Dip Switch Demo

You will learn how to use GPIO as the input to identify the status of dip switches to control the digital tube to display the corresponding numbers by this demo.

As shown in Figure 1, you can use four dip switches on the extension board to control four 7-segment digital tubes to achieve the cyclic display of the numbers 1 to 4.

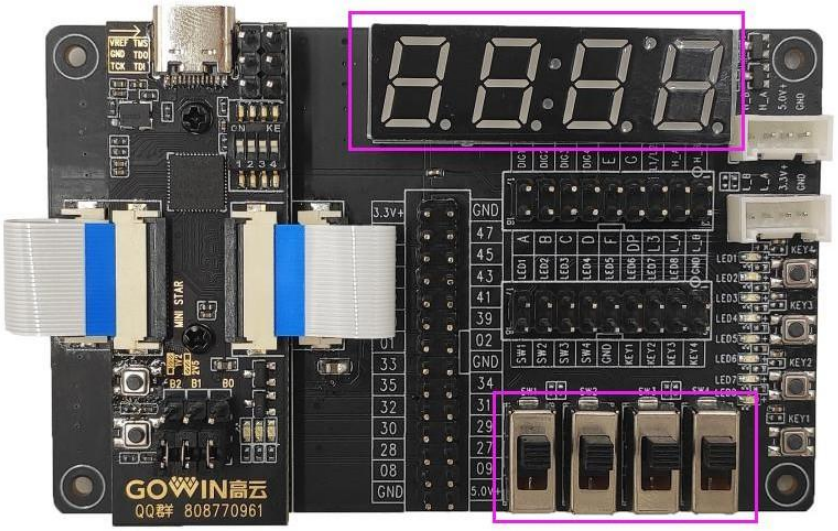

Figure 1 Mini-Star Development Board（GW1NSR-LV4CQN48P）and Extension

This demo introduction includes four parts:

1. GPIO Introduction
2. Hardware Design
3. Software Design
4. Download and Verification

GPIO Introduction

Gowin GW1NSR-LV4CQN48P is used as the platform in this demo. From DS861, GW1NSR series of FPGA Products Datasheet, you can learn the GPIO module and know there is a Cortex-M3 RISC embedded in this chip. The key to this demo is how to control the GPIO as the output. The design information listed below is excerpted from the datasheet.

The SoC microprocessor system communicates with the GPIO block through the AHB bus. The GIPO block interconnects with the FPGA. GPIO provides a 16 bits I/O interface with the following properties:

- Programmable interrupt generation capability. You can configure each bit of the I/O pins to generate interrupts;
- Bit masking support using address values;
- Registers for alternate function switching with pin multiplexing support;
- Thread safe operation by providing separate set and clear addresses for control registers.

The GPIO register is as shown in Table 3-21. The GPIO base address is 0x40010000.

The following table lists GPIO registers.

Table 3-21 GPIO Register

Name	Base Offset	Type	Data Width	Reset Value	Description
DATA	0x0000	Read/ Write	16	0x----	Data value [15:0]
DATAOUT	0x0004	Read/ Write	16	0x0000	Data output register value [15:0]
OUTENSET	0x0010	Read/ Write	16	0x0000	Output enable set [15:0] Write 1: Set the output enable bit. Write 0: No effect. Read 1: Indicates the signal direction as output. Read 0: Indicates the signal direction as input.
OUTENCLR	0x0014	Read/ Write	16	0x0000	Output enable clear [15:0]
ALTFUNCSET	0x0018	Read/ Write	16	0x0000	Alternative function set [15:0] Write 1: Sets the ALTFUNC bit. Write 0: No effect. Read 0: GPIO as I/O Read 1: ALTFUNC Function
ALTFUNCCLR	0x001C	Read/ Write	16	0x0000	Alternative function clear [15:0]
INTENSET	0x0020	Read/ Write	16	0x0000	Interrupt enable set [15:0] Write 1: Sets the enable bit. Write 0: No effect. Read 0: Interrupt disabled. Read 1: Interrupt enabled.
INTENCLR	0x0024	Read/ Write	16	0x0000	Interrupt enable clear [15:0] Write 1: Clear the enable bit. Write 0: No effect. Read 0: Interrupt disabled. Read 1: Interrupt enabled.
INTTYPESET	0x0028	Read/ Write	16	0x0000	Interrupt type set [15:0]
INTTYPECLR	0x002C	Read/ Write	16	0x0000	Interrupt type clear [15:0]
INTPOLSET	0x0030	Read/ Write	16	0x0000	Polarity-level, edge interrupt request configuration [15:0]
INTPOLCLR	0x0034	Read/ Write	16	0x0000	Polarity-level, edge interrupt request configuration [15:0]
INTSTATUS/ INTCLEAR	0x0038	Read/ Write	16	0x0000	Read interrupt status register Write 1: Clear the interrupt request

