# 图文解说:如何将STM32的标准库编译成lib库

以前一直使用 STM32 的标准库,需要一步步地将代码加进去,将编译选项设置好,然后再编译整个工程。这个编译过程是一个相当慢的过程!完全编译大约需要一支烟的时间。每次建立工程都这么编译,是一个相当浪费时间和香烟的过程。

于是,我有了将库编译成 1 ib 文件的想法。本文就是我将 STM32F4 的标准库 编译成 1 ib 文件并在工程中使用的过程。

#### 适用对象:

1、熟悉库,不想再看库里边代码

- 2、有稳定的库,库文件更新不频繁
- 3、库文件多,每次编译时间长

下面是我将 STM32F4 的标准库编译成 1ib 并在工程中使用的过程:

#### 1、建立创建 1ib 的工程

#### 2、将库文件拷贝到工程目录:

将库里边目录\STM32F4xx\_StdPeriph\_lib v1.0.2\STM32F4xx\_StdPeriph\_Driver下的 inc 和 src 两个文件夹拷贝到预创建 工程的目录。我计划在目录 E:\学习\ARM\库\stm32f4-2 中创建库。于是我将两 个文件夹拷贝到了这里。

#### 3、选择芯片: STM32F407VG

Vendor: STMicroelectronics Device: STM32F407VG Toolset: ARM		Description	
STM32F407/E STM32F407/E STM32F407/E STM32F407/E STM32F407/ZG STM32F407ZG STM32F407ZG STM32F415VG STM32F415VG STM32F415VG STM32F417/E STM32F417/E STM32F417/E STM32F417/E	•	Core - ARM 32-bit Cortex-M4 CPU with FPU - Adaptive real-time accelerator (ART Accelerator) - 168 MHz maximum frequency, 210 DMIPS/1.25 DMIPS/MHz - Memories - Up to 1 Mbyte of Rash memory - Up to 1 192+4 Kbytes of SRAM - Rexoble static memory controller (supports Compact Rash, SRAM, PSRAM, NOR, NAND memories) LCD parallel interface, 8080/6800 modes Clock, reset and supply management - 1.7 V to 3.6 V application supply and I/Os - POR, PDR, PVD and BOR - 4 to 25 MHz crystal oscillator - Internal 16 MHz factory-trimmed R	•

## 4、选择 NO。因为这不是可运行的程序,这里不需要加入启动文件。

μVision	
?	Copy 'startup_stm32f4xx.s' to Project Folder and Add File to Project ?
	OFweek lee.ofweek.com
	<b>甩子工</b> 程网

## 5、创建完工程后,工程结构如图:

本地磁盘 (Ei	) • 学习	▶ ARM ▶ 库 ▶	stm	32f4-2 🕨		
共享 🕶	刺梁	新建文件夹				
名称	31		Ŧ	传改日期	英型	大小
🔰 inc				2013/7/14 16:30	文件夹	0
📕 src				2013/7/14 16:30	文件夹	OFweek lee.ofweek.com
stm32f4	4lib.uvproj			2013/7/14 16:22	續sion4 Pro	ject 0 KB
stm32f4	4lib_Targe	t 1.dep		2013/7/14 16:26	DEP文件	电子虹程网

6、MDK 中点击工具栏上的 型 设置工程结构,并将库文件加入工程:

为了通用,我将库中所有的C文件都加入了工程

reponents, Environment and Books	11 H L	and Add Tales to Group The	
reject Components  Felders/Batemaises   Bodes			* 8 0 8*
Prinet Tagets (X) () () Groue (X) () ()	Res X ≠ 4 macc + mil34a_adcc + mil34a_can c	Sft " misca strallNoc.adca strallNoc.adca	修改日期 2012/3/22 15:4 2012/3/22 15:4 2012/3/22 15:4
	sen 23 Augusto Santa San	stm3294as,crcs stm3294as,cryp.c stm3294as,cryp.dm.c stm3294as,cryp.dm.c stm3294as,cryp.dm.c stm3294as,dm.c stm3294as,dm.c stm3294as,dm.c	2012/3/22 15-4 2012/3/22 15-4 2012/3/22 15-4 2012/3/22 15-4 2012/3/22 15-4 2012/3/22 15-4 2012/3/22 15-4
Set as Euroret Target	And Res Add Res Add Res	stm206 dmac stm206 dmac stm206 fweek ee.o	3012/3/22 154 2012/3/22 154 5012/3/22 154 5012/3/22 154 5012/3/22 154 5012/3/22 154 5012/3/22 154 5012/3/22 154
(% Faorel	Nelp	Enance Here II	程网

