

FPGA 黑金开发平台

用户手册

AX309

ALINX

文档版本控制

文档版本	修改内容记录
REV1.0	创建文档
REV1.1	第 30 页, J2 扩展口表格, 29 改为 R1, 30 改为 R2
REV1.2	第 6 页, 第 11 页, 64Mbit 改为 16Mbit
REV1.3	第 21 页, LED 电平部分描述修正

版权申明：

本手册版权归属芯驿电子公司(上海)有限公司所有, 并保留一切权力。非经本公司(书面形式), 同意, 任何单位及个人不得擅自摘录或修改本手册部分或全部内容, 违者我们将追究其法律责任。

感谢您购买黑金开发板, 在使用产品之前, 请仔细地阅读该手册并且确保知道如何正确使用该产品, 不合理的操作可能会损坏开发板, 使用过程中随时参考该手册以确保正确使用。

此手册不断更新中, 建议您使用时下载最新版本。

软件版本：

本手册使用 Xilinx 官方提供的 ISE 14.7 版本, 请从 Xilinx 网站下载正确的版本和使用我们提供的软件版本安装。

黑金官网：

[Http://www.alinx.com.cn](http://www.alinx.com.cn)

黑金动力社区:

<http://www.heijin.org>

黑金官方淘宝店：

<http://oshcn.taobao.com>

联系方式：

021-67676997

黑金微信公众号：

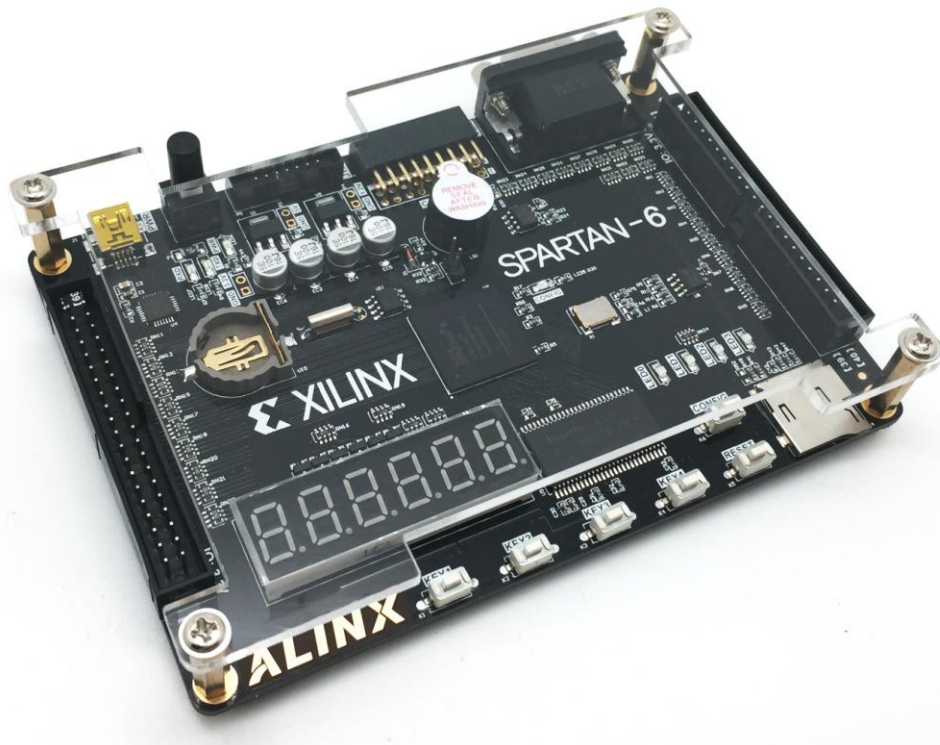
ALINX-HEIJIN



目 录

一、	简介.....	5
二、	电源.....	7
三、	FPGA.....	8
四、	50M 有源晶振.....	10
五、	QSPI Flash.....	11
六、	SDRAM.....	12
七、	EEPROM 24LC04.....	15
八、	实时时钟 DS1302.....	16
九、	USB 转串口.....	17
十、	VGA 接口.....	18
十一、	SD 卡槽.....	20
十二、	LED.....	21
十三、	按键.....	22
十四、	摄像头接口.....	24
十五、	数码管.....	25
十六、	蜂鸣器.....	27
十七、	扩展口.....	28

