Digital Tube Demo

You will learn how to use GPIO as the output to control the digital tube to display numbers by this demo. As shown in Figure 1, the four 7-segment digital tubes on the extension board are used to achieve the cyclic display of numbers 0 to 9.

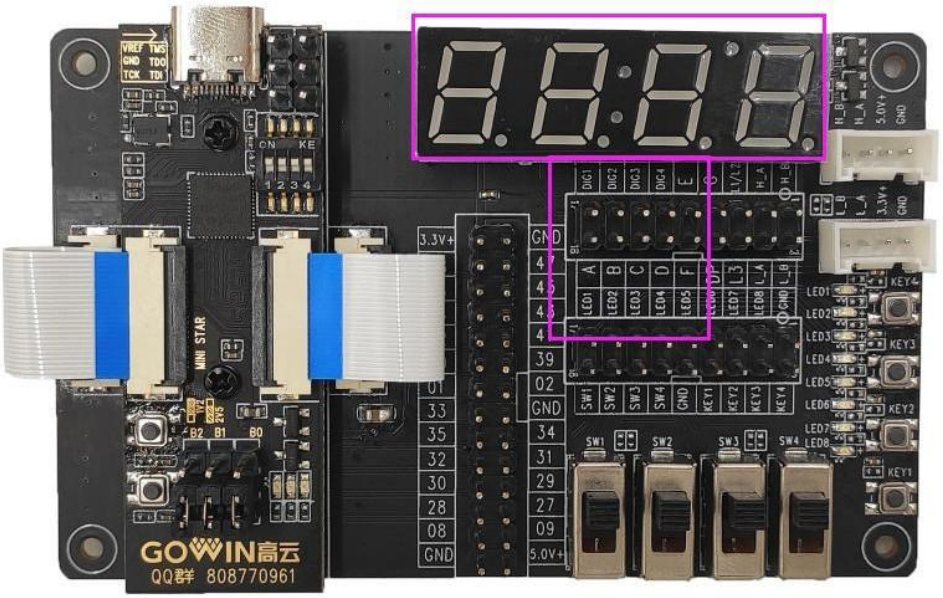

Figure 1 Mini-Star Development Board（GW1NSR-LV4CQN48P）and Extension

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

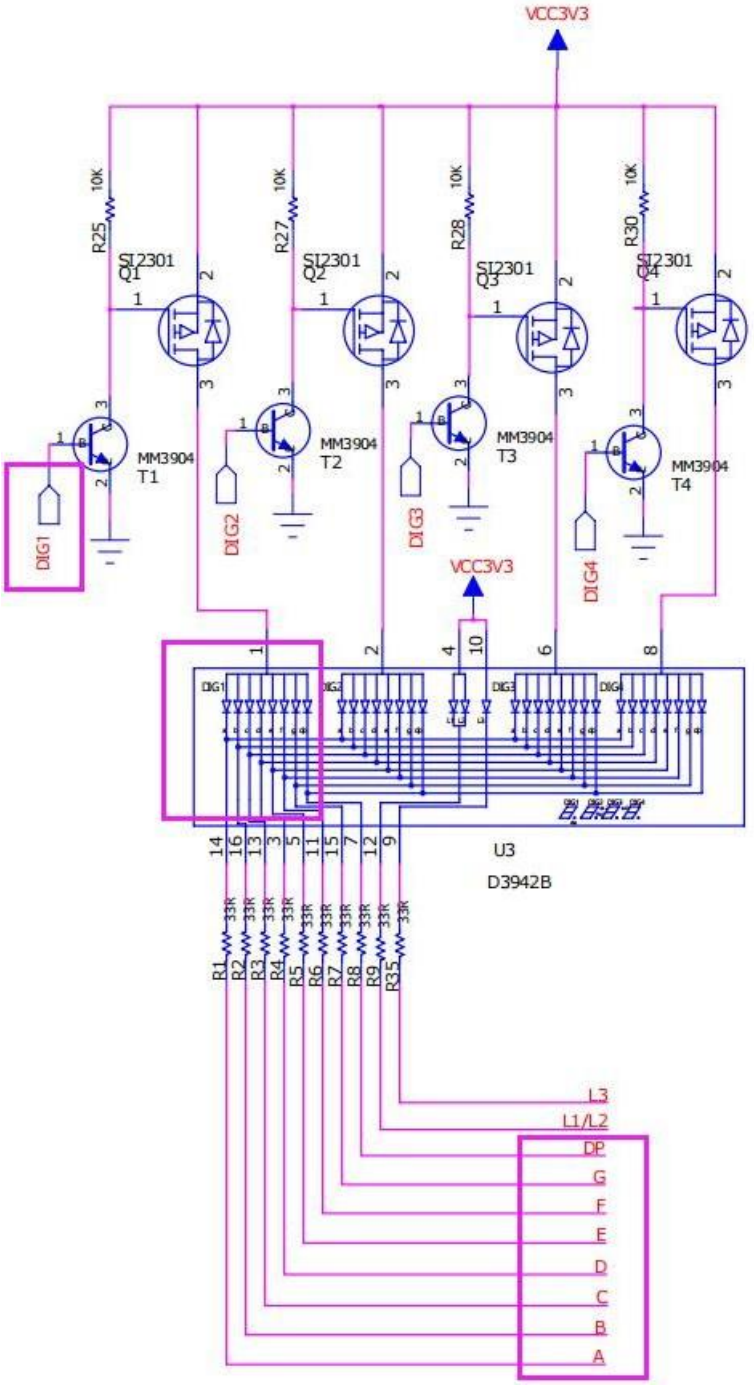

Figure 2 Digital Tube Schematic

In Figure 2, it is a common anode 8-segment digital tube, and the common terminal DIG1, DIG2, DIG3, and DIG4 control four digital tubes respectively, active-high. The other terminal is controlled by A, B, C, D, E, F, G, and DP, active-low.

