

第3章

CHAPTER 3

STM32 单片机应用基础

3.1 STM32 单片机简介

英特尔公司在 20 世纪 80 年代推出了 MCS-51 系列单片机,该系列单片机包括了很多 品种,如 8031、8051、8751、8032、8052、8752 等,其中 8051 是最典型的产品,该系列其他单片 机都是在 8051 的基础上进行功能的增、减而来的,所以人们习惯于用 8051 来称呼 MCS-51 系列单片机。英特尔公司将 MCS-51 的核心技术授权给了很多其他公司,这些公司使用以 MSC-51 架构为核心,根据自身的需求,设计出自己的 51 单片机,如 ATMEL 的 AT89C51、 国产宏晶的 STC89C51。

51 单片机在工业界和教育界产生过巨大的影响,时至今日依然经常可以见到 51 单片 机的身影。但 51 单片机是 8 位的,随着电子和计算机技术的发展,其并不能满足所有的应 用场合。与此同时英国的 ARM 公司在 20 世纪 90 年代做出了一个意义深远的决定:自己 不制 造 芯 片,只将 芯 片 的 设 计 方 案 授 权 给 其 他 公 司 来 生 产。意 法 半 导 体 公 司 (STMicroelectronics,ST 公司)正是在这种情况下推出了基于 ARMv7 架构的 32 位单片 机,即 STM32 系列单片机。STM32 系列单片机产品多样化、性价比高、开发简单,迅速占 领了中低端单片机市场,成为单片机市场一颗耀眼的明星。

STM32系列单片机除了在单片机内部集成了ARM公司提供的微处理器内核外,还在 单片机内部集成了大量的片上外设,如USART(串口)、I²C、SPI等。STM32系列单片机从 微处理器的内核上可以分为Cortex-M0、M3、M4、M7等,每个内核又根据性能和功耗分为 几个系列,如F0、F1、F2、F3、F4、F5、F6、F7等属于高性能型,L0、L1、L4等属于低功耗型。 单片机内部集成的片上外设是ST公司自己开发的电路模块,如存储器模块、ADC模块和 GPIO模块等,这些模块各个具体单片机是不同的,应用的时候要查阅单片机芯片的使用手 册(查阅不是全文阅读)。ST公司基于Arm Cortex内核的32位MCU和MPU系列产品 如图 3.1.1所示。

STM32 单片机的开发方式有以下三种:

(1) 寄存器模式:这种方式直接操作单片机内部各个电路模块对应的控制寄存器,例如,设置单片机第 H 组的 GPIO 的模式寄存器 MODER,需要编写类似"GPIOH_MODER=



• Cortex-M0+无线协处理器 ■ 集成双Arm Cortex-A7和Cortex-M4内核 ◆含Arm Cortex-M7和Cortex-M4双核产品

图 3.1.1 ST 公司基于 Arm Cortex 内核的 32 位 MCU 和 MPU 系列产品

0x00100000"这样的代码。这种开发方式需要先搞清楚单片机的内部结构和寄存器每一位的含义,早期的 MCS-51 单片机或 MSP430 单片机开发一般采用这种方式。但是,对 STM32 这种结构较为复杂的单片机,初学者如果没有 MCS-51 单片机或 MSP430 单片机开发经验,一般很难上手。

(2)标准外设库(SPL)函数模式: ARM 公司为 Cortex-M 内核提供了 Cortex 微控制器 软件接口标准(CMSIS),ST 公司又为单片机上的外设编写了驱动库函数,开发者无须过多 关注寄存器每一位的具体含义,只需搞清楚库函数的参数和作用即可编写类似"GPIOH. GPIO_MODE = GPIO_MODE_OUT"这样的代码,开发者很容易理解上面的语句是把 GPIO 的模式设置为输出,这相对于寄存器开发模式又更进一步接近人类语言,便于开发者 和程序阅读者理解。

(3) HAL/LL 库函数模式: HAL(Hardware Abstract Layer)和 LL(Low Layer)库是 ST 公司推出的另外一种驱动库,相对于 SPL 库的优势在于配置了单片机的图形化配置软件 STM32CubeMX,可以对单片机的资源和外设进行图形化配置,并针对不同的集成开发 环境(IDE)工具软件生成基于 HAL/LL 库的外设初始化程序和 IDE 程序框架,开发者只需 要把主要精力用于应用程序的编写,节省了大量的时间和精力,特别适合没有单片机开发经 验的初学者快速入门。

本书采用 HAL/LL 库函数模式,使用 STM32CubeMX 软件配置单片机硬件资源,采用 STM32CubeIDE 编写应用程序,采用 STM32CubeProgrammer 工具烧录单片机可执行 文件。

3.2 STM32 单片机片上资源和开发板

使用 STM32 单片机和使用其他任何一款芯片一样,首先要搞清楚芯片的主要功能、引 脚分配以及功能寄存器的配置。各种图书和网络资料从不同的角度介绍了 STM32 单片机 的结构、功能和用法,具有一定的参考价值;但出于权威性和知识产权考虑,优先从 ST 公司 官网获取资料。本书不像其他教材一样详细介绍所有片上资源,而是在用到时逐步讲解,建议读者购买一块 STM32 单片机开发板,配合本书按步骤边阅读边操作。

3.3 STM32CubeMX

无论是寄存器开发模式还是标准外设库函数开发模式,都需要掌握大量寄存器相关知识,给初学者造成一种 STM32 开发很难、门槛很高、不容易学通的感觉,很多人在开始不久就放弃了。STM32CubeMX 的出现解决了这一问题,利用 STM32CubeMX 可以轻松配置 STM32 单片机的硬件寄存器,快速上手。

3.3.1 下载和安装 STM32CubeMX

STM32CubeMX 可以到官网下载安装文件,也可以在一些开发论坛下载。 STM32CubeMX 提供三种操作系统的若干版本,如图 3.3.1 所示,一般选择对应操作系统的最新版本下载。

	产品型号 ▲	一般描述	÷	供应商 🍦	下载	÷	All versions		÷
+	STM32CubeMX-Lin	STM32Cube init code generator for Linux		ST	Get latest		选择版本	\sim	
+	STM32CubeMX-Mac	STM32Cube init code generator for macOS		ST	Get latest		选择版本	~	
+	STM32CubeMX-Win	STM32Cube init code generator for Windows		ST	Get latest		选择版本	~	