FPGA 黑金开发平台又迎来了一位小兄弟 AX309，这款开发板是隶属入门级产品，主要针对 FPGA 初学者。AX309 使用的是 XILINX 公司的 SPARTAN6 系列芯片，型号为 XC6SLX9-2FTG256C，是 256 脚的 FBGA 封装。整个开发板的配置实用，有两路黑金标准的 AX 扩展口，一共有 $34 \times 2 = 68$ 个 IO，另外也引出了 5V 电源，3.3V 电源，还有多路 GND，对于喜欢 DIY 的玩家来说，是个非常不错选择。另外，黑金众多的配套模块也可以直接连接到这款 FPGA 开发板的扩展口上，比如 ADDA 模块，4.3 寸液晶屏，音频模块，摄像头等等，为玩家学习提供了更多的选择。下面我们就对 AX309 做一个详细的介绍。



一、简介

在这里，对这款 FPGA 开发平台进行简单的功能介绍。

此款开发板使用的是 Xilinx 公司的 Spartan6 系列 FPGA，型号为 XC6SLX9，256 个引脚的 FBGA 封装。此款 FPGA 的资源如下图所示：

Device	Logic Cells ⁽¹⁾	Configurable Logic Blocks (CLBs)			DSP48A1 Slices ⁽³⁾	Block RAM Blocks		CMTs ⁽⁵⁾	Memory Controller Blocks (Max) ⁽⁶⁾	Endpoint Blocks for PCI Express	Maximum GTP Transceivers	Total I/O Banks	Max User I/O
		Slices ⁽²⁾	Flip-Flops	Max Distributed RAM (Kb)		18 Kb ⁽⁴⁾	Max (Kb)						
XC6SLX4	3,840	600	4,800	75	8	12	216	2	0	0	0	4	132
XC6SLX9	9,152	1,430	11,440	90	16	32	576	2	2	0	0	4	200
XC6SLX16	14,579	2,278	18,224	136	32	32	576	2	2	0	0	4	232
XC6SLX25	24,051	3,758	30,064	229	38	52	936	2	2	0	0	4	266
XC6SLX45	43,661	6,822	54,576	401	58	116	2,088	4	2	0	0	4	358

其中，主要的参数，

参数	数值
逻辑单元 Logic Cells	9152
乘法器 DSP48	16
可配置逻辑块 CLBs	90Kb
Block RAM	576Kb
时钟单元 CMTs	2
可用 IO 数量	200
内核电压	1.15V-1.25V(推荐 1.2V);
工作温度	0-85°C