The relationship between the field and code of the common anode 7-segment digital tube is shown in Figure 3.

Number	dp ,g,f,e,d,c,b, a	Character Code
0	1 1 0 0 0 0 0 0	C0H
1	1 1 1 1 1 0 0 1	F9H
2	1 0 1 0 0 1 0 0	A4H
3	1 0 1 1 0 0 0 0	B0H
4	1 0 0 1 1 0 0 1	99H
5	1 0 0 1 0 0 1 0	92H
6	1 0 0 0 0 0 1 0	82H
7	1 1 1 1 1 0 0 0	F8H
8	1 0 0 0 0 0 0 0	80H

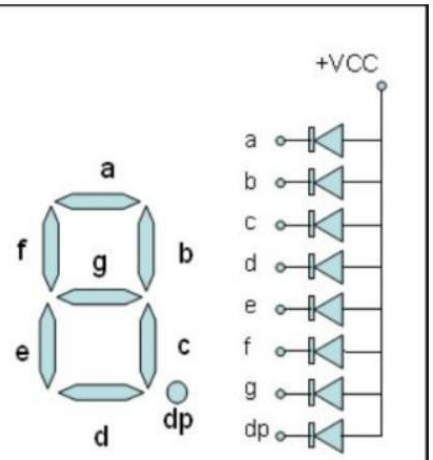

Figure 3 Code Table

The control signals of the digital tube are connected to the FPGA pins, after programming, you can observe a digital tube displaying the numbers 0~9. The pin connection is shown in Figure 4 and Table 1, and the digital tube and FPGA connection is as shown in Figure 5.

