GVM32F030 工程建立向导

Rev1.3

本文以一个 LED 例程的建立来讲解 GVM32F030 工程的建立工程。

一、在 <u>GVM32F030 StdPeriph Lib V1.4.1\Projects\Examples</u> 文件夹中新建一个文件夹, 命名为 LED。

二、打开 Keil MDK (我这里采用 Keil v5);点击 Project -- New uVision Project;选择工程 文件的路径 (刚才新建的文件夹下),命名工程为 LED;点击保存。

Create New Project	ct		-						×
😋 🖉 🗢 📕 « Pr	ojects	 Examp 	oles 🕨 LED		-	· • • 搜索	LED		٩
组织 ▼ 新建文	件夹								0
 ☆ 收藏夹 ↓ 下载 ■ 桌面 2 最近访问的位 		名称		^	修改 没有与搜索条件匹替	日期 配的项。	类型		大小
 □ 库 □ Git □ 视频 □ 图片 □ 文档 □ ① 示載 	Ŧ	•							•
文件名(N): 保存类型(T):	LED Proje	ct Files (*.u	vproj; *.uvpro	ojx)					•
🔊 隐藏文件夹						(R	存(S)	取消	

三、在 Select Device for Target 'Target 1' 对话框中,选择 ARM -- Cortex MO 内核, 然后点击 OK。

Select Device for Target 'Target 1'		x
CPU Software Packs Vendor: ARM Device: ARMCM0 Toolset: ARM Search:	▼	
ARM ARM Cortex M0 ARM Cortex M0 ARM Cortex M0 plus ARM Cortex M3 ARM Cortex M4 ARM Cortex M4 ARM Cortex M7 ARM SC000 ARM SC300 Cortex STMicroelectronics	Des <u>oription:</u> The Cortex-M0 processor is an entry-level 32-bit ARM Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including: - simple, easy-to-use programmers model - highly efficient ultra-low power operation - excellent code density - deterministic, high-performance interrupt handling - upward compatibility with the rest of the Cortex-M processor family.	* TC
	OK Cancel Hel	Lp

四、对工程添加固件库文件。固件库文件在 <u>GVM32F030 StdPeriph Lib V1.4.1\Libraries</u> 文件夹下,新建 main.c(可从 Examples 中拷贝)最终的工程文件结构如下:

🖃 🔧 Project: LED
🖨 🞥 Target 1
🖨 🦾 CMSIS
startup_gvm32f030.s
system_gvm32f030.c
🚊 🗁 User
main.c
🚊 🗁 StdPeriph_Driver
gvm32f030_gpio.c
gvm32f030_misc.c
gvm32f030_sysctrl.c

		Xtal (MHz): 1	2.0	Code C ARM	Generation Compiler:	Use default	compiler versi	on 💌
Operating system: IN	lone		-	_				
System Viewer File:					se Cross-I	Module Optimiza	tion	
ARMCM0.svd				I D U	se MicroL	IB [Big Endian	
Use Custom File								
Read/Only Memory	Areas			Read/	Write Men	nory Areas		
default off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
□ ROM1:			- C	Г	RAM1:			
ROM2:			0	Г	RAM2:			
□ ROM3:		í –	- C	Г	RAM3:		í –	
orchip			_		on-chip	1		
IROM1: 0x	0	0x8000	•	~	IRAM1:	0x10000000	0x1000	
IBOM2:			С	Г	IRAM2:			

五、配置项目的 Flash 和 RAM 的地址范围,打开项目配置,按如下图所示配置:

六、根据 GVM32F030 的具体型号,定义相应的宏定义。 GVM32F030x8,USE_STDPERIPH_DRIVER

)evice Target Output Listing User C/C++ Asm Linke	r Debug Vtilities
Preprocessor Symbols	
Define: GVM32F030x8,USE_STDPERIPH_DRIVER	
Undefine:	
Language / Code Generation	
Execute-only Code Strict ANSI C	<u>W</u> amings:
Optimization: Level 0 (-00)	int 🛛 🛛 All Warnings 💌
☐ Optimize <u>f</u> or Time ☐ <u>P</u> lain Char is Signed	🔲 Thum <u>b</u> Mode
Split Load and Store Multiple Read-Only Position Inde	ependent 🗌 No Auto Includes
✓ One ELF Section per Function	ependent 🗌 C99 Mode
IncludeLibraries\CMSIS\Include:Libraries\CMSIS\Devic	e\GV\gvm32f030;\\\Libraries\G\
Paths Miss	
Controls	
Compiler -ccpu Cortex-M0li -g -O0apcs=interworksplit_sections -I.	\\\Libraries\CMSIS\Include -I\
control \\Libraries\CMSIS\Device\GV\gvm32f030 -1\\Libraries\	GVM32F030_StdPeriph_Driver\inc
sung	
OK Cancel	Defaults
OK Cancel	Defaults
OK Cancel	Defaults
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H	Defaults
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" {	Defaults
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif	Defaults
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device	Defaults I
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application	Defaults D
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6)	Defaults 1
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application 77 #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */	Defaults I
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application /* #defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif	Defaults 1
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x6 */ #endif	Defaults 1
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif	Defaults 1 e used in your
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #endif	Defaults 1
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #if !defined (GVM32F030x8) && !defined (GVM32F030xx device used in #endif #if defined(GVM32F030x8) && defined(GVM32F030x6)	Defaults 1
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x6 */ #endif #if !defined (GVM32F030x6 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #endif #if defined(GVM32F030x8) && defined(GVM32F030x6) #error "Only one GVM32F030 device selected (in gvm32f030.h file)" #endif	Defaults 1 e used in your
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #endif #if defined(GVM32F030x8) && defined(GVM32F030x6) #error "Only one GVM32F030 device selected (in gvm32f030.h file)" #endif	Defaults 1 e used in your your application (in gvm32f030.h f
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #if defined(GVM32F030x8) && !defined(GVM32F030x6) #error "Only one GVM32F030 device selected (in gvm32f030.h file)" #if !defined USE_STDPERIPH_DRIVER /**	Defaults 1 e used in your your application (in gvm32f030.h f
OK Cancel #ifndefGVM32F030_H #defineGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ /* #define GVM32F030x6 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #endif #if defined(GVM32F030x8) && defined(GVM32F030x6) #error "Only one GVM32F030 device selected (in gvm32f030.h file)" #endif #if !defined USE_STDPERIPH_DRIVER /** * @brief Comment the line below if you will not use the peripherals In this case, these drivers will not be included and the annication	Defaults 1 e used in your your application (in gvm32f030.h f s drivers. tion code will
OK Cancel #ifndefGVM32F030_H #ifdefcplusplus extern "C" { #endif /* Uncomment the line below according to the target GVM32F030 device application */ #if !defined (GVM32F030x8) && !defined (GVM32F030x6) /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ /* #define GVM32F030x8 */ #endif #if !defined (GVM32F030x8 */ #endif #if !defined (GVM32F030x8) && !defined (GVM32F030x6) #error "Please select first the target GVM32F030xx device used in #endif #if defined(GVM32F030x8) && defined(GVM32F030x6) #error "Only one GVM32F030 device selected (in gvm32f030.h file)" #if !defined USE_STDPERIPH_DRIVER /** *@Dief Comment the line below if you will not use the peripherals In this case, these drivers will not be included and the applicat be based on direct access to peripherals registers	Defaults 1 e used in your your application (in gvm32f030.h f s drivers. tion code will

七、添加头文件包含路径。需要添加的路径有:

GVM32F030_StdPeriph_Lib_V1.4.1\Libraries\CMSIS\Include

<u>GVM32F030</u> StdPeriph Lib V1.4.1\Libraries\CMSIS\Device\GV\gvm32f030 GVM32F030 StdPeriph Lib V1.4.1\Libraries\GVM32F030 StdPeriph Driver\inc

Folder Setup	?	×
Setup Compiler Include Paths:	2 🗙	∱
\\Libraries\CMSIS\Include \\\Libraries\CMSIS\Device\GV\gvm32f030 \\Libraries\GVM32F030_StdPeriph_Driver\inc		
OK Cancel		

八、将 GVM32F030 的 Flash 烧写算法文件 GVM32F030_32K.FLM 拷贝到 Keil 的安装目 录 C:\Program Files (x86)\Keil_v5\ARM\Flash 下。

九、配置调试工具,这里以 Jlink 为例。

😗 Options for Target 'Target 1'	
Device Target Output Listing User C/C++ A	Asm Linker Debug Utilities
○ Use <u>Simulator</u> <u>with restrictions</u> <u>Settings</u> □ Limit Speed to Real-Time	© Use: J-LINK / J-TRACE Cortex
Load Application at Startup Initialization File:	Load Application at Startup Run to main() Initialization File: Edit
Restore Debug Session Settings	Restore Debug Session Settings Breakpoints Toolbox Watch Windows Memory Display System Viewer
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL Dialog DLL: Parameter: TARMCM1.DLL pCM0
OK	icel Defaults Help

点击 Setting,若弹出对话框选择 No

🔜 J-Link	V4.98e Device Selection
<u>^</u>	The selected device "ARMCM0" is unknown to this version of the J-Link software. In most cases, this is not a problem and can be safely ignored. Proper device selection is required to use the J-Link internal flash loaders for flash download or unlimited flash breakpoints. For some devices which require a special handling, selection of the correct device is important. Do you want to manually select a device ?
	In case of doubt, click "No".

