OPL1000

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

PinMux Tool User Guide

OPULINKS

REVISION HISTORY

Date	Version	Contents Updated
2018-05-09	0.1	Initial Release
2018-05-30	0.2	• Update section 3.4 because PWM port setting is modified in v0.5 version • SPI setting is updated in v0.5, hence section 3.2 is updated.
2018-08-01	0.3	• Add two additional IO resource, IO16/IO17 • Update all images
2018-09-07	0.4	• Update GPIO tab setting according to v0.7 version SW.
2018-09-14	0.5	• Add include file generation description according to v0.8 version SW.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of Document Application

This document outlines the usage method for OPL1000 pin-Mux tool. Pin mux configuration software is used for the setup of OPL1000 external register parameters and IO pin setting. Peripherals include PWM, AUX(SAR ADC), SPI (master), UART (flow control optional) , I2C(master or slave) and GPIO.

1.2. Abbreviations

Abbr.	Description
AUX	Auxiliary ADC Module
CPHA	Clock PHAse
CPOL	Clock POLarity
DevKit	Develop Kit OPL1000 board
FW	FirmWare · Embedded Software Operating on CPU
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit bus I2C
PWM	Pulse-Width Modulation
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver / Transmitter

1.3. References

[1] OPL1000 data sheet, OPL1000-DS-R04.pdf

2. INTERFACE INTRODUCTION

OPL1000 provides external pins used for configuring three types of communication ports (UART, I2C and SPI), as well as particular signal port (such as PWM, AUX/ADC and GPIO). As these pin configurations are multi-purpose, e.g. a particular signal line, that can be configured as UART, can also be configured as AUX/ADC or GPIO port. Pin-Mux tool offers a way to help customers flexibly and conveniently define multi-purpose mode and port parameter configuration. Pin-Mux tool output is a .c document (OPL1000_pin_mux_define.c) and "ini" configuration document. ".c" document contains pin multi-purpose setup and designated port parameter configuration table, and based on this configuration table, users can utilize corresponding API to complete OPL1000 pin multi-purpose set up and port work-mode configuration.

OPL1000 Pin-Mux Interface is shown as Figure 1.

Figure 1: Pin-Mux tool Interface

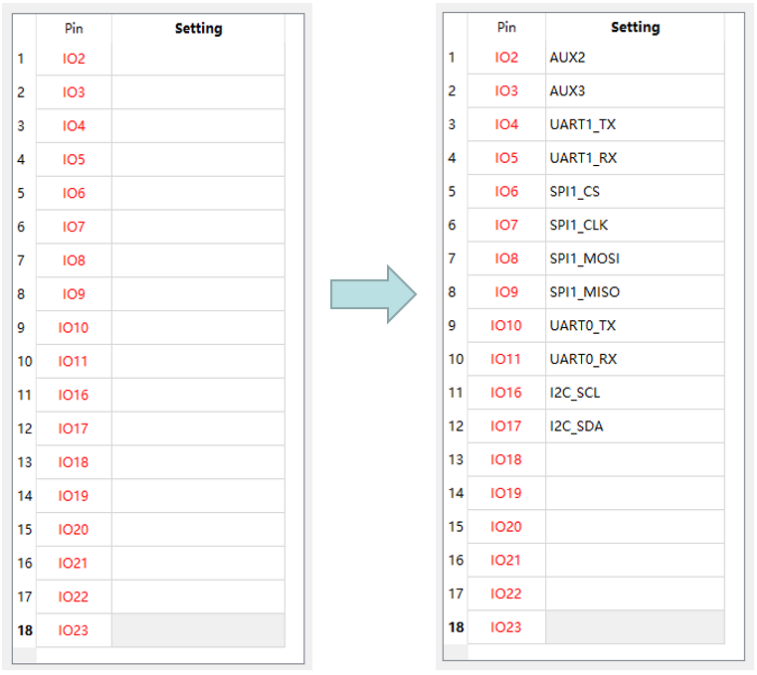
Interface includes four parts:

1. Pin configuration result list: This is the output result after right-side pins are configured.
2. External pin and parameter configuration dialogue box: It contains 7 tag pages, used to select external resources, including UART, SPI, I2C, PWM, AUX/ADC and GPIO, etc., and set their pin output according to their needs.
3. Version message and user manual: Indicate the current software version number and display of user manual of this software.
4. Select corresponding external pin, as a particular pin can only be configured as an external resource at a time.
5. Output ".c", ".h" and ".ini": Once external resource and parameter configuration is done, click "Build" button to generate ".c", ".h" and ".ini" documents.

2.1. Pin Configuration Result List

As this list is generated by IO take page dialogue box configuration, when the "configuration" option is not selected, pin setting list would be empty, as shown in Figure 2. Once external IO pin is setup, that is having selected corresponding multi-selection boxes, pin setting list displays pin distribution result.

Figure 2: Configured IO List



	Pin	Setting
1	IO2	
2	IO3	
3	IO4	
4	IO5	
5	IO6	
6	IO7	
7	IO8	
8	IO9	
9	IO10	
10	IO11	
11	IO16	
12	IO17	
13	IO18	
14	IO19	
15	IO20	
16	IO21	
17	IO22	
18	IO23	

	Pin	Setting
1	IO2	AUX2
2	IO3	AUX3
3	IO4	UART1_TX
4	IO5	UART1_RX
5	IO6	SPI1_CS
6	IO7	SPI1_CLK
7	IO8	SPI1_MOSI
8	IO9	SPI1_MISO
9	IO10	UART0_TX
10	IO11	UART0_RX
11	IO16	I2C_SCL
12	IO17	I2C_SDA
13	IO18	
14	IO19	
15	IO20	
16	IO21	
17	IO22	
18	IO23	

2.2. External Pin and Parameter Configuration Dialogue

External pin and parameter configuration dialogue box contains seven tag pages, with external selection and parameter configuration tag pages such as UART, SPI, I2C, PWM, AUX/ADC, and GPIO, where IO tag page is the external pin selection dialogue box. IO tag page selection is displayed on the left-side pin setting list.