图 3.3.1 STM32CubeMX 提供三种操作系统版本

下载后双击安装文件,STM32CubeMX的安装界面如图 3.3.2 所示,第一个界面如 图 3.3.2(a)所示,单击"Next",后续如有协议需要单击同意,按提示安装直至完成,不建议 更改安装目录,不建议在安装路径中使用中文。安装的最后一步如图 3.3.2(b)所示。



(a) 安装开始界面

图 3.3.2 STM32CubeMX 的安装界面

STM32CubeMX Ins	tallation Wizard	-		×
STM32CubeMX Installa	tion done			МΧ
—	Installation has completed successfully.			
	An uninstaller program has been created in:			
life.augmented	C:\Program Files\STMicroelectronics\STM32Cube\STM32Cub	eMX\Unins	taller	
STM32 Cube				
	Generate an automatic installation	script]	
STMicroelectronics		[Ø	Done
	(b)安装结束界面			

图 3.3.2 (续)

启动 STM32CubeMX 后界面如图 3.3.3 所示, 左侧为打开已有工程, 中间为新建工程, 右侧为检查更新软件和安装嵌入式软件包(MCU 驱动库), 单击"INSTALL/REMOVE"后 界面如图 3.3.4 所示, 选择单片机对应系列的驱动库安装即可。界面最上面的"File" 菜单 栏提供与工程文件相关的操作, "Window"和"Help" 菜单栏提供一些窗口和辅助操作。

STM32CubeMX Untitled					-	o x
STM32 CubeMX	File	Window	Help	œ	f 🖸 🏏 🔆	< 57
Home					Secol. Sec	
Existing Projects		New Project		Manage s	oftware installation	IS
Open Existing Projects	E.	I need to :		Check	for STM32CubeMX a	and
					CHECK FOR UPDATES	1
		Start My projec	t from MCU			
		ACCESS TO N	ICU SELECTOR	Install	or remove embedded	S0
		Start Mu projon	from CT Roard		INSTALL / REMOVE	
		ACCESS TO BE	ARD SELECTOR			
		Start My project	t from Example			MB b
		ACCESS TO EXA	MPLE SELECTOR	1	D- MILL	

图 3.3.3 启动 STM32CubeMX 后的界面

3.3.2 新建一个 STM32CubeMX 项目

新建 STM32CubeMX 项目有三种方式:一是新建基于 STM32 系列 MCU 的项目;二 是新建基于 ST 公司开发板的项目;三是新建基于其他公司 MCU 迁移到 STM32 系列

MX En	bedded Software Packages Man	ager				×
-	STM32Cube MCU Packa	ges and embedded soft	ware packs releas	es		+ -
	Releases Information was	ast refreshed less than or	ne hour ago.			
577 S	M32Cube MCU Packages	STMicroelectronics R	RoweBots SEGGE	ER emotas	portGmbH wol	fSSL
	Description			Installed	Version Availat	ble Version
•	STM32F3					1
•	STM32F4					
	STM32Cube MCU Package for S	STM32F4 Series (Size : 7	73.0 MB)		1	.27.1
Deta	S					_
Fro	n Local From Url		Refre	sh Install	Remove	Close

图 3.3.4 单击"INSTALL/REMOVE"后界面

MCU的项目。本节新建基于 STM32 系列 MCU 的项目,在 STM32CubeMX 的启动界面中的"New Project"下面选择"ACCESS TO MCU SELECTOR"按钮,进入 MCU 型号选择界面,如图 3.3.5 所示,在"Commercial Part Number"后面输入自己的 MCU 型号,筛选到 MCU 型号后选择,然后单击图 3.3.6 右上角的"Start Project",弹出如图 3.3.7 所示的 MCU 配置界面。

New Project from a MC	U/MPU						×
MCU/MPU Selector	Board Selector Example Sele	ctor Cross Selector					
MCU/MPU Filters							
* 6	ඩ බො ව	Featu Block	Diagr Docs & Resi	ourc CAD Resourc	Datash	🗋 B 🕞 Start Proj	
Commercial Part Number	STM32F429 ~ STM32F429AGH6	*	0				
۹ 🗌	STM32F429AGH6TR STM32F429AIH6		STM32	、 雷人			
PRODUCT INFO	STM32F429BET6 STM32F429BGT6		a.				
Segment	STM32F429BIT6		1755	See Al			
Series	STM32F429BIT7 STM32F429IEH6		a com				
Line	STM32F429IET6		STM32U5 ultr	a-low-power MCU se	eries 57		
Marketing Status	STM32F429IGH6 STM32F429IGT6		with compres	nensive STM32Cube	ecosystem		
Price	STM32F429IIH6						
Package	STM32F429IIH6TR	MCUs/MPUs List: 37 items		Display similar item	S.	合 Expor	t
Core	STM32F429IIT61	Commercial Part No Commercial Part No	Reference Market	in X Unit Price for X Bo	X Package >	Flash X RAM XI/OX Frequ.	×
Conrocassor	STM32F429NEH6	STM32F429AGH6	STM32F429 Active	9.0364	UFBGA 169 7x7x0.6 P	1024 K 256 KB 130 180 M	
Coprocessor	STM32F429NGH6	- STM32F425AGHOTK	STM32F425 Active	10 2262	UEBCA 169 7x7x0.6 P	2049 k 255 kD 130 100 M	
	STM32F429NIH6	☆ STM32F425AIF10	STM32F429 Active	9.9977	UPBGA 109 7X7X0.0 P	512 kB 256 kB 169 180 M	
MEMORY	STM32F429NIH6J	☆ STM32F429BGT6	STM32F429 Active	10.0775	LOFP 208 28x28x1.4	1024 k 256 kB 168 180 M	
Elach From 512 to	STM32F429NIH6TR		STM32F429 Active	11 2674	LOFP 208 28x28x1.4	2048 k 256 kB 168 180 M	1
• • • • • • • • • • • • • • • • • • •	• 2040 (nD)103)		STM32F429 Active	12 0562	LOFP 208 28x28x1 4	2048 k 256 kB 168 180 M	
512	2048	☆ STM32F429IEH6	STM32F429 Active	8.382	UFBGA 176+25 10x10	512 kB 256 kB 140 180 M	1
EEPROM = 0 (Byte	es)	☆ STM32F429IET6	STM32F429 Active	8.2928	LQFP 176 24x24x1.4	512 kB 256 kB 140 180 M	