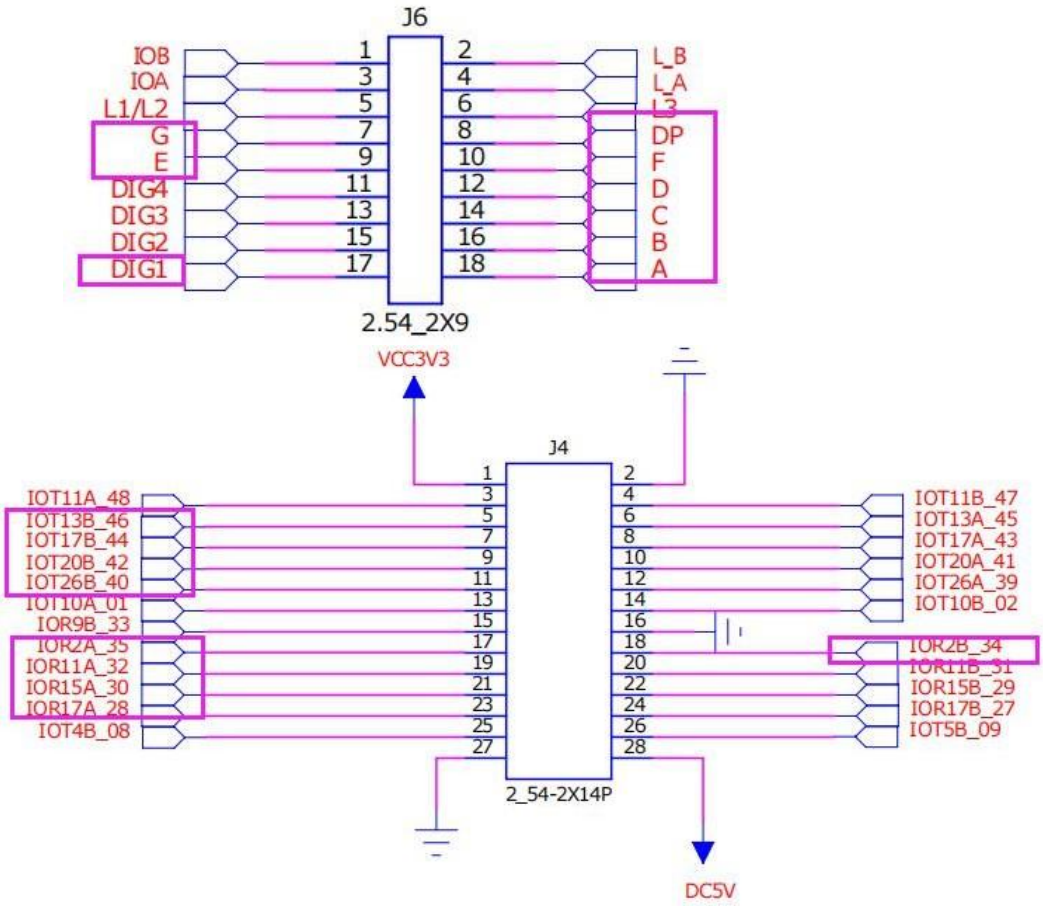

Figure 4 Pin Connection

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 1 Pin Definition

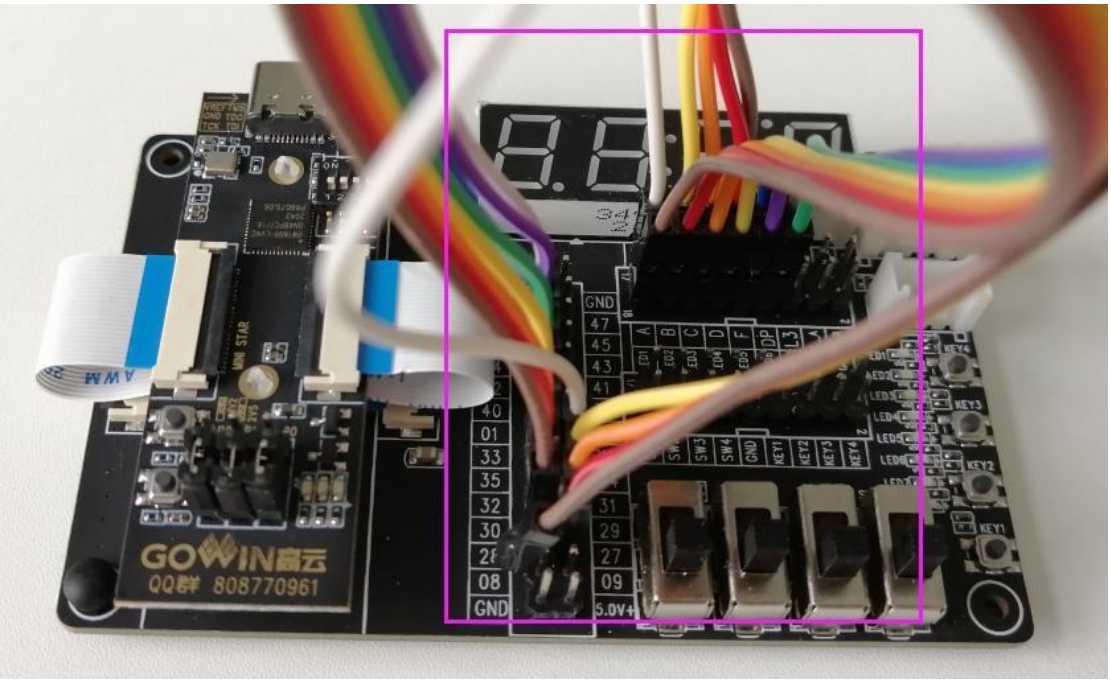

Figure 5 Connection of Digital Tube and FPGA

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 led 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 configuration is shown in Figure 6.

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		drag or ty...		False	LVCMOS18
5	gpio_io[13]	inout		drag or ty...		False	LVCMOS18
6	gpio_io[14]	inout		drag or ty...		False	LVCMOS18
7	gpio_io[15]	inout		drag or ty...		False	LVCMOS18
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 6 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 led project. Open Led.uvprojx in the Keil_led\PROJECT folder.