Figure 3: External Tag Options

For various external resource selection and parameter configuration, please refer to Chapter 3.

3. EXTERNAL RESOURCE SELECTION AND PARAMETER CONFIGURATION

This chapter outlines the external selection and parameter configuration function such as UART, SPI, I2C, PWM, AUX/ADC, and GPIO, etc.

The number of external resources supported by OPL1000 is shown as the table below.

Table 1: Number of External Resources Supported by OPL1000

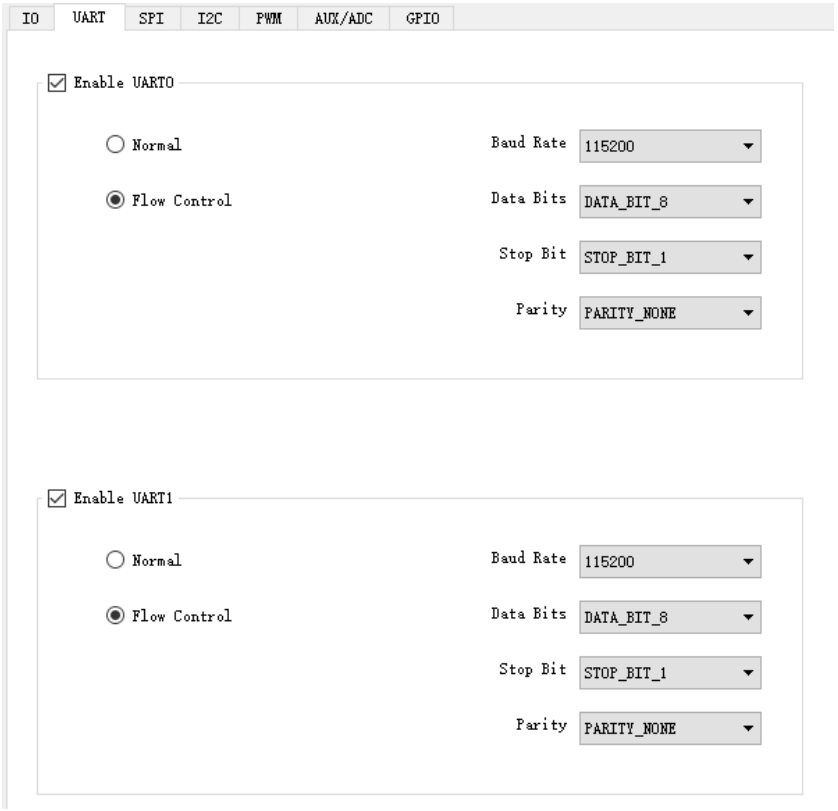
External Resources	Number
UART	2
I2C	1
SPI	2
PWM	6
AUX/ADC	10
GPIO	18

3.1. UART Selection and Parameter Configuration

OPL provides 2 ways of UART, i.e. UART0 and UART1. The setup is as shown in Figure 4.

By selecting "Enable UART0/UART1" multi-selection dialogue boxes, it indicates that UART external is enabled, and that allows users to indicated pin configuration of every signal line in IP tag page.

Figure 4: UART Selection and Parameter Configuration



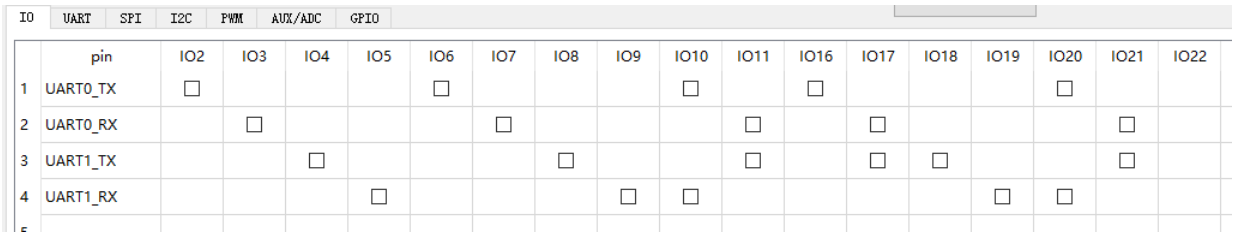
As OPL1000 UART offers two types of working modes, one is Normal, and the other is the intensified mode for flow traffic control.

When selecting Normal mode, UART has two signal lines requiring pin configuration, as shown in Figure 5.

To select the working mode for increasing volume traffic control, UART has four signal lines with pin configuration, as shown in Figure 6.

