

# OPL1000

ULTRA-LOW POWER 2.4GHZ WI-FI + BLUETOOTH SMART SOC

## DEVKIT Getting Start Guide



OPULINKS

<http://www.opulinks.com/>

Copyright © 2017-2018, OpuLinks. All Rights Reserved.

---

OPL1000-DEVKIT-getting-start-guide-R02 | Version V1.2

Date	Version	Contents Updated
2018-05-10	0.1	<ul style="list-style-type: none"> <li>Initial Release</li> </ul>
2018-05-17	0.2	<ul style="list-style-type: none"> <li>Update section 3.1.2, 3.2.2 because download tool use method has been updated.</li> <li>Add Figure10 to introduce how to verify AT UART is working</li> <li>Update Figure 12 content</li> <li>Add section 3.4 to introduce how to enable APS UART log info output</li> <li>Split section 3.3 to 3.3.1 and 3.3.2.</li> <li>Add section 3.3.2 to introduce J-link emulator selection</li> </ul>
2018-05-31	0.3	<ul style="list-style-type: none"> <li>Mini USB port will be acted as APS UART port when downloaded user APP after using v1.0.1.19 SDK</li> </ul>
2018-06-05	0.4	<ul style="list-style-type: none"> <li>Add a new section 3.1</li> <li>Update section 3.2.2.</li> </ul>
2018-07-13	0.5	<ul style="list-style-type: none"> <li>Update section 3.2</li> </ul>
2018-08-03	0.6	<ul style="list-style-type: none"> <li>Update section 2 and section 3.5 for A1 chip</li> </ul>
2018-08-15	0.7	<ul style="list-style-type: none"> <li>Update section2, figure2;</li> <li>Update section3.2 and section 3.3</li> </ul>
2018-11-09	0.8	<ul style="list-style-type: none"> <li>Update section3.4.1 and section3.4.2 to enable ICE interface</li> </ul>
2018-11-20	0.9	<ul style="list-style-type: none"> <li>Update section3.4.1 to disable watchdog</li> </ul>
2019-04-22	1.0	<ul style="list-style-type: none"> <li>Add LM80 introduction</li> </ul>
2019-06-21	1.1	<ul style="list-style-type: none"> <li>Update description of AT and APS serial ports.</li> </ul>
2019-07-08	1.2	<ul style="list-style-type: none"> <li>Update Fw_binary folder to FW_Pack folder</li> </ul>

TABLE OF CONTENTS

1. 介绍 2

1.1. 文档应用范围 2

1.2. 缩略语 2

1.3. 参考文献 2

2. DEVKIT 和 LM80 概要介绍 3

2.1. DEVKIT 概要介绍 3

2.2. LM80 概要介绍 4

3. 使用 DEVKIT 6

3.1. 安装 USB 驱动 6

3.1.1. Win10/Win7 系统驱动安装 6

3.1.2. WinXP/Vista 系统驱动安装 11

3.2. APS 串口连接和使用 13

3.2.1. APS 串口连接 14

3.2.2. 通过 APS 观察固件调试打印信息 14

3.3. AT 串口连接和使用 15

3.3.1. 通过 AT 串口更新固件 15

3.3.2. 使用 AT 串口执行 AT 指令 17

3.4. SWD 端口 18

3.4.1. 禁用 watchdog 18

3.4.2. M3 ICE 端口连接 19

4. 使用 LM80 21

4.1. 安装 USB 驱动 21

4.1.1. Win10/Win7 系统驱动安装 21

4.1.2. WinXP 系统驱动安装 24

4.2. APS 串口连接和使用 25

4.2.1. APS 串口连接 25

4.2.2. 通过 APS 观察固件调试打印信息 26

4.3. AT 串口连接和使用 26

4.3.1. 通过 AT 串口更新固件 27

4.3.2. 使用 AT 串口执行 AT 指令 29

4.4. SWD 端口 29

4.4.1. 禁用 watchdog 29

4.4.2. M3 ICE 端口连接 30

LIST OF FIGURES

Figure 1: DEVKIT 板组成介绍 ..... 3

Figure 2: DEVKIT 扩展 IO map..... 4

Figure 3: LM80 组成介绍 ..... 4

Figure 4 : LM80 扩展 IO map ..... 5

Figure 5: DEVKIT 设备端口 ..... 6

Figure 6: Win10/Win7 驱动安装 – 搜索驱动程序..... 7

Figure 7 : Win10/7 驱动安装搜索..... 7

Figure 8: Win10/Win7 驱动安装 – 选择从驱动程序列表选取 ..... 7

Figure 9: Win10/Win7 驱动安装 – 为 CP210x 更新驱动程序..... 8

Figure 10: Win10/Win7 驱动安装 –指定安装目录 ..... 9

Figure 11: Win10/Win7 驱动安装 – 选择 USB inf 文件..... 9

Figure 12: Win10/Win7 驱动安装 – 驱动更新完成 ..... 9

Figure 13: Win10/Win7 驱动安装 – 检查驱动版本信息 ..... 10

Figure 14: WinXP/Vista 驱动安装 – 执行驱动安装程序 ..... 11

Figure 15: WinXP/Vista 驱动安装 – 接受协议 ..... 11

Figure 16: WinXP/Vista 驱动安装 – 检查驱动版本信息..... 12

Figure 17: DEVKIT APS 串口连接 ..... 13

Figure 18: APS 串口接线实例..... 14

Figure 19: APS 串口输出打印信息 ..... 14

Figure 20: DEVKIT AT 串口设备 ..... 15

Figure 21:载入 M3/M0 Bin 文件进行合并操作..... 15

Figure 22: Patch Bin 文件下载..... 16

Figure 23: 启动后 APS 串口输出 log 信息..... 16

Figure 24: 执行 AT 指令 ..... 17

Figure 25: 禁用 watchdog..... 18

Figure 26: DEVKIT 板上 M3 ICE 连线图..... 19

Figure 27: J-link ICE 仿真器正确识别 ..... 19

Figure 28 : LM80 串口信息 ..... 21

Figure 29 : WIN7/10 安装 CH340 驱动语言选择 ..... 21

Figure 30 : 安装界面..... 22

Figure 31 : 驱动安装进行界面 ..... 22

Figure 32 : 驱动安装完成界面 ..... 23

Figure 33 : 驱动程序..... 24

Figure 34 : XP 驱动安装界面 ..... 24

Figure 35 : 驱动安装成功界面 ..... 25

Figure 36: APS 串口接线实例..... 25

Figure 37: APS 串口输出打印信息 ..... 26

Figure 38: DEVKIT AT 串口设备 ..... 26

Figure 39:载入 M3/M0 Bin 文件进行合并操作..... 27

Figure 40: Patch Bin 文件下载..... 27

Figure 41: 启动后 APS 串口输出 log 信息..... 28

Figure 42: 执行 AT 指令 ..... 29

Figure 43 : Watchdog 禁用开关..... 30

Figure 44 : LM80 板上 M3 ICE 信号接线图 ..... 30

Figure 45 : LM80 板上 M3 ICE 连线图 ..... 31

Figure 46: J-link ICE 仿真器正确识别 ..... 31

Figure 47: 断点设置调试..... 31

LIST OF TABLES

Table 1: DEVKIT M3 SWD 信号连接 ..... 19

Table 2: LM80 M3 SWD 信号连接 ..... 30

# 1. 介绍

## 1.1. 文档应用范围

OPL1000 DEVKIT 用于评估 OPL1000 芯片的功能、开发应用程序。本文介绍了 DEVKIT 和 LM80 板子的组成，如何使用 DEVKIT 及 LM80 板子提供的端口进行固件下载和应用程序调试。DEVKIT 和 LM80 都采用 OPL1000 低功耗芯片，DEVKIT 由开发母版和子模块组成，而 LM80 子板作为最小系统，可以单独使用，模板仅仅保证了电压的稳定，以及串口转换功能。

## 1.2. 缩略语

Abbr.	Explanation
APP	APPLication 应用程序
APS	Application Sub-system 应用子系统，在本文中亦指 M3 MCU
AT	Attention 终端命令指令集
DevKit	Development Kit OPL1000 评估开发板
EVB	Evaluation Board 评估板
FW	FirmWare 固件，处理器上运行的嵌入式软件
ICE	In-Circuit Emulator 在线仿真调试工具
LM80	由亮明科技制作 LM80 板
RX	Receive 接收
SWD	Serial Wire Debug 串行线调试
TX	Transmit 发送

## 1.3. 参考文献

[1] Download tool 使用指南 OPL1000-patch-download-tool-user-guide.pdf

[2] 亮明科技官网：<http://www.lbluetooth.com>

[3] 亮明官方淘宝店铺：<https://shop114109610.taobao.com/>

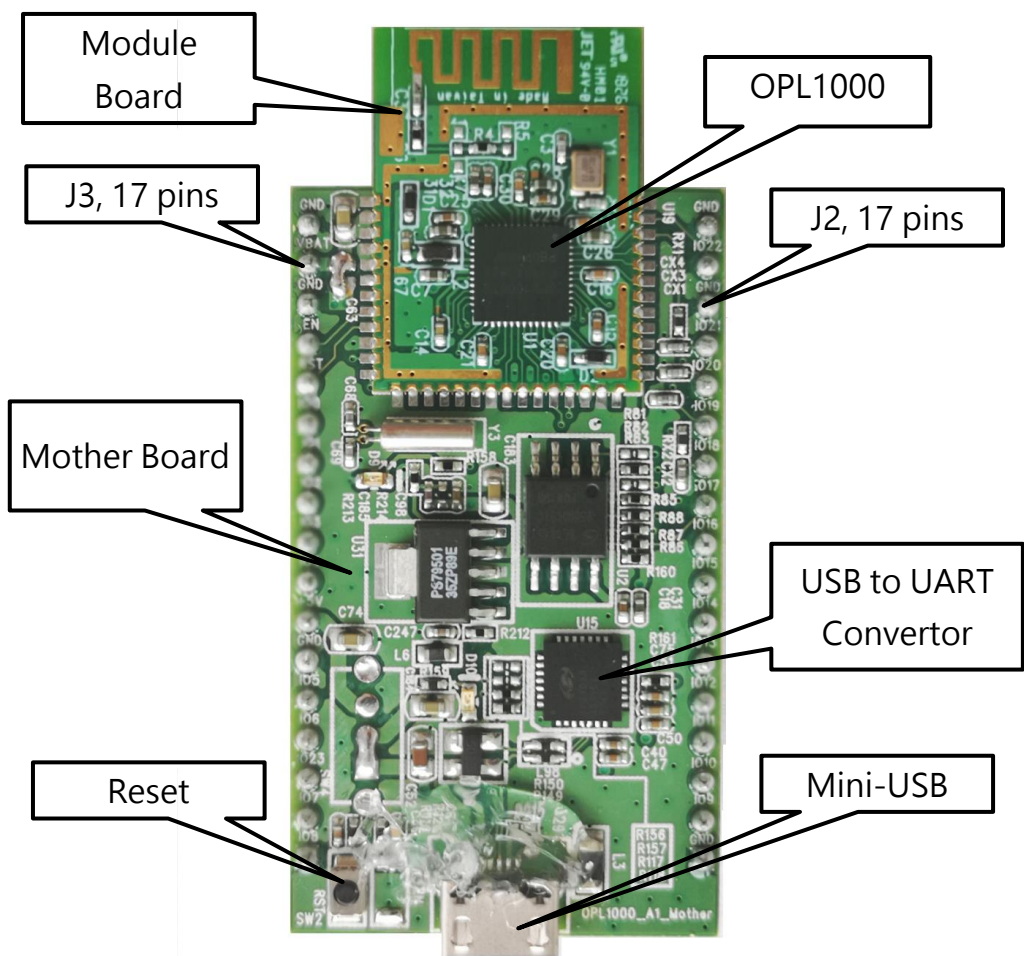
## 2. DEVKIT 和 LM80 概要介绍

### 2.1. DEVKIT 概要介绍

DEVKIT 板包括一个开发母板和 OPL1000 模块子板。母板包括 USB 转 UART 转换芯片、Flash 芯片和电源适配模块。OPL1000 模块子板包括 OPL1000 SOC 芯片和外部晶振。如

