

# SYM-Link 仿真器用户手册

适用范围:

● SYM32 全系列芯片



#### 1 概述

SYM-Link 是砂翊微电子推出的一款针对 SYM32 系列 MCU 的集仿真器和编程器于一体的开发 工具。支持在线仿真、在线编程及离线编程,该工具小巧便携、性能可靠、操作简单,适用于 工程研发及生产批量编程。其主要有以下特点:

- USB2.0 全速免驱接口
- 支持 CMSIS-DAP SWD 在线仿真功能
- 可选择为目标板供电模式 / 目标板自供电模式
- 具备电平转换电路,适配目标板任意工作电压
- 支持全速运行、单步调试和断点设置等
- 支持在线升级固件
- 支持在线编程
- 支持离线编程



2 硬件连接

Step1. 使用 USB 线缆将 SYM-Link 连接到电脑,如下图所示:



连接成功后可以在设备列表中查看本设备。其查看路径为:【设置】-【蓝牙和其他设备】-【设备】-【其他设备】,如下图所示:



Step2. 将仿真器的调试接口和目标板的 SWD 接口连接,如下图所示。





Step3. 通过仿真器的跳线可控制目标板的供电情况:

- 向目标板供电模式:将 SYM-Link 的 JP3 的【VTC】短接到【3V3】,此模式下 SYM-Link 通过 VT 向目标板提供 3.3V 电压。
- 目标板自供电模式:将 SYM-Link 的 JP3 的【VTC】悬空,此模式下目标板通过 VT 向 SYM-Link 的电平转换电路供电。



注意:编程器最多可向目标板提供 100mA 电流;若目标板工作电流大于 100mA,请采用外部电源对目标板进行供电。



#### 3 在 MDK 环境使用 SYM-Link

本章节介绍如何配置 MDK(Keil) 集成开发环境,实现通过 SYM-Link 仿真器对目标芯片进行 仿真调试。

Step1. 连接仿真器:参照硬件连接章节,连接 SYM-Link 仿真器、目标板和电脑。



Step2. 选择仿真器:打开 MDK(Keil)工程,点击【Project】-【Options for Target】在弹出的对话框中按照下图所示进行配置。

Ψ,	\Examples\GPIO\GPIO_Blink\MDK\GPIO_Blink.uvprojx - μVision	
File Edit View Project Fla	ash Debug Peripherals Tools SVCS Window Help	
📄 💕 🛃 🥔 🕺 🛍	ッ で   ← →   陀 隐 隐 陰 譯 譯 /////////////////////////////	
🕸 🕮 🗰 🗳 - 📖 🔤 🤂	SPIO_Blink 🖂 🕺 📥 🖶 🗇 🎕	
Project		
🗉 🏘 Project: GPIO_Blink		` _
🖻 💭 GPIO_Blink	Device Target Output Listing User   C/C++   Asm   Linker Debug   Utilities	1
Startup     Startup     system_sy	○ Use Simulator       with restrictions       Setting3       Image: Setting Seting Setting Setting Setting Setting Setting Setting	
User	Load Application at Startup     Run to main()     Initialization File:     Initialization F	ıc
<ul> <li>main.c</li> <li>interrupts</li> <li>Driver</li> <li>sym32f03</li> <li>sym32f03</li> <li>sym32f03</li> <li>sym32f03</li> </ul>	Restore Debug Session Settings       Edit         Restore Debug Session Settings       Restore SiLabs UDA Debugger         Image: Silabs UDA Debugger       Silabs UDA Debugger         Image: Silabs UDA Debugger       Image: Silabs UDA Debugger	
Doc CMSIS	CPU DLL:     Parameter:     Driver DLL:     Parameter:       SARMCM3.DLL     SARMCM3.DLL     SARMCM3.DLL	
	Dialog DLL:     Parameter:     Dialog DLL:     Parameter:       DARMCM1.DLL     -pCM0+     TARMCM1.DLL     -pCM0+	
	Warn if outdated Executable is loaded Wanage Component Viewer Description Files	
	OK Cancel Defaults Help	