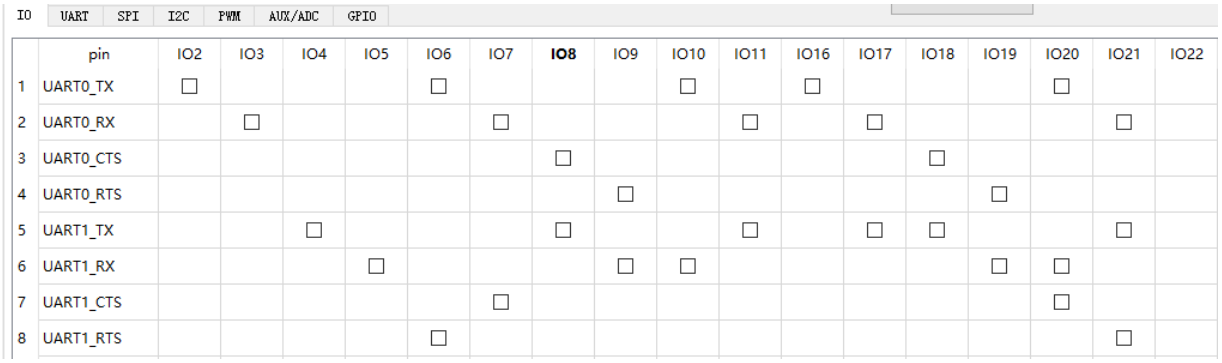
Serial-port configuration includes: baud rate, data bit, Termination bit, Calibration bit, as each parameter is selected from pull-down menu.

Figure 5: UART Signal Line Configuration under Normal Mode



	pin	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21	IO22
1	UART0_TX	☐				☐				☐		☐				☐		
2	UART0_RX		☐				☐				☐		☐				☐	
3	UART1_TX			☐				☐			☐		☐	☐			☐	
4	UART1_RX				☐				☐	☐					☐	☐		

Figure 6: UART Signal Line Configuration under Flow Traffic Control Mode



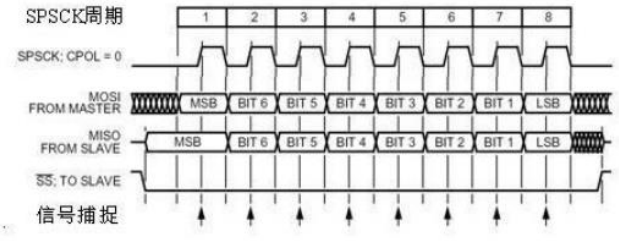
	pin	IO2	IO3	IO4	IO5	IO6	IO7	**IO8**	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21	IO22
1	UART0_TX	☐				☐				☐		☐				☐		
2	UART0_RX		☐				☐				☐		☐				☐	
3	UART0_CTS							☐						☐				
4	UART0_RTS								☐						☐			
5	UART1_TX			☐				☐			☐		☐	☐			☐	
6	UART1_RX				☐				☐	☐					☐	☐		
7	UART1_CTS						☐									☐		
8	UART1_RTS					☐											☐	

3.2. SPI Selection and Parameter Configuration

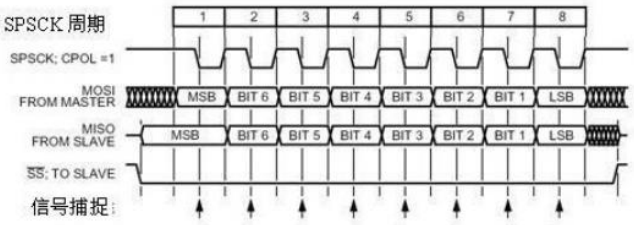
OPL1000 supports two kinds of SPI’ s, i.e. SPI1 and SPI2. Having clicked to enable SPI1/SPI2 multi-selection dialogue box, it means that certain external resources shall be configured as designated SPI ports. SPI supports two kinds of work modes, Single mode and Quad mode. Quad mode corresponds to high-speed SPI communication modes.

Under Single mode, CPOL and CPHA definition has 4 working time-sequences, as shown in Figure 7.

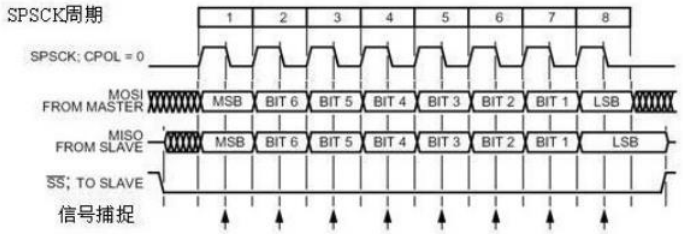
Figure 7: Four Working Time-Sequences under Single Mode