Note！

For the details, you can see 3.11.10 GPIO in DS861, GW1NSR series of FPGA Products Datasheet.

Hardware Design

Figure 2 shows the dip switch schematic. As shown in Figure 2, there are two statuses for the dip switch: Power VCC3V3 and GND. The signals are connected to the FPGA pins through SW1, SW2, SW3, and SW4; and the dip switch uses the input function of IO ports.

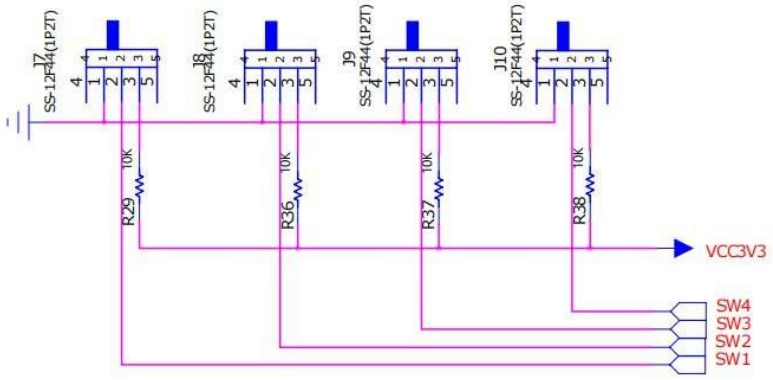

Figure 2 Dip Switch Schematic

The schematic of four 7-segment digital tubes is shown in Figure 3, and it is a common anode 8-segment digital tube. The common terminal DIG1, DIG2, DIG3, and DIG4 control 4 digital tubes respectively, active-high; the other terminal is controlled by A, B, C, D, E, F, G, and DP, active-low.

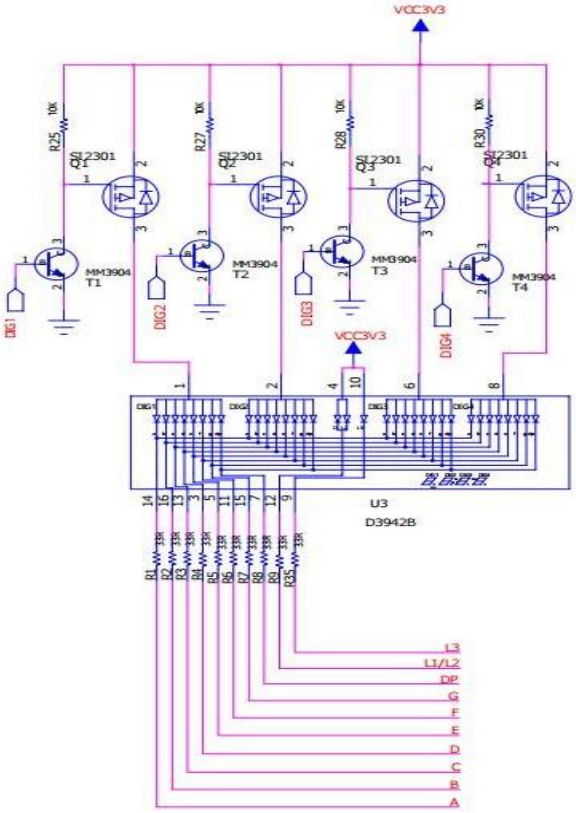

Figure 3 Digital Tube Schematic

The coding of common anode 7-segment digital tube is referred to the previous demo. The dip switch and digital tube control signal ports are connected to the FPGA pins. The connection is shown in Table 1 and Table 2, and the physical picture is shown in Figure 4.