Step3. 配置仿真器: 在弹出的对话框中按照下图第1、2、3处进行配置,即可看到第4处所示的连接成功提示。

CMSIS-DAP Cortex-M Target Driver S	etup	×
Debug Trace Flash Download CMSIS-DAP - JTAG/SW Adapter SYM-Link CMSIS-DAP	SW Device 4 IDCODE Device Name SWDIO Ox0BC11477 ARM CoreSight SW-DI	P Move
Firmware Version 2.0.0 2 SWJ Port SW Max Clock: 1MHz	Automatic Detection ID CODE:     Manual Configuration Device Name:     Add Delete Update	AP: 0x00
Debug Connect & Reset Options Connect Normal ▼ Reset ▼ Reset after Connect Log Debug Accesses Sta	Autodetect  Cache Options Cache Code Cache Memory	Download Options Verify Code Download Download to Flash
	OK Cancel	Help

Step4. 配置编程方式:点击【Flash DownLoad】标签栏,根据需要选择第2处所示的下载的流程。根据实际芯片容量设置下图 3 处的 RAM Size。根据实际芯片型号设置下图 4 处的 Flash 下载算法。设置完成之后,点击【OK】关闭上述对话框。

CMSIS-DAP Cortex-M Target Driver Setup	×
Debug   Trade (Flash Download)	
Download Function Erase Full Chip Erase Sectors Do not Erase RAM for Algorithm Start 0x20000000 Size: 0x00002000 Size: 0x00002000	
Programming Algorithm	
Description     Device Size     Device Type     Address Range	
SYM32F030 /2k On-chip Flash 00000000H - 00011FFFH	
Start 0x00000000 Size: 0x00012000	
Add Remove	
OK Cancel Help	



Step5. 程序下载:如下图所示点击【DownLoad】,即可下载编译完成的程序到目标芯片。

File Edit View Project Flash Debug Peripherals	Tools SVCS	Window Help
n 29 19 (→ →   20 10 <del>  20 40</del> ) ×   € 🖬 🖬 🗃 🗐	後年年	//= //= 🖉 audio_hid_class_han 🗸 🗟 🥐 🔍 🗸 + 🕒 📀 🔗 🚓 🕶 🖾 🔍
🕸 🎬 📽 🕶 🔜   🗱   GPIO_Blink 🛛 🗸 🕺	🛔 🗟 🔶 🐡	> 🚳
roject	<b>д 🛛</b>	main.c startup_sym32f030.s
■ Project: GPIO_Blink		59
🖻 🐖 GPIO_Blink		60
E G Startun		61 - /***********************************
		62 ** \brief Main function of project
system_sym32f030.c		63 **
startup_sym32f030.s		64 ** \return uint32 t return value, if needed
🖃 🗁 User		65 **

Step6. 仿真调试:如下图所示点击【Debug】按钮开始进入仿真调试界面。在仿真调试界面可以进行单步执行、全速运行、设置断点等操作,如下图所示。





#### 4 在 EWARM 环境使用 SYM-Link

本章节介绍如何配置 EWARM(IAR) 集成开发环境,实现通过 SYM-Link 仿真器对目标板芯片进行仿真 调试。



Step1. 连接仿真器:参照硬件连接章节,连接 SYM-Link 仿真器、目标板和电脑

Step2. 选择仿真器: 打开 EWARM(IAR)工程,点击【Project】-【Options】在弹出的对话框中 按照下如所示进行配置。

😌 GPIO_Blink -	IAR Embedded Workbench IDE - Arm 8.32.1	
File Edit <b>1</b> iew	Project J-Link Tools Window Help	
11 11 🖬 🖬 🎽		
Workspace	Options for node "GPIO_Blink"	×
Release		
Files  GPI0_Blink  GDI  GDI  CDI0_Blink  C	Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions There Custom Build Build Actions Device description file Perices St-LINK Third-Party Driver TI MSP-FET TI XDS Device default SPROJ_DIR\$\\\\.\.\deSupport\EWARM\SYM32F030.svd	Factory Settings
		51



Step3. 配置仿真器:点击左侧项目栏中的【CMSIS DAP】,在右侧标签页中点击【Interface】,选择连接模式为"SWD",如下图所示:

Options for node "GPIO Blink"