图 3.3.5 MCU 型号选择界面

U/MPU Selector Board Selector	Example Selector	r Cross Selector		
U/MPU Filters				
★ 🖸 🛱	J	Featu Block Diagr	Docs & Resourc CAD Resourc	🖺 Datash 📑 B 🕞 Start Proj
Commercial Part Number	GT6 V	☆ STM32F4 Series		
۵	→ + -	STM32F407IGT6	High-performance foundation line FPU, 1 Mbyte of Flash memory, 1 Ethernet, FSMC	e, Arm Cortex-M4 core with DSP and 68 MHz CPU, ART Accelerator,
PRODUCT INFO	~	ACTIVE	Unit Price for 10kU (US\$): 8.7517	
Segment	>	Product is in mass		LQFP 176 24x24x1.4 mm
Series	>			
Line	>	The STM32F405xx and S operating at a frequency of	TM32F407xx family is based on the high-perf	ormance Arm [®] Cortex [®] -M4 32-bit RISC core
				A FEE BARDER FRANCE COURTER FEED
Marketing Status	>	uchish sunnada all Arm sin	als presision data pressesing instructions on	ad data timos. It also implemente a full set af
Marketing Status Price	>	uhish sunnarta all Arm ais		d data kunon it alan implemente o full ont of
Marketing Status Price Package	> > >	MCUs/MPUs List: 1 item	- Display similar items	d data hunon Histori implemente a full cat of
Marketing Status Price Package Core	> > > >	MCUs/MPUs List: 1 item	Display similar items Reference MarketinX Unit Price forX 80X M32F4070 Artise 8 7477	21 Export Package × Finah × RAM × 102 kF 107 188
Marketing Status Price Package Core Coprocessor	> > > >	MCUs/MPUs List: 1 item Commercial Part No ☆		the formation of the state of
Marketing Status Price Package Core Coprocessor WEMORY	> > > > >	MCUs/MPUs List: 1 item COmmercial Part No ↑	Display similar items Reference MarketinX Unit Price forX M32F407IActive 8.7517	the standard point on it (۲۰۰۶ stands processing) کرد را Expon Package Rash X Rash X RAM Xuo XFree LQFP 176 24x24x1.4 1024 k 192 kB 140 168 f
Marketing Status Price Package Core Coprocessor MEMORY Flash = 1024 (kBytes) 1024	> > > >	Lukisk normande all tem Sin MCUs/MPUs List: 1 item Commercial Part No ^{+−} STM32F407/GT6 ST	+ Display similar items Attenne Unit Arte for a location Materian Marketina V Unit Proce for a location M32F407I Active 8.7517	한 4 44 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

图 3.3.6 选择 MCU 后的界面

STM32Cub	eMX Untitled: STM32F407IGTx									- (n x
STM32	File	Window	Help				1	f	y	\star	577
Home >	STM32F407IGTx Vintitled - Pino	ut & Configuration >					GE	NERATE CO	DDE		
	Pinout & Configuration	Clock Co	nfiguration		Project M	anager		T	ools		
		✓ Software I	Packs	✓ Pinout							
Q	(Q)			Dinout view	In System v	ew					
Categories	H-2L										- 1
System C	ore >										- 1
Analog	>		_0								- 1
Timers	>										- 1
Connectivi	ty >										- 1
	- <u>-</u>										- 1
Multimedia	a		111								- 1
Security	>		1111								- 1
Computing	g >										- 1
Middlewar	e >		5790 8190 2193 8191								- 1
			273 276 276	STM3	F407IGTx						- 1
				LG	FP176						- 1
			=			²					
											- 1
		۲	r 9	D> <0	Dall 1	= 0		1			- 1
		Q	LJ Q		List		~	1			

图 3.3.7 MCU 配置界面

MCU 配置界面包括四部分:引脚与配置界面"Pinout & Configuration",配置 MCU 的 系统内核、外设和引脚;时钟配置界面"Clock Configuration",配置 MCU 的时钟信号频率; 项目管理界面"Project Manager",对项目进行设置;工具界面"Tools",对 MCU 功耗等性 能进行分析。

3.3.3 STM32CubeMX 项目设置示例

本节通过点亮发光二极管的示例展示 STM32CubeMX 项目配置方法。发光二极管电路如图 3.3.8 所示,发光二极管的阴极分别与单片机的 PH10、PH11 和 PH12 引脚相连。

1. Pinout & Configuration 设置

Pinout & Configuration 设置区域包含系统核心 (System Core)、模拟电路(Analog)、定时器 (Timers)、互联(Connectivity)、多媒体(Multimedia) 等内核和片上外设的设置区域,单击图 3.3.7 左侧 ">"符号后会展开若干模块的名称,单击每个模块名称后就出现了该模块的详细设置,如图 3.3.9 所示。