Figure 1 所示：

Figure 1: DEVKIT 板组成介绍





Opulinks DEVKIT 提供 mini USB 转串口功能，mini USB 同时提供供电功能。用户可以轻松使用 USB 控制 OPL1000，快速进行功能评估以及完成产品开发。DEVKIT 母板提供了若干扩充 GPIO 管脚，在线开发用的 ICE mode 管脚及 flash 烧录用的 UART Tx 及 Rx 管脚。OPL1000 预置为 Normal function mode，可快速切换为 ICE mode，另外提供 flash 烧录软件。扩展 GPIO 管脚可配置为 GPIO、ADC、SPI、I2C 等功能。扩充排针 J2,J3 配置底视图 (Bottom view)如 Figure 2 所示。

Figure 2: DEVKIT 扩展 IO map

J2						ANT	J3					
ICE Mode	PWM	I2C	ADC	Pin Name	Pin No		Pin No	Pin Name	ADC	SPI	UART	Flash Prg
	Yes			GND	pin 17	Bottom View	pin 17	GND				
				GPIO22	pin 16		pin 16	+3V				
				GND	pin 15		pin 15	GND				
M3_CLK				GPIO21	pin 14		pin 14	CHIP_EN				
M3_DAT				GPIO20	pin 13		pin 13	RST_N				
M0_DAT				GPIO19	pin 12		pin 12	GPIO0(REV)				UART_Prg_Tx
M0_CLK				GPIO18	pin 11		pin 11	GPIO1(REV)				UART_Prg_Rx
		SDA	Yes	GPIO17	pin 10		pin 10	GPIO2	Yes	MOSI	TxD	
		SCLK	Yes	GPIO16	pin 9		pin 9	GPIO3	Yes	MISO	RxD	
				GPIO15	pin 8		pin 8	GPIO4	Yes	CLK		
				GPIO14	pin 7		pin 7	Ex_5V				
				GPIO13	pin 6		pin 6	GND				
	Yes			GPIO12	pin 5		pin 5	GPIO5	Yes	CS		
				GPIO11	pin 4		pin 4	GPIO6	Yes			
				GPIO10	pin 3		pin 3	GPIO23				
	Yes			GPIO9	pin 2		pin 2	GPIO7	Yes	CS		
				GND	pin 1	USB	pin 1	GPIO8	Yes			

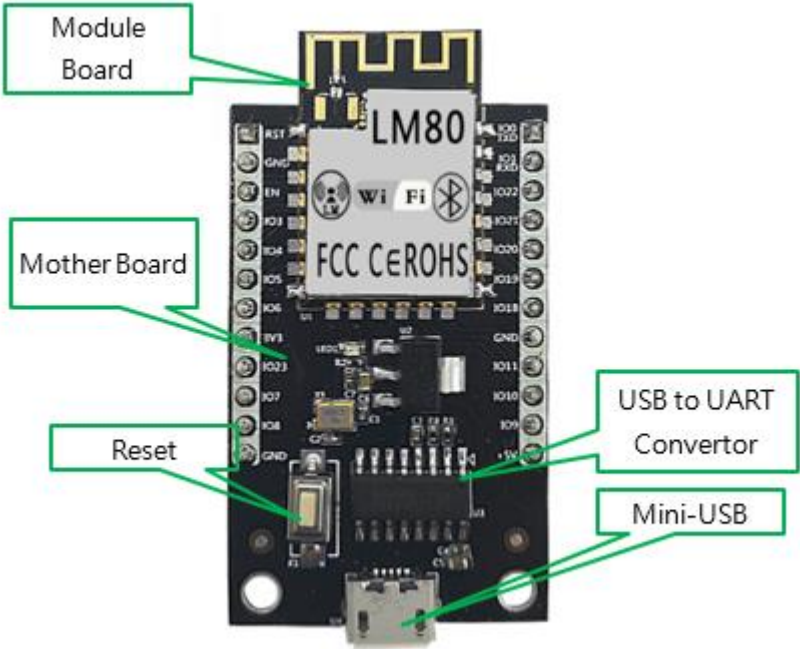
注 1：UART\_Prg 串口波特率为 115200 bps。

注 2：chip Enable (CHIP\_EN)和 Reset (RST\_N)都可以视为 Reset 功能。

2.2. LM80 概要介绍

LM80 板包括一个开发母版和 LM80 模块子板。母版包括 USB 转 UART 转换芯片，电源适配稳压模块。

Figure 3: LM80 组成介绍



LM80 提供 mini USB 转串口功能，mini USB 同时提供供电功能。用户可以轻松使用 USB 控制 LM80 模块，快速进行功能评估以及完成产品开发。LM80 母板提供了若干扩充 GPIO 管脚，在线开发用的 ICE mode 管脚及 flash 烧录用的 UART Tx 及 Rx 管脚。OPL1000 预置为 Normal function mode，可快速切换为 ICE mode，另外提供 flash 烧录软件。扩展 GPIO 管脚可配置为 GPIO、ADC、SPI、I2C 等功能。扩充排针 J2,J3 配置底视图 (Bottom view) 如 Figure 4 所示，其中子板作为一个独立的最小系统可以脱离母板独立使用，主要用于超低功耗的测试和评估，也可以在其完成产品开发。

Figure 4：LM80 扩展 IO map

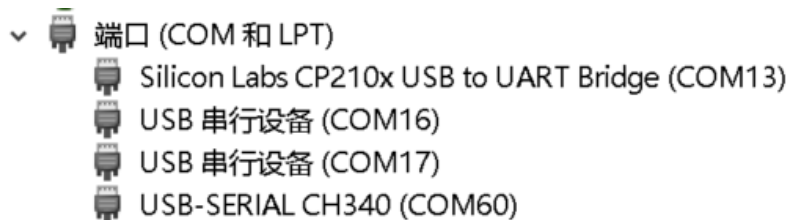
Flash Prg	ICE Mode	PWM	I2C	ADC	Pin Name	Pin NO	ANT	Pin NO	Pin Name	ADC	SPI	UART
UART_Prg_Tx					GPIO0	pin12	Bottom View	pin12	RST			
UART_Prg_Rx					GPIO1	pin11		pin11	GND			
		YES			GPIO22	pin10		pin10	EN			
	M3_CLK				GPIO21	pin9		pin9	GPIO3	YES		
	M3_DAT				GPIO20	pin8		pin8	GPIO4	YES	MISO	RxD
	M0_DAT				GPIO19	pin7		pin7	GPIO5	YES		
	M0_CLK				GPIO18	pin6		pin6	GPIO6	YES		
					GND	pin5		pin5	3V3			
			SDA	Yes	GPIO11	pin4		pin4	GPIO23			
			SCLK	Yes	GPIO10	pin3		pin3	GPIO7	YES		
		YES			GPIO9	pin2		pin2	GPIO8	YES		
					+5V	pin1		pin1	GND			

## 3. 使用 DEVKIT

OPL1000 DEVKIT 板提供了三个通信端口用于用户程序开发。它们分别是：

1. AT 串口: AT 串口用于升级固件以及发送 AT 命令给 OPL1000。在 MP1.8 之前, 它默认为和 mini-USB 连接的串口, 对应的 IO pin 是 IO0/1; 而在 MP1.8 之后, 接收 AT 命令的功能则默认在 UART1 上实现, 对应的 IO pin 是 IO8/9。在目前 DEVKIT 板使用的 USB 转串口芯片为 Silicon Labs CP210x 芯片, 用户可以通过查找设备管理器“COM 和 LPT 端口”得知 mini-USB 连接的串口编号是多少。例如下图所示的设备端口列表中, OPL1000 DEVKIT AT 串口为 COM13。

Figure 5: DEVKIT 设备端口



2. APS 串口: 该串口用于输出内部调试信息, 可用于离线调试应用程序. 在 MP1.8 之前, 默认在 UART1 上实现, 对应的 IO pin 是 IO8/9; 而在 MP1.8 之后, 它默认为和 mini-USB 连接的串口, 对应的 IO pin 是 IO0/1.
3. Cortex M3 SWD 调试接口。

### 3.1. 安装 USB 驱动

OPL1000 DEVKIT 使用 Silicon Labs CP210x USB 转 UART 桥接芯片。为使 Mini-USB 正常工作, 需要根据用户使用的操作系统类型安装对应的驱动软件。驱动软件目录为: Tool\CP210x\_Windows\_Drivers