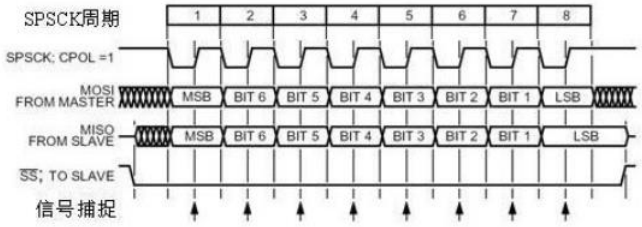
CPHA=0，CPOL=0时的数据/时钟时序图



CPHA=0，CPOL=1时的数据/时钟时序图



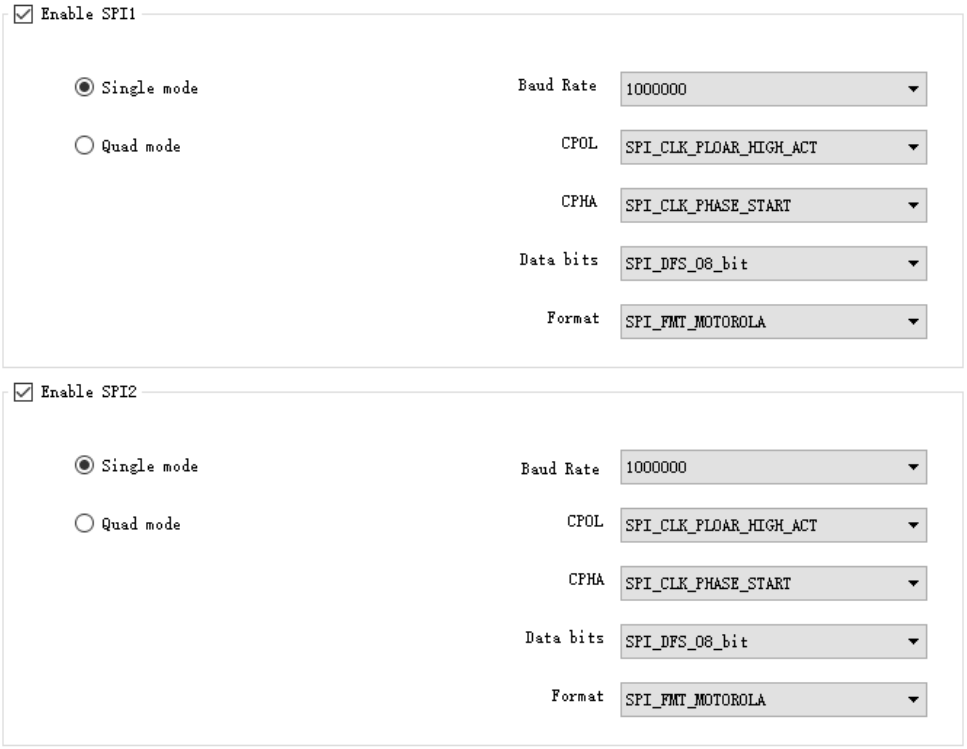
CPHA=1，CPOL=0时的数据/时钟时序图



CPHA=1，CPOL=1时的数据/时钟时序图

SPI resource selection and parameter configuration are as shown in Figure 8.

Figure 8: SPI Selection and Parameter Configuration



Under Single mode, SPI port has four signal line requiring configuration, as shown in Figure 9.

Figure 9: Single Mode Pin Configuration

IO | UART | SPI | I2C | PWM | AUX/ADC | GPIO

	pin	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21	IO22
1	SPI1_CLK						☐											
2	SPI1_CS					☐												
3	SPI1_MISO								☐									
4	SPI1_MOSI							☐										
5	SPI2_CLK			☐											☐			
6	SPI2_CS				☐									☐				
7	SPI2_MISO	☐															☐	
8	SPI2_MOSI		☐													☐		

3.3. I2C Selection and Parameter Configuration

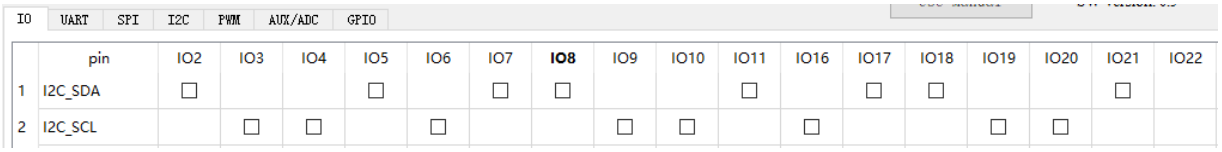
OPL1000 supports one way of I2C, for Master and Slave.

By enabling "Enable I2C", it indicates that selection of I2C port configuration. The parameter configuration of I2C is as shown in Figure10.

Figure 10: I2C Parameter Configuration

I2C has two signal lines requiring configuration, which can be selected from 16 IO pins, as shown in Figure 11.

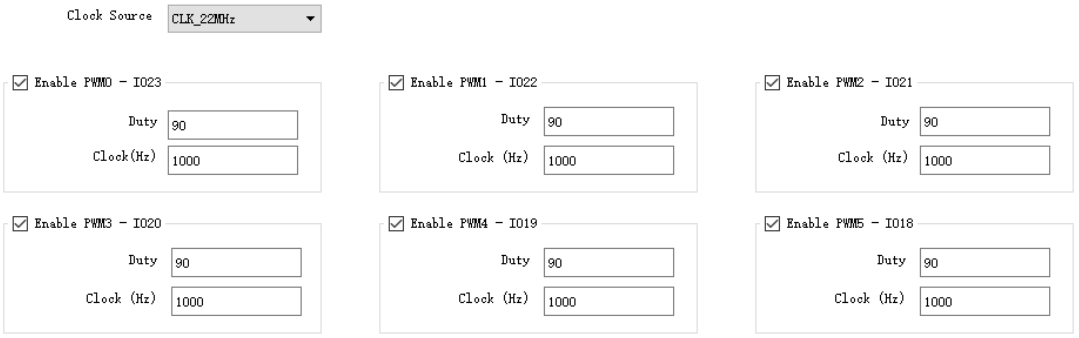
Figure 11: I2C Pin Multi-Purpose Setup



3.4. PWM Selection and Parameter Configuration

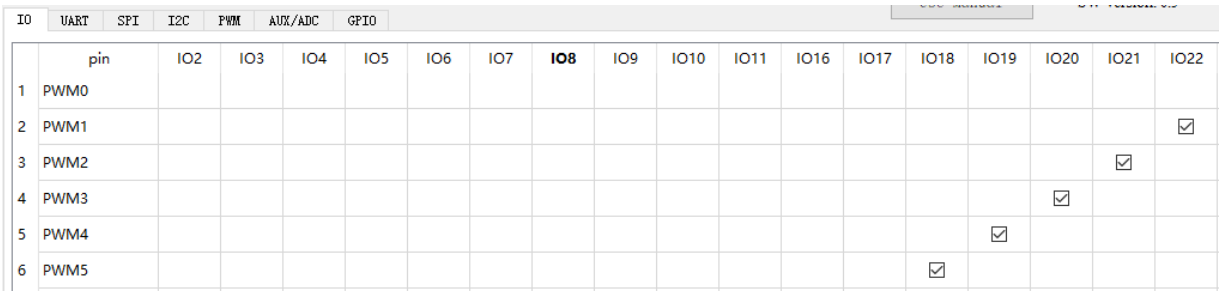
OPL1000 supports sixe PWM ports. By enabling "Enable PWMx" multi-selection dialogue box, it means that certain PWM port is selected. Please note that PWM port and external pin mapping are of one-to-one corresponding relationship. For example, PWM0 means that IO23 pin is designated for PWM0. PWM selection and parameter configuration is shown in Figure 12.

