

AN0001

通用系列 使用第二复位向量IAP升级

By Home Appliance Product Line

简介

IAP(In Application Programming)即在应用程序编程,是一种自举程序。由于产品固化后不容易采用传统下载器更新固件使得许多产品中内置Bootloader程序用于远程更新固件,有的应用产品在产品固化后,只预留了SPI、UART等通信接口,所以如果需要固件更新,只能考虑使用预留的通信接口进行更新固件操作,在一些上网设备或者无线设备上,也会使用到该功能,例如通过BLE、WIFI等无线通信进行固件更新,我们称为OTA(over the air technology))空中下载技术,也是利用到了IAP升级的原理。 HolyChip HA系列支持第二复位向量的MCU的功能均支持此功能。

实现IAP功能时,即用户程序需要在运行中对自身的功能程序进行更新操作,需要在设计固件程序时编写两个项目代码,第二个项目程序不执行正常的功能操作,而只是通过某种通信方式(如UART、SPI、IIC等通信接口)接收程序或数据,执行对第一部分代码的更新,称之为Bootloader程序;第一个项目代码才是真正的功能代码,称之为APP程序。这两部分项目代码都同时烧录在Flash的不同地址范围,并且可以通过软件复位的方式随意切换。本应用笔记将以HC88T3541为范例介绍此功能。

- 本应用笔记适用芯片: HC89F30xC、HC89F3xx1、HC89F1431、HC89F24x1、HC89F34x1C系 列芯片。
- 本应用笔记对应范例: HC88T3541 Self-update V1.0.0.0。
- 相关数据手册、工具及技术文档下载网址: <u>http://www.holychip.cn/</u>。





目录

1	原理	理及流程图	3
	1.1	原理	3
	1.2	APP 程序流程图	3
	1.3	Bootloader 程序流程图4	ŧ
2	函数	数介绍	5
	2.1	APP 函数介绍5	5
	2.2	Bootloader 函数介绍	5
3	软值	牛使用	3
	3.1	CRC 校验	3
	3.2	Hex To Bin 使用)
4	程序	芧协议定义及范例相关指令	2
5	范依	列应用流程	3
6	KE	IL 工程配置及注意项	5
	6.1	APP 程序工程配置16	5
	6.2	Bootloader 程序工程配置19)
	6.3	注意事项21	Ĺ
7	版本	本说明	2

1 原理及流程图

1.1 原理

MCU 在运行 APP 程序时,等待上位机发送的升级指令,接收指令后通过复位重读代码选项指令转 至 Bootloader 程序区,随后 Bootloader 程序接收更新程序数据,通过 IAP 的操作方式对 APP 程序区更新。

可通过复位指令在 APP 以及 Bootloader 程序中相互切换:

APP → Bootloader: 复位重读代码选项

Bootloader → APP: 复位不重读代码选项

1.2 APP 程序流程图

APP 程序流程图如下图:



1.3 Bootloader 程序流程图

Bootloader 程序流程图如下图:



图 1-2 Bootloader 程序流程图

2 函数介绍

2.1 APP 函数介绍

2.1.1 SystemInit

描述:系统初始化函数

C语言原型: void SystemInit(void);

输入参数:无

返回值:无

2.1.2 CRC_CalcCRC

描述: CRC 校验函数

C语言原型: unsigned int CRC_CalcCRC(unsigned char *fucp_CheckArr,unsigned int fui_CheckLen);

输入参数: *fucp_CheckArr: CRC 校验数据首地址

fui_CheckLen: CRC 校验数据长度

返回值: CRC 校验结果,低位在前,高位在后

2.1.3 ResetReadOption

描述:复位重读代码选项函数 C语言原型: void ResetReadOption (void); 输入参数:无

返回值:无

2.1.4 UART_Send_Str

描述: Uart 写字符串函数
C 语言原型: void UART_Send_Str(unsigned char *fucp_Str);
输入参数: *fucp_Str: 写入的数据的首地址
返回值: 无

2.2 Bootloader 函数介绍

2.2.1 SystemInit

描述:系统初始化函数

C语言原型: void SystemInit(void);

输入参数:无

返回值:无

2.2.2 CRC_CalcCRC

描述: CRC 校验函数

C语言原型: unsigned int CRC_CalcCRC(unsigned char *fucp_CheckArr,unsigned int fui_CheckLen);