#### 3.1.1. Win10/Win7 系统驱动安装

CP210x 桥接芯片的 Win10/Win7 系统驱动程序路径

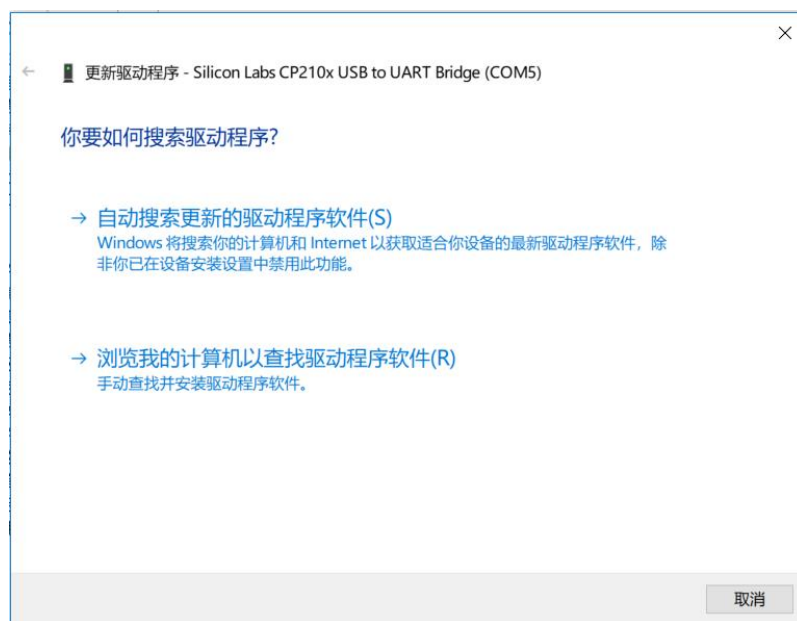
Tool\CP210x\_Windows\_Drivers\Win7\_Win10\_x64

Win10 操作系统会自动安装 CP210x 桥接芯片驱动。但是这个驱动和 download 工具使用的串口模块库有冲突, 需要进行版本降级。即使用 Win7\_Win10\_x64 目录下的驱动程序。操作步骤为:

Step1 首先打开设备管理器，选择串口设备“Silicon Labs CP210x USB to UART Bridge”，右键选择‘更新驱动程序’，点击‘浏览我的计算机以查找驱动程序软件’

Figure 6: Win10/Win7 驱动安装 – 搜索驱动程序

Figure 7 : Win10/7 驱动安装搜索



Step2 点击‘让我从计算机上的可用驱动程序列表中选择’，如下图所示。

Figure 8: Win10/Win7 驱动安装 – 选择从驱动程序列表中选择



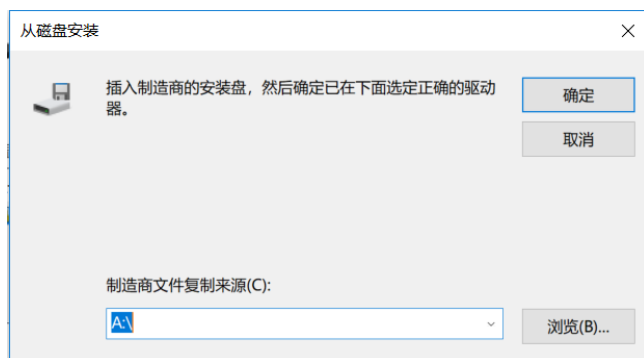
Step3 由于 CP210x 桥接芯片已经在系统中有注册，因此在 Figure 9 “显示兼容硬件”列表中有 Silicon Labs CP210x USB to UART Bridge。点击‘从磁盘安装’

Figure 9: Win10/Win7 驱动安装 – 为 CP210x 更新驱动程序



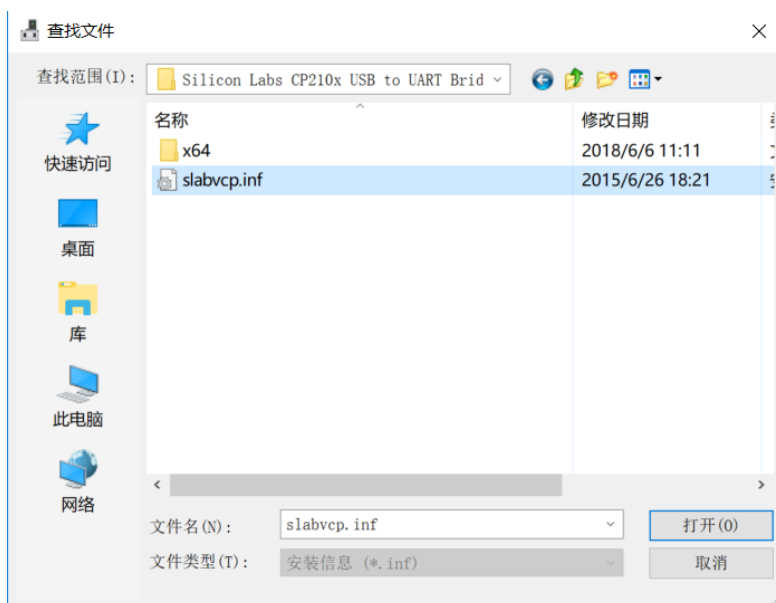
Step4 在弹出窗口中点击‘浏览’选择 Tool\CP210x\_Windows\_Drivers\Win7\_Win10\_x64 目录

Figure 10: Win10/Win7 驱动安装 –指定安装目录



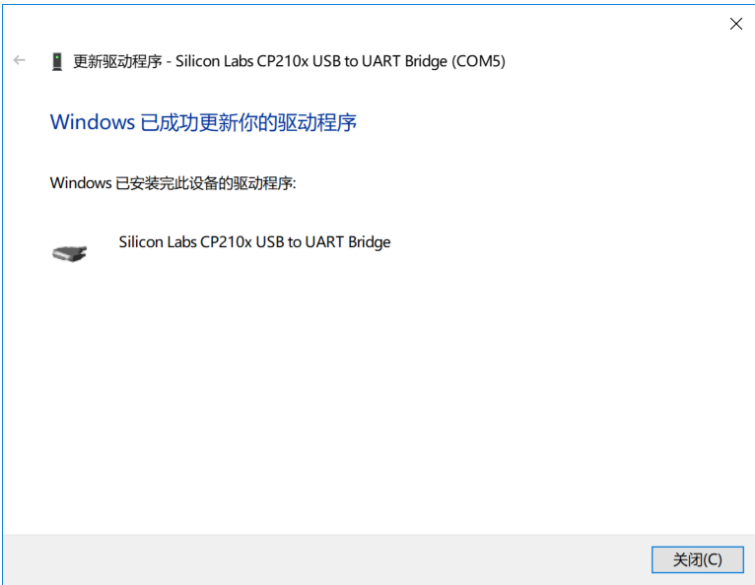
Step5 选择 `Tool\CP210x_Windows_Drivers\Win7_Win10_x64` 目录下的 `slabvcp.inf` 文件，点击“打开”。`slabvcp.inf` 包含了 CP210x 芯片的 USB 设备信息和驱动配置信息。

Figure 11: Win10/Win7 驱动安装 – 选择 USB inf 文件



Step6 点击“确定”按钮，“下一步”，按钮。点击“关闭”按钮，安装完成。得到 Figure 12 所示画面。表示驱动更新完成。

Figure 12: Win10/Win7 驱动安装 – 驱动更新完成



Step7 检查驱动版本信息，是否为 6.7.1.0。右键点击串口设备“Silicon Labs CP210x USB to UART Bridge”，在属性中选择“驱动程序”标签页。如果版本为 6.7.1.0 则表示驱动安装正确。

Figure 13: Win10/Win7 驱动安装 – 检查驱动版本信息



### 3.1.2. WinXP/Vista 系统驱动安装

CP210x 桥接芯片的 WinXP/Vista 系统驱动程序路径 Tool\CP210x\_Windows\_Drivers\WinXP\_Vista。

整个安装包括三个步骤：

Step1 运行 WinXP\_Vista 目录下的 'CP210xVCPInstaller\_x86.exe' 程序

Step2 在弹出的对话框界面 Figure 14 上 点击“下一步”。得到

Figure 15 所示画面。然后选择“我接受这个协议”，“下一步”。

Figure 14: WinXP/Vista 驱动安装 – 执行驱动安装程序



Figure 15: WinXP/Vista 驱动安装 – 接受协议





Step3 安装结束后点击“完成”

最后检查驱动版本信息。右键点击串口设备“Silicon Labs CP210x USB to UART Bridge”，在属性中选择“驱动程序”标签页。如果得到 Figure 16 所示 6.7.0.0 版本号就对了。

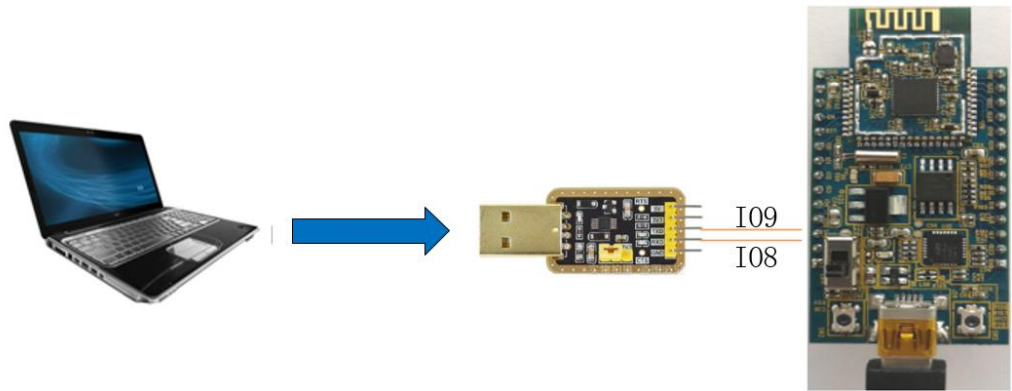
Figure 16: WinXP/Vista 驱动安装 – 检查驱动版本信息



3.2. APS 串口连接和使用

DEVKIT 板两侧提供了两排扩展接口，其中包含 APS（Debug\_prg）串口，实现和 M3 MCU 串口通信功能。APS 串口可以输出固件 log 打印功能。

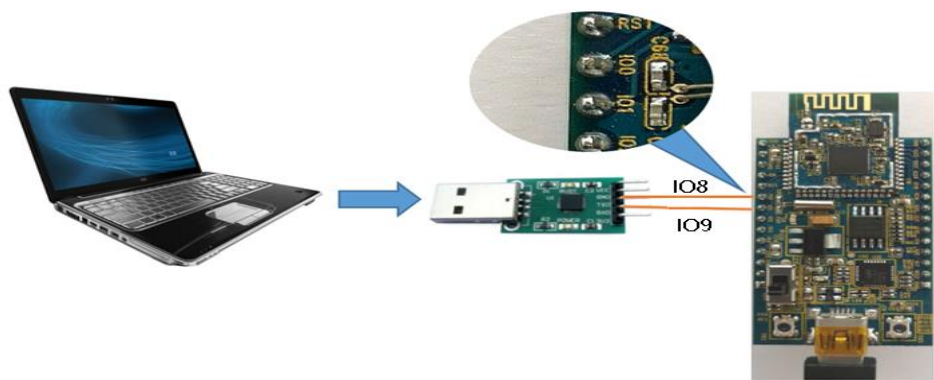
Figure 17: DEVKIT APS 串口连接



3.2.1. APS 串口连接

APS 串口连接使用 IO8 和 IO9 两根管脚。IO8 是 APS 串口的 TX 输出信号线，接 UART 转接板的输入 RX 信号线。IO9 是 APS 串口 RX 信号线，接 UART 转接板的 TX 信号线。接线如下图所示。

