本擦除单位，Flash构件封装了初始化、擦除、写入等基本接口函数。

2．关于ADC模块。ADC将模拟量转换为数字量，以便计算机可以通过这个数字量间接对应实际模拟量进行运行与处理。与A/D转换编程直接相关的技术指标主要有：转换精度、是单端输入还是差分输入等。STM32L431芯片内部含有一个12位ADC模块，共有19个单端输入通道，16个可组合的差分输入通道。

3．关于DAC模块。DAC将数字量转换为模拟量，以便计算机可以通过数字量控制实际的诸如音量大小等模拟量输出。与编程相关的DAC主要技术指标是分辨率，一般情况下，分辨率使用DAC位数来表示。STM32L431芯片内部含有一个12位DAC模块，共有2个输出通道。

3.17 第9章

1．关于SPI模块。SPI是四线制主从设备双工同步通信，4条线分别是：串行时钟线 SCK、主机输入/从机输出数据线 MISO、主机输出/从机输入数据线MOSI和从机选择线。编程时注意主机和从机必须使用同样的时钟极性与时钟相位，才能正常通信。时钟极性与时钟相位的设置要做到：确保发送数据在一周期开始的时刻上线，接收方在1/2周期的时刻从线上取数，这样是最稳定的通信方式。

2．关于I2C模块。I2C字面上的意思是集成电路之间，主要用于同一块电路板内集成电路模块之间的连接和数据传输。I2C采用双向2线制（SDA、SCL）串行数据传输方式，简化了硬件连接。注意二线均需接1.5～10KΩ的上拉电阻，接入总线的设备必须共地。一般情况下，I2C设备使用7位地址，地址0000000一般用于发出总线广播。

3．关于TSC模块。STML431RCT6的TSC是一种电容转移计数传感器，当有感应物与初始化为TSC通道的引脚相连的电极时，通过观察TSC模块记录的采样电容的达到阈值所经历的电荷转移周期数的变化，就可以判断是否触摸，可用于触摸键盘的设计。还提供了一种GPIO模拟TSC的方法，当GPIO引脚处于既无上拉也无下拉的状态时，通过触摸引脚，会发生一定变化，可以判断触摸，用于无触摸功能的MCU实现简单的触摸感应输入。

3.18 第10章

1．关于CAN总线 。CAN总线常用于汽车电子中，它属于半双工通信。制作与测试芯片的CAN构件时，为了保证给未知软件提供可信的硬件环境，可以使用不带收发器芯片及隔离电路的连接方式。实际使用时是收发器芯片及隔离电路的差分线路。理解CAN总线通信原理有一定深度，可以从理论到应用，再从应用到理论的反复中不断理解。

2．关于DMA 。DMA是可以使数据不经过CPU直接在存储器与I/O设备之间、不同存储器之间进行传输的一种方式，一般用于比较深入的编程中，可以节约CPU的占用时间。例如，要把内存中的2000个字节送入串口发送出去，可以使用DAM编程方式，让DAM去做这件事，完成之后产生一个中断，CPU就知道已经传输完毕，在DMA传输期间，CPU可以做别的事情。

3．关于位带操作。常规的编程中不会使用到位带操作，只有对位操作十分频繁且对时间要求很严格的条件下才会使用到，属于高级编程范畴。由于对内存的访问通常以字或字节为单位进行，要改变内存中的一位，需要通过“读-改-写”的过程，用时长。位带操作用于解决这个问题，其实质是内存位带区的一位，对应于一个位带别名区的一个字地址，对位带别名区字地址的赋值（0或1），等同于干预了位带区的那一位。需要注意的是编程时对位带区的单元变量必须使用关键字volatile来修饰，指示C编译器不对此变量进行优化处理。

3.19 第11章

1．关于STM32L431时钟系统。STM32L431的时钟系统包括六个时钟源，其中HSI16、MSI、HSE、PLL可以提供系统时钟，LSI、LSE、HIS可以驱动部分外设，PLL，PLLSAI1都有三个独立的输出，对于每个时钟源来说，在未使用时都可单独打开或关闭，以降低功率。本书例程使用内部MSI 时钟将STM32L431的系统频率初始化为48MHz，作为MCU运行的总线工作时钟。

2．关于复位模块与看门狗模块。复位模块可以在出现异常时使得芯片回复到最初已知状态，以对系统进行保护，需要了解不同的复位源以及各个复位发生的条件。关于看门狗模块，在应用系统研发阶段，一般先关闭看门狗功能，避免不必要复位的发生。只有在系统开发完成，调试正常准备投入使用时，才开启看门狗的功能，规范的使用看门狗可以有效的防止程序跑飞。

3．关于电源控制模块与CRC校验模块。STM32L431支持多种低功率模式，用户可以选择具体的低功率模式，以在低功率、短启动时间和可用唤醒源之间寻求最佳平衡。CRC校验模块提供了一种硬件CRC校验计算方法。

3.20 第12章

本章作为扩展及讲座性内容，给出了嵌入式系统稳定性问题、外接传感器及执行部件的编程方法、实时操作系统的应用、嵌入式人工智能的应用、NB-IoT的应用，还给出了4G、Cat1、Wi-Fi及WSN的应用简介等，这些内容来自实际应用开发的基本概括。目的是了解嵌入式系统实际应用的相关知识及有关领域，为实际应用提供借鉴。