 $\times$ 



## Step4. 存储器配置:点击【CMSIS-DAP】-【Memory Configuration】在弹出的对话框中根据 实际芯片设置 Flash 和 RAM,如下图所示:

Edit Vier	V Project CMSIS-	DAP Tools W	indow Help	>毎月人	V > C >	B 🖲 😑 🖸 🛛	
🖹 🔛 🛃 mory Cor is importar	figuration	urget memory is de	- < Q	> \$ HE <	V > 2 V	D 🕶 🖸	, I
mory Cor is importar	nfiguration	urget memory is de	7 17 11 1			_	
is importar	it for C-SPY that the ta	urget memory is de					
t is importar	it for C-SPY that the tε	arget memory is de					
			escribed fully and	accurately. Your	r project settings	normally specify thi	is, as follows.
Factory ran	jes						
Selected	, de∨ice description file	e in Project Option	s:				
in second	ran company	Examples\GF	PIO\GPIO_Blink\E	WARM\\\\\Id	eSupport\EWAF	N/SYM32F030.svd	ł
	the following defaults						
it specilies	the following detault	nemory ranges.					1
Zone	Name	9	Start 1	End	Type	Size	Extra
				Flas	sh和RAI	M的大小	<b>焉</b> 要
Jsed range	s			Flas 根描	<b>sh和RA</b> I 配示态	M的大小智	<b>焉</b> 要 署
Jsed rage	s memory configuratio	1 that will be used.	You can modify t	Flas	<b>sh和RA</b> I 居实际芯	M的大小 片进行配	焉要 置
Jsed rage This is the	s memory configuration	1 that will be used.	You can modify t	Flas	sh和RAI 皆实际芯	M的大小领 片进行配	売 要 置
Jsed rage This is the Zone	s memory configuration Start	1 that will be used.	You can modify t Cache Type	Flas hise needed. Size	sh和RAI 居实际芯	M的大小 片进行配	焉 要 置
Jsed range This is the Zone Memory	s memory configuration Start 0x00000000	1 that will be used. End 0x00011FFF	You can modify t Cache Type ROM Tlash	Flas 根胡 his es needed. Size 72 kbyte	sh和RAI 官实际芯 es	M的大小 片进行配 <sup>Comment</sup> Flash	<b>焉</b> 要 置
Jsed rage This is the Zone Memory Memory	s memory configuration Start 0x00000000 0x20000000	1 that will be used. End 0x00011FFF 0x20001FFF	You can modify t Cache Type ROM Flash RAM	Flas his as needed. Size 72 kbyte 8 kbytes	sh和RAI 居实际芯	M的大小 片进行配 <sup>Comment</sup> Flash RAM	焉要 置
Jsed rage This is the Zone Memory Memory Memory Memory	s memory configuration Start 0x0000000 0x2000000 0x4000000	1 that will be used. End 0x00011FFF 0x20001FFF 0x5FFFFFFF	You can modify t Cache Type ROM Flash RAM Uncached SF	Flas his as needed. Size 72 kbyte 8 kbytes R 512 Mby	sh和RAI 古实际芯 Extra es s tes	M的大小 片进行配 Comment Flash RAM Periphera	寄要 置



### Step5. 程序下载:点击【Project】-【Download】-【Dwonload active application】,即可下载编译完成的程序到目标芯片。

SPIO\_Blink - IAR Embedded Workbench IDE - Arm 8.32.1

File Edit View	Pro	ject CMSIS-DAP Tools W	indow Help	_
12 🖻 🖬 🗿 🛔	<b>D</b>	Add Files		、> 与 旺 く 🟮 > 🔄 🖻 🌘 🖷 🔕 🕘 晶 📜
Workspace	6	Add Group		30 hal systick.c
<u> </u>	[+]	Import File List		
Kelease		Add Project Connection		
Files		Edit Configurations		*****
	×	Remove		t main( <b>void</b> )
–⊕ 🖬 Driver				
	U	Create New Project		PIO InitTypeDef GPIO InitStruct;
L = Oser	0	Add Existing Project		
	Φ	Options	Alt+F7	<pre>lemClr(&amp;GPIO InitStruct, sizeof(GPIO Init</pre>
		Version Control System	•	
				//Open GPIOx Clk
		Make	F/	AL_SYSCTRL_GPIOA_CLK_ENABLE();
		Compile	Ctrl+F/	
		Rebuild All		(/初始化SysTick
	2	Clean		AL_InitTick(800000);
	•	Batch build	F8	_
		C-STAT Static Analysis	+	PIO_InitStruct.Pins = GPIO_PIN_7;
	8	Stop Build	Ctrl+Break	PIO_InitStruct.Mode = GPIO_MODE_OUTPUT_
			CL IV D	PIO_InitStruct.Speed = GPIO_SPEED_HIGH;
	U	Download and Debug	Ctrl+D	AL_GPIO_INIt(SYM_GPIOA, &GPIO_INItStruct
		Debug without Downloading		
		Attach to Running larget	CL L D	
	9	Make & Restart Debugger	Ctrl+R	WAL OPTO Write Ding (SVM OPTON OPTO DI
	Č	Restart Depugger	Ctri+Sniπ+K	
	Ч	Download	•	Download active application
		SFR Setup		
		CMSIS-Manager		crase memory



