

# OPL1000

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

## AT/Debug UART Switch Command Application Note



OPULINKS

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

Copyright © 2018, OpuLinks. All Rights Reserved.

Date	Version	Contents Updated
2018-10-11	0.1	<ul style="list-style-type: none"><li>Initial Release</li></ul>

TABLE OF CONTENTS

1. 介绍 4

1.1. 文档应用范围 4

1.2. 缩略语 4

1.3. 参考文献 4

2. 如何使用 5

3. 如何設定與開發 6

3.1. 設定初始的狀態值 6

3.2. 指令函式 6

3.3. 設定 IO 的功能 7

LIST OF FIGURES

Figure 1 : AT UART work mode .....5

Figure 2: AT UART switch to Debug UART.....5

LIST OF TABLES

Table 1: AT/Debug UART Switch Command .....5

## 1. 介绍

### 1.1. 文档应用范围

本文介绍在相同的 IO 脚位上，如何利用指令切换 AT UART 和 Debug UART 模式。

### 1.2. 缩略语

Abbr.	Explanation
UART	Universal Asynchronous receiver-transmitter

### 1.3. 参考文献

[1] DEVKIT 快速使用指南 OPL1000-DEVKIT-getting-start-guide.pdf

[2] OPL1000 硬件设计手册 OPL1000-HDK.pdf

## 2. 如何使用

當固件設定的 IO0 和 IO1 是 AT UART，IO8 和 IO9 是 Debug UART，IO8 和 IO9 可以依照用戶設計而製定其它 IO 腳位。在 AT UART 打入“at+switchdbg”可以改變 IO0 和 IO1 到 Debug UART。在 Debug UART 中，打入“switchat”可以改變 IO0 和 IO1 到 AT UART。可以參考

· 用戶在使用上可以更清楚明白。

Table 1: AT/Debug UART Switch Command

指令 \ IO 腳位	IO 0、1	IO 8、9
at+switchdbg	Debug UART	AT UART
Switchat	AT UART	Debug UART

當用戶一開始開機的時候，IO0 和 IO1 是 AT UART 功能。如 Figure 1 所示。敲入 at 的指令時，會出現 ok 的回應。表示当前工作模式是 AT UART 功能。

Figure 1 : AT UART work mode

```
>><CHECK>
SPI load patch, last index 614 result 2
BootMode 10
>at
OK
```

在 AT UART 功能下，用戶可以透過 at+switchdbg 的指令切換到 Debug UART。當用戶切換到 Debug UART 時，可以敲 at 指令，會得到 at 的回應，如 Figure 2 所示。當用戶重新啟動板子時，必須在重新敲入切換 Debug UART 的指令。

Figure 2: AT UART switch to Debug UART

```
>at+switchdbg
C!
Switch: Dbg UART
>
>atat
```

## 3. 如何設定與開發

### 3.1. 設定初始的狀態值

初始化的定義在 `hal_pin_config_project.h`

預設初始化 IO0 和 IO1 為 AT UART 模式，如下：

```
/* Select IO 0/1 UART mode. For switching UART use */  
  
#define HAL_PIN_0_1_UART_MODE    IO01_UART_MODE_AT    //IO01_UART_MODE_AT  
                                //IO01_UART_MODE_DBG
```

當使用者想要切換 IO0 和 IO1 為 Debug UART 時，必需修改如下：

```
/* Select IO 0/1 UART mode. For switching UART use */  
  
#define HAL_PIN_0_1_UART_MODE    IO01_UART_MODE_DBG    //IO01_UART_MODE_AT  
                                //IO01_UART_MODE_DBG
```

### 3.2. 指令函式

下列函式皆為控制交換 AT UART 和 Debug UART 的行為：

- `at_cmd_at_switch_to_dbg` 函式在 `at_cmd_table_ext.c` 裡頭。
- `ParseSwitchAT_DBGCommand_patch` 函式在 `cli_patch.c` 裡頭。



### 3.3. 設定 IO 的功能

- 下表中的源碼定義在 *at\_cmd\_switch\_uart1\_dbguart* 的函式之中，在 *at\_cmd\_common\_patch.c* 裡頭。
- 當前轉換範例的 IO 數字為 8、9，但用戶可以依據用戶的設計，設定成符合用戶的 IO 腳位，不侷限於 IO 腳位為 8、9。
  - a. UART 行為中，拉高電位表示不作動。Tx 會自主把 pin 腳位拉高或低，但 Rx 不會自主把 pin 腳位的電流拉高或拉低，並且外部也沒有接 pull high 的原件。要讓 Rx 的腳位在預設的情況下不作動，故在設定上必須設定為 *PIN\_DRIVING\_HIGH*。
  - b. 其它腳位設定為 float 的原因，當腳位設定為 HIGH 或 LOW 時，腳位必須提供較大的電流讓腳位作動或不作動，這也是因為腳位設定成 *PIN\_DRIVING\_FLOAT* 的原因。

```
if (g_eIO01UartMode == UART_AT)
{
    /* AT UART 在 0、1 switch to Debug UART */
    Hal_Pin_ConfigSet(0, PIN_TYPE_UART_APS_TX, PIN_DRIVING_FLOAT);
    Hal_Pin_ConfigSet(1, PIN_TYPE_UART_APS_RX, PIN_DRIVING_FLOAT);

    /* Debug UART 在 8、9 switch to AT UART */
    Hal_Pin_ConfigSet(8, PIN_TYPE_UART1_TX, PIN_DRIVING_FLOAT);
    Hal_Pin_ConfigSet(9, PIN_TYPE_UART1_RX, PIN_DRIVING_HIGH);
}
else
{
    /* AT UART 在 8、9 switch to Debug UART */
    Hal_Pin_ConfigSet(8, PIN_TYPE_UART_APS_TX, PIN_DRIVING_FLOAT);
    Hal_Pin_ConfigSet(9, PIN_TYPE_UART_APS_RX, PIN_DRIVING_HIGH);

    /* Debug UART 在 0、1 switch to AT UART */
    Hal_Pin_ConfigSet(0, PIN_TYPE_UART1_TX, PIN_DRIVING_FLOAT);
    Hal_Pin_ConfigSet(1, PIN_TYPE_UART1_RX, PIN_DRIVING_FLOAT);
}
```

## CONTACT

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