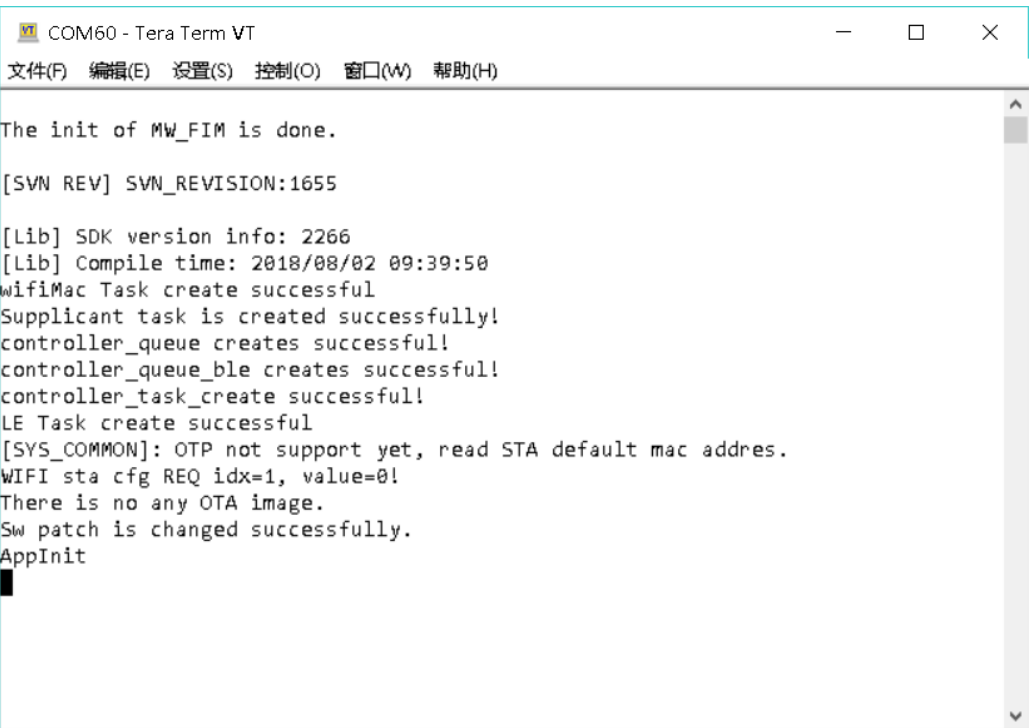
Figure 18: APS 串口接线实例



3.2.2. 通过 APS 观察固件调试打印信息

使用 Tera Term 连接 Debug\_Prg 串口（本例中为 COM60），波特率设置为 115200。复位 DEVKIT 板在信息输出窗口可以得到固件的打印信息。如 Figure 19 所示。

Figure 19: APS 串口输出打印信息



3.3. AT 串口连接和使用

DEVKIT 板上的 mini USB 提供供电，固件下载，以及 MP1.8 之前提供 AT 命令通信功能（注：MP1.8 及其之后的版本在 APS 口输入 AT 命令）。AT 串口所采用的 USB 转串口控制芯片为 CP210X，正确安装芯片驱动后，连接 DEVKIT 板。在 PC 设备管理器中可以观察到 CP210x 串口设备。下图给出的例子中 COM13 为 AT 串口，另外一个 CH340 串口设备 COM60 连接的是 APS 串口。

Figure 20: DEVKIT AT 串口设备

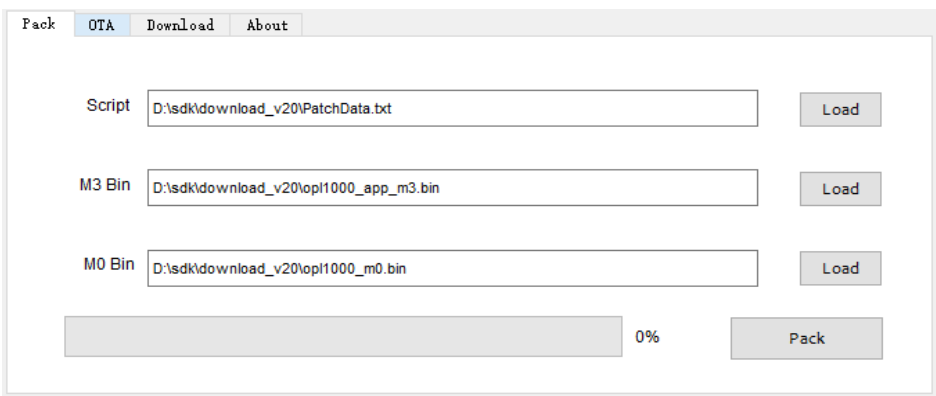


3.3.1. 通过 AT 串口更新固件

从编译工程到下载固件至 DEVKIT 板有 4 个步骤，以编译下载 hello\_world 示例工程为例:

- 1. 使用 keil uVision(建议版本不低于 5.23)软件编译 SDK 的示例工程 hello\_world。  
目录：SDK\APS\_PATCH\examples\get\_started\hello\_world  
编译完成以后，在工程目录 Output\Objects 获得 opl1000\_app\_m3.bin。
- 2. 将编译得到的 opl1000\_app\_m3.bin 保存到 FW\_Binary 目录下面。在 FW\_Pack 目录下有固件合并脚本文件 PatchData.txt 和 M0 Bin。在 Pack 标签页点击 Script,M3 Bin, M0 Bin 文本框右侧的 Load 按钮，依次载入固件合并脚本文件、M3 bin 和 M0 bin 文件。点击 Pack 按钮。Pack 动作会把几个独立的 bin 文件合成为一个可供下载的 opl1000.bin 文件，存放在 download tool 同目录下的 Patch 子文件夹。

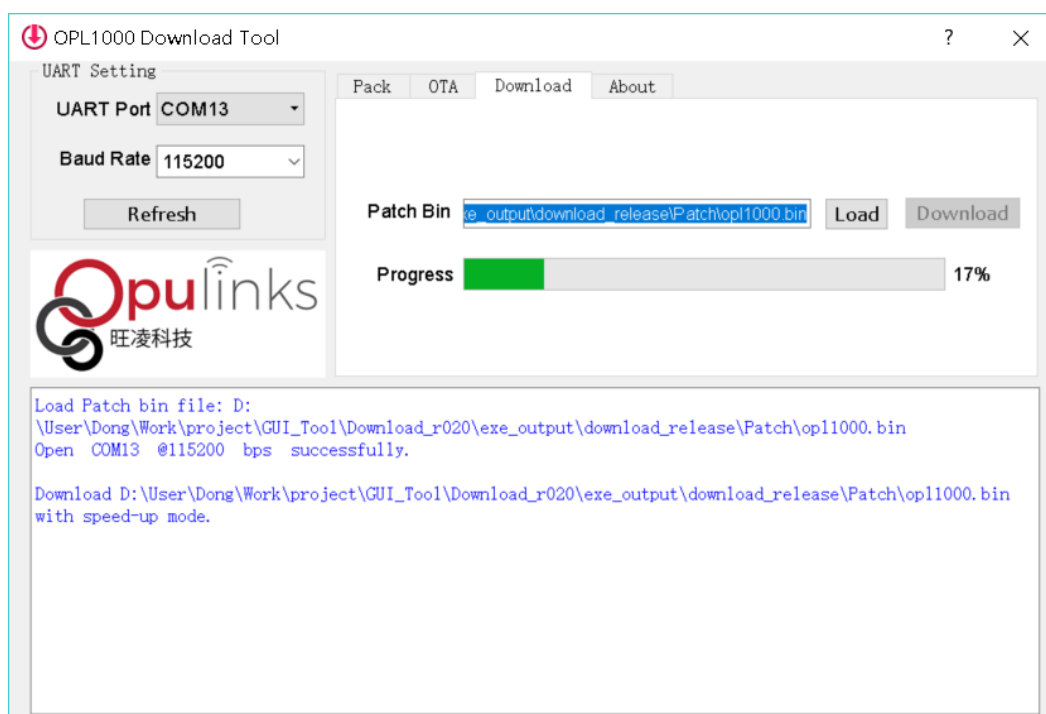
Figure 21:载入 M3/M0 Bin 文件进行合并操作



- 选择 AT 串口端口号，波特率默认 115200bps。切换到 Download 选项，Patch Bin 路径已经正确填充为 opl1000.bin 文件，点击 Download 按钮，并在 5 秒之内复位 DEVKIT 板。download tool 自动识别到 DEVKIT 板复位以后，开始下载 opl1000.bin。进度条到达 100%，表示下载 opl1000.bin 成功。

对 OTA 固件下载过程和上面的一样。只是在 load 固件文件的时候，需要选取 opl1000\_ota.bin (OTA image 文件)。

Figure 22: Patch Bin 文件下载



- 下载完成后 DEVKIT 板会自动复位，Flash 的固件载入到 RAM 中执行。在“UART Port”选择 APS 串口端口，选择 115200 波特率，再次复位 DEVKIT 板，在串口调试工具中看到下图输出信息则表明固件下载正确。