Step6. 仿真调试:点击【Download and Debug】按钮进入仿真调试界面,在仿真调试界面可以进行单步执行、全速运行、设置断点等操作,如下图所示。

sembly CMSIS-DAP Tools Window Help

<ul> <li>&lt; Q, &gt; ₅ ⊨≡ &lt; </li> </ul>	> < >			
Disassembly 🗸 🕈 🗙	main.c X	<b>•</b>	Registers 3	<b>→</b> 中 ×
Go to 🗸 🗸	main()	fo	<find register=""></find>	~
Disassamhlu	66	**		Value
	67	***************************************		Value 0::0000EE2E
0w270: 0w4811	68	int32 t main( <b>void</b> )		0x0000FF7F
0x270. 0x4011 0x272: 0xf7ff 0xffal	60		SPEED	0x00000000
GPIO InitStruct Pins =	09	1	# PDR	0×00000000
0x276: 0x4668	70	GPIO_InitTypeDef GPIO_InitStruct;	± PUR	0x00000000
0x278: 0x2180	71		▲ FRH	0x00000000
0x27a: 0x80c1	72	MemClr(&GPIO_InitStruct, sizeof(GPIO	🗄 AFRL	0x00000000
GPIO_InitStruct.Mode =	72		H ANALOG	0x00009F7F
0x27c: 0x2120	15		<b>∃</b> DRIVER	0x00000000
0x27e: 0x7001	74	//Open GPIOx Clk	* RISEIE	0x00000000
GPIO_InitStruct.Speed =	75	HAL SYSCTRL GPIOA CLK ENABLE();	<b>FALLIE</b>	0x00000000
0x280: 0x2101	76		<b>HIGHIE</b>	0x00000000
0x282: 0x7101	70	11 tatta lka mi 1	+ LOWIE	0x00000000
HAL_GPIO_Init(SYM_GPIOA	//	// <i>WIXEWLSYSTICK</i>	# ISR	0x00000000
0x284: 0x2490	78	HAL InitTick(8000000);		0x0000FFFF
0x206: 0x0564	79	_	TITEP	0x00000000
0x28a 0x0020	00	CDIO InitStruct Ding - CDIO DIN 7.		0x00000000
0x28c: 0xf7ff_0xff18	00	GFIO_INICSCIUCC.FINS = GFIO_FIN_/,		0x00002000
HAL GPIO WritePins(S	81	GPIO_InitStruct.Mode = GPIO_MODE_OUI	PIN15	0
0x290: 0x2201	82	GPIO InitStruct.Speed = GPIO SPEED HI	- PIN14	0
0x292: 0x2180	83	HAL GPIO Init (SYM GPIOA, &GPIO InitSt	- PIN13	0
0x294: 0x0020	0.4		- PIN12	0
0x296: 0xf7ff 0xff7:	04		PIN11	0
HAL_Delay(100);	85	while (1)	- PIN10	0
Ox29a: 0x2064	86	] {	- PIN9	0
0x29c: 0xf000 0xf83e	87	HAL GPTO WritePins (SYM GPTOA GPT	- PIN8	0
HAL_GPIO_WritePins(			PIN7	1
0x2a0: 0x2200	• 88	HAL_DELAY(100);	- PIN6	0
0x2a2: 0x2180	89	HAL_GPIO_WritePins(SYM_GPIOA, GPI	PIN5	0
Ux2a4: UxUU2U	90	HAL Delay(100);	PIN4	U
UX286: UXI/II UXII6:	01		PIN3 DIN3	0
0w2aa: 0w2064	91		PIN2	0
0x2ac: 0x2004 0x2ac: 0xf000 0xf834	92	}	PINO	0
0x2b0: 0xe7ee	93	- }	# ODRLOWBYTE	0x80
0x2b2: 0xbf00	94		<b>DORHIGHBYTE</b>	0x00
0x2b4: 0x40010030	QE		BRR	0x00000000
0x2b8: 0x007a1200	95		BSRR	0x00000000
if ( *(uint16_t*)(0x000;	96 6	/**************************************	<b>⊞ TOG</b>	0x0000000



### 版本记录

版本	修订日期	修订说明
Rev1.0	2022-07-22	初始版本