## AG32 开发板使用入门



在使用开发板前,请确认已经安装好开发环境。 安装环境过程,请参考文档《AG32开发环境搭建.pdf》

#### 一、上电:

给开发板 5V 供电,打开开关,可以看到电源旁边的小红灯亮起。

#### 二、使用 example 例程:

打开 example 例程,在 SDK 路径下:

D:\xxxxx\AgRV\_pio\platforms\AgRV\examples\example (注意这里的两重 example)

(J)	资源管理器 ···	🄯 platformio.ini 🗙 🛛 c example_gpio.c
-	∨EXAMPLE [ຊີເຊີບອີ	o platformio.ini
Q	> .pio	1 [setup]
/-	> .vscode	<pre>2 boards_dir = boards</pre>
0	) src	<pre>3 board = agrv2k_103</pre>
Po	<ul> <li>gitignore</li> </ul>	<pre>4 board_logic.ve = example_boar 5 #board logic.compress = true</pre>
$\sim$	example_board_bank.ve	6
<b>ø</b> ⁄	■ example_board.asf	7 #ips_dir =/ips
_	⊑ example_board.post.asf	8 #ip_name = analog_ip
ß	≡ example_board.pre.asf	9 #logic_dir = logic
	E example board.ve	10
do	nlatformio ini	11 framework = agrv_sdk
~	Platformioan	12 program = agm_example

由于开发板使用的是 407 芯片,需要先修改 platformio.ini 中的 board 类型:

### board = agrv2k\_103

修改为:

#### board = agrv2k\_407

为了验证简单化,可以先把 example\_board.ve 中的配置暂时删除,只留下前两项: SYSCLK 100 HSECLK 8

三、烧录 VE 文件和代码 bin: 烧录程序可选 jlink 和官方烧录器两种; (串口烧录这里不讨论)

#### 如果使用 jlink:

- 连线:用 jlink 线连接开发板的 jlink 接口。
- 配置:确认 platformio.ini 中的以下两行配置是打开的:

# debug\_tool = jlink-openocd upload\_protocol = jlink-openocd 安装插件:第一次使用 jlink,需要先安装插件【zadig-2.8.exe】,安装参考下图: (该插件在 sdk 路径的根目录下)

📴 Zadig	3 <u></u>	
<u>D</u> evice <u>Options</u> <u>H</u> elp		
J-Link 1		∼ □Edit
Driver Jlink (v2.70.8.0)	j. 1. 7600. 16385) More WinUs	Information SB (libusb)
USB ID 1366 0101 Repla	ce Driver	-win32 K
	4 WinUs	<u>SB (Microsoft)</u>
i devices found.		Zadig 2.7.765

注:如果第一步从下拉列表中找不到【J-Link】项,可以把下拉列表打开,插拔 Jlink 几次,找列表中的变化项。列表中的那个变化项,就是要更新驱动的项。

```
烧录:
```

新开发板第一次使用,要先烧录 VE 配置。(不烧录 VE 而先烧录程序 bin,会报错) 烧录 VE:



烧录程序:

Ch	PLATFORMIO ···	🏺 platformio.ini × 🛛 🖺 example_board.v			
U.	~ project tasks ひ 🗟 🗗	🤯 platformio.ini			
O	> 🗟 Default	1 [setup]			
/-	〉 등리 dev	2 boards_dir = boards 3 board = agrv2k_103			
0					
Po	✓ General	<pre>4 board_logic.ve = example_</pre>			
$\sim$	O Build	6			
æ^	O Upload	7 #ips_dir =/ips			
-	O Monitor	<pre>8 #ip_name = analog_ip</pre>			
В	O Upload and Monitor	9 #logic_dir = logic			
6926553	O Clean	10			
2		11 framework = agrv_sdk			
<b>V</b>	O Clean All	12 program = agm_example			
	> Dependencies	13			

#### 仿真:

点击仿真按钮,可以进入仿真调试。可单步运行到 main 函数的结尾。

×	文件(E) 编辑(E)	选择(S)	查看(V)	转到( <u>G</u> )	运行(图)	终端①	帮助(出)	example.c - example - V
Ω	PIO De	bug 🗸	@ ··	ᅘ platfor	mio.īni	<b>≅</b> exa	mple_board.ve	C example.c ×
	∨ 鋰		ð	src > C	example.c	> 🕅 main	(void)	
ک 13 %				25 ] 26 27 i 28 { 29 30	) int main ( // This board :	(void) s will i init():	nit clock and	uart on the board
9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	✓ 蓋視			31 32 33 34 35 36	// The // GPIC plic_is // Any INT_Set	default D0_isr() sr[BUT_G interru tIRQThre	<pre>isr table is , and can be r PIO_IRQ] = But pt priority ne shold(MIN_IRQ_</pre>	plic_isr. The default e-assigned. :ton_isr; eds to be greater than PRIORITY);



