EK-STM32 板的第一个例程

买了这块板好久好久了,可是就是没有时间让它转起来。最近不是太忙准备 让他转起来,这下就有了我的入门第一例。我不为了别的,只为了让和我一样的 人快速入门,省得费时间。

前提条件:将 stm32 的样例下载,名为 um0427.zip,解压开。

第一步,创建一个文件夹,并参照"如何使用 stm32 的软件库在 IAR 的 EWARM 下进行的应用开发。



쎀		止	
퐈	<u> </u>	ル	,

		и. УН
Sa Save Workspace Close Workspace Clos	ded Workbench IDE Project Tools Window Help Add Files Add Group Import File List Edit Configurations Remove Create New Project Add Existing Project	 点击菜单File> New>Workspace 创建一个新的工程 点击菜单Project> Create New Project 创建一个项目
B+ aar Printh Ctrl+P B+ C+ Recent Files + D_L Recent Workspaces + Dut Exit - Describtion Creates an empty project.	Options Alt+F7 Source Code Control > Make F7 Compile Ctrl+F7 Rebuild All Clean Batch build F8 Stop Build	3. 保存项目到设定的项目根目录中
STM32 32位微控制器 STM32 释放您的创造力 其中的第二小项,可以创建 Create New Project	www.stmicroele 这一个空的工程,如下图	2007年12月 7 ctronics.com.cn/mcu
Tool chain: ARM	•	
Project templates: Empty project asm C++ Externally built executable		

之后,保存到创建项目的目录中。

第三步:添加文件,具体需要添加什么,看例程中的 readme.txt 中的说明。

```
38 How to use it
39 ==========
```

40 In order to make the program work, you must do the following: 41 - Create a project and setup all your toolchain's start-up files

42 - Compile the directory content files and required Library files:

- + stm32f10x_lib.c 43
- + stm32f10x gpio.c 44
- + stm32f10x_rcc.c 45
- + stm32f10x_nvic.c 46
- 47 + stm32f10x flash.c

GPIO 需要添加,如上五个头文件,如下图



再添加源程序,

添加完成后如下图,



第四步:根据硬件连接修改软件,由于 EK-STM32F 开发板上的指示灯连接到如 下: ld5-led2 pc4-pc7, 在开发板后面标明。 故将原程序中的 GPIO_Pin_6, 7、8、9, 分别改为4、5、6、7。 注: 原例程中的文件为只读文件, 去掉只读属性, 改程序如下: contigute force, force, force and force ab cacedo paon p GPI0_InitStructure.GPI0_Pin = GPI0_Pin_4 | GPI0_Pin_5 | GPI0_Pin_6 | GPI0_Pin_7; GPIO InitStructure.GPIO Speed = GPIO Speed 50MHz; GPIO InitStructure.GPIO Mode = GPIO Mode Out PP; GPIO_Init(GPIOC, &GPIO_InitStructure); while (1) ł /* Turn on led connected to PC.06 pin */ GPI0_SetBits(GPI0C, GPI0_Pin_4); /* Insert delay */ Delay(OxAFFFF); /* Turn on led connected to PC.07 and PC.08 pins */ GPI0_SetBits(GPI0C, GPI0_Pin_5 | GPI0_Pin_6); /* Turn off led connected to PC.06 pin */ GPIO_ResetBits(GPIOC, GPIO_Pin_4); /* Insert delay */ Delay(OxAFFFF); /* Turn on led connected to PC.09 pin */ GPIO_SetBits(GPIOC, GPIO_Pin_7); /* Turn off led connected to PC.07 and PC.08 pins */ GPIO ResetBits(GPIOC, GPIO Pin 5 | GPIO Pin 6); /* Insert delay */ Delay(0xAFFFF); /* Turn off led connected to PC.09 pin */

```
/* furn off fea connectea to Pt.09 pin
GPI0_ResetBits(GPI0C, GPI0_Pin_7);
```

第五步 设置芯片类型

	_					
Category:						
General Options						
C/C++ Compiler		Target	Output	Library Configuration	Library options	MIS
Assembler		- Proo		rippt		
Custom Build		FIUC	essui va	nani		
Build Actions		0.0	Core	Cortex-M3	*	
Linker						
Debugger		 • • • 	Device	ST STM32F10x		"∎⊷
Simulator				,		

第六步: 设置包含文件路径,如下图

r		- detoly settings
	General Options	
	C/C++ Compiler	Language Optimizations Output List Preprocessor Diagnostic
	Assembler	
	Custom Build	J Ignore standard include directories \$TOOLKIT_DIR\$\INC\
	Build Actions	
	Linker	Additional include directories: (one per line)
	Debugger	E:\Learn_Program\STM32\GPI0\library\inc
	Simulator	E:\Learn_Program\STM32\GPI0\
	Angel	
	IAR ROM-monitor	

```
第七步:
```

设置链接文件,就选择前边我们加入的 lnkarm_ram.xcl,有两个文件一个为 RAM, 另外一个为 Flash,根据自己下载的目标确定。

Assembler Custom Build Build Actions Linker Debugger Simulator Angel	Linker command file Override default E:\Learn_Program\STM32\first\Inkarm_flash.xcl Command file configuration tool Override default program entry			
第八步: 选择调试工具,为第三方工具。				
Custom Build Build Actions Linker	Driver Third-Party Driver			
Debugger Simulator Angel	Setup macros			
第九步: 选择驱动文件路径				
Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace LMI FTDI Macraigor RDI	IAR debugger driver plugin C:\Manley\Drivers\STLink\STM32Driver.dll			

第十步:

编译程序,同时系统会提示保存项目,命名并保存。

Log communication

第十一步:

如果没有错误出现,则可以进行在线调试了,点击调试,会出现如下图。

\$TOOLKIT_DIR\$\cspycomm.log



至此,可以完全仿真了,点击全速运行。观察四个指示灯。 我是刚入门,好多东西不明白,错误之处请大家指处,诚挚感谢!让我们共同进 步!!!!