Figure 23: 启动后 APS 串口输出 log 信息

```
BootMode 10 go to normal path

The init of MW_FIM is done.

[Lib] SDK version info: 1516
[Lib] Compile time: 2018/05/10 17:49:03

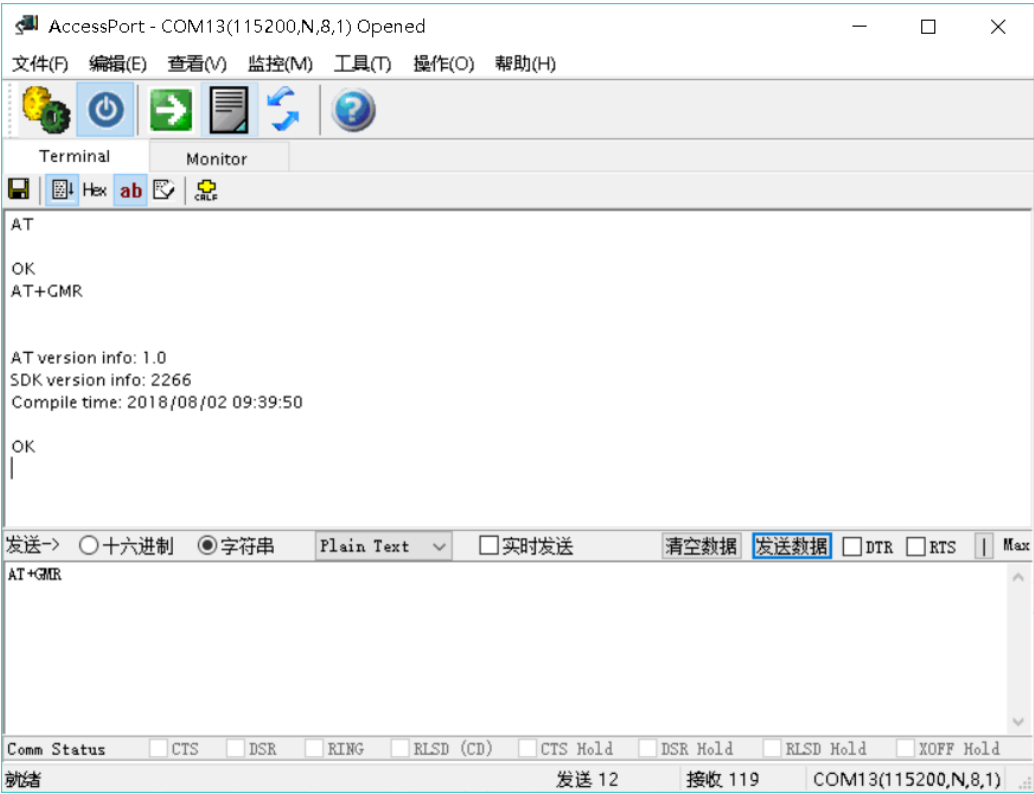
[SVN REV] SVN_REVISION:809
wifiMac Task create successful
Supplicant task is created successfully!
controller_queue creates successful!
controller_queue_ble creates successful!
controller_task_create successful!
LE Task create successful
Sw patch is changed successfully.
Hello world 1
Hello world 2
Hello world 3
Hello world 4
Hello world 5
```

3.3.2. 使用 AT 串口执行 AT 指令

打开串口工具，选择 AT 串口号，波特率 115200。正常情况下，键入 ENTER (回车)，出现命令提示符 >，输入 at，得到 OK 的返回，则说明 AT 功能正常。

注：AT 命令的执行在 MP1.8 及其之后版本中要在 APS 口输入。

Figure 24: 执行 AT 指令



### 3.4. SWD 端口

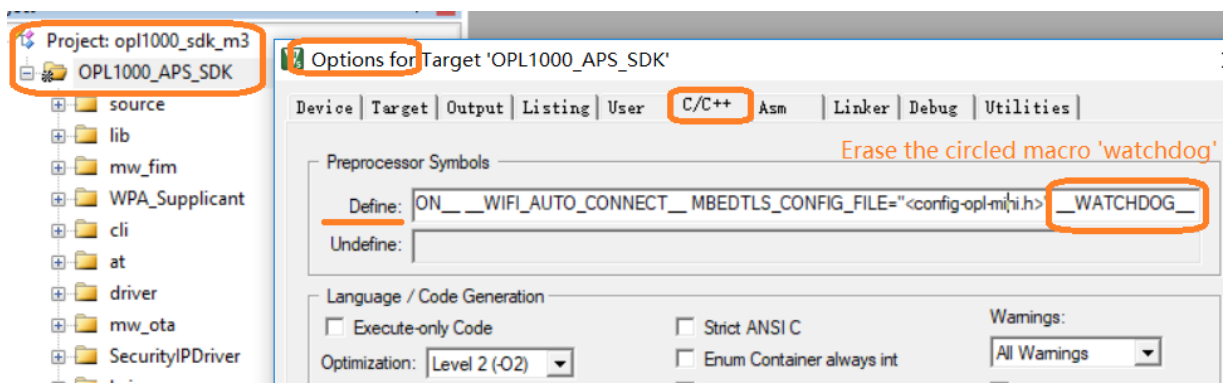
如果需要使用在线调试应用程序，则需要使用 ICE 仿真器，DEVKIT 板支持 M0 和 M3 四线 SWD 调试方式。由于用户 APP 在 M3 上执行，因此实际需要使用 M3 ICE 端口。后续章节将介绍如何禁用 watchdog 以及 M3 ICE 端口连接方式。

#### 3.4.1. 禁用 watchdog

在默认情况下 OPL1000 A1 开启了 watchdog 功能。但用户可根据需要将 watchdog 功能关闭，以免在调试过程中被复位，方便使用 J-link 仿真器在线调试。方法如下：

1. 使用 keil uVision(建议版本不低于 5.23)软件打开 opl1000\_sdk\_m3 工程文件（目录：A1\SDK\APS\project\opl1000\），打开该工程的 options，点击'C/C++' tag，在'Define'域中删除 '\_\_WATCHDOG\_\_'宏定义，然后 rebuild 该工程；

Figure 25: 禁用 watchdog



2. 用 keil 打开 hello\_world 工程（目录：SDK\APS\_PATCH\examples\get\_started\hello\_world），rebuild 它。再参照 3.3.1 节的方法用 download tool 把生成的 opl1000\_app\_m3.bin 和 m0 bin 文件合并成为 opl1000.bin 文件，再将它烧录到 A1 板上。

烧录完毕，重启 A1 板后，watchdog 功能就被禁用了。

注：在 rebuild 该工程前，请确认以下设置：

在 hal\_pin\_config\_project.h 文件中 IO20,IO21 做如下设置

```
#define HAL_PIN_TYPE_IO_20 PIN_TYPE_ICE_M3_DAT // PIN_TYPE_NONE
#define HAL_PIN_TYPE_IO_21 PIN_TYPE_ICE_M3_CLK // PIN_TYPE_NONE
```

3.4.2. M3 ICE 端口连接

M3 ICE 端口接线对应关系如表 Table 1 所示。

Table 1: DEVKIT M3 SWD 信号连接

编号	SWD 信号	DEVKIT 板 J2 排针	J-Link 仿真器管脚	说明
1	3.3V	--	1	3.3 V 电源
2	GND	GND	4 - 20	GND
3	SWD_CLK	IO21	9	时钟信号
4	SWD_DAT	IO20	7	数据线

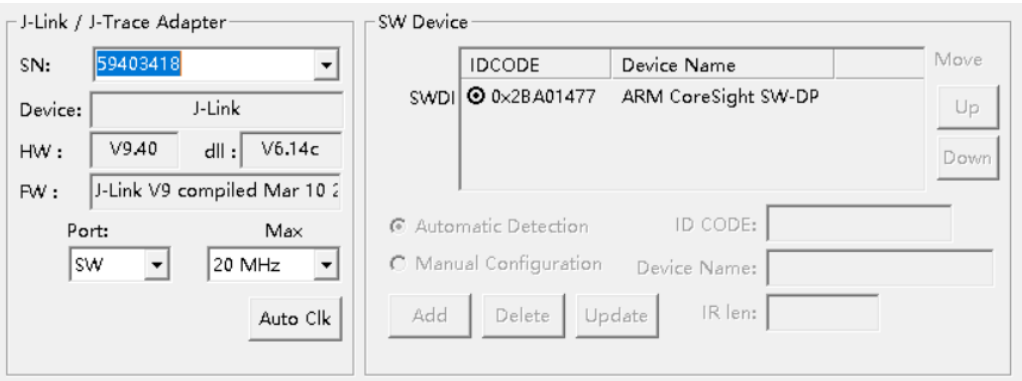
Notes: 对于多芯片 J-Link 仿真器，需确保编号为 1 的电源线也正确地连接。

Figure 26: DEVKIT 板上 M3 ICE 连线图



连接好 J-Link 仿真器和 DEVKIT 板，在 keil 工程的 debug 界面里面如果检测到 SW Device 的序列号，则说明连接正确，可以正常使用 SWD 开发和调试工程。如图 Figure 27 所示。

Figure 27: J-link ICE 仿真器正确识别





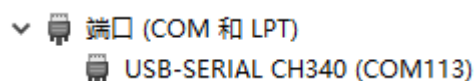


## 4. 使用 LM80

LM80 板提供了三个通信端口用于用户程序开发。它们分别是：

1. AT 串口: AT 串口用于升级固件以及发送 AT 命令给 OPL1000。在 MP1.8 之前, 它默认为和 mini-USB 连接的串口, 对应的 IO pin 是 IO0/1; 而在 MP1.8 之后, 接收 AT 命令的功能则默认在 UART1 上实现, 对应的 IO pin 是 IO8/9。目前 LM80 板使用的 USB 转串口芯片为 CH340G 芯片, 用户可以通过查找设备管理器“COM 和 LPT 端口”得知 mini-USB 连接的串口编号是多少。例如下图所示的设备端口列表中, LM80 板 AT 串口为 COM113。

Figure 28 : LM80 串口信息



2. APS 串口。该串口用于输出内部调试信息, 可用于离线调试应用程序。在 MP1.8 之前, 默认在 UART1 上实现, 对应的 IO pin 是 IO8/9; 而在 MP1.8 之后, 它默认为和 mini-USB 连接的串口, 对应的 IO pin 是 IO0/1。
3. Cortex M3 SWD 调试接口。

### 4.1. 安装 USB 驱动

LM80 开发板使用 CH340G 转 UART 桥接芯片。为使 Mini-USB 正常工作, 需要根据用户使用的操作系统类型安装对应的驱动软件。驱动软件目录为: Tool\CH340\_Windows\_Drivers