输入参数: *fucp_CheckArr: CRC 校验数据首地址

fui_CheckLen : CRC 校验数据长度

返回值: CRC 校验结果,低位在前,高位在后

2.2.3 ResetNoReadOption

描述:复位不重读代码选项函数

C语言原型: void ResetNoReadOption (void);

输入参数:无

返回值:无

2.2.4 UART_Send_Str

描述: Uart 写字符串函数

C语言原型: void UART_Send_Str(unsigned char *fucp_Str);

输入参数: *fucp_Str: 写入的数据的首地址

返回值:无

2.2.5 Flash_EraseBlock

描述:扇区擦除,约消耗 5ms 的时间

C语言原型: void Flash_EraseBlock(unsigned int fui_Address);

输入参数: fui_Address: 被擦除扇区的任意一个地址

返回值:无

2.2.6 FLASH_WriteData

描述: 写入一个字节到 Flash 里面

C语言原型: void FLASH_WriteData(unsigned char fuc_SaveData, unsigned int fui_Address);

输入参数: fui_Address : Flash 地址

fucp_SaveData: 写入的数据

返回值:无

2.2.7 Flash_WriteArr

描述: 写入任意长度的数据到 Flash 里面

C语言原型: void Flash_WriteArr(unsigned int fui_Address, unsigned char fuc_Length, unsigned char

*fucp_SaveArr);

输入参数: fui_Address : Flash 地址

fuc_Length: 写入数据长度

*fucp_SaveArr: 写入的数据存放区域的首地址

返回值:无

3 软件使用

压缩包解压后有一个工具文件夹,其中有本次例程中需要使用的常用工具。

3.1 CRC 校验

1、打开 GCRC.exe, 如下图所示

🚣 格西CRC计算	器 1.0						Σĭ
算法:	CRC8	-]		Þ	-	0
多项式(HEX):	07	-	初始值(HEX):	00			•
数据反转:	MSB First	•	异或值(HEX):	00			-
数据:	HEX	-]				
结果:							_
🛃 检测到最新	所版本软件					计算	

图 3-1 GCRC 界面

2、选择算法,如下图所示

🚣 格西CRC计算	器 1.0						23
算法:	CRC16	•			Þ	-	0
多项式(HEX):	1021	-	初始值(HEX):	FFFF			-
数据反转:	LSB First	•	异或值(HEX):	0000			-
数据:	HEX	•					
结果:							_
🛃 检测到最新	所版本软件					计算	

图 3-2 GCRC 配置

3、输入待计算数据,点击计算获得计算结果,如下图所示

🚣 格西CRC计算	器 1.0					X
算法:	CRC16	-]			0
多项式(HEX):	1021	-	初始值(HEX):	FFFF		-
数据反转:	LSB First	-	异或值(HEX):	0000		-
数据:	HEX	-				
AA 05 01						
结果:						
2AAF						
🛒 检测到最新	所版本软件				计算	

图 3-3 输出计算结果

4、其他计算不一一举例,计算方式与此相同。

3.2 Hex To Bin 使用

1、打开 Bin_Hex.exe,如下图所示



图 3-4 Bin_Hex 界面



	Hex<=>Bin				_	~
(5)玟(牛	(E)编辑 (O)设	置 (H)帮助				
🛩 🗅	🔳 🖿 🌺	?				
	11 打开文件			×		
	查找范围([]):	● 更新程序	▼ ← 🗈 💣 🗉	•		
	名称	^	修改日期	类型		
	Pro.hex		2019/7/25 10:55	HEX 文		
	٢			>		
	< 文件名(<u>N</u>):	Pro.hex	1	, то		
	< 文件名(N): 文件类型(I):	Pro.hex 16进制文件(*.hex)	Tt z	、 开(<u>O</u>)		
	< 文件名(<u>N</u>): 文件 <u>类型</u> (<u>T</u>):	Pro.hex 16进制文件(*.hex)	TE T	► 开(<u>O</u>) 欧消		