Figure 12: PWM Parameter Configuration



As PWM and pins are of one-to-one corresponding relationship, therefore if pin configuration is of multi-purpose, PWM should be selected first, before selecting other communication port signal line configuration (as they of more choices), as shown in Figure 13, where 6 ports of PWM0 ~ PWM5 and an I2C communication port are selected.

Figure 13: PWM Pin Multi-Purpose Selection

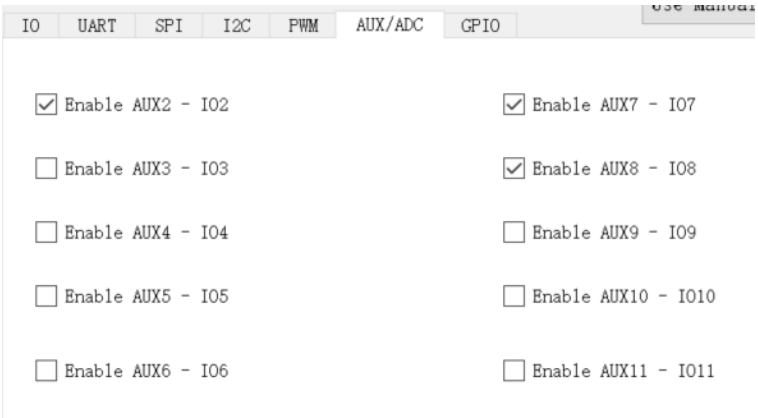


	pin	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21	IO22
1	PWM0																	
2	PWM1																	☑
3	PWM2																☑	
4	PWM3															☑		
5	PWM4														☑			
6	PWM5													☑				

3.5. AUX/ADC Selection and Parameter Configuration

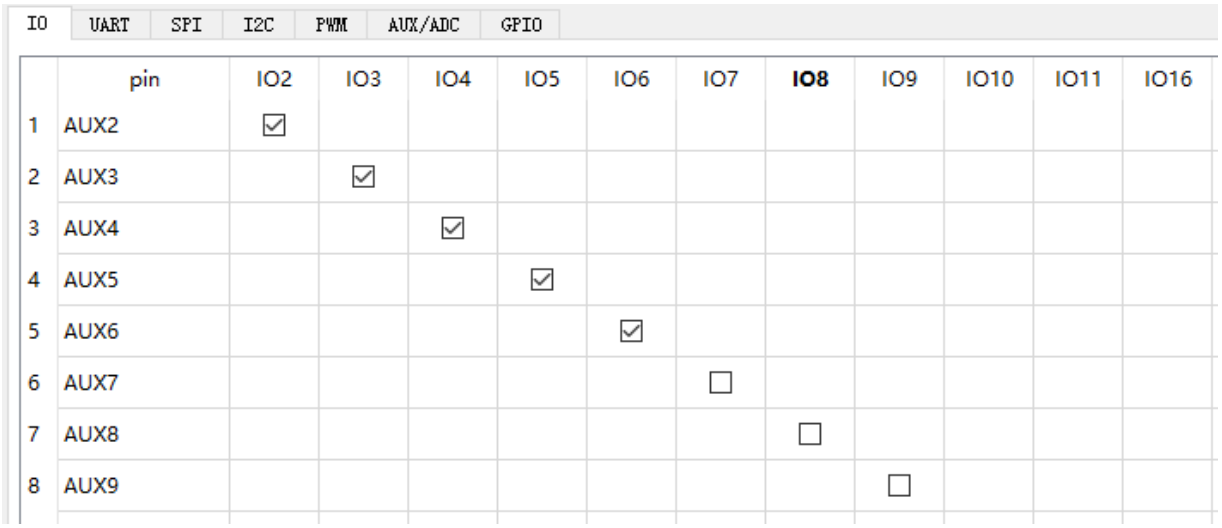
AUX/ADC is used to configure support of ADC port. ADC port is a signal line simulating signal input, therefore there is no parameter configuration. OPL1000 supports 10 AUX/ADC ports. And just like PWM, AUXn and IO pin are of one-to-one corresponding relationship, as shown in Figure14.

Figure 14: AUX/ADC Selection



Just like PWM, due to the one-to-one mapping relationship with IO pin, when configuring AUX port, its pin will be selected first, and then configured for other communication ports. In Figure15, AUX2, AUX7, AUX8 and I2C signal lines need to be configured. By selecting AUX port configuration first, I2C can be configured flexibly amongst the remaining pins.