Table 1 Digital Tube Pin Definition (Output)

Digital Tube	A	B	C	D	E	F	G	DP	DIG1	DIG2	DIG3	DIG4
FPGA	30	32	35	33	42	44	46	40	34	31	29	27
GPIO	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[11]	[10]	[9]	[8]

Table 2 Dip Switch Pin Definition (Input)

Dip Switch	SW1	SW2	SW3	SW4
FPGA	45	43	41	39
GPIO	[15]	[14]	[13]	[12]

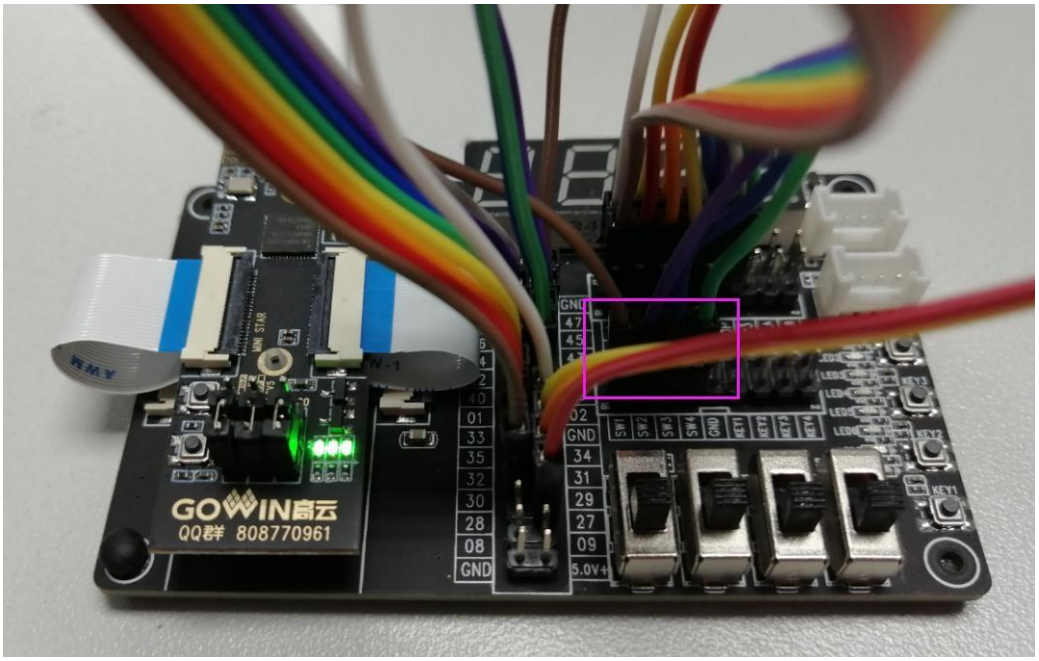

Figure 4 Connection of Dip Switch, Digital Tube, and FPGA

Software Design

The software design includes two parts: FPGA internal hardware logic and Cotex-M3 software control code, which can be modified on the basis of the digital tube project.

FPGA Internal Logic Design

This HDL code does not need to be modified, and you only need to modify the pin connection according to the table, which has been described

above. The pin definition modified according to Table 1 and Table 2 is shown in Figure 5.

I/O Constraints

	Port	Direction	Diff Pair	Location	Bank	Exclusive	IO Type
1	gpio_io[0]	inout		30	2	False	LVCMOS33
2	gpio_io[10]	inout		31	2	False	LVCMOS33
3	gpio_io[11]	inout		34	2	False	LVCMOS33
4	gpio_io[12]	inout		39	1	False	LVCMOS33
5	gpio_io[13]	inout		41	1	False	LVCMOS33
6	gpio_io[14]	inout		43	1	False	LVCMOS33
7	gpio_io[15]	inout		45	1	False	LVCMOS33
8	gpio_io[1]	inout		32	2	False	LVCMOS33
9	gpio_io[2]	inout		35	2	False	LVCMOS33
10	gpio_io[3]	inout		33	2	False	LVCMOS33
11	gpio_io[4]	inout		42	1	False	LVCMOS33
12	gpio_io[5]	inout		44	1	False	LVCMOS33
13	gpio_io[6]	inout		46	1	False	LVCMOS33
14	gpio_io[7]	inout		40	1	False	LVCMOS33
15	gpio_io[8]	inout		27	2	False	LVCMOS33
16	gpio_io[9]	inout		29	2	False	LVCMOS33
17	reset_n	input		20	3	False	LVCMOS18

Figure 5 FPGA Pin Configuration

Then click Place & Route to generate the logic file fpga_led.fs.