2、打开压缩包中的更新程序文件夹选择其中的 hex 文件,如下图所示

- 图 3-5 更新 hex 文件
- 3、将文件保存在相同文件夹下,如下图所示

51Hex<=>Bin		_		\times
(F)文件 (E)编辑	(O)设置 (H)帮助			
😂 🗀 🔡 Pa	A& ?			
: 03000000203B : 0C003B00787FE4 : 10002100788E51 : 0A003100837FFA : 1000030003EF94 : 00001300011CED : 0100200022ED : 00003001FF	CO F05BFD758107020021F3 90FE917401F090FF0374C2F0B28D 7E0012000380F5C1 00E940400417492C706E01D700C 4C70F7EF1F70E61E80E33E			^
	■ 另存为 ×			
	保存在(1): 更新程序 ← € 삼 四▼			
	名称 个 修改日期 类型 没有与搜索条件匹配的项。			
	文件名(N): Pro 保存(S) 保存类型(T): 二进制文件(*.bin) ▼ 取消			~
< 保存文件			≂.6 ⊅0.1	>
1000000	陸序文件大小パイチャ	1	ו:[ע ס:ד	1

图 3-6 保存文件



3/H 51Hex<=>Bin		- [1 ×
(F)文件 (E)编辑 (O)设置 (H)帮助		
😅 🗀 🔛 🖣 🔺	8 <mark>?</mark>		
030000002003BC0 0C003B00787F44F61 10002100788E5190 0A003100837FFA7E1 1000030003EF94001 00001300011CED4C' 0100200022BD 00000001FF	0 0 0 0 0 0 0 0 0 0 0 0 0 0		
	11 打开文件	X	
	查找范围(I): 更新程序		
	名称 ^ Pro.Bin	修改日期 类型 2020/2/27 14:04 BIN 文	
	< 文件名(N): Pro.Bin	> 打开(O)	
	文件类型(T):	▼ 取消	
			>
「开文件		程序文件大小、71字节 行:9	万川·1

4、再次通过 Hex To Bin 打开刚刚保存的 Bin 文件,如下图所示

图 3-7 打开保存文件

5、从而得到转换好的 Bin 文件,如下图所示

(F)文件 (E)编辑 (O)设置 (H)帮助 I2 00 38 D3 FF 84 00 EE 94 00 EF 95 77 70 EF 10 70 69 17 76 81 07 02 00 12 00 38 80 F5 78 7F E4 F6 D8 F7 76 81 07 02 00 21 10 13 14 12 14 14 12 12	×		_	-																															n	Bi	=>	x<:	He:	511	иH	
Image: Constraint of the state of																													助	l)帮!	(H	ł	跧置	D)ì	(C	iii	编	(E)\$		件	F)ズ	(
D2 00 3B D3 EF 94 00 EE 94 00 40 14 70 3C 7C 06 ED 10 70 01 C ED 4C 70 FF 1F 70 B6 1E 80 E5 E2 75 86E 51 90 F6 91 74 F1 EF 19 70 374 C2 F0 E2 25 77 FA 7E 00 12 00 03 80 F5 78 7F E4 F6 D8 F0 76 61 07 02 00 21																																	?	8	1	Ĩ				2	Z	1
	^																					06 E3 F0 D8	7C 80 C2 F6	3C 1E 74 E4	'D (16) 13 ' 'F H	17 E 70 77	14 70 FF 78	40 1F 90 F5	00 3F 70 30	4 0 7 E 1 F 3 8	9. F 0	EE 70 74 00	10 H 10 T 11 T 12 0	4 0 9 4 9 1 0 2	94 ED FE 00	EF 1C 90 7E 02	3 1 1 1 1 1 7 1	D3 01 51 FA 07	3B 70 8E 7F 81	10 D 5 3 5		0: E12: B: F1
<	~																																									~
程序文件大小71字节 行:6 列:1		5: 6 列:1	行		5	节	1字节	小71	件大/	下文件	呈序3	種	_	_		_	 _		_		_	_	_	_	_		_	_	-	_		_			_			-	-		-	Ē

图 3-8 查看转换好的文件

4 程序协议定义及范例相关指令

命名	А	B1	C1	D1D2	Ex	F1F2								
序号	1	2	3	4	5	6								
举例	0xAA	05	01	-	-	0x2A0xAF								
序号	名称		说明											
1	帧头	数据帧头,若	故据帧头,若帧头错误返回相应提示											
2	数据长度	数据总长度												
3	比么	0x01: 进入 B	ootloader 程序											
5	1日.之	0x02:开始A	PP 程序更新											
Α	待更新 APP	加世地人北平		车也人 可勿败										
4	程序地址	如此1日マートル	如 APP 但 / 史/ 文史	胡1日 マ, 可心咍	Ĩ									
5	待更新 APP	如此指令非开	始 APP 程序更新	新指令,可忽略	T									
5	程序数据	待更新 APP 程序数据最大 128												
6	CRC 校验值	校验值低位在	校验值低位在前,高位在后											