图为整个系统的结构示意图 1.1 所示：

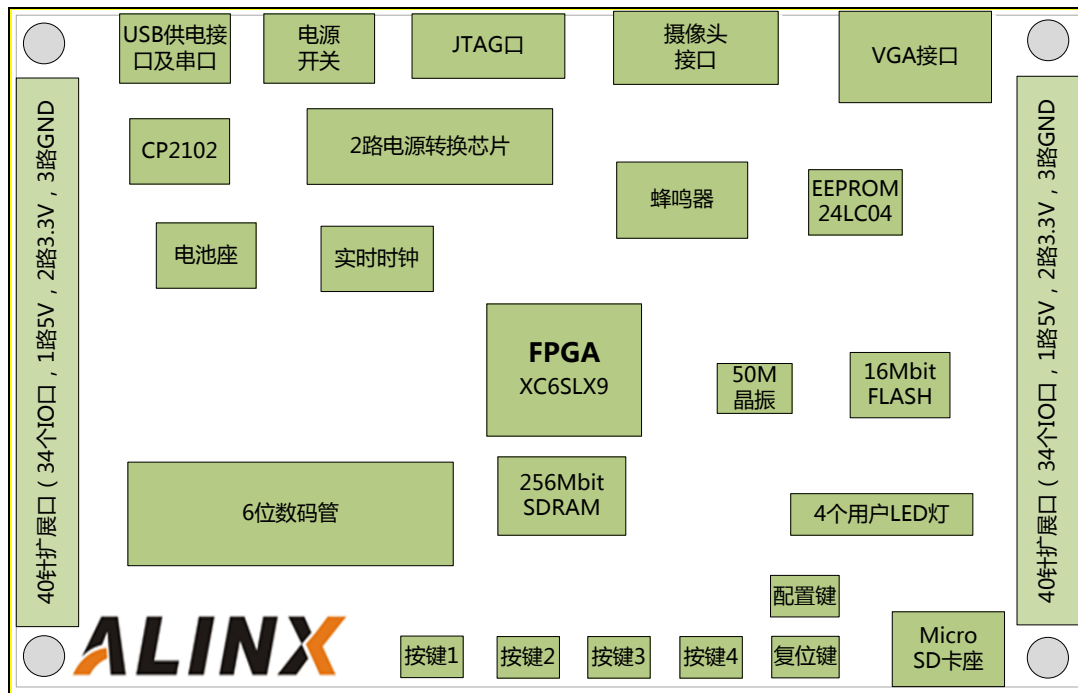


图 1.1 开发板系统结构图

通过这个示意图，我们可以看到，我们这个开发平台所能实现的功能。

- USB 接口供电，同时实现 USB 转串口功能；
- 一片大容量的 256Mbit SDRAM,可作为数据的缓存;
- 一片 16Mbit 的 SPI FLASH, 可用作 FPGA 配置文件和用户数据的存储;
- 一个摄像头接口，可以接 500 万的 OV5640 摄像头；
- 一路 VGA 接口，VGA 接口为 16bit，可以显示 65536 种颜色，可以显示彩色图片等信息。

- 一片的 RTC 实时时钟，配有电池座，电池的型号为 CR1220。
- 一片 IIC 接口的 EEPROM 24LC04;
- 4 个红色 LED，可实现流水灯功能;
- 4 个独立用户按键；
- 板载 50M 的有源晶振，给开发板提供稳定的时钟源；
- 2 路 40 针的黑金标准 AX 扩展口 (2.54mm 间距)，其中 34 个 IO 口，1 路 5V 电源，2 路 3.3V 电源，3 路 GND。可同时接两个扩展模块，例如 4.3 寸 TFT 模块和 AD/DA 模块等扩展模块。
- 预留了 JTAG 口，可对 FPGA 进行调试和程序固化。
- 1 路 Micro SD 卡座，支持 SPI 模式。
- 1 个 6 位数码管，可以 6 位数字的动态显示。