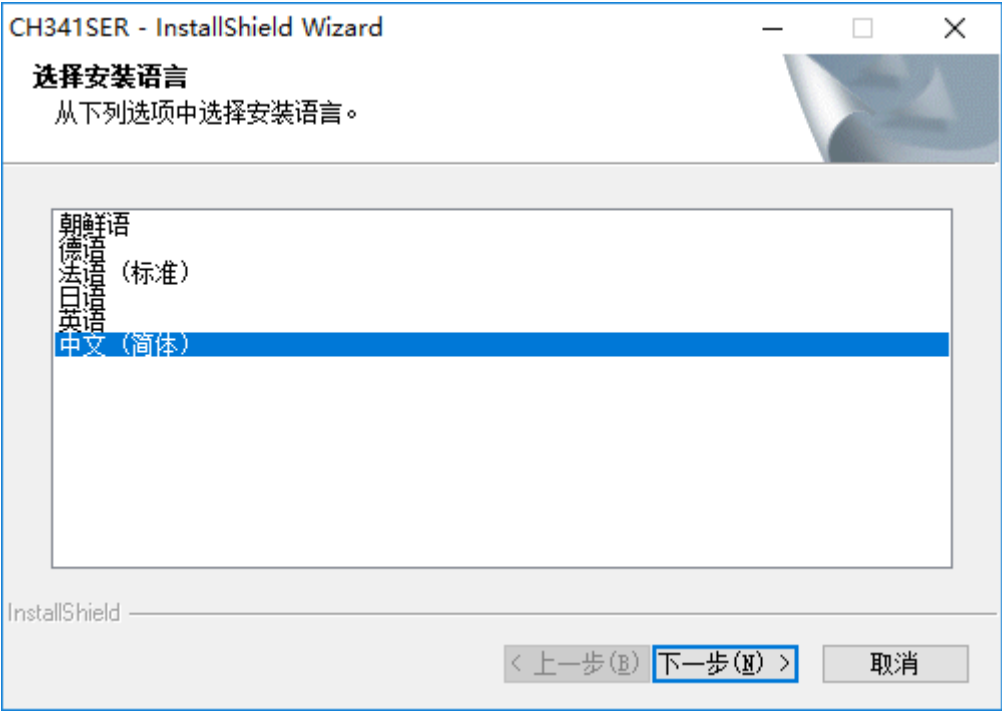
#### 4.1.1. Win10/Win7 系统驱动安装

CH340 桥接芯片的 Win10/Win7 系统驱动程序路径

Tool\CH340\_Windows\_Drivers\Win7\_8\_10\_Drivers

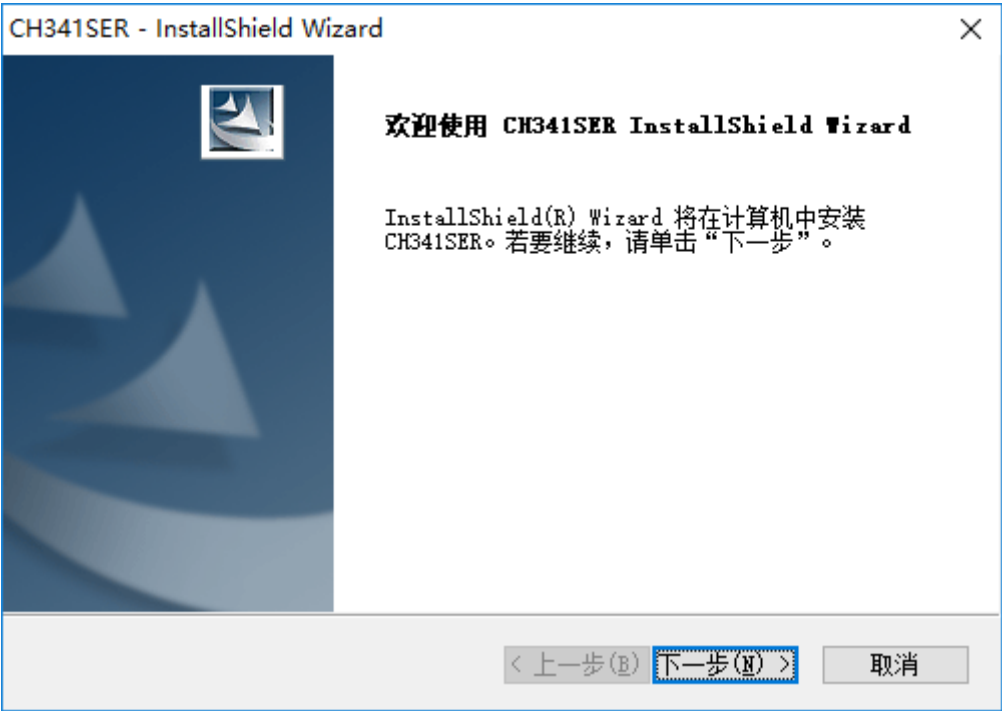
- 运行上面目录下的安装文件, 弹出安装语言, 选择需要的语言, 如【中文】, 点击【下一步】

Figure 29 : WIN7/10 安装 CH340 驱动语言选择



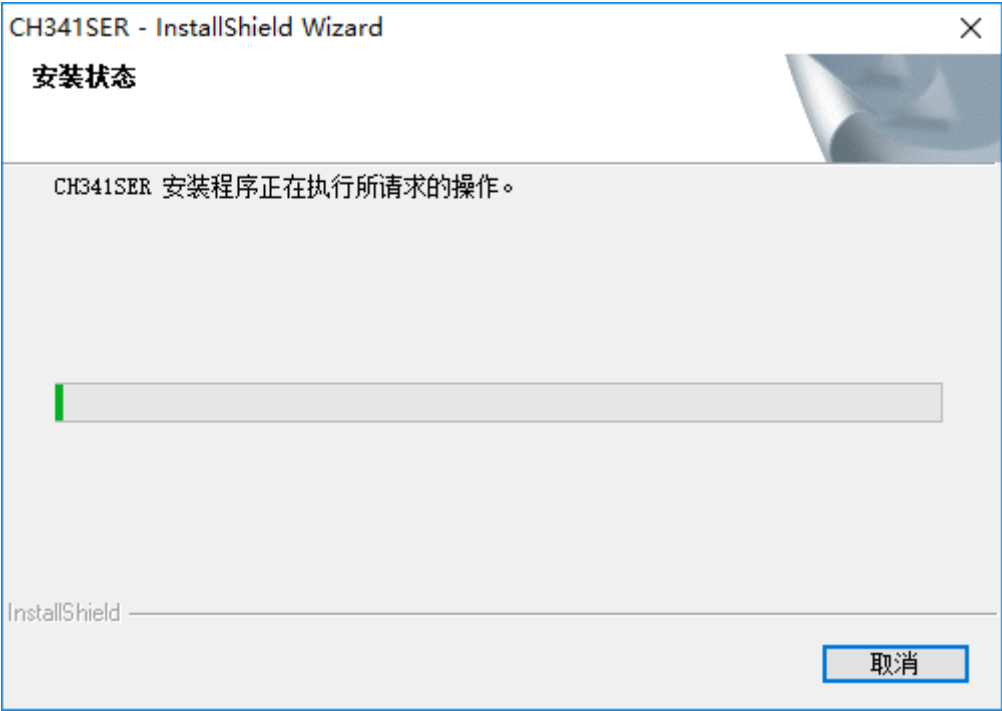
➤ 安装界面 · 继续【下一步】

Figure 30：安装界面



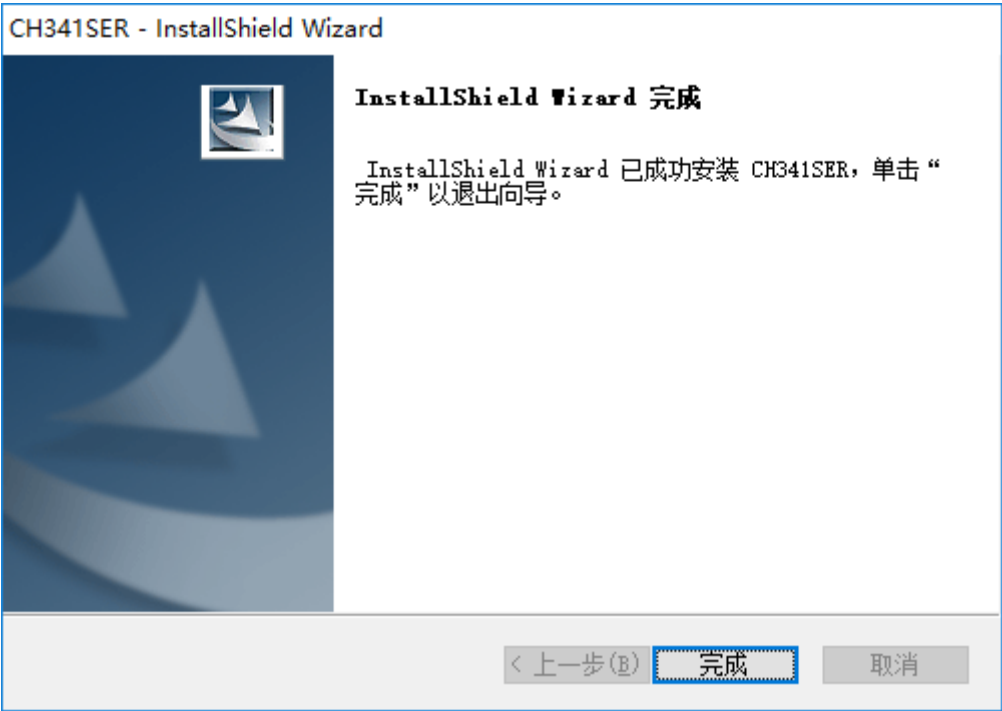
➤ 安装进行中

Figure 31：驱动安装进行界面



➤ 安装完成

Figure 32：驱动安装完成界面



➤ 安装完成后，对于 Windows8/8.1 会出现“数据无效”的错误提示，原因是由于微软对硬件设备管理的服务独立出来，如果安装驱动没有开启相关服务，那么就可能会弹出这个错误提示。  
解决办法如下：

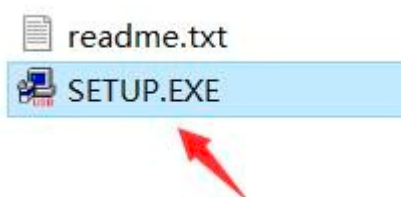
1. 按键盘上的 Winkey+R，在弹出的“运行”对话框中输入“services.msc”，亦可通过“计算机管理”窗口下找到“服务”；
2. 在服务列表中找到“Device Install Service”和“Device Setup Manager”两个服务项，确认这两个服务是否开启，如果没有开启，可以点击左侧的“启动”此服务。

#### 4.1.2. WinXP 系统驱动安装

CH340 桥接芯片的 WinXP 系统驱动程序路径 Tool\CH340\_Windows\_Drivers\WinXP\_Drivers。整个安装包括三个步骤：