Figure 15: AUX/ADC Multi-Purpose Selection



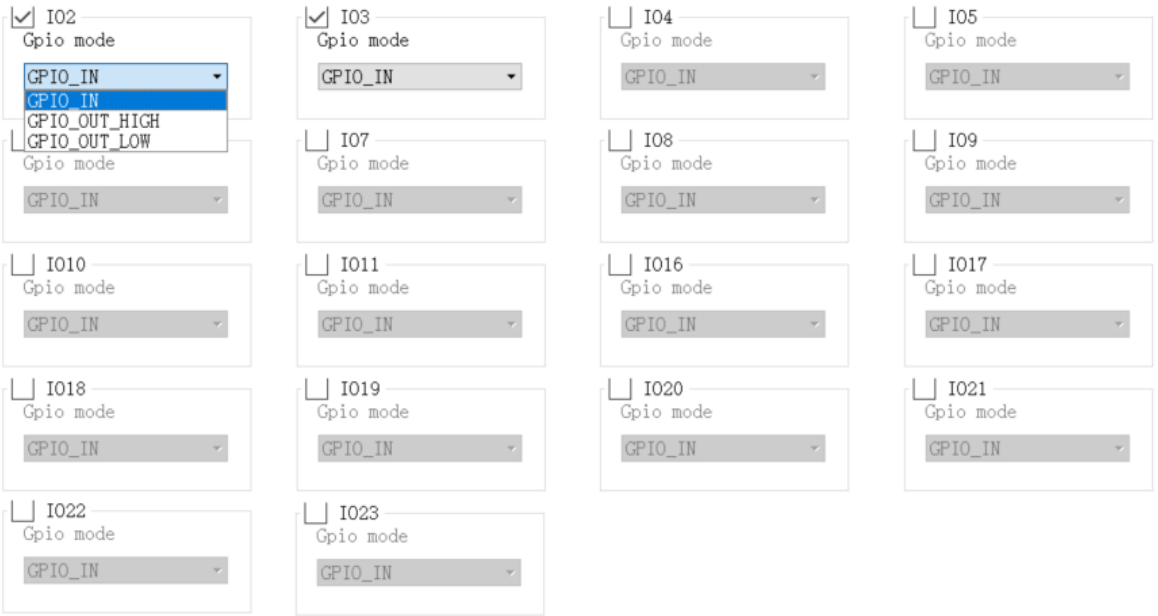
3.6. GPIO Selection and Parameter Configuration

OPL supports 18 GPIO signal selection. It means that the offered 18 pins can all be configured as GPIO signal. The three configurations options for GPIO signal working mode are:

(1) Input Signal GPIO_IN, (2) Output Signal, with configuration of high level GPIO_OUT_HIGH, and (3) Output signal, with configuration of low level GPIO_OUT_LOW. When users are selecting GPIO, circuit design must also be considered at the same time.

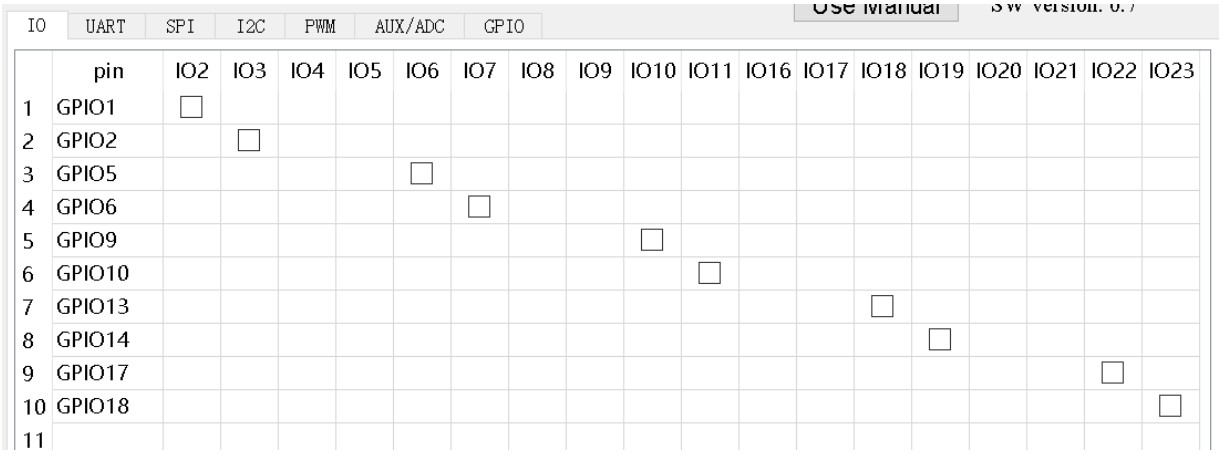
GPIO and IO pin are also of one-to-one corresponding relationship. Selection and parameter configuration are shown in Figure 16.

Figure 16: GPIO Selection and Parameter Configuration



GPIO pin multi-purpose configuration is as shown in Figure 17. As PWM, AUX/ADC and others similar are configured with other communication pins, GPIO needs to be selected first, before defining communication pins.