表格 4-1

HC88T3541 Self-update V1.0.0.0 例程升级指令

进入Bootloader	AA 05 01 2A AF
	AA 4E 02 00 00 02 00 3B D3 EF 94 00 EE 94 00 40 14 7D 3C 7C 06 ED 1D 70 01 1C
写入更新程序	ED 4C 70 F7 EF 1F 70 E6 1E 80 E3 22 75 8E 51 90 FE 91 74 01 F0 90 FF 03 74 C2 F0
	B2 83 7F FA 7E 00 12 00 03 80 F5 78 7F E4 F6 D8 FD 75 81 07 02 00 21 69 98

表格 4-2

5 范例应用流程

- 打开 HC88T3541 Self-update V1.0.0.0 文件夹中的 APP 文件夹中的 Keil 工程文件编译后下载到 芯片。
- 打开 HC88T3541 Self-update V1.0.0.0 文件夹中的 Bootloader 文件夹中的 Keil 工程文件编译后下 载到芯片。
- 3、 完成以上步骤,重新给芯片上电,可以测得 P02 口输出低电平。
- 4、 打开工具文件夹中的串口工具—"UartAssist.exe",将芯片的串口引脚与串口工具连接,芯片的 P06 口接串口工具的 RXD, P07 口接串工具的 TXD。
- 5、 打开 UartAssist.exe 并做好相关配置。

串口波特率 9600、无校验位、8 位数据位、1 停止位;接收区选择非十六进制显示、发送区选择按十六进制发送。串口配置如下图所示

· ·	串口调试助手	(C∎精装版	V3. 8. 3)		(×
申日设置 串ロ号 COM2 ▼ 波特率 9600 ▼ 校验位 NONE ▼	串口数据接收				
数据位 ⁸					
 接收区设置 接收转向文件. 显示接收时间 十六进制显示 暂停接收显示 保存数据 清除显示 					
发送区设置 「 启用文件数据源… 「 自动发送附加位					
 反法元日初月呈 反 按十六进制发送 □ 数据流循环发送 发送间隔 300 毫秒 文件载入 清除输入 					发送
☞ 就绪!		发ì	<u></u> ž:0	接收:0	复位计数

图 5-1 串口配置图

6、 在发送窗口输入 AA 05 01 2A AF 指令,并点击发送。指令定义详见程序协议定义及范例相关指 令章节。指令可直接在升级指令.TXT 中获取复制。如下图所示

••	串口调试助手((Ⅲ精装版 ▼3.8.3	3)	×
串口设置 串口号 COM2 ▼ 波特率 9600 ▼ 校验位 NONE ▼ 数据位 8 ▼ 停止位 1 ▼ 後收区设置 「接收转向文件 豆示接收时间 「十六进制显示 「暫停接收显示 <u>保存数据</u> <u></u> 富馀显示 发送区设置	串口数据接收 Entering ISP Mode The current mode is I	SP mode.		
 □ 自动发送附加位 □ 发送完自动清空 ▽ 按十六进制发送 □ 数据流循环发送 > 发送间隔 300 毫秒 > 支件表) 速移会) 	AA 05 01 2A AF			发送
」 → → → → → → → → → → → → → → → → → → →	I	发送:5	接收:53	复位计数

图 5-2 发送升级指令

7、此时可以看到接收端口返回相应的指令并且可以测得 P02 输出高电平、P03 输出低电平。然后 再在发送窗口输入 AA 4E 02 00 00 02 00 3B D3 EF 94 00 EE 94 00 40 14 7D 3C 7C 06 ED 1D 70 01 1C ED 4C 70 F7 EF 1F 70 E6 1E 80 E3 22 75 8E 51 90 FE 91 74 01 F0 90 FF 03 74 C2 F0 B2 83 7F FA 7E 00 12 00 03 80 F5 78 7F E4 F6 D8 FD 75 81 07 02 00 21 69 98 指令并点击发送。如下图 所示