四、MCU实践源代码导引

电子资源下载：

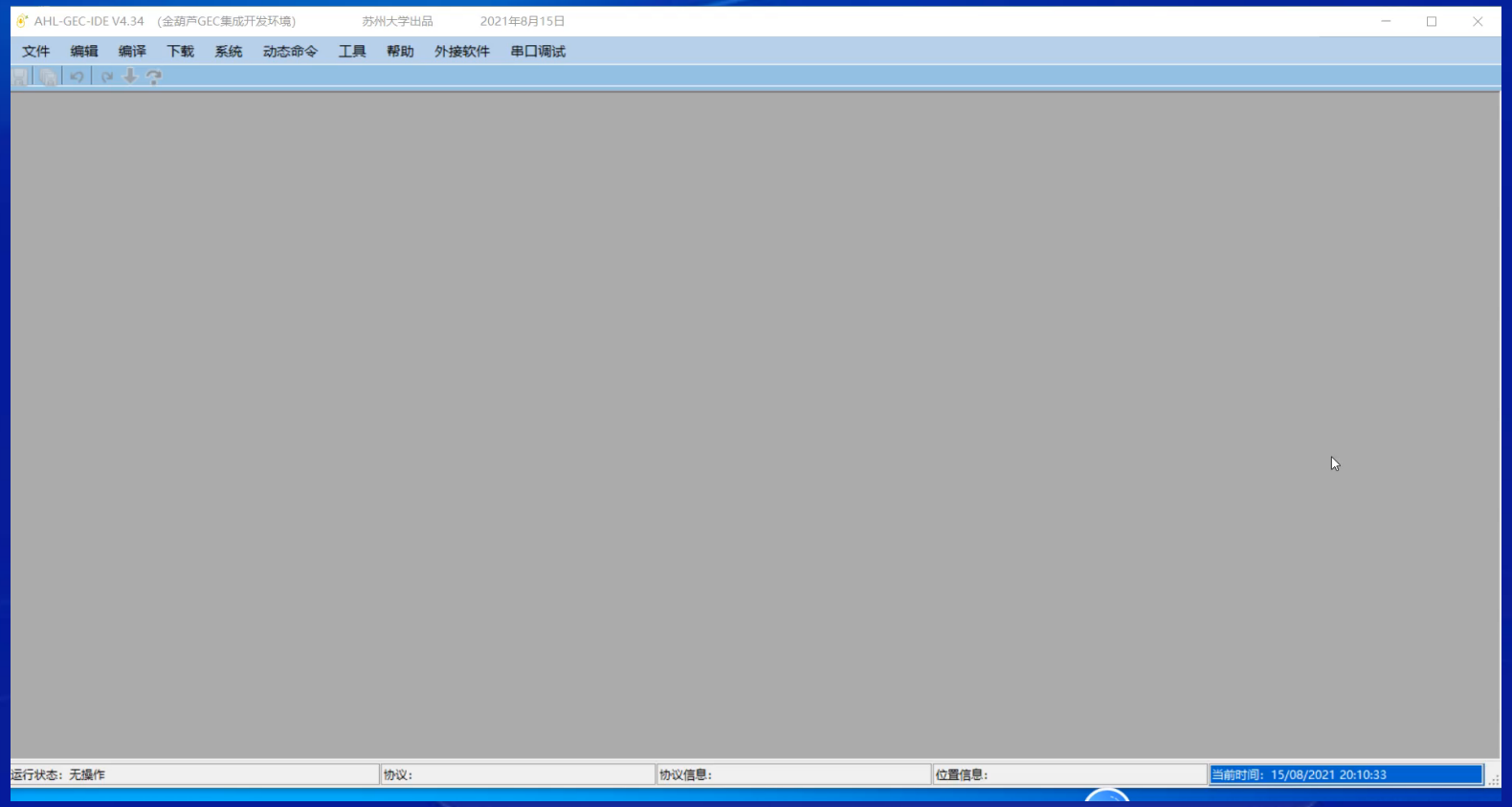
http://sumcu.suda.edu.cn/qrss6b/list.htm

网上教学资源：AHL-MCU-6

文件夹		内容
01-Information		内核及芯片文档
02-Document		补充阅读材料、硬件使用说明等
03-Hardware		硬件文档
04-Software	CH01	硬件测试测试程序（含MCU方及PC方程序）
	CH02	认识汇编语句生成的机器码
	CH04	直接地址方式干预发光二极管；构件方式干预发光二极管；汇编编程方式干预发光二极管
	CH06	直接地址方式串口发送数据、构件方式串口发送数据、利用串口接收中断进行数据接收
	CH07	内核Systick定时器；带日历功能实时时钟RTC；Timer模块基本定时器；脉宽调制PWM、输入捕捉、输出比较；PC方配套测试程序
	CH08	Flash、ADC、DAC；PC方配套测试程序
	CH09	SPI、I2C、TSC；GPIO模拟TSC
	CH10	CAN、DMA、位带操作
	CH11	系统时钟程序的注解、看门狗、CRC
	CH12\RTOS	实时操作系统的延时函数、事件、消息队列、信号量、互斥量
	CH12\EORS	嵌入式人工智能：物体认知系统
	CH12\NB-IoT	窄带物联网NB-IoT
	CH12\IoT	物联网的4G、Cat1、Wi-Fi、WSN通信方式
05-Tool		AHL-STM32L431板载TTL-USB芯片芯片驱动程序
06-Other		C#2019串口测试程序；C#快速应用指南下载导引

该网上教学资源适时更新，下载路径：百度搜索“苏州大学嵌入式学习社区”官网→“金葫芦专区”→“嵌入式书 6 版”

五、MCU实践操作演示


AHL-GEC-IDE V4.34 (金葫芦GEC集成开发环境) 苏州大学出品 2021年8月15日
文件 编辑 编译 下载 系统 动态命令 工具 帮助 外接软件 串口调试
运行状态：无操作
协议：
协议信息：
位置信息：
当前时间：15/08/2021 20:10:33

Thank You!