二、电源

AX309 开发板通过 USB 供电，用 MINI USB 线将开发板跟电脑的 USB 连接，按键电源开关，既可以给开发板供电。开发板上的电源设计示意图如下：

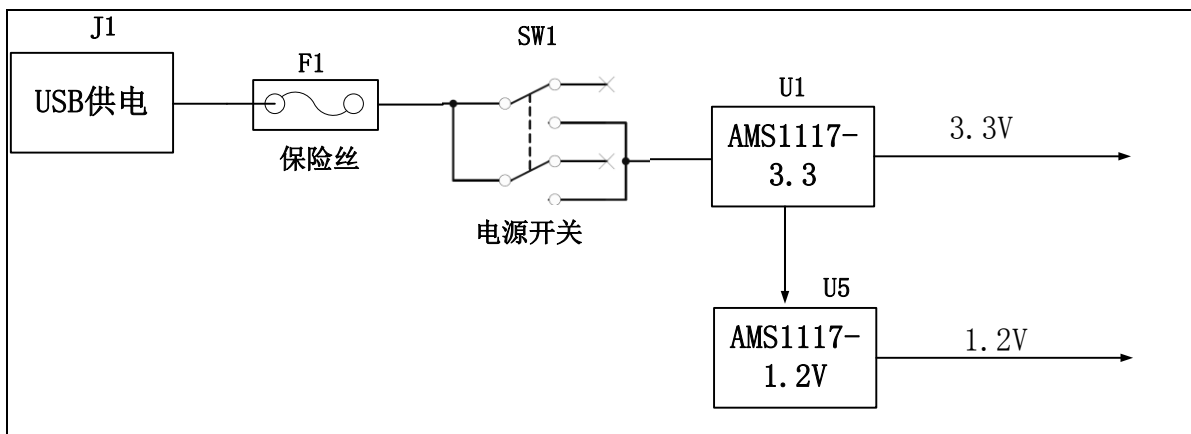


图 2.1 原理图中电源接口部分

开发板用 USB 供电，通过两路 LDO 电源芯片产生 +3.3V，+1.2V 两路电源，满足 FPGA 的 BANK 电压和内核电压。

我们在设计 PCB 的时候，采用 4 层 PCB，预留了独立 GND 层，使得整个开发板具有完整的地平面，保证了开发板具有非常好的稳定性。在 PCB 板上我们预留了各个电源的测试点，以使用户确认板上的电压。

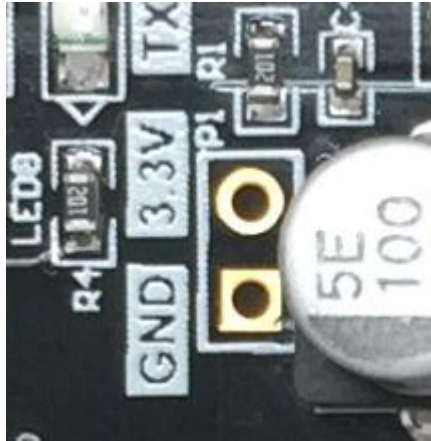


图 2.2 实物图中的电源测试点

三、FPGA

前面已经介绍过了，我们所使用的 FPGA 型号为 XC6SLX9-2FTG256C，属于 Xilinx 公司 Spartan-6 的产品。此型号为 BGA 封装，256 个引脚。再次说明一下 FPGA 引脚的定义。很多人使用 FPGA 都是非 BGA 封装的，比如 144 引脚，208 引脚的 FPGA 芯片，他们的引脚定义是由数字组成，比如 1 到 144，1 到 208 等等，而当我们使用 BGA 封装的芯片以后，引脚名称变为由**字母+数字**的形式，比如 E3，G3 等等，因此我们在看原理图的时候，看到的**字母+数字**这种形式的，就是代表了 FPGA 的引脚。说完这个，我们来看与 FPGA 有关系的各个部分的功能。图 3.1 为开发板所用的 FPGA 芯片实物图。

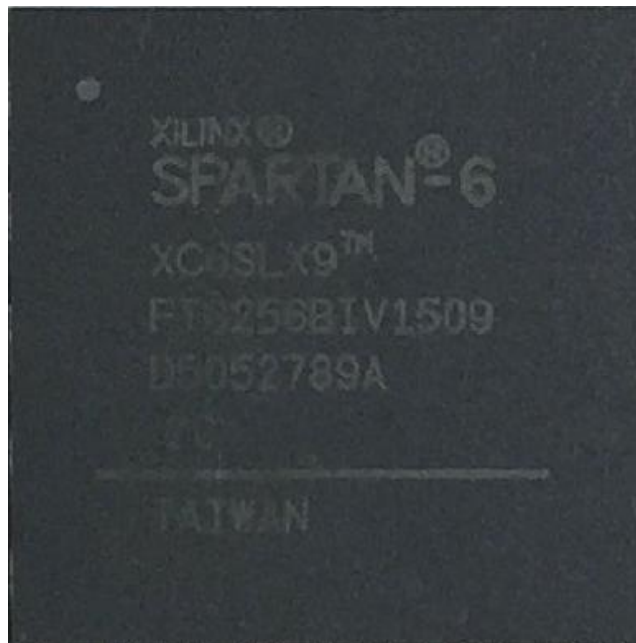


图 3.1 FPGA 芯片实物

1) JTAG 接口

首先我们来说 FPGA 的配置和调试接口：JTAG 接口。JTAG 接口的作用是将编译好的程

序 .bit)下载到 FPGA 中或把 FLASH 配置程序(.mcs)下载到 SPI FLASH ,Bit 文件下载到 FPGA 后,掉电以后就会丢失,需要上电重新下载才可以。但下载到 FLASH 的 MCS 文件以后,掉电以后就不会丢失,重新上电后 FPGA 会读取 FLASH 中的配置文件并运行。

