

OPL1000

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

AT/Debug UART Switch Command Application Note



OPULINKS

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

Copyright © 2019, Opulinks. All Rights Reserved.

OPL1000-AT-UART-Switch-Command-R01 | Draft Version V01

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

TABLE OF CONTENTS

1. Introduction _____ 4

1.1. Scope of Document Application _____ 4

1.2. Abbreviations _____ 4

1.3. References _____ 4

2. How to use _____ 5

3. How to set up and develop _____ 7

3.1. Set Initial Status Values _____ 7

3.2. Command Functions _____ 7

3.3. Setting IO Functions _____ 8

LIST OF FIGURES

Figure 1 : AT UART work mode5

Figure 2: AT UART switch to Debug UART6

LIST OF TABLES

Table 1: AT/Debug UART Switch Command5

1. INTRODUCTION

1.1. Scope of Document Application

This document outline the method to use command to switch between “AT UART” and Debug UART” modes on the same pin.

1.2. Abbreviations

Abbr.	Explanation
UART	Universal Asynchronous receiver-transmitter

1.3. References

[1] OPL1000-DEVKIT-getting-start-guide.pdf

[2] OPL1000-HDK.pdf

2. HOW TO USE

When the values set for IO0 in firmware is "AT UART", and those for IO8 and IO9 is "Debug UART", IO8 and IO9 can designated other IO pins according to user design. When entering "at+switchdbg" under "AT UART", it can change IO0 and IO1 to Debug UART. Under "Debug UART", when entering "switchat", it will change the values of IO0 and IO1 to "AT UART". User can refer the Table 1 to get more clear in usage.

Table 1: AT/Debug UART Switch Command

cmd \ IO pin	IO 0 · 1	IO 8 · 9
at+switchdbg	Debug UART	AT UART
Switchat	AT UART	Debug UART

When user just powers up, IO0 and IO1 are AT UART functions, as shown in Figure 1. When entering "at" command, the "ok" response will appear which means that the current working mode is "AT UART" function.

Figure 1 : AT UART work mode

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

Under "AT UART" function, users can switch to "Debug UART" through "at+switchdbg" command. When users switch to "Debug UART", "at" command can be issued to get "at" response, as shown in Figure 2. When users re-start the development board, the command of switching to "Debug UART" must be re-entered.

Figure 2: AT UART switch to Debug UART

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


3. HOW TO SET UP AND DEVELOP

3.1. Set Initial Status Values

Initialization is defined in "hal_pin_config_project.h".

The default initialization of IO0 and IO1 is under AT UART mode, as shown below:

```
/* 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
```

When users would like to switch IO0 and IO1 to "Debug UART", the following modifications are required:

```
/* 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. Command Functions

The following functions are actions of control of switch between AT UART and Debug UART:

- "at_cmd_at_switch_to_dbg" function is in "at_cmd_table_ext.c"
- "ParseSwitchAT_DBGCommand_patch" function is in "cli_patch.c".

3.3. Setting IO Functions

- The source codes shown in the table below are defined in the function of "at_cmd_switch_uart1_dbguart", in "at_cmd_common_patch.c".
- The IO numbers in the current switch example are 8 and 9, but user can designate IO pin to suit users' design needs, and not limited to the IO pin of 8 and 9.
 - a. In UART behaviors, pull-high potential means not activated. Tx will autonomously raise or lower pin location. However, Rx will not autonomously increase or decrease the voltage of pin, without external connection to pull-high resistor component. In order to allow Rx pin not to be activated in default, the setting must be designated as "PIN_DRIVING_HIGH".
 - b. The reason for setting other pins as "float", as pin is set as "HIGH" or "LOW", pin location must provide larger current to enable the pin to be "activated" or "not-activated", which is the reason the pin is set as "PIN_DRIVING_FLOAT".

```

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