图 5-3 写入更新程序

8、数据接收窗口提示数据写入成功后,重新给芯片上电,此时 P03 口电平持续翻转,程序更新成功。



6 KEIL工程配置及注意项

6.1 APP 程序工程配置

1、 STARTUP.A51 文件中的起始地址需配置为0。如下图所示

🕸 Project: Pro	116	2C C51STARTUR	SEGMENT	CODE
Target 1	117	DETACK	SECHENT	TDATA
	11/	PSTACK	SEGMENT	IDAIA
🖻 🐙 Source Group 1	118			
	119		RSEG	?STACK
	120		DS	1
	121			
🕀 📄 Public.c	122		EXTRN C	ODE (?C START)
🕀 📄 Uart.c	123		PUBLIC	?C_STARTUP
	124			_
	125		CSEG	AT 0
	126	?C_STARTUP:	LJMP	STARTUP1
	127			
	128		RSEG	?C_C51STARTUP
	129			
	130	STARTUP1:		
	131			
	132	IF IDATALEN <>	0	
	133		MOV	RO,#IDATALEN - 1
	134		CLR	A
	135	IDATALOOP:	MOV	@RO,A
	136		DJNZ	R0, IDATALOOP
	137	ENDIF		

图 6-1 STARTUP.A51 文件配置

2、 OPTION 中的 C51 选项卡需要将中断地址配置为 0x0000。如下图所示

🖂 🕺 📥 🖷 🗇 🍏 🏟	
🔣 Options for Target 'Target 1'	×
Device Target Output Listing User C51 A51 1	BL51 Locate BL51 Misc Debug Vtilities
Preprocessor Symbols	
Define:	
Undefine:	
Code Optimization Level: 8: Reuse Common Entry Code Emphasis: Favor speed Global Register Coloring Linker Code Packing (max. AJMP / ACALL) Don't use absolute register accesses	Warninglevel 2 Bits to round for float compare: 3 ✓ Interrupt vectors at address: 0x0000 ✓ Keep variables in order ✓ Enable ANSI integer promotion rules
Include Paths	

图 6-2 C51 选项卡配置

3、 内存区间可配置为默认区间。如下图所示

🕅 Options for Target 'Target 1' X					
Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities					
Use Memory Lavout from	n Taroet Dialog				
	Code Range: 0X0000-0X7FFF	—			
Space Base Segments:	Xdata Range: 0X0000-0X03FF				
<u>C</u> ode:					
<u>X</u> data					
Pdata:					
P <u>r</u> ecede:					
<u>B</u> it:					
-	图 6-3 内存区间配置				

4、 HC-LINK 下载选择程序所有存储区,如下图



图 6-4 程序所有存储区配置

5、 代码选项中需要使能第二复位向量并且将第二复位向量地址设置为 0x7000, 如下图所示



图 6-5 第二复位向量配置

6.2 Bootloader 程序工程配置

- 🖃 🔧 Project: Pro DATA 109 SP 81H 😑 ᇶ Target 1 DPL DATA 82H 110 111 DPH DATA 83H 🖮 ᇶ Source Group 1 112 STARTUP.A 113 NAME ?C_STARTUP 🗉 🗋 main.c 114 • Public.c 115 ?C C51STARTUP SEGMENT CODE 116 🗄 📄 Uart.c 117 ?STACK SEGMENT IDATA 🗄 📄 Flash.c 118 119 RSEG ?STACK 120 DS 1 121 EXTRN CODE (?C_START) PUBLIC ?C_STARTUP 122 123 124 125 CSEG AT 7000H 126 ?C STARTUP: LJMP STARTUP1 127
- 1、 STARTUP.A51 文件中的起始地址需配置为 7000H。如下图所示

图 6-6 STARTUP.A51 配置

2、 OPTION 中的 C51 选项卡需要将中断地址配置为 0x7000。如下图所示

Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities
Preprocessor Symbols
Define:
Code Optimization

图 6-7 C51 选项卡配置

3、 内存区间地址需设置为 0x7000-0x7FFF。如下图所示

📱 Options for Target 'Target 1'						
Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities						
Use Memory Layout from Target Dialog						
			Code Range: 0X7000-0X7FFF			
Space	Base	Segments:	Xdata Range: 0X0000-0X03FF			
Code:						
<u>X</u> data						
<u>P</u> data:						
Precede:						
图 6-8 内存区间地址配置						