图 3.2 就是 JTAG 口的原理图部分,其中涉及到 TCK,TDO,TMS,TDI 这四个信号。这四个信号由 FPGA 引脚引出通过 33 欧姆电阻连接到 JTAG 连接器,起到对 FPGA 芯片的保护作用。

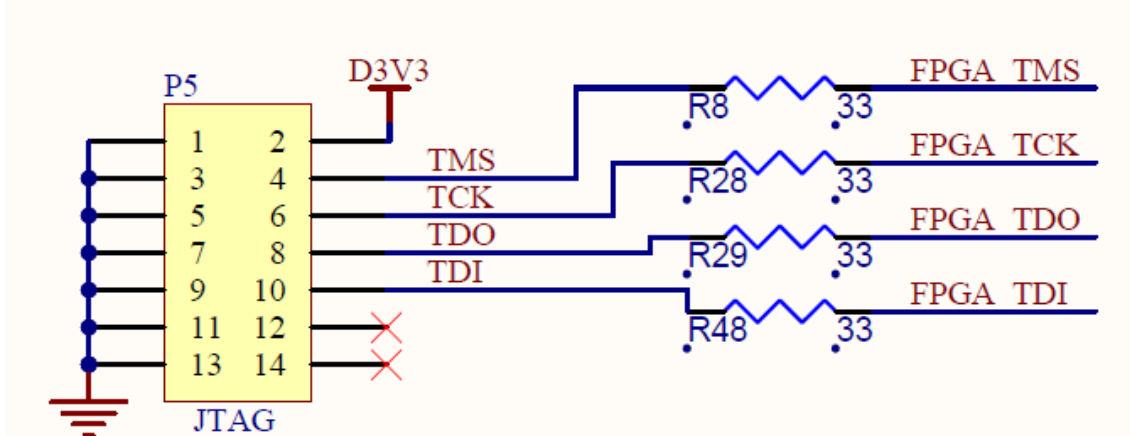


图 3.2 原理图中 JTAG 接口部分

JTAG 接口采用 14 针的 2.0mm 标准的连接器,图 3.3 为 JTAG 接口在开发板上的实物图



图 3.3 JTAG 接口实物图

2) FPGA 电源和 GND 引脚

接下来,我们说一下 FPGA 的电源引脚部分,其中包括每一个 bank 的电源引脚,内核电压引脚和辅助电压引脚,VCCINT 为 FPGA 内核供电引脚,接 1.2V;VCCAUX 为 FPGA 辅助供电引脚,可以接 3.3V 或 2.5V,我们这里接 3.3V;VCCIO 是 FPGA 的每个 BANK 的供电电压,其中 VCCIO0 是 FPGA 的 BANK0 的供电引脚,同理,VCCIO1~ VCCIO3 分别是 FPGA 的 BANK~BANK3 的供电引脚,在开发板中,VCCIO 都接了 3.3V 电压,也就是说,这款开