STM32CubeMX 的最大功能就是将以往文字化的模块配置方式图形化,把"填空题"改成 "选择题",极大地降低了初学者的学习难度。

File Window Help Home STM32F429IGTx Untitled - Pinout & Configuration Pinout & Configuration Clock Configuration Pinout & Configuration > Software Packs System Care Categories A->Z DMA GPIO MVC RCC SYS WWDG Analog Times Connectivity Multimedia Security Computing Warning: This peripheral has no parameters to be configured. 	MX STM32CubeMX Untitle	ed*: STM	32F429IGTx			
Home STM32F429IGTx Untitled - Pinout & Configuration Pinout & Configuration Clock Configuration Q SYS Mode and Configuration Q Other Packs System Core Mode Debug Disable > System Core System Wake-Up DMA System Wake-Up Timebase Source SysTick WWDG SysTick Analog > Timers > Security > Connectivity > Multimedia > Security > Computing >	STM32		File	Window	Help	
Pinout & Configuration Clock Configuration Q Image: System Core SYS Mode and Configuration System Core Image: System Wake-Up DMA Opeoug Disable Image: System Wake-Up Image: System Wake Image: System Wake-Up Image: System Wake-Up Image: System Wake Image: System Wake-Up Image: System Wake-Up Image: System Wake Image: System Wake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Wake Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Vake-Up Image: System Va	Home 🔰 STM32F	=429IGT	🗙 🔪 Untitled - Pi	nout & Configuration >		
Software Packs SYS Mode and Configuration Categories A->2 System Core System Core Debug Disable Origon Debug Disable System Wake-Up Timebase Source SysTick Analog Analog Analog Analog Analog SYS MwDG Multimedia Security Security Connectivity A Connectivity A Connectivity Autimedia Analog Analog		Pinou	t & Configuration		Cloc	< Configuration
Q SYS Mode and Configuration Categories A->Z System Core Oblag DMA Oblag OPIO System Wake-Up Timebase Source SysTick SYS WWDG Analog Timers Connectivity Security Security Commuting Commuting Warning: This peripheral has no parameters to be configured.						✓ Software Packs
Categories A>Z Mode System Core Debug Disable System Wake-Up Timebase Source SysTick MVDG NVIC RCC SYS WWDG Analog Timers Connectivity Multimedia Security Security Computing Warning: This peripheral has no parameters to be configured. 	۹ ~	0		SYS Mode and	Configuration	
System Core DMA Debug Disable OMA GPIO Impose Source SysTick Impose Source System Wake-Up Timebase Source SysTick Analog > Analog > Connectivity > Configuration Security > Configuration Multimedia > Security > Mustime Source > Mustime Source > Security > Mustime Source > Mustime Source > Security > Mustime Source >	Categories A->Z			Mod	e	K
Image: System Wake-Up DMA GPIO WDG NVIC RCC Image: System Wake-Up Timebase Source System Wake-Up Multimedia Security	System Core	~	Debug Disable			~
DMA GPIO GPIO WDG NVIC RCC SYS WWDG Analog > Timers > Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.		_	System Wake-U	р		
GPIO WDG NVIC RCC ✓SYS WWDG	DMA	- 1	Timebase Source S	ysTick		\sim
MUIC RCC Image: System WWDG Analog Image: Timers Image: Timers <	GPIO	- 1				
RCC VWDG Analog Timers Connectivity Multimedia Security Security Warning: This peripheral has no parameters to be configured.	NVIC	- 1				
WWDG Analog Timers Connectivity Connectivity Multimedia Security Security Computing Warning: This peripheral has no parameters to be configured.	RCC	- 1				
Analog > Timers > Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.	WWDG	- 1				
Analog > Timers > Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.		- 1				
Timers > Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.	Analog	>				
Timers > Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.		_				
Connectivity > Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.	Timers	<u> </u>				
Multimedia > Security > Computing > Warning: This peripheral has no parameters to be configured.	Connectivity	>				
Security Configuration Computing > Marning: This peripheral has no parameters to be configured.	Multimedia	>				
Security > Configuration Computing > Computing >						
Computing > A Warning: This peripheral has no parameters to be configured.	Security	>		Configu	ration	
Comparing	Computing	>	w	arning: This peripheral has	no parameters to be conf	igured.
	MC-Lill.					
Middleware /	Middleware	<u> </u>				

图 3.3.9 Pinout & Configuration 设置区域

在利用 STM32CubeMX 进行项目开发时,SYS 和 RCC 是任何一个项目都必须要设置的模块。SYS 模块设置如图 3.3.9 所示,主要设置调试接口和时钟源,本书采用串口调试,所以将图 3.3.9 中的 Debug 方式由"Disable"改为"Serial Wire",其余设置不变。RCC 模块 主要设置系统时钟源,设置界面如图 3.3.10 所示,本项目计划采用单片机的外部高速时钟 源,所以将图 3.3.10 所示 High Speed Clock(HSE)由"Disable"改为"Crystal/Ceramic Resonator",其余设置不变。