Figure 17: GPIO Pin Multi-Purpose Configuration



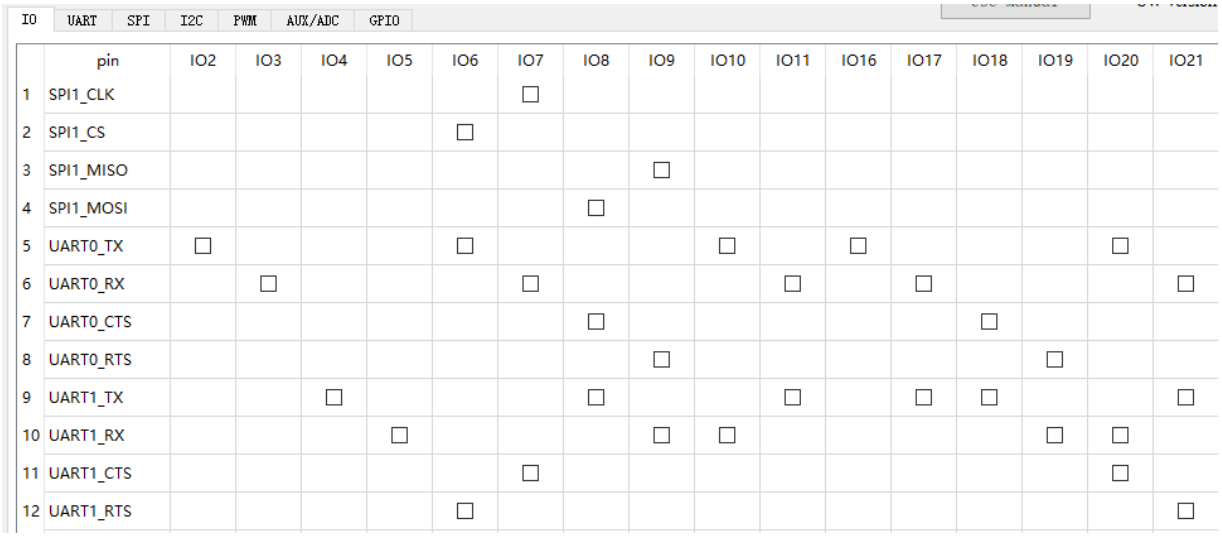
	pin	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21	IO22	IO23
1	GPIO1	☐																	
2	GPIO2		☐																
3	GPIO5					☐													
4	GPIO6						☐												
5	GPIO9									☐									
6	GPIO10										☐								
7	GPIO13													☐					
8	GPIO14														☐				
9	GPIO17																	☐	
10	GPIO18																		☐
11																			

4. IO PIN SELECTION

As Chapter 3 outlines the selection of all external (communication ports and signal ports), and parameter configuration, having chosen which ports to configure, the selected external resources and options used to configure pin are displayed in IO tag pages. When signal line pin of a particular port is selected, i.e. the corresponding multi-selection boxes ticked, then it means this IP pin is occupied by this signal line, and other signal lines would not be able to access it. The same multi-selection box will be inhibited. As shown in Figure 18, once CLK of SPI1 is designated and multi-selection dialogue boxes such as IO7, SPI2_CLK, UART0_RX, and I2C_SDA in the same column as IO7 shall be inhibited.

Through IO tag list, users can clearly and intuitively know which pins can be designated to which signal line. And by clicking multi-selection dialogue box, it ensures that every selected signal line will be designated with pin resource.

Figure 18: External IO Pin Definition



	pin	IO2	IO3	IO4	IO5	IO6	IO7	IO8	IO9	IO10	IO11	IO16	IO17	IO18	IO19	IO20	IO21
1	SPI1_CLK						☐										
2	SPI1_CS					☐											
3	SPI1_MISO								☐								
4	SPI1_MOSI							☐									
5	UART0_TX	☐				☐				☐		☐				☐	
6	UART0_RX		☐				☐				☐		☐				☐
7	UART0_CTS							☐						☐			
8	UART0_RTS								☐						☐		
9	UART1_TX			☐				☐			☐		☐	☐			☐
10	UART1_RX				☐				☐	☐					☐	☐	
11	UART1_CTS						☐									☐	
12	UART1_RTS					☐											☐

5. GENERATE PIN MULTI-PURPOSE DEFINITION DOCUMENT

When port definition and pin distribution is defined, click "Build" button, to generate "OPL1000_pin_mux_define.c" document, "hal_pin_config_project.h" header document and a "ini" document. ".c" and ".h" documents are used for subsequent port initialization and pin multi-purpose configuration, while "ini" document recording the same message to be used for other purposes such as the automatic testing for pin multi-purpose module.

Figure 19: Generating Document

As the names of ".c" and ".h" documents are unchanged, as "ini" document contains data and time messages. Therefore, in the dialogue box it only displays the name of "ini" document, as ".c" and ".h" documents are stored in "pinmux.exe" under the same folder.

According to a particular application-defined pin multi-purpose solution, having utilizing pinmux tool to generate ".c" and ".h" documents, these two documents can be copied to users' own application project folder, before utilizing the corresponding API to complete pin multi-purpose configuration.

The content of OPL1000_pin_mux_define.c document is shown in Figure 19 and Figure 20.

Figure 20: OPL1000_pin_mux_define.c Part1

```
T_OPL1000_Periph OPL1000_periph = {
  1,{
     {UART_IDX_0,
      OPL1000_IO20_PIN,
      OPL1000_IO21_PIN,
      BLANK_PIN,
      BLANK_PIN,
      115200,
      DATA_BIT_8,
      PARITY_NONE,
      STOP_BIT_1,
      UART_SIMPLE},
     {UART_IDX_MAX,
      BLANK_PIN,
      BLANK_PIN,
      BLANK_PIN,
      BLANK_PIN,
      0,
      DATA_BIT_8,
      PARITY_NONE,
      STOP_BIT_1,
      UART_SIMPLE}
    },
  1,{I2C_SPEED_FAST,
    OPL1000_IO19_PIN,
    OPL1000_IO18_PIN,
    I2C_07BIT,
    0x7A,
    MASTER_HAS_STOP},
  2,{
     {SPI_IDX_1,
      OPL1000_IO6_PIN,
      OPL1000_IO7_PIN,
      OPL1000_IO9_PIN,
      OPL1000_IO8_PIN,
      BLANK_PIN,
      BLANK_PIN,
      1000000,
      SPI_CLK_PLOAR_HIGH_ACT,
      SPI_CLK_PHASE_START,
      SPI_FMT_MOTOROLA,
      SPI_DFS_08_bit,
      QMODE_DISABLE},
     {SPI_IDX_2,
      OPL1000_IO5_PIN,
      OPL1000_IO4_PIN,
      BLANK_PIN,
      BLANK_PIN,
      OPL1000_IO10_PIN,
      OPL1000_IO11_PIN,
      1000000,
      SPI_CLK_PLOAR_HIGH_ACT,
      SPI_CLK_PHASE_START,
      SPI_FMT_MOTOROLA,
      SPI_DFS_08_bit,
      QMODE_ENABLE},
    },
 // continue ...
```