发板 FPGA 引脚均为 3.3V 输入和输出。连接图如图 3.4 所示。

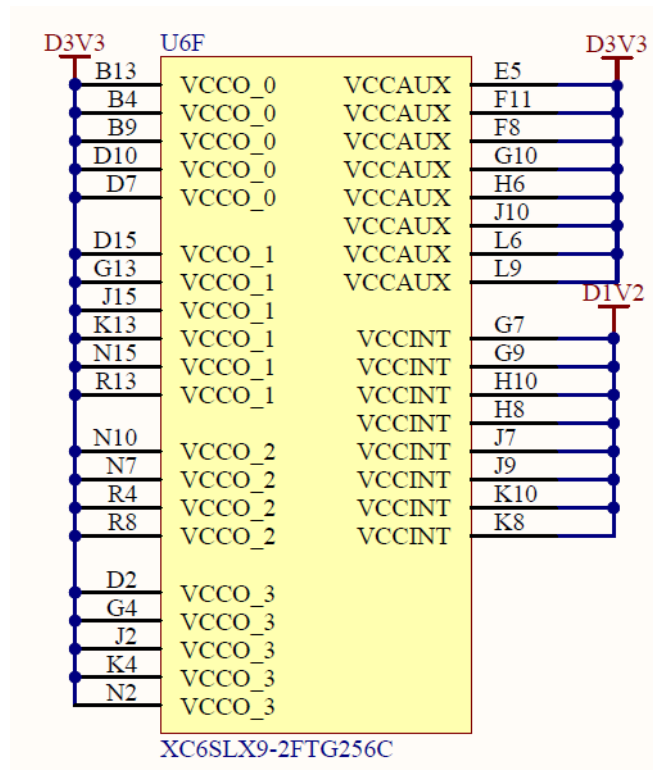


图 3.4 FPGA 电源引脚

另外，FPGA 还有很多引脚需要连接 GND，保证 FPGA 内部有一个平稳的参考地。FPGA 连接的 GND，如图 3.5 所示。

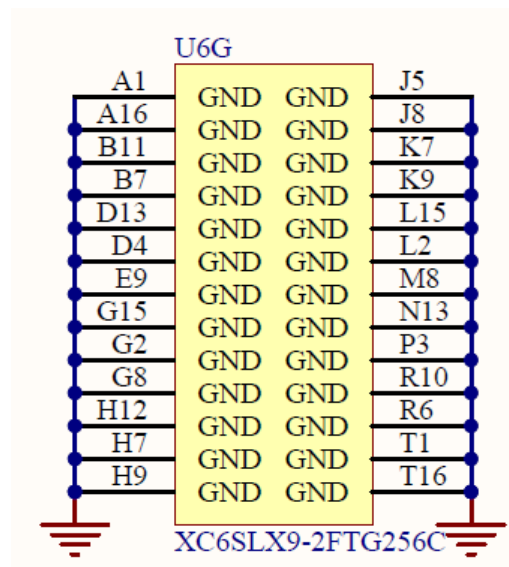


图 3.5 FPGA GND 引脚

四、50M 有源晶振

图 4.1 即为我们上述提到的给开发板提供时钟源的 **50M** 有源晶振电路。晶振输出连接到 FPGA 的全局时钟(GCLK Pin T8)，这个 GCLK 可以用来驱动 FPGA 内的用户逻辑电路，用户

可以通过配置 FPGA 内部的 PLLs 和 DCMs 来实现更高的时钟。

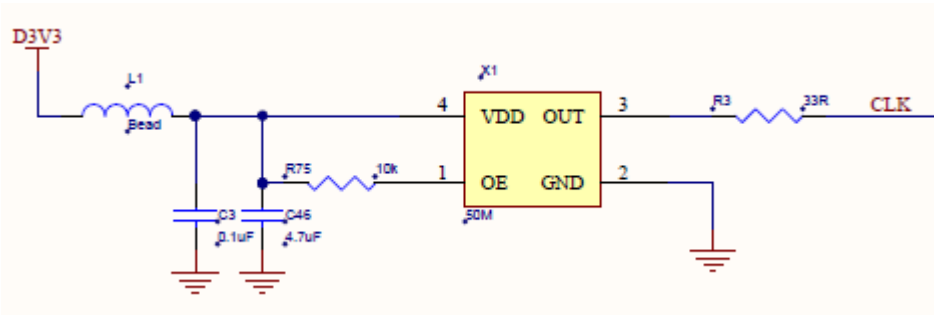


图 4.1 50M 有源晶振

图 4.2 为有源晶振实物图



图 4.2 50M 有源晶振实物图

时钟引脚分配：

引脚名称	FPGA 引脚
CLK	T8

五、QSPI Flash

开发板上使用了一片 16Mbit 大小的 SPI FLASH 芯片，型号为 M25P16，它使用 3.3V CMOS 电压标准。由于它的非易失特性，在使用中，SPI FLASH 可以作为 FPGA 系统的启动镜像。这些镜像主要包括 FPGA 的 bit 文件、软核的应用程序代码以及其它的用户数据文件。