本例中用到了 GPIO 引脚,所以还要设置 GPIO 模块。单击图 3.3.9 中 System Core

MX STM32CubeMX Do	cuments.io	c: STM32F429IGTx				
STM32		File	Wind	low	Help	
Home 🔰 STM3:	2F429IGT>	Cocuments.iod	: - Pinout & C	onfiguration	\rangle	
Pinou	t & Coni	figuration		Clock Conf	iguration	
			````	<ul> <li>Software Pa</li> </ul>	acks	✓ Pine
Q	~ ۞		RCC Mode	and Configuratio	n	
Categories A->Z				Mode		
System Core	~	High Speed Clock (HSE	E) Disable			~
	- 1	Low Speed Clock (LSE	) Disable			~
DMA	- 1	Master Clock Output	ut 1			
GPIO	- 1	Master Clock Output	ut 2			
NVIC		Audio Clock Input (	I2S_CKIN)			- 1
RCC ✔ SYS WWDG	-1					
Analog	>					
Timers	, I		Co	nfiguration		
Connectivity	>	Reset Configuration	A Llear Canet	anto 🖉 NV/C	Sattings	
Multimedia	>	Configure the below parameter	eters :		Oottings	
Security	>	Q Search (Ctrl+F)	$\bigcirc$			0
Computing	>	<ul> <li>System Parameters</li> <li>VDD voltage (V)</li> </ul>	1	3.3 V		- H
Jonipulity	<u> </u>	Instruction Cach	ne	Enabled		U
Middleware	>	Prefetch Buffer		Enabled		
		Flash Latency(V	VS)	0 WS (1 CPU	cycle)	
		✓ RCC Parameters		a a testara Anna (1994-199		I

图 3.3.10 RCC 模块设置界面

下面的 GPIO 模块,可以看到有些 GPIO 端口已经被设置,这是因为 SYS 设置中采用串口 调试占用了 PA13 和 PA14, RCC 设置中使用外部高速时钟源占用了 PH0 和 PH1, 如 图 3.3.11 所示。

本例中三个发光二极管连接到了 PH10、PH11 和 PH12,所以还需要设置这三个端口。 在图 3.3.11 中单击右下方芯片的 PH10 引脚,在弹出的选项中选择"GPIO_Output"选项, 即设置好了 PH10,如图 3.3.12 所示。按照同样的方法将 PH11 和 PH12 设置为输出模式。

在图 3.3.12 中间的"GPIO Mode and Configuration"设置区域将引脚 PH10、PH11 和 PH12 的别名设置为 LED0、LED1 和 LED2,设置完成后该区域如图 3.3.13 所示。这样设置的好处是:以后在单片机的应用程序设计中可通过对 LED0、LED1 和 LED2 赋值而改变引脚 PH10、PH11 和 PH12 的电平,程序可读性更高。

#### 2. Clock Configuration 设置

Clock Configuration 设置主要是设置单片机系统的时钟,在图 3.3.12 上单击"Clock Configuration"后出现如图 3.3.14 所示界面,首先在 HSE 模块左边的"input frequency"处输入石英晶体的实际频率,笔者手头的开发板采用 25MHz 晶体振荡器,所以输入"25"。本



44 📲 电子系统设计与实习





图 3.3.12 设置 PH10 为"GPIO_Output"选项

例中只是演示 GPIO 的用法,对时钟频率无特殊要求,可在 HCLK(MHz)范围内(180MHz) 设置 150MHz。

	GPIO Mode and Configuration									
	Configuration									
Group By Peripherals										
S GPIO S RC	SPI0 SPI0 SYS									
Search Signals					C Show or	h Madifi	od Dino			
Search (Ctri+F)						ily wouli	eu rins			
Pin Name 🕈	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximu	. User	.Modif			
PH10	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LED0	$\checkmark$			
PH11	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LED1	~			
PH12	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LED2	V			







# 3.3.4 STM32CubeMX 项目生成代码

在 STM32CubeMX 项目设置的最后一步为生成项目代码,首先单击图 3.3.7 上侧中间的"Project Manager"菜单,出现如图 3.3.15 所示界面,单击"Project"菜单,设置项目名称和 项目保存的位置。不要将项目保存在有中文名的文件路径下,项目名称也不要取中文名。本书在 D 盘里新建了一个 STM32Cube 文件夹,将项目保存在 D:\STM32Cube 里,设置项 目名 称 为 MXProject_LED,并 将 应 用 软 件 开 发 工 具 "Toolchain/IDE"设 置 为 "STM32CubeIDE"(因为下一步计划采用 STM32CubeIDE 作为开发工具),如图 3.3.15 所示。

# 46 利 电子系统设计与实习

STM32CubeMX Untitled*: STM32F429IGTx				
STM32	e Window	Help		
Home 🔪 STM32F429IGTx 🔪 U	ntitled - Project Manager >			
Pinout & Conf	iguration	Clock Configuration		
Project Code Generator	Project Settings Project Name Project Location Application Structure Toolchain Folder Location Toolchain / IDE	MXProject_LED D:\STM32Cube Advanced D:\STM32Cube\MXProject_LED\ STM32CubeIDE		

图 3.3.15 设置项目保存地址和名称

设置完项目保存地址和名称后单击"Code Generator"菜单,勾选"Generate peripheral initialization as a pair of '. c/. h' files per peripheral"前面的小方框,为生成代码做准备,如图 3.3.16 所示,这样做是为了让 STM32CubeMX 生成相关程序框架和包含". c"和". h"的程序代码。

IIX STM32CubeMX Untitled*: STM32F429IGTx								
STM32	File	Window	Help					
Home > STM32F429IGT>	🖒 🔪 Untitled - Proj	ect Manager >						
Pinout & Configuration Clock Configuration								
Project	<ul> <li>STM32Cube I</li> <li>Copy all u:</li> <li>Copy only</li> <li>Add neces</li> </ul>	WCU packages and embedd sed libraries into the project the necessary library files asary library files as reference	ied software packs folder :e in the toolchain project configuration file					
Code Generator	Generated file ☑ Generate p	eripheral initialization as a	pair of '.c/.h' files per peripheral					
Advanced Settings	☐ Backup pr	eviously generated files whe r Code when re-generating viously generated files when	n re-generating					



最后单击图 3.3.7 右上角的"GENERATE CODE"菜单生成项目代码,此时 STM32CubeMX 已经给应用软件源程序做好了一个框架,并且"编写"好了系统初始化的代码,下一步就可以在应用软件开发 IDE 中进一步添加代码。打开 D:\STM32Cube 文件夹, 可以看到文件夹里多了一个名为 MXProject_LED 的文件夹,文件夹里的文件和文件夹如 图 3.3.17 所示。

MXProject_LED				P	
⊕ 新建 ~			以排序、 三 直看、		
$\leftarrow \rightarrow \checkmark \uparrow$		> 此电脑 > 新加卷 (D:) > STM32Cube >	MXProject_LED		
✓ ▲ 主文件夹			修改日期	类型	大小
■ 桌面	*	Core	2022/10/23 22:25	文件夹	
↓下載		Trivers	2022/10/23 22:25	文件夹	
		^{IDE} .cproject	2022/10/23 22:25	CPROJECT 文件	25 KB
	~	.mxproject	2022/10/23 22:25	MXPROJECT 文件	8 KB
	*	IDE .project	2022/10/23 22:25	PROJECT 文件	2 KB
CubeProject		MX MXProject_LED	2022/10/23 22:25	STM32CubeMX	5 KB
■ 发表的论文		STM32F429IGTX_FLASH.Id	2022/10/23 22:25	LD 文件	6 KB
▶ 视频		STM32F429IGTX RAM.ld	2022/10/23 22:25	LD 文件	6 KB
1 音乐			12-1 1351		

图 3.3.17 MXProject_LED 文件夹的界面

# 3.4 STM32CubeIDE

利用 STM32CubeMX 只是对 STM32 单片机的硬件进行了初始化,并且做了一个源程序的框架,要完成软件的全部功能还需要编写应用程序,这就要使用源程序的编辑和编译集成开发环境。STM32 单片机的集成开发环境有很多,如 IAR、Keil 等,本书采用 STM32CubeIDE 开发环境。

### 3.4.1 下载和安装 STM32CubeIDE

ST 公司官方提供 Linux、macOS 和 Windows 等几种版本的 STM32CubeIDE,本书以 Windows 版本为例介绍。下载、解压后以管理员身份运行即可安装。安装过程中若提示错 误可能是计算机用户名设置中文名导致环境变量中有中文,修改环境变量即可正常安装。

安装完成后,双击桌面生成的图标启动 STM32CubeIDE。出现如图 3.4.1 所示界面,显示要选择一个文件目录设置为工作空间。更改 STM32CubeIDE 的工作空间目录为 D:\STM32Cube,与 3.3 节中 STM32CubeMX 项目相同。

IDE STM32CubeIDE Launcher	×
Select a directory as workspace	
STM32CubeIDE uses the workspace directory to store its preferences and development artifaction of the store o	cts.
Workspace: ⁹ C:\Users\#22\STM32CubeIDE\workspace_1.10.1 V Bro	wse
Use this as the default and do not ask again	
Launch	Cancel

图 3.4.1 STM32CubeIDE 更改工作空间界面

### 48 ┥ 电子系统设计与实习

更改工作空间文件夹后单击下一步,进入初始界面,如图 3.4.2 所示。界面左边是新建 或打开工程的快速入口,中间是一些关于 STM32CubeIDE 的快速链接,右边是一些 ST 公 司相关网站链接。选择"Start new project from STM32CubeMX file(.ioc)",选择 D:\ STM32Cube\MXProject_LED 文件夹中以".ioc"结尾的文件,如图 3.4.3 所示,然后单击 "Finish"。打开 STM32CubeMX 工程文件后,STM32CubeIDE 的界面如图 3.4.4 所示。

MXproject_LED - STM32	CubeIDE	×
File Edit Source Refact	or Navigate Search Project Run Window Help x	
STM32CubeID	E Home	^
Start a project	Welcome to STM32CubeIDE	Support & Community
Start new project	STM32CubeIDE         Your entry point for all information         Information         on our development tots         STM32CubeIDE         Variance development tots         STM32CubeIDE         Understand         Understand         Stracestand         Stracestand	Twitter Facebook Youtube ST Home ST Community ST Longevity Commitment ST Longevity Commitment
	Quick links	STM32CubeMon-RF     STM32CubeMon-UCPD     STM32CubeProg
Import STM32Cube example	STM32CubelDE resource portal on wiki     STM32CubelDE manuals	Application Tools
	STM32 videos	<ul> <li>eDesignSuite</li> <li>AlgoBuilder</li> <li>ST-MC-Suite</li> </ul>
	W STM32 MPU wiki	
	W STM32 MCU wiki	
<		>

图 3.4.2 STM32CubeIDE 初始界面

在图 3.4.4 的界面中出现了 STM32CubeMX 的界面,其实 STM32CubeIDE 里集成了 STM32CubeMX 功能,不过界面较小,并且功能不如单独的 STM32CubeMX 强大,建议使用独立的 STM32CubeMX 软件。

### 3.4.2 在 STM32CubeIDE 里编写主程序

单击图 3.4.4 中"Core"左侧的">"符号,可以看到更多的文件夹和文件,双击"main.h" 文件可以看到文件的详细内容,如图 3.4.5 所示。

在图 3.4.5 中可以看到定义了 LED0、LED1 和 LED2 变量,并且对应好了引脚 PH10、 PH11 和 PH12。

### 第3章 STM32单片机应用基础 ∥▶ 49

STINSLCUDEI	MX .ioc file	
File: D:\STM	32Cube\MXProject_LED\MXProject_LED.ioc	Browse
Project		
Project Name	MXProject_LED	
🖉 Use defau	It location	
Location:	D:/STM32Cube	Browse
Targeted O Executa	Binary Type	
Targeted		
O STM32	Cube O Empty	
STIVISE		