(D)	▷ PIO Debug 🗸 🐯	💀 🍯 platform	nio.ini	■ example_board.ve	C example.c ×			
-	∨ 委量	src > C es	xample.c > (	🕈 main(void)				
Q	∨ Local	27 int main(void)						
	> Global	28	// Thic	uill init clock and	want on the board			
દુરુ	> Static	30	board_in:	it();	uart on the board			
0000		31						
2		32	// The de	efault isr table is	plic_isr. The default entr			
21		33	// GPI00	isr(), and can be	re-assigned.			
-	✓ 监視	34	plic_isr	[BUT_GPIO_IRQ] = Bu	tton_isr;			
RY .		35	// Any in	nterrupt priority n	eeds to be greater than MIN			
		36	INT_SetI	RQThreshold(MIN_IRQ	_PRIORITY);			
		37	// Enable	e interrupt from BU	T_GPIO			
		D 38	INT_Enab.	leIRQ(BUT_GPIO_IRQ,	PLIC_MAX_PRIORITY);			
		20						

#### 如果使用官方烧录器:

连线:

连线后的样子如下图:



配置:

确认 platformio.ini 中的以下两行配置是否打开

debug\_tool = cmsis-dap-openocd

upload\_protocol = cmsis-dap-openocd

使用官方烧录器, PC 是免驱动的。

烧录:

烧录 VE 和烧录程序 bin,同上边的 Jlink 的烧录过程。 不再重复,请参考上边。

调试:

调试的启动和过程,也同上边的 Jlink 的调试过程。 不再重复,请参考上边。

四、查看 led 灯:

在 example 样例程序的 main 函数中,最后是调用函数 TestGpio()。进入 TestGpio()函数,里边是对 LED 灯的闪灯操作。

由于我们在第三步时,删除了 example\_board.ve 里的全部 IO 配置。现在把 led 的 pin 脚 映射恢复回来。copy 下边的设置到 ve 文件:

GPIO4\_1 PIN\_34 # LED1 GPIO4\_2 PIN\_33 # LED2 GPIO4\_3 PIN\_32 # LED3 GPIO4\_4 PIN\_31 # LED4

copy 后效果如下图:

🈻 platformio.ini		≡ exar	nple_board.ve	×	c example.c
≣ exa	ample_board.	/e			
1	SYSCLK 1	90			
2	HSECLK 8				
3					
4	GPI04_1	PIN_34 #	LED1		
5	GPI04_2	PIN_33 #	LED2		
6	GPI04_3	PIN_32 #	LED3		
7	GPI04_4	PIN_31 #	LED4		
8					
9					

此时,再次烧录 ve 文件。 (烧录 VE 的方式,如上边第三步)

烧录成功后,可以看到左下角的几个 led 闪烁起来。

五、查看 log 输出:

在以上的基础上,修改以下三项:

1. Platformio.ini 中:

确认 logger\_if 配置是打开的:

```
logger_if = UART0
build_flags = -DBAUD_RATE=115200
```

以上两项分别设置: log 输出通过 uart0 输出、输出的波特率是 115200.

2. Example\_board.ve 中:

Copy 以下的串口 IO 配置到 ve 中去:

#### UARTO\_UARTRXD PIN\_69

UARTO\_UARTTXD PIN\_68

修改后图示如下:

🤴 plati	formio.ini	$\equiv$ example_board.ve $\times$	c
≣ exa	mple_board.ve	e	
1	SYSCLK 10	0	
2	HSECLK 8		
3			
4	GPI04_1 P	PIN_34 # LED1	
5	GPI04_2 P	PIN_33 # LED2	
6	GPI04_3 P	PIN_32 # LED3	
7	GPI04_4 P	PIN_31 # LED4	
8			
9	UARTØ_UAR	TRXD PIN_69	
10	UARTØ_UAR	TTXD PIN_68	
11			

3. 在 example\_gpio.c 中的 TestGpio()函数中, while(1)里增加一句 log:



最后,烧录 VE,烧录程序 bin。

然后,用串口线,接到开发板的串口 0(上边的那组)上,在 PC 端的串口工具(波特率 115200)上可以看到 log 的输出信息。

以上,只是展示了拿到开发板后验证 LED 灯和 log 通过串口 0 输出的样例。 更多的驱动使用,请参考文档《AG32 驱动的使用.pdf》