SPI FLASH 的具体型号和相关参数见表 5.1。

位号	芯片类型	容量	厂家
U8	M25P16	16M bit	ST

表 5.1 SPI Flash 的型号和参数

SPI Flash 原理图如图 5.2 所示，

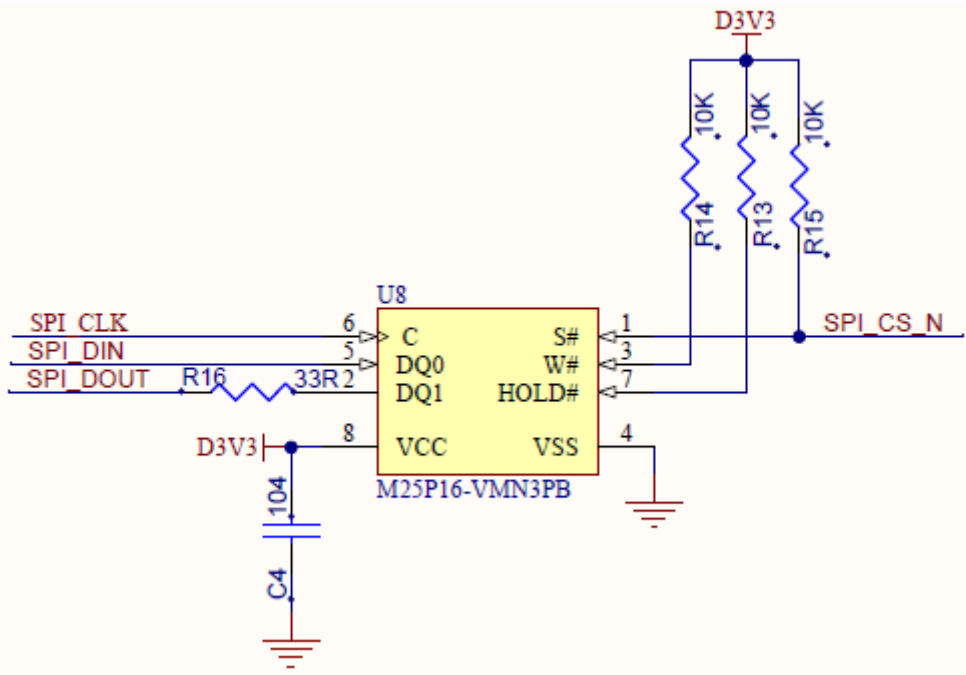


图 5.2 SPI Flash 连接示意图

SPI Flash 的硬件实物图，如图 5.3 所示

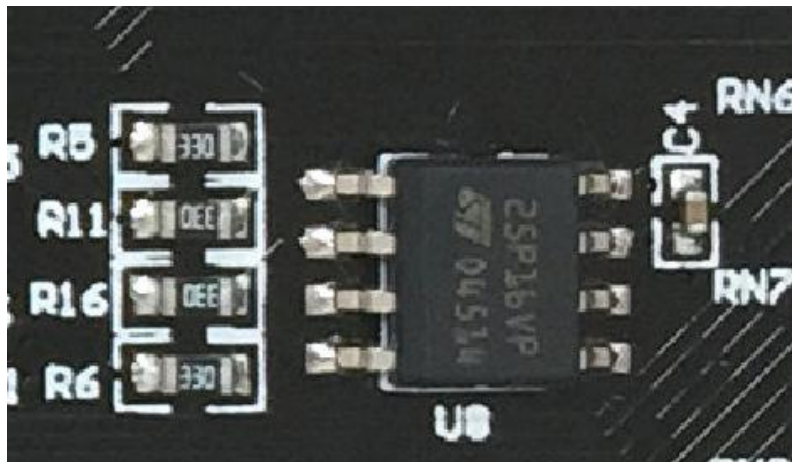


图 5.3 SPI Flash 实物图

配置芯片引脚分配：

引脚名称	FPGA 引脚
SPI_CLK	R11
SPI_CS_N	T3
SPI_DIN	T10
SPI_DOUT	P10

六、SDRAM

开发板板载了一片 SDRAM 芯片,型号 :HY57V2562GTR ,容量 :256Mbit(16M*16bit),

16bit 总线。SDRAM 可用于数据缓存，比如摄像头采集到的数据，暂存到 SDRAM 中，然后通过 VGA 接口进行显示。这里面 SDRAM 就是用于数据缓存的。