图 3.4.4 打开 STM32CubeMX 工程文件后 STM32CubeIDE 的界面

IDE STM32Cube - MXProject_LED/Core/In	nc/main.h - S	TM32CubeIDE				—		×
File Edit Source Refactor Navigate Search Project Run Window Help								
☐ ▼ 🖩 🕼   🗞 ▼ 🗞 ▼ 📾 : 🗃 ▼ 셀 ▼ 祠 ▼ ♡ ♡ ♡ ♡ ▼ ○ ▼   🗗	8° - C°   ()	• @ •   * •	0 - 9 - 10		<i>4</i> 7 ▼ [ <b>]</b>	ା ୩ ସ	: 🖻	
Project Explorer ×	* □ <b>■</b> ■ ■	IXProject_LED.ioc	: 🖸 main.c	h main.	h ×			- 8
<ul> <li>MXProject_LED</li> <li>MXProject_LED</li> <li>M project_LED</li> <li>M project_LED</li> <li>M project_LED</li> <li>M stm32f4xx_hal_conf.h</li> <li>M stm32f4xx_ti.h</li> <li>Frc</li> <li>M gpio.c</li> <li>M stm32f4xx_hal_msp.c</li> <li>M stm32f4xx_ti.c</li> <li>M stm32f4xx_ti.c</li> <li>M system_stm32f4xx_ti.c</li> <li>M system_stm32f4xx_c</li> <li>M system_stm32f429igtx.s</li> <li>M Stm32F4xx_HAL_Driver</li> <li>M XProject_LED.ioc</li> <li>STM32F429IGTX_FLASH.Id</li> <li>STM32F429IGTX_RAM.Id</li> </ul>	0       52         53       54         556       57         58       59         60       61         62       63         64       65         67       68         69       70         71       72         73       74         75       0         0       item	<pre>/* Exported void Error_ /* USER COD /* USER COD /* USER COD /* Private #define LED #define LED #define LED #define LED #define LED /* USER COD /* USER COD #ifdefcp } #endif /*  X T 5 35</pre>	functions pr Handler (void) E BEGIN EFP * E END EFP */ defines 0_Pin GPIO_PIT 0_GPIO_POrt G 1_Pin GPIO_PI 1_GPIO_POrt G 2_Pin GPIO_PIT 2_GPIO_POrt G E BEGIN Private Lusplus MAIN_H */	N_10 PIOH N_11 PIOH N_12 PIOH te defines	es */ */ ; Build Anal	 × ≞ Static S		
	Desc	ription	^	N	1emory Region	s Memory De	etails	
				F	Region	Start addre	is En	id addr
Writable	,	Smart Insert	1:1:0	1				

图 3.4.5 "main.h"文件的详细内容

单击"main.c"文件可以看到详细内容如图 3.4.6 所示(为便于展示删除了部分注释)。 "main.c"文件包含了"main.h"头文件,声明并定义了 SystemClock_Config(void)函数和 MX_GPIO_Init(void)函数,给系统时钟和部分引脚做了初始化,这些工作都是 STM32CubeMX软件替程序设计者做的,程序设计者只需要在 While 循环里添加功能代码 即可,极大地减小了程序设计者的工作量和降低了程序设计者的入门难度。

本例的设计任务是让发光二极管依次点亮和熄灭,所以在 While 循环中添加了部分代码,如图 3.4.7 所示。其中 HAL_Delay()为库函数,表示延迟以毫秒为单位的 CPU 时间。

HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin, GPIO_PIN_RESET)函数向 GPIO引脚写人数据,第一个参数为 GPIO 端口号,本例选择了 LED0 的端口号也就是 GPIOH;第二个参数为 GPIO引脚编号,本例选择了 LED0_Pin,也就是 PH10;第三个参 数为高低电平,本例选择了 GPIO_PIN_RESET,也就是向 PH10 写入了低电平,二极管点 亮。这样选择是因为在"main.h"中定义了 LED0_GPIO_Port 为 GPIOH, LED0_Pin 为 PH10,这是 STM32CubeMX 里配置后生成的。这样做的好处是:如果更换引脚,只需要在 STM32CubeMX 配置,而不需要更改主程序。

HAL_Delay(1000)将单片机空转 1000ms,发光二极管的状态保持。

### 第3章 STM32单片机应用基础 II▶ 51







图 3.4.7 While 循环中的部分代码

HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin, GPIO_PIN_SET)函数向 GPIO 引脚写入高电平,将发光二极管熄灭。

HAL_Delay(1000)将单片机时钟空转 1000ms,发光二极管的状态保持。

以上程序合起来的功能就是让发光二极管亮 1s, 灭 1s, while(1)内的代码是循环执行的,执行的效果就导致二极管不停闪烁。

### 3.4.3 在 STM32CubeIDE 里构建项目和下载程序

### 1. 选择构建方式

主程序编写完成后即可构建项目,构建的方式有 Debug 和 Release, Debug 用于在线调试, Release 用于最终发布。本例只是调试程序,所以选择 Debug,选择 Project→Properties→ C/C++Build→Configuration 菜单命令,选择构建项目方式,如图 3.4.8 所示。

pe filter text	C/C++ Build	<> ▼ <> ▼
Resource Builders C/C++ Build C/C++ General CMSIS-SVD 设置 Project References	Configuration: Debug [Active] Var Debug [Active] Release Builder Setti [All configurations]	nage Configuration
Kun/Debug Settings	Builder Builder type: External builder Use default build command Build command: make Configure build arguments in the Behavior tab.	Variables
	Makefile generation Generate Makefiles automatically Build location Build directory: \${workspace_loc:/MXProject_LED}/Debug	kefiles
	Workspace File syste	m Variables
	Restore Do	efaults Apply

图 3.4.8 选择构建项目方式

### 2. 连接计算机与开发板

STM32CubeIDE 只支持 J-LINK 和 ST-LINK 仿真器下载程序到 STM32 单片机,所以 建议读者购买开发板与 J-LINK 或 ST-LINK 仿真器。下载前先用 J-LINK 或 ST-LINK 仿 真器将开发板与计算机的 USB 连接,如图 3.4.9 所示。

### 3. 检测和更新仿真器的固件

开发板上电后计算机会自动检测仿真器并提示是否更新仿真器的固件程序,如果没有



图 3.4.9 开发板与计算机的 USB 连接方式

反应,则可以选择 Help→ ST-LINK 更新菜单命令进行检测和更新仿真器的固件,如图 3.4.10 所示。



图 3.4.10 检测和更新仿真器的固件

#### 4. 程序下载

项目构建无误、开发板连接成功后,将可执行程序下载到开发板。选择 Run→Debug 菜 单命令或者快捷工具栏上的 Debug 按钮即可下载并调试程序。本例在下载成功后若观察 到 3 个发光二极管依次闪烁(重复点亮并熄灭),则可判定实验成功。



# 3.5 STM32 单片机开发实例

本节通过中断应用实例进一步介绍利用 STM32CubeMX、STM32CubeIDE 以及 HAL 库函数开发 STM32 单片机的方法。

单片机的主程序一般是一个 while(1)循环,CPU 循环运行 while(1)循环内的语句,当 某个外设或内部条件发生异常时,才跳转到相应的异常处理程序去处理和应对,处理完毕后 继续运行 while(1)循环内的语句,这个过程称为中断。STM32 单片机根据型号不同有不同 的中断来源,具体需要查阅单片机的参考手册,但都有 GPIO 外部中断。CPU 处理中断的 过程以及中断程序的编写方法都是类似的,所以本节以 GPIO 外部中断为例进行讲解。

#### 1. 单片机外围电路

单片机 STM32F429 外围电路如图 3.5.1 所示,GPIO 端口 PA0 外接触发按键,按键悬 空时,PA0 为低电平,按键按下时,PA0 为高电平。PH10、PH11 和 PH12 外接发光二极管, 当 PH10、PH11 和 PH12 为低电平时,发光二极管点亮,否则熄灭。





#### 2. 设计目标

当按下按键时,发光二极管闪烁三下后熄灭,闪烁时间间隔1s。

#### 3. 工作原理

单片机 STM32F429 的外部中断(EXTI)有 23 个中断源,其中 EXTI0~EXTI15 对应 GPIO 引脚中断,如 EXTI0 对应 PX0(X 的取值可以为 A、B、C、D、E、F、G、H 和 I),EXTI1 对应 PX1(X 的取值可以为 A、B、C、D、E、F、G、H 和 I),其余一直到 EXTI15 类似。由此可 知,PA0 可以引起 EXTI0 中断。

GPIO中断程序设计过程(图 3.5.2)如下:①利用 GPIO 引脚初始化函数 HAL_GPIO_ Init()设置 PAO 为中断工作方式;②利用 HAL_NVIC_SetPriority()函数设置中断优先 级;③利用 HAL_NVIC_EnableIRQ()函数启用中断;④利用 EXTIO_IRQHandler()函数 中的 HAL_GPIO_EXTI__IRQHandler()设置 EXTO 的中断服务例程(CPU 参数入栈、清 除中断标志、跳转到中断服务函数和 CPU 参数出栈等一系列 CPU 的操作,本例的设计者 不用关心);⑤编写中断服务函数 HAL_GPIO_EXTI_Callback()(CPU 检测到中断发生后 跳转到中断服务函数 HAL_GPIO_EXTI_Callback()让发光二极管闪烁)。



#### 图 3.5.2 中断程序设计过程

### 4. 设计实现

图 3.5.2 中 HAL_GPIO_Init()、HAL_NVIC_SetPriority()、HAL_NVIC_EnableIRQ()、 EXTI0_IRQHandler()利用 STM32CubeMX 设置生成,流程如下:

(1) 在 D 盘新建 MX_Project_inter 文件夹。

(2)在STM32CubeMX中创建项目,保存名称为MXProject_inter.ioc,路径为新建的MX_Project_inter文件夹,具体步骤参考 3.3 节。配置 Debug 接口为 Serial Wire,设置HCLK为168MHz。

(3) 设置按键和 LED 对应的 GPIO 设置如图 3.5.3 所示, LED 对应引脚和 3.3 节相同, 按键对应 GPIO 端口 PA0 设置为上跳沿触发外部中断。

🛛 🛇 GPIO 🛛 🛇 F	RCC SYS NVIC			
Search Signals	7		□ Sha	w only Modified Pins
Pin Name	GPIO mode	GPIO Pull-up/Pull-down	User Label	Modified
PA0/WKUP	External Interrupt Mode with Rising edge trigger detection	Pull-up	KeyInter	✓
PH10	Output Push Pull	No pull-up and no pull-down	LED0	~
PH11	Output Push Pull	No pull-up and no pull-down	LED1	~
PH12	Output Push Pull	No pull-up and no pull-down	LED2	$\checkmark$

#### 图 3.5.3 按键和 LED 对应的 GPIO 设置

(4) 在 System Core 组件面板里单击 NVIC,设置中断优先级,如图 3.5.4 所示。首先 在 Priority Group 下拉列表里选择优先级分组,将表示中断优先级的 4 位二进制数分为 2+ 2 位,前两位用于抢占优先级,后两位用于次优先级(也可以做其他分配方式)。然后在 "EXTI line0 interrupt"打钩,并将抢占优先级和次优先级都设置为 1(也可以做其他级别设 置,但不要将抢占优先级的级别设为 0,因为 HAL_Delay()函数用到的定时器中断已经设 置为 0 这个最高优先级)。

NVIC Mode and Configuration									
Configuration									
NVIC Code generation									
Priority Group	Priority Group 2 bits for pre-emption priority 2 bits for subpriority v Sort by Premption Priority and Sub Priority Sort by interrupts names								terrupts names
Search	Search (Ctrl+F)	$\odot$	$\odot$	Show	available ir	nterrupts $ \smallsetminus $		Force DM	A channels Interrupts
	NVIC Interrupt Table					Enabled	Preempt	ion Priority	Sub Priority
Non maskable i	nterrupt					$\checkmark$	0		0
Hard fault interr	upt					~	0		0
Memory manag	ement fault					$\checkmark$	0		0
Pre-fetch fault,	memory access fault					$\checkmark$	0		0
Undefined instru	iction or illegal state					$\checkmark$	0		0
System service	call via SWI instruction					$\checkmark$	0		0
Debug monitor						$\checkmark$	0		0
Pendable reque	st for system service					$\checkmark$	0		0
Time base: Sys	tem tick timer					$\checkmark$	0		0
PVD interrupt through EXTI line 16						0		0	
Flash global interrupt						0		0	
RCC global interrupt						0		0	
EXTI line0 interr	EXTI line0 interrupt					$\checkmark$	1		1
FPU global inte	rrupt						0		0

#### 图 3.5.4 在 NVIC 里设置中断优先级

(5) 设置完成后选择应用开发环境为 STM32CubeIDE 并生成应用程序框架。

(6) 生成应用程序框架后打开 main. c 可以看到如下代码,系统初始化后是一个 while(1)

### 56 ◀ 电子系统设计与实习