选择 SW 模式

J-Link / J-Trace Adapter	_SW Device	
SN:	IDCODE Device Name	Move
Device: J-Link ARM	SWDI(🗿 OxOBB1 ARM CoreSight SW-DP	Up
HW : V8.00 dll V4.98e		Down
FW : J-Link ARM V8 compiled]		
orc. Max	💿 Automatic Detectic ID CODE:	
SW 💌 5 MHz 💌	C Manual Configurati Device Name:	
Auto Clk	Add Delete Update IR len:	
Connect & Reset Options Connect: Normal 💌 (eset: No 🔽 Reset after Conn	Cache Options Download C rmal Cache Code Verify Cache Memory Download	Dptions Code Downlo d to Flash
Connect & Reset Options Connect: Normal Veset: No V Reset after Conn Interface	mal ▼ Cache Options Download C ▼ Cache Code Verify ▼ Cache Memory Downlos	Dptions Code Downlo d to Flash Misc
Connect & Reset Options Connect: Normal Veset: No Veset after Conn Interface C USB C TCP/IP	Cache Options Download C Cache Code Cache Memory Download Settings	Pptions Code Downlo d to Flash Misc
Connect & Reset Options Connect: Normal Veset: No Reset after Conn Interface USB O TCP/IP Network IP-Addre Scan	Cache Options Download C Tmal Cache Code Verify Cache Memory Download Settings Port Autodetect	Dptions Code Downlo dd to Flash Misc <u>Flink Info</u>
Connect & Reset Options Connect: Normal Veset: No Reset after Conn Interface USB O TCP/IP Scan Character Conn TCP/IP Network IP-Addre 127	Cache Options Tral Cache Code Cache Code Cache Memory Download C Cache Memory Download	Dptions Code Downlo d to Flash Misc JLink Info JLink Cmd
Connect & Reset Options Connect: Normal Veset: No Veset: Normal Veset: No Veset after Conn Interface USB O TCP/IP Scan State: ready	Cache Options Tral Cache Code Cache Memory Download C Cache Memory Settings Settings Port O . 0 . 1 : 0 Ping	Dptions Code Downlo d to Flash Misc TLink Info JLink Cmd

52

在 Flash Download 标签页中,将 New Device 256kB Flash 选中,Remove 掉。 RAM for Algorithm 配置为 Start:0x10000000 Size:0x1000

Cortex JLink/JTrace Target Driver Setup	×
Debug Trace Flash Download Download Function O Erase Full Chip Image: Program Image: Comparison of Erase Sectors Image: Verify Image: Start: Ox1000000 Image: Comparison of Erase Image: Reset and Run Image: Start: Ox1000000	
Programming Algorithm Bescription Device Size Device Type Address Range GVM32F030 32kB Flash 32k On-chip Flash 00000000H - 00007FFFH	
Start: 0x0000000 Size: 0x00008000 Add Remove	
确定 取消 应用(A

Description	Flash Size	Device Type	Origin	
C2538x 256 KB	256k	On-chip Flash	MDK Core	
C2538xx 512 KB	512k	On-chip Flash	MDK Core	
CY8C40xx (16kB) Flash	16k	On-chip Flash	MDK Core	
CY8C42xx 1MACRO (32kB)	32k	On-chip Flash	MDK Core	
CY8C5xxx Flash	256k	On-chip Flash	MDK Core	
CY8C5xxx Configuration	32k	On-chip Flash	MDK Core	
CY8C5xxx CFG NVL	4B	On-chip Flash	MDK Core	
CY8C5xxx EEPROM	2k	On-chip Flash	MDK Core	
CY8C5xxx Flash Protection	256B	On-chip Flash	MDK Core	
CY8C5xxx WO NVL	4B	On-chip Flash	MDK Core	
EFM32 Gecko/Tiny Gecko	128k	On-chip Flash	MDK Core	
EFM32 Giant Gecko	1M	On-chip Flash	MDK Core	
EFM32 Leopard Gecko	256k	On-chip Flash	MDK Core	
EFM32 Wonder Gecko	256k	On-chip Flash	MDK Core	
EFM32 Zero Gecko	32k	On-chip Flash	MDK Core	
GVM32F030 32kB Flash	32k	On-chip Flash	MDK Core	
		1005000 001/ 51 M		
:\Program Files (x86)\Keil_v5\/	ARM Mash \GVN	132F030_32K.FLM		

点击 Add,在列表中选中 GVM32F030 32kB Flash,点击 Add

最后点击确定。

十、至此配置完成。可以编译调试工程了。