7、完成后 MDK 下的目录结构:



8、MDK 下设置输出选项:

工程选项中设置输出,选择输出 lib 到目录 E:\学习\ARM\库 \stm32f4-2\lib\:

vice   Target Output   Listing   User	C/C++   Asm   Linker   Debug   Utilities
Select Folder for Objects	Name of Executable: stm32/4lib
C Create Executable: .Vib\stm32/4lb	
Debug Information	Create Batch File
🗖 Create HEX File	
P Browse Information	
Create Executable: .\lb\stm32l4lb.lb	
	OFweek ee.ofweek.
	电子工程

#### 9、设置C语言预编译宏和引用目录:

因为要使用 STM32F4 标准库,预编译选项设置: USE\_STDPERIPH\_DRIVER, STM32F4XX

将刚才拷贝的 inc 和工程根目录文件夹加入引用:

Ofweek 电子工程网

Defin	Folder Setup		5 ×
Undefin	Setup Compiler Include Paths:	10	< + +
Languag	.Vinc Vstm32f4-2		
Optimizat			2
C Opte			
□ Split			
□ One			
Include			
Pathe			
Controls	1		
Compile	OK	Cancel	

# 10、设置完成后:

Define: DSE_STDPERIPH_DRIVEP	R,STM32F4XX	
Definization: Level 0 (-00)	Strict ANSI C     Enum Container always int     Plain Char is Signed     Read-Only Position Independent     Read-Write Position Independent	Warnings: Cunspecified> Thumb Mode No Auto Includes 
Misc Controls Compiler control 4 D:\tools\Keil\ARM\RV31\ string	-apcswinterwork -1.\inc -1\atm3214-2	Weel ( lee ofweek con

# 11、将 stm32f4xx\_conf.h 文件拷贝到工程:

这个文件需要在标准库提供的示例工程中找:

我使用的是

 $\TM32F4-Discovery_FW_V1. 1. 0\Project\Peripheral_Examples\ADC3_DMA\stm 32f4xx_conf.h$ 

这个文件引用了库文件中所有的头文件。因为不包含在库中,我将这个文件 拷贝到 E:\学习\ARM\库\stm32f4-2 文件夹。

### 12、到此,工程设置完成。按 F7 编译,经过一支烟的时间即可生成库的 1ib。

生成完成后, MDK 工程中:

Build Output	
compiling	stm32f4xx_adc.c
compiling	stm32f4xx_can.c
compiling	stm32f4xx_crc.c
compiling	stm32f4xx_cryp.c
compiling	stm32f4xx_cryp_aes.c
compiling	stm32f4xx_cryp_des.c
compiling	stm32f4xx_cryp_tdes.c
compiling	stm32f4xx_dac.c
compiling	stm32f4xx_dbgmcu.c
compiling	stm32f4xx_dcmi.c
compiling	stm32f4xx_dma.c
compiling	stm32f4xx_exti.c
compiling	stm32f4xx_flash.c
compiling	stm32f4xx_fsmc.c
compiling	stm32f4xx_gpio.c
compiling	stm32f4xx_hash.c
compiling	stm32f4xx_hash_md5.c
compiling	stm32f4xx_hash_sha1.c
compiling	stm32f4xx_i2c.c
compiling	stm32f4xx_iwdg.c
compiling	stm32f4xx_pwr.c
compiling	stm32f4xx_rcc.c
compiling	stm32f4xx_rng.c
compiling	stm32f4xx_rtc.c
compiling	stm32f4xx_sdio.c
compiling	stm32f4xx_spi.c
compiling	stm32f4xx_syscfg.c
compiling	stm32f4xx_tim.c
compiling	stm32f4xx_usart.c
compiling	stm32f4xx_wwdg.c p 7 7 10
creating 1	
".\lib\str	$m_{32}$ relief $m_{32}$ reli

13、工程输出目录:

名称	修改日期	类型	大小
misc.crf	2013/7/14 16:51	CRF 文件	431 KB
inisc.d	2013/7/14 16:51	D文件	2 KB
misc.o	2013/7/14 16:51	0 文件	450 KB
🔛 stm32f4lib.lib	2013/7/14 16:53	Object File Library	14,131 KB
stm32f4xx_adererf	2013/7/14 16:51	BEST ON	438 KB
stm32f4xx_adc.d	2013/7/14 16:51	D 文件	2 KB
stm32f4xx_adc.o	2013/7/14 16:51	BP 7 T	196 X8
stm32f4xx_can.crf	2013/7/14 16:51	CRF 文件	- 418 KB