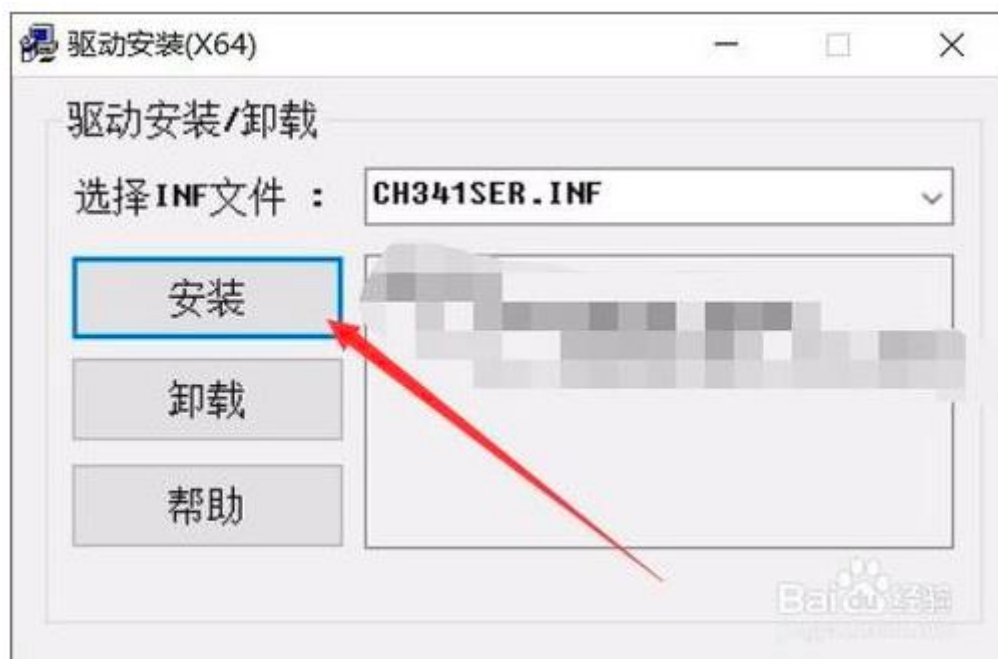
1. 在上述目录找到驱动程序“SETUP.exe”，双击运行

Figure 33：驱动程序



2. 跳出图中界面，点击“安装”控件，如图所示：

Figure 34：XP 驱动安装界面



3. 等待片刻，如果成功安装会跳出图中的界面，点击确定，如图所示；

Figure 35：驱动安装成功界面



4. 连接我们的开发板到电脑
5. 第一次接入，会显示正在安装，等待片刻后即可完成安装

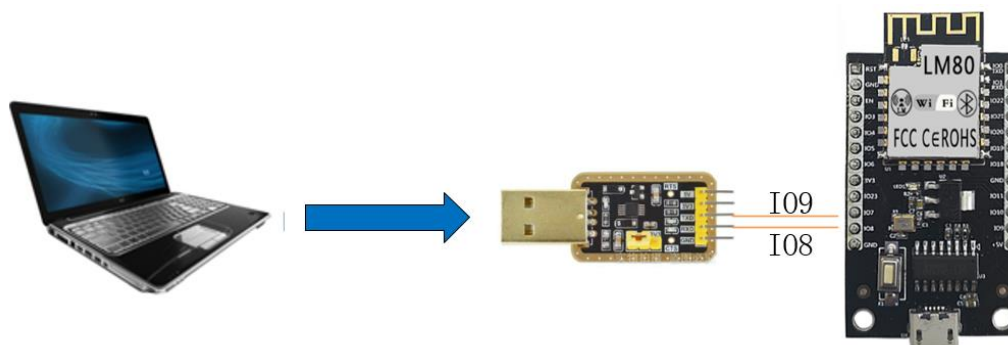
## 4.2. APS 串口连接和使用

LM80 板两侧提供了两排扩展接口，其中包含 APS（Debug\_prg）串口，实现和 M3 MCU 串口通信功能。APS 串口可以输出固件 log 打印功能。

### 4.2.1. APS 串口连接

APS 串口连接使用 IO8 和 IO9 两根管脚。IO8 是 APS 串口的 TX 输出信号线，接 UART 转接板的输入 RX 信号线。IO9 是 APS 串口 RX 信号线，接 UART 转接板的 TX 信号线。接线如下图所示。

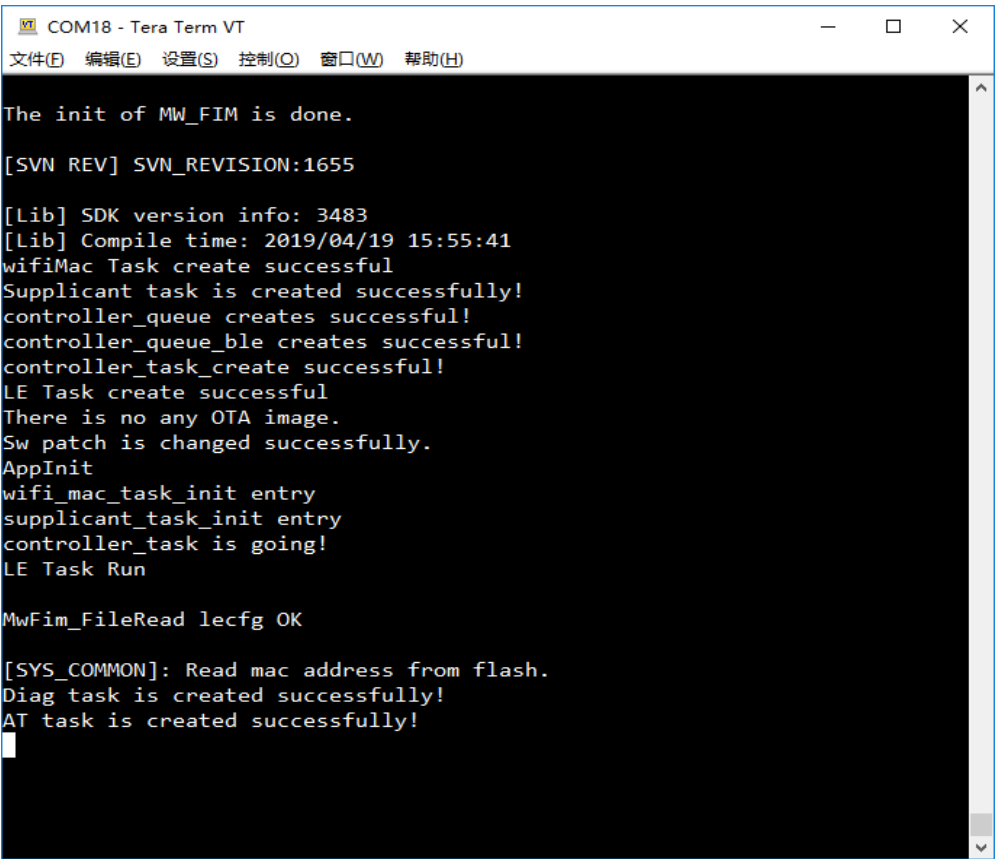
Figure 36: APS 串口接线实例



4.2.2. 通过 APS 观察固件调试打印信息

使用 Tera Term 连接 Debug\_Prg 串口 ( 本例中为 COM18 ) , 波特率设置为 115200 。复位 DEVKIT 板在信息输出窗口可以得到固件的打印信息 。如 Figure 37 所示 。

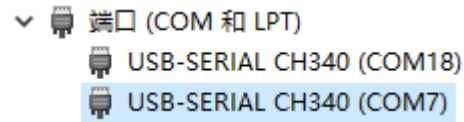
Figure 37: APS 串口输出打印信息



4.3. AT 串口连接和使用

LM80 板上的 mini USB 提供供电 , 固件下载 , 以及 MP1.8 之前提供 AT 命令通信功能 ( 注 : MP1.8 及其之后的版本在 APS 口输入 AT 命令 ) 。AT 串口所采用的 USB 转串口控制芯片为 CH340 , 正确安装芯片驱动后 , 连接 LM80 板 。在 PC 设备管理器中可以观察到串口设备 。下图给出的例子中 COM7 为 AT 串口 , 另外一个 CH340 串口设备 COM18 连接的是 APS 串口 。

Figure 38: DEVKIT AT 串口设备



## 4.3.1. 通过 AT 串口更新固件

从编译工程到下载固件至 LM80 板有 4 个步骤，以编译下载 hello\_world 示例工程为例:

1. 使用 keil uVision(建议版本不低于 5.23)软件编译 SDK 的示例工程 hello\_world。  
目录：SDK\APS\_PATCH\examples\get\_started\hello\_world  
编译完成以后，在工程目录 Output\Objects 获得 opl1000\_app\_m3.bin。  
2. 将编译得到的 opl1000\_app\_m3.bin 保存到 FW\_Pack 目录下面。在 FW\_Pack 目录下有固件合并脚本文件 PatchData.txt。在 Pack 标签页点击 Script,M3 Bin, M0 Bin 文本框右侧的 Load 按钮，依次载入固件合并脚本文件、M3 bin 和 M0 bin 文件。点击 Pack 按钮。Pack 动作会把几个独立的 bin 文件合成为一个可供下载的 opl1000.bin 文件，存放在 download tool 同目录下的 Patch 子文件夹。

Figure 39:载入 M3/M0 Bin 文件进行合并操作

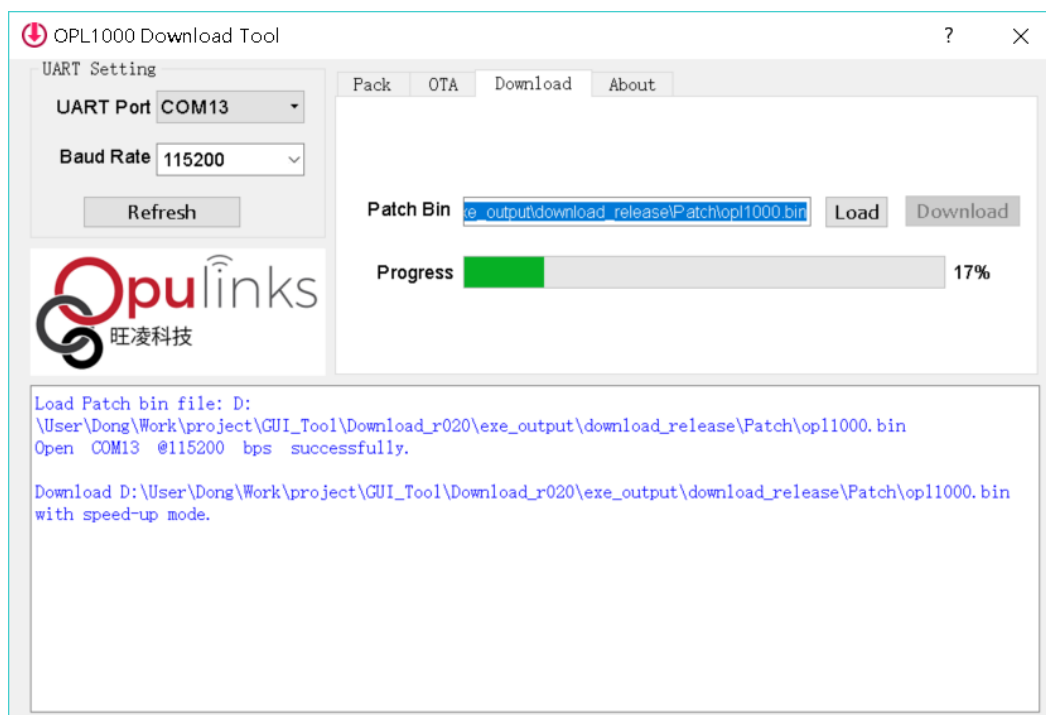


3. 选择 AT 串口端口号，波特率默认 115200bps。切换到 Download 选项，Patch Bin 路径已经正确填充为 opl1000.bin 文件，点击 Download 按钮，并在 5 秒之内复位 DEVKIT 板。  
download tool 自动识别到 LM80 板复位以后，开始下载 opl1000.bin。进度条到达 100%，表示下载 opl1000.bin 成功。