Cotex-M3 Software Control Design

You can modify this design on the basis of the digital tube project. Open Led.uvprojx in the Keil_led\PROJECT folder.

1. Set GPIO register; set GPIO[15:12] as the input of dip switch and GPIO[11:0] as the output of digital tube.

```
GPIO0->OUTENSET = 0x0fff;
```

2. Read the status of the dip switch and save it in the upper four bits [15:12] of the input data register DATA, and shift the upper four bits data right by four bits to write the status of the dip switch in the high-byte lower four bits [11:8] of the output register, which corresponds to the common terminal DIG1, DIG2,DIG3, and DIG4 of the digital tube; then set the lower 8 bits of the output register to 0 , which means that the A to DP segment of the digital tube is all low (the coded value of 8). The code is as follows:

```
s = (GPIO0->DATA>>4) & 0xff00;
GPIO0->DATAOUT = s;
```

When the dip switch is set to low, DIG1 to DIG4 will be low and the ommon anode digital tube will be off. When the dip switch is set to high, DIG1 to DIG4 will be high and the ommon anode digital tube will display number 8. The physical picture is shown in Figure 6, in which SW1 and SW2 are set to low, SW3 and SW4 are set to high; that is, digital tube 1 and 2 are off, digital tube 3 and 4 display.

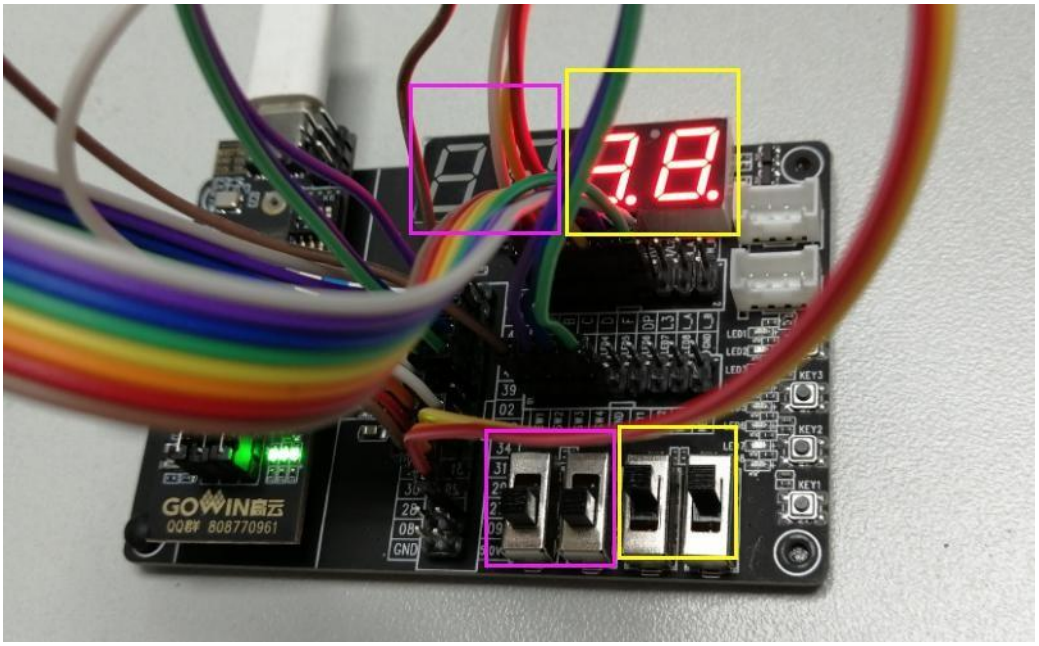

Figure 6 Dip Switch Controls Digital Tube On/Off

3. Based on the second step, modify GPIO->DATAOUT output data register to realize digital tube displaying different numbers.

According to the coding principle of common anode digital tube, the A to DP of digital tube correspond to GPIO[7:0] (active-low); DIG1 to DIG4 correspond to GPIO[11:8] (active-high); and the numbers 1 to 4 are coded as 0xf9,0xa4,0xb0,0x99 in order.