文件 stm32f4lib.lib 就是我们生成的 lib 文件

## 14、将库文件加入该工程

15、新建工程,我命名成 stm32f4use,处理器依旧选择 STM32F407VG。

Vendor: STMcroelectronics Device: STM32F407VG Toolset: ARM	Description	
STM32F407/E     STM32F407/G     STM32F407/G     STM32F407/E     STM32F407/E     STM32F407/ZE     STM32F415/G     STM32F415/G     STM32F415/G     STM32F415/G     STM32F415/G     STM32F417/E     STM32F417/E     STM32F417/E     STM32F417/E     STM32F417/E	Core - ARM 32-bit Cortex-M4 CPU with FPU - Adaptive real-time accelerator (ART Accelerator) - 168 MHz maximum frequency, 210 DMIPS/1.25 DMIPS/MHz - Memory protection unit Memories - Up to 1 Mbyte of Rash memory - Up to 1 192+4 Kbytes of SRAM - Rexble static memory controller (supports Compact Rash, SRAM, PSRAM, NOR, NAND memories) LCD parallel interface. 8080/6800 modes Clock, reset and supply management - 1.7 V to 3.6 V application supply and I/Os - POR, PDR, PVD and BOR - 4 to 25 MHz crystal oscillator - Internal 16 MHz factory-trimmed RC	* m
4 III +	· · · · · · · · · · · · · · · · · · ·	

16、这是选择 yes,因为这是一个可执行的工程:



17、将标准库示例工程的

\STM32F4-Discovery\_FW\_V1.1.0\Project\Peripheral\_Examples\ADC3\_DMA中的system\_stm32f4xx.c拷贝到工程目录(E:\学习\ARM\库\stm32f4-2)中

18、再次点击菜单上的 望过 设置工程目录结构,将刚才生成的 1ib 库加入 到工程中:

• • • • •
- 博政日期 2013/7/14 1#53
Celt se.ofweek.com

19、将示例工程

\STM32F4-Discovery\_FW\_V1.1.0\Project\Peripheral\_Examples\ADC3\_DMA 中的 system\_stm32f4xx.c 拷贝到工程目录。

20、将 main.c 和 system\_stm32f4xx.c 加入到工程

extect Components   Builders/Fetensions   Builde	Ad Ad	Id Files to Group 'user'	
in the second seco	東抗	10000 ( 🔒 100000-0	• • • • • •
Prest Tapts (X 1 4 Gran (X 1 4 6	× • •	ne 15 maine yatarni, strai 2/Maxe.e	機改田順 2013/7/54 16-30 2013/7/14 16-53 2013/7/14 16-55 2011/7/54 16-55 2011/10/28 10-32
		OFweek	e.ofweek.com
Set as Current Target	Add Res 2014		e.ofweek.com

21、完成后的目录结构:



### 22、加入编译选项

与上边生成 1 ib 相似, 预编译选项设置: USE\_STDPERIPH\_DRIVER, STM32F4XX

引用目录:..\inc;..\stm32f4-2(这里的.\inc;..文件夹就是刚才建立库时 候的文件夹)

输出可执行文件:

Ofweek 电子工程网

Create Batch File
Create Batch File

### 23、添加几行简单的代码

```
//点亮一个LED
#include <stm32f4xx.h>
#include "stm32f4xx_conf.h"
#include "stm32f4xx_tim.h"
int main()
{
GPI0_InitTypeDef GPI0_InitStructure;
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPI0D, ENABLE);
GPI0_InitStructure.GPI0_Mode = GPI0_Mode_OUT;
GPI0_InitStructure.GPI0_OType = GPI0_OType_PP;
GPI0_InitStructure.GPI0_Pin = GPI0_Pin_12;
GPI0_InitStructure.GPI0_PuPd = GPI0_PuPd_NOPULL;
GPI0_InitStructure.GPI0_Speed = GPI0_Speed_2MHz;
```

GPI0\_Init(GPI0D,&GPI0\_InitStructure);

GPI0\_SetBits(GPI0D, GPI0\_Pin\_12);

while(1);

}

24、工程设置完成,按F7试试。现在编译速度快起来了,点个烟的时间就 编译完了