```
循环。
 # include "main.h"
 # include "gpio.h"
 void SystemClock Config(void);
 int main(void)
 HAL Init();
 //时钟初始化
 SystemClock Config();
 //GPIO 初始化
 MX_GPI0_Init();
 while (1)
 {
 }
 }
 (7) 打开 main.h 可以看到如下宏定义,为应用程序编写名称提供依据。
 # define KeyInter Pin GPIO PIN 0
 # define KeyInter GPIO Port GPIOA
 # define KeyInter_EXTI_IRQn EXTI0_IRQn
 # define LED0 Pin GPIO PIN 10
 # define LED0 GPIO Port GPIOH
 # define LED1 Pin GPIO PIN 11
 # define LED1 GPIO Port GPIOH
 # define LED2 Pin GPIO PIN 12
 # define LED2 GPIO Port GPIOH
```

(8) 打开 STM32CubeIDE 开发工具,进入开发环境后首先切换工作空间到 MX_Project_inter 文件夹,然后根据 MXProject_inter.ioc 文件新建应用程序工程。打开 gpio.c 文件,在"/* USER CODE BEGIN 2 */"和"/* USER CODE END 2 */"之间的代码沙箱里编写中断服务函数,主要内容是如果检测到 PA0(KeyInter_Pin)的中断标志后将三个发光二极管依次点亮 1s。主程序 While(1)循环中为空。中断服务程序代码如下:

```
void HAL GPIO EXTI Callback(uint16 t GPIO Pin)
{
 if (GPIO Pin == KeyInter Pin)
 HAL GPIO WritePin(LEDO GPIO Port, LEDO Pin, GPIO PIN RESET);
 HAL Delay(1000);
 HAL GPIO WritePin(LEDO GPIO Port, LEDO Pin, GPIO PIN SET);
 HAL Delay(1000);
 HAL GPIO WritePin(LED1 GPIO Port, LED1 Pin, GPIO PIN RESET);
 HAL Delay(1000);
 HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
 HAL_Delay(1000);
 HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, GPIO_PIN_RESET);
 HAL_Delay(1000);
 HAL GPIO WritePin(LED2 GPIO Port, LED2 Pin, GPIO PIN SET);
 HAL Delay(1000);
 }
}
```

(9)编写完成后保存,执行 Run→Debug 菜单命令或者单击快捷工具栏上的 Debug 按 钮即可下载并调试程序(要保证程序没有错误以及开发板连接到计算机并上电)。本例在下 载成功后每按下一次 PA0 引脚上的按键就观察到三个发光二极管依次点亮 1s 并熄灭。

### 小结

本章对 STM32 单片机的基本概念、开发语言与工具、设计应用系统的流程进行了详细介绍,并实际设计了外部引脚中断应用实例。通过本章的学习读者可以在参考 STM32 单 片机硬件资源的基础上初步设计 STM32 应用系统。

### 思考题

1. STM32 单片机与 Arduino 和 MCS-51 单片机之间的区别是什么?

2. STM32 单片机应用的流程是什么?

3. 到官网或其他论坛查阅 STM32 单片机的硬件资源和开发方法,进一步学习 STM32 单片机的应用。

# 扩展阅读:我国 MCU 产业的发展



扩展阅读