Firstly, the status value of the dip switch is stored in the upper four bits of DATA [15:12], and the DATA register value is shifted to the right by 8 bits, and then an AND operation is performed with 0x80, 0x40, 0x20 and 0x10 in turn; the high and low level of the dip switch are judged according to whether the operation result is equal to 0; then the detected status will be corresponded to the DIG common terminal of digital tube, and the processed data will be transferred to the output register to realize different numbers display controlled by dip switches. The reference code is shown in Figure 7 below.

4. After bulid, the download file led.bin is generated.

```
sw1= (GPIO0->DATA>>8) & 0x80;
sw2= (GPIO0->DATA>>8) & 0x40;
sw3= (GPIO0->DATA>>8) & 0x20;
sw4= (GPIO0->DATA>>8) & 0x10;

if(sw1!=0) key1= 0x08;
else key1= 0x00;

if(sw2!=0) key2= 0x04;
else key2= 0x00;

if(sw3!=0) key3= 0x02;
else key3= 0x00;

if(sw4!=0) key4= 0x01;
else key4= 0x00;
```

```
s= (key1<<8) | 0x00f9;       //1
GPIO0->DATAOUT = s;
Delay(833300);
//Delay(833300);

s=  (key2<<8) | 0x00a4;      //2
GPIO0->DATAOUT = s;
Delay(833300);
//Delay(833300);

s=  (key3<<8) | 0x00b0;      //3
GPIO0->DATAOUT = s;
Delay(833300);
//Delay(833300);

s=  (key4<<8) | 0x0099;      //4
GPIO0->DATAOUT = s;
Delay(833300);
// Delay(833300);
```

Figure 7 Reference Code

Download and Verification

Use Gowin Software to download, and the running is as shown in Figure 8. The FPGA hardware platform file is fpga_led.fs, and the Cotex-M3 software file is led.bin, so be careful to choose the correct file path and bulid file.

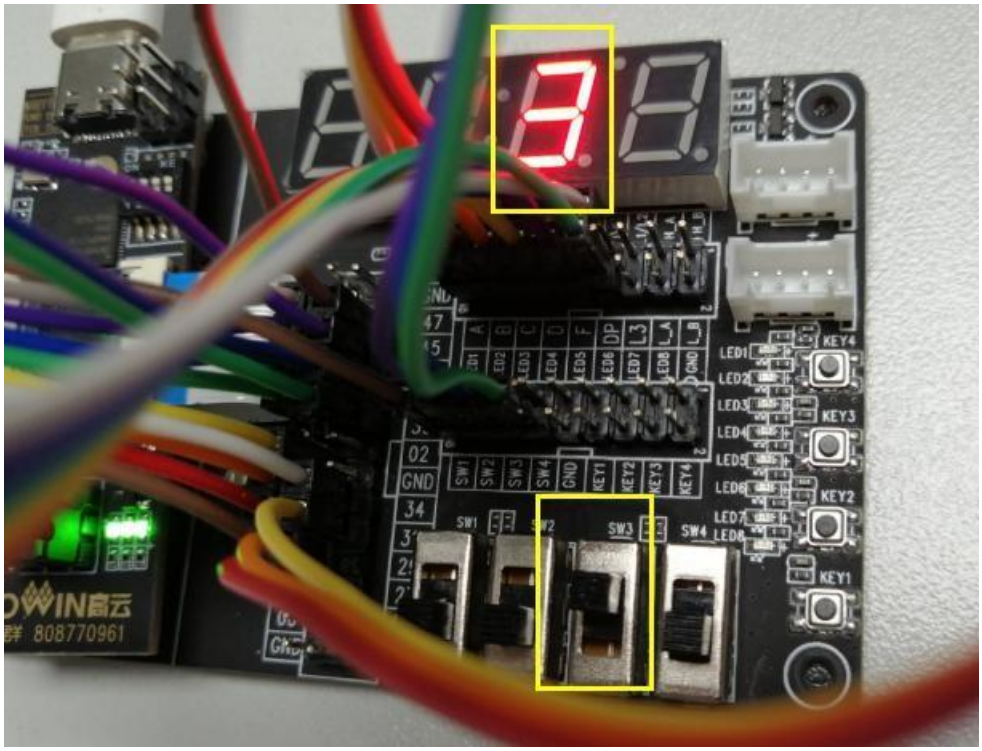

Figure 8 Demo Running