对 OTA 固件下载过程和上面的一样。只是在 load 固件文件的时候，需要选取 opl1000\_ota.bin (OTA image 文件)。

Figure 40: Patch Bin 文件下载





4. 下载完成后 LM80 板会自动复位，Flash 的固件载入到 RAM 中执行。在“UART Port”选择 APS 串口端口，选择 115200 波特率，再次复位 DEVKIT 板，在串口调试工具中看到下图输出信息则表明固件下载正确。

Figure 41: 启动后 APS 串口输出 log 信息

```

BootMode 10 go to normal path

The init of MW_FIM is done.

[Lib] SDK version info: 1516
[Lib] Compile time: 2018/05/10 17:49:03

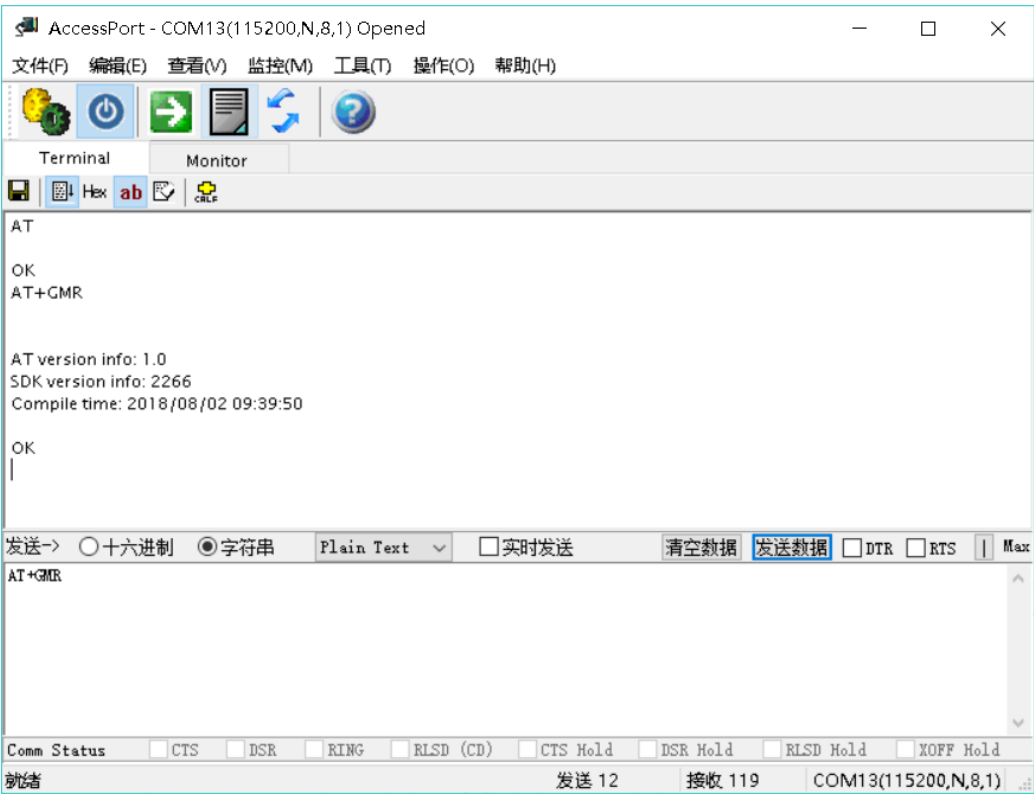
[SVN REV] SVN_REVISION:809
wifiMac Task create successful
Supplicant task is created successfully!
controller_queue creates successful!
controller_queue_ble creates successful!
controller_task_create successful!
LE Task create successful
Sw patch is changed successfully.
Hello world 1
Hello world 2
Hello world 3
Hello world 4
Hello world 5

```

4.3.2. 使用 AT 串口执行 AT 指令

打开串口工具，选择 AT 串口号，波特率 115200。正常情况下，键入 ENTER (回车)，出现命令提示符 >，输入 at，得到 OK 的返回，则说明 AT 功能正常。

Figure 42: 执行 AT 指令



4.4. SWD 端口

如果需要使用在线调试应用程序，则需要使用 ICE 仿真器，LM80 板支持 M0 和 M3 四线 SWD 调试方式。由于用户 APP 在 M3 上执行，因此实际需要使用 M3 ICE 端口。后续章节将介绍如何禁用 watchdog 以及 M3 ICE 端口连接方式。

4.4.1. 禁用 watchdog

在默认情况下 OPL1000 A1 开启了 watchdog 功能，因为系统出现 hung 的时候，watchdog 会工作，系统就会执行复位操作。但用户可根据需要将 watchdog 功能关闭，以免在调试过程中被复位，方便使用 J-link 仿真器在线调试。方法如下：

- 1. 用 keil 打开 hello\_world 工程 ( 目录 : SDK\APS\_PATCH\examples\get\_started\hello\_world ) , 打开 main\_patch,将 Hal\_Wdt\_Stop()释放掉即可。

Figure 43 : Watchdog 禁用开关

```
static void Main_MiscModulesInit(void)
{
    //Hal_Wdt_Stop();    //disable watchdog here.
}
```

- 2. 再参照 3.3.1 节的方法用 download tool 把生成的 opl1000\_app\_m3.bin 和 m0 bin 文件合并成为 opl1000.bin 文件 , 再将它烧录到 A1 板上。

烧录完毕 , 重启 A1 板后 , watchdog 功能就被禁用了。

注 : 在 rebuild 该工程前 , 请确认以下设置 :

- 3. 在 hal\_pin\_config\_project.h 文件中 IO20,IO21 做如下设置
- 4. #define HAL\_PIN\_TYPE\_IO\_20 PIN\_TYPE\_ICE\_M3\_DAT // PIN\_TYPE\_NONE
- 5. #define HAL\_PIN\_TYPE\_IO\_21 PIN\_TYPE\_ICE\_M3\_CLK // PIN\_TYPE\_NONE

4.4.2. M3 ICE 端口连接

M3 ICE 端口连接如下图所示。接线对应关系如表 Table 2 所示。

Figure 44 : LM80 板上 M3 ICE 信号接线图

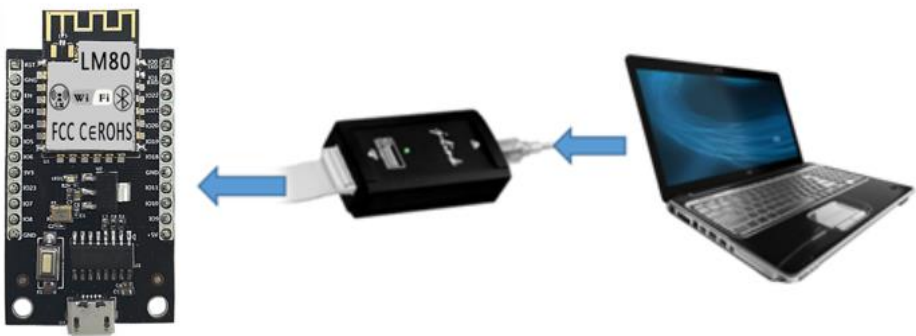


Table 2: LM80 M3 SWD 信号连接

编号	SWD 信号	LM80 排针	J-Link 仿真器管脚	说明
1	3.3V	--	1	3.3 V 电源
2	GND	GND	4 - 20	GND

编号	SWD 信号	LM80 排针	J-Link 仿真器管脚	说明
3	SWD_CLK	IO21	9	时钟信号
4	SWD_DAT	IO20	7	数据线

Notes: 对于多芯片 J-Link 仿真器，需确保编号为 1 的电源线也正确地连接。

Figure 45 : LM80 板上 M3 ICE 连线图



连接好 J-Link 仿真器和 LM80 板，在 keil 工程的 debug 界面里面如果检测到 SW Device 的序列号，则说明连接正确，可以正常使用 SWD 开发和调试工程。如图 Figure 46 所示。

Figure 46: J-link ICE 仿真器正确识别

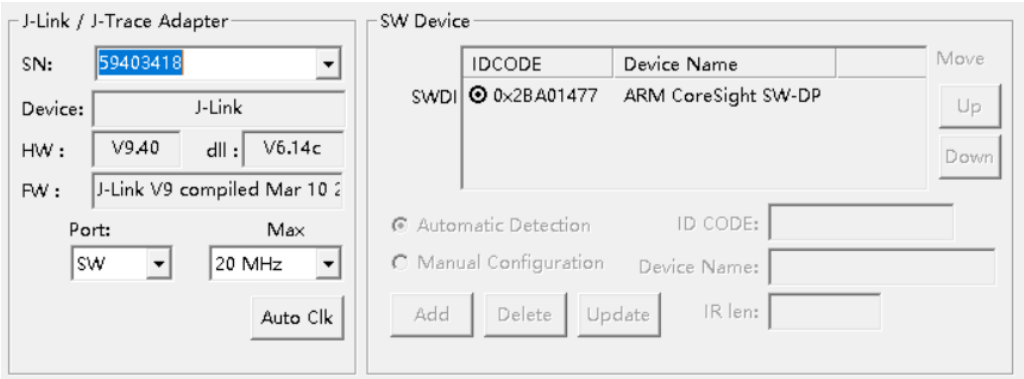


Figure 47 : 断点设置调试

```

main_patch.c  hal_pin_config_project.h
250 // create the thread for AppThread_2
251 tThreadDef.name = "App_2";
252 tThreadDef.pthread = Main_AppThread_2;
253 tThreadDef.tpriority = OS_TASK_PRIORITY_APP; // osPriorityNormal
254 tThreadDef.instances = 0; // reserved, it is no used
255 tThreadDef.stacksize = OS_TASK_STACK_SIZE_APP; // (512), unit: 4-byte, the size is 512*4 bytes
256 g_tAppThread_2 = osThreadCreate(&tThreadDef, NULL);
257 if (g_tAppThread_2 == NULL)
258 {
259     printf("To create the thread for AppThread_2 is fail.\n");
260 }
261
262 // create the message queue for AppMessageQ
263 tMessageDef.queue_sz = APP_MESSAGE_Q_SIZE; // number of elements in the queue
264 tMessageDef.item_sz = sizeof(S_MessageQ); // size of an item
265 tMessageDef.pool = NULL; // reserved, it is no used
266 g_tAppMessageQ = osMessageCreate(&tMessageDef, g_tAppThread_2);
267 if (g_tAppMessageQ == NULL)
268 {
269     printf("To create the message queue for AppMessageQ is fail.\n");
270 }
271

```

## CONTACT

[sales@Opulinks.com](mailto:sales@Opulinks.com)