1. Set GPIO register; set GPIO[15: 0] as the output.

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

2. Set the GPIO->DATAOUT output data register, and the digital tube can display numbers.

According to the coding in Figure 3 and the pin definition in Table 1, the A to DP of the digital tube correspond to GPIO[7:0], and the DIG1 to DIG4 correspond to GPIO[11:8], so the upper 8 bits need to be set to high level, and the lower 8 bits correspond to the coding in Figure 3. For example, the number 0 corresponds to the code 0xc0, and the upper bit is 0xff, which corresponds to the code 0xffc0, so that four digital tubes can display the number 0 at the same time. The reference code is shown in Figure 7.

```
GPIO0->DATAOUT = 0xf8c0;      //0
Delay(833300);
Delay(833300);


GPIO0->DATAOUT = 0xf4f9;      //1        GPIO0->DATAOUT = 0xf492;     //5
Delay(833300);                         Delay(833300);
Delay(833300);                         Delay(833300);

GPIO0->DATAOUT = 0xf2a4;      //2        GPIO0->DATAOUT = 0xf282;     //6
Delay(833300);                         Delay(833300);
Delay(833300);                         Delay(833300);

GPIO0->DATAOUT = 0xf1b0;     //3         GPIO0->DATAOUT = 0xf1f8;     //7
Delay(833300);                         Delay(833300);
Delay(833300);                         Delay(833300);

GPIO0->DATAOUT = 0xf899;     //4         GPIO0->DATAOUT = 0xf880;     //8
Delay(833300);                         Delay(833300);
Delay(833300);                         Delay(833300);

GPIO0->DATAOUT = 0xf492;     //5         GPIO0->DATAOUT = 0xf490;     //9
Delay(833300);                         Delay(833300);
Delay(833300);                         Delay(833300);
```

Figure 7 Digital Tubes Display Numbers 0-9 in Turn

3. Use the array to code the digital tube, and it is also possible to display the numbers 0-9. The reference code is shown in Figure 8.

```
unsigned int segcode[]={0xffc0,0xfff9,0xffa4,0xffb0,     //0
                        0xfff9,0xfff9,0xffa4,0xffb0,     //1
                        0xffa4,0xff90,0xff80,0xff90,     //2
                        0xffb0,0xff90,0xff80,0xff90,     //3
                        0xff99,0xff92,0xff82,0xfff8,     //4
                        0xff92,0xff92,0xff82,0xfff8,     //5
                        0xff82,0xff90,0xff80,0xff90,     //6
                        0xfff8,0xff90,0xff80,0xff90,     //7
                        0xff80,0xff90,0xff80,0xff90,     //8
                        0xff90,0xff80,0xff90,0xff90      //9
                       };

                    for(i=0;i<40;i++)
                   {
                    s= segcode[i] | 0xff00;
                    GPIO0->DATAOUT = s;
                    Delay(833300);
                    //Delay(833300);
                   }
```

Figure 8 Reference Code

Note!

ARM core is a 32-bit processor, and different types of data correspond to different types of storage; ARM system in non-aligned storage access will generate data acquisition issues, and you will learn the coding and address alignment issue in this case.

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

Download and Verification

Use Gowin Software to download, and the running is as shown in Figure 9. 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.

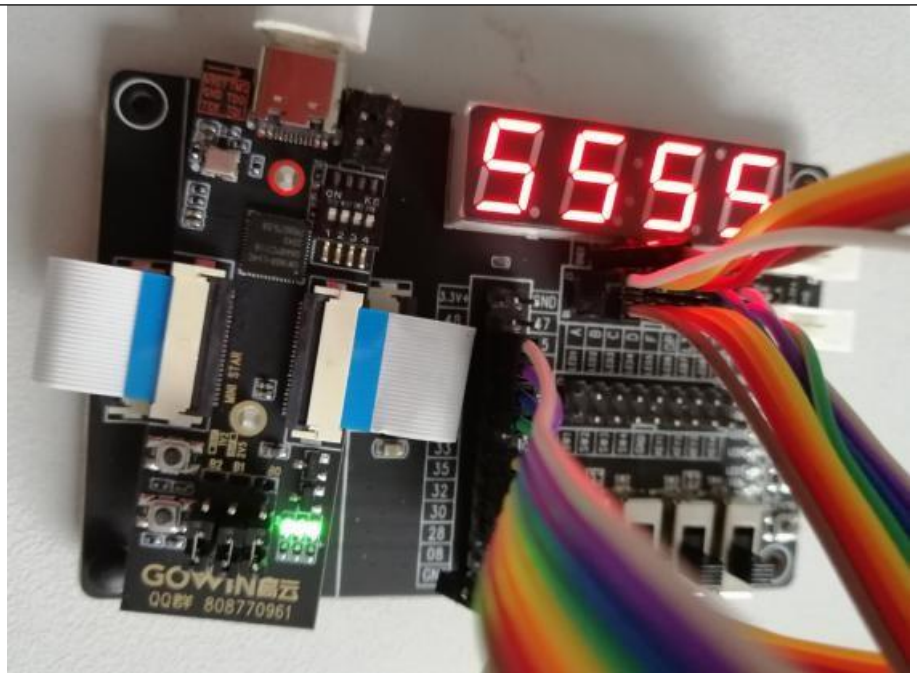

Figure 9 Demo Running