4、 页擦写选择第 28、29、30、31 页,并需要打开 Option 擦写开关。如下图所示

₩ Options for Target 'Target 1'						×		
Device Towart 0	HC-LINK V4.0.3.0	\times	L.					
Device Targer U 文件 没客 语言(Language) 都助				4	页号	地址		^
Configure Flash M			н.	15	14	0x3800-0x3BFF		
Use Target D	「端口设置			16	15	0x3C00-0x3FFF		
	端口: USP.SEPIAL CH240 (COM7)			17	16	0x4000-0x43FF		
	SMEA: 1030-3ETIME CI1340 (COM7)		2	18	17	0x4400-0x47FF		
Init File:				19	18	0x4800-0x4BFF		
				20	19	0x4C00-0x4FFF		
C Use External	MCU:HC88T3541 FLASH_SC	SWD		21	20	0x5000-0x53FF		
Command:		复位、由酒	Π.	22	21	0x5400-0x57FF		
Arguments:			н.	23	22	0x5800-0x5BFF		
	SN_DATA ID_DATA 烧录限制	打开PM51	н.	24	23	0x5C00-0x5FFF		
				25	24	0x6000-0x63FF		
Carfarra haras I	烧录设置◀	Cache Options		26	25	0x6400-0x67FF		
-Configure image F	☑ 擦除 擦除页设置	Cache Data		27	26	0x6800-0x6BFF		
Output File:	LOAD IV 烧录		Ц,	28	27	0x6C00-0x6FFF		
		🔽 Cache Xdata 🗡		29	28	0x7000-0x73FF		
Image Files Root	臣 고왕도님파	E Contra Contra	ш	30	29	0x7400-0x77FF	Y	
	▶ ▶ 報后上电	V Lache Lode		31	30	0x7800-0x7BFF		
	确定取消	恢复默认		32	31	0x7C00-0x7FFF	 ✓ 	*
OK Cancel Defaults					确定	取消		

图 6-9 页擦写配置

5、 代码选项卡中需要使能第二复位向量并且将第二复位向量地址设置为 0x7000(新版 link 插件擦 写页设置非全擦时,无法修改 OPTION。则可以不需要设置 OPTION)。如下图所示



图 6-10 第二复位向量配置

6、关于起始地址、中断地址、程序空间地址、页擦写地址、第二复位向量的地址等配置规则详见 各 MCU Datasheet 第二复位向量配置章节。以 HC88T3541 为例,第二复位向量空间大小只能 为 1K、2K、4K、8K、16K、32K。如果将第二复位向量空间大小定义为 4K,那么 APP 程序 地址为 0x0000-0x7FFF,第二复位向量地址设置为 7000。Bootloader 程序地址为 0x7000-0x7FFF。 那么 Bootloader 工程中的起始地址应该设置为 7000H,中断地址需设置为 0x7000,内存区间地 址需设置为 0x7000-0x7FFF,页擦写选择第 28、29、30、31 页,第二复位向量地址设置为 0x7000。

6.3 注意事项

- 1、 Bootloader 程序中关于 IAP 的操作并不严谨,具体的操作方式请严格参照《AN0003_HA_通用 系列 IAP 使用》。
- 2、 APP 程序和 Bootloader 程序烧录时请先烧录 APP 程序再烧录 Bootloader 程序。
- 3、 在使用脱机工具 HC-PM51 烧录时可将 APP 和 Bootloader 程序同时烧录进入 MCU,具体操作 方式请参照《UM0101_HC-PM51 V5.0_工具使用手册》。
- 4、 使用第二复位向量 IAP 升级只能升级 APP 程序,无法升级代码选项。

7 版本说明

版本	日期	描述		
V1.0.0.0	2020/02/27	初版		
V1.0.1.0	2020/03/02	版本更新		
V1.02	2020/07/15			
V1.03	2020/10/15	删去 HC89F3xx1 系列,新增 HC88T3xx1 系列		
V1.04	2020/12/30 增加 HC89F3xx1、HC88T6xx1 系列			
V1.05	2021/11/02	删去 HC88T3xx1、HC89F30xB、HC88T6xx1、HC88F1xx1 系列		
	2021/11/03	新增 HC89F30xC 系列		
V1.06	5 2024/11/13 增加 HC89F1431、HC89F24x1、HC89F34x1C 系列			

HOLYCHIP公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。 HOLYCHIP不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任,HOLYCHIP的产品不 是专门设计来应用于外科植入、生命维持和任何HOLYCHIP产品产生的故障会对个体造成伤害甚至死亡 的领域。如果将HOLYCHIP的产品用于上述领域,即使这些是由HOLYCHIP在产品设计和制造上的疏忽 引起的,用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用,并且用 户保证HOLYCHIP及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子

2024年11月