Figure 21: OPL1000_pin_mux_define.c Part2

```
  0,{{BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0},
      {BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0},
      {BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0},
      {BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0},
      {BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0},
      {BLANK_PIN,CLK_32KHz,CFG_SIMPLE,0,0,0,0,0,0,0,0}},

 0,{BLANK_PIN,BLANK_PIN,BLANK_PIN,BLANK_PIN,BLANK_PIN,BLANK_PIN,BLANK_PIN,BLANK_PIN
 ,BLANK_PIN,BLANK_PIN},
  0,{{BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP},
      {BLANK_PIN,IO_OUTPUT,PULL_UP}}
 };
```

hal_pin_config_project.h header document defines the functional attributes of the 24 pins, from IO0 to IO23, as Figure 22 lists the definitions of Io0 to IO4.

Figure 22: hal_pin_config_project.h Header Document Pin Function Definition

```
// IO type select
#define HAL_PIN_TYPE_IO_0   PIN_TYPE_NONE         // PIN_TYPE_NONE
                             // PIN_TYPE_GPIO_INPUT
                             // PIN_TYPE_GPIO_OUTPUT_LOW
                             // PIN_TYPE_GPIO_OUTPUT_HIGH
                             // PIN_TYPE_UART0_CTS
                             // PIN_TYPE_UART1_TX
                             // PIN_TYPE_I2C_SCL
                             // PIN_TYPE_SPI2_IO_3
                             // PIN_TYPE_AUX_0
                             // PIN_TYPE_UART_APS_TX
                             // PIN_TYPE_UART_MSQ_RX
                             // PIN_TYPE_ICE_M3_DAT
                             // PIN_TYPE_ICE_M0_CLK

#define HAL_PIN_TYPE_IO_1   PIN_TYPE_NONE         // PIN_TYPE_NONE
                            // PIN_TYPE_GPIO_INPUT
                            // PIN_TYPE_GPIO_OUTPUT_LOW
                            // PIN_TYPE_GPIO_OUTPUT_HIGH
                            // PIN_TYPE_UART0_RTS
                            // PIN_TYPE_UART1_RX
                            // PIN_TYPE_I2C_SDA
                            // PIN_TYPE_SPI2_IO_2
                            // PIN_TYPE_AUX_1
                            // PIN_TYPE_UART_APS_RX
                            // PIN_TYPE_UART_MSQ_TX
                            // PIN_TYPE_ICE_M3_CLK
                            // PIN_TYPE_ICE_M0_DAT

#define HAL_PIN_TYPE_IO_2   PIN_TYPE_UART0_TX      // PIN_TYPE_NONE
                            // PIN_TYPE_GPIO_INPUT
                            // PIN_TYPE_GPIO_OUTPUT_LOW
                            // PIN_TYPE_GPIO_OUTPUT_HIGH
                            // PIN_TYPE_UART0_TX
                            // PIN_TYPE_I2C_SDA
                            // PIN_TYPE_SPI2_IO_1
                            // PIN_TYPE_AUX_2
                            // PIN_TYPE_UART_APS_TX
                            // PIN_TYPE_UART_MSQ_RX
                            // PIN_TYPE_ICE_M3_DAT
                            // PIN_TYPE_ICE_M0_CLK

#define HAL_PIN_TYPE_IO_3   PIN_TYPE_UART0_RX      // PIN_TYPE_NONE
                            // PIN_TYPE_GPIO_INPUT
                            // PIN_TYPE_GPIO_OUTPUT_LOW
                            // PIN_TYPE_GPIO_OUTPUT_HIGH
                            // PIN_TYPE_UART0_RX
                            // PIN_TYPE_I2C_SCL
                            // PIN_TYPE_SPI2_IO_0
                            // PIN_TYPE_AUX_3
                            // PIN_TYPE_UART_APS_RX
                            // PIN_TYPE_UART_MSQ_TX
                            // PIN_TYPE_ICE_M3_CLK
                            // PIN_TYPE_ICE_M0_DAT

#define HAL_PIN_TYPE_IO_4   PIN_TYPE_UART1_TX      // PIN_TYPE_NONE
                            // PIN_TYPE_GPIO_INPUT
                            // PIN_TYPE_GPIO_OUTPUT_LOW
                            // PIN_TYPE_GPIO_OUTPUT_HIGH
                            // PIN_TYPE_UART1_TX
                            // PIN_TYPE_I2C_SCL
                            // PIN_TYPE_SPI2_CLK
                            // PIN_TYPE_AUX_4
                            // PIN_TYPE_UART_APS_TX
                            // PIN_TYPE_UART_MSQ_RX
                            // PIN_TYPE_ICE_M3_DAT
                            // PIN_TYPE_ICE_M0_CLK
```

6. VERSION NUMBER AND USER MANUAL

By clicking "Use Manual" button will utilize the built-in Explore of Window system to load the user manual of this software, as shown in the diagram below.

CONTACT

sales@Opulinks.com