SDRAM 的硬件连接方式如图 6.1 所示

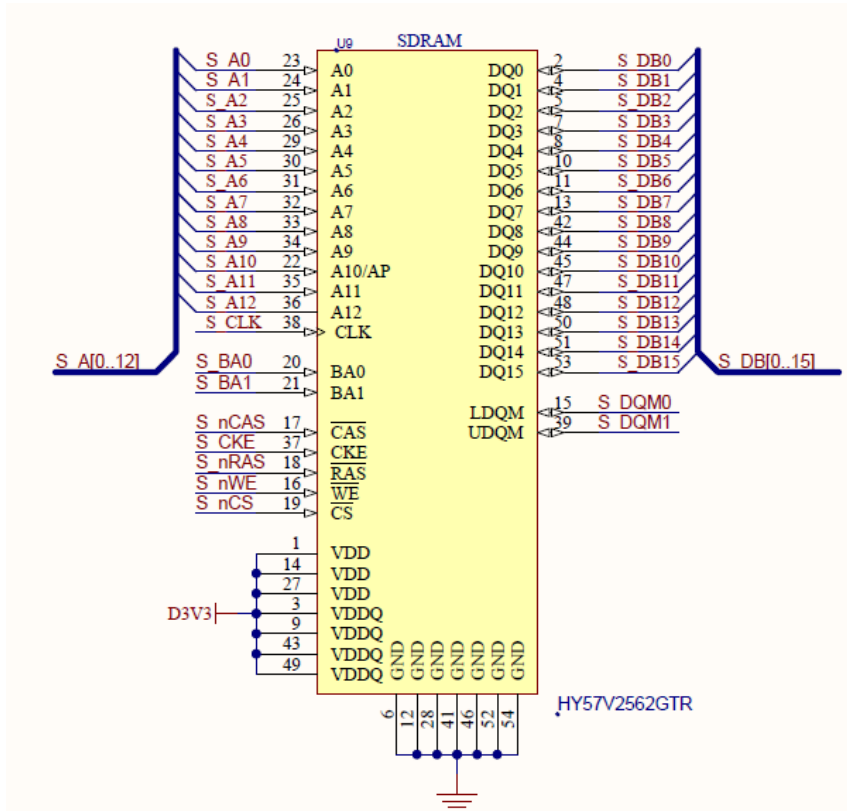


图 6.1 SDRAM 原理图部分

图 6.2 为 SDRAM 实物图



图 6.2 SDRAM 实物图

SDRAM 引脚分配：

引脚名称	FPGA 引脚
S_CLK	H4
S_CKE	H2
S_NCS	G1
S_NWE	E1
S_NCAS	F2
S_NRAS	F1
S_DQM<0>	E2
S_DQM<1>	H1
S_BA<0>	G6
S_BA<1>	J6
S_A<0>	J3
S_A<1>	J4
S_A<2>	K3
S_A<3>	K5
S_A<4>	P1
S_A<5>	N1
S_A<6>	M2
S_A<7>	M1
S_A<8>	L1
S_A<9>	K2
S_A<10>	K6
S_A<11>	K1
S_A<12>	J1
S_DB<0>	A3
S_DB<1>	B3
S_DB<2>	A2
S_DB<3>	B2
S_DB<4>	B1
S_DB<5>	C2
S_DB<6>	C1

S_DB<7>	D1
S_DB<8>	H5
S_DB<9>	G5
S_DB<10>	H3
S_DB<11>	F6
S_DB<12>	G3
S_DB<13>	F5
S_DB<14>	F3
S_DB<15>	F4

七、EEPROM 24LC04

开发板板载了一片 EEPROM，型号为 24LC04，容量为：4Kbit (2*256*8bit)，由 2 个 256byte 的 block 组成，通过 IIC 总线进行通信。板载 EEPROM 就是为了学习 IIC 总线的通信方式。EEPROM 一般用在仪器仪表等设计上，用作一些参数的存储，掉电不丢失。这种芯片操作简单，具有极高的性价比，所以虽然容量比高，但价格非常便宜，对于那些对成本要求很高的产品来说，是个不错的选择。图 7.1 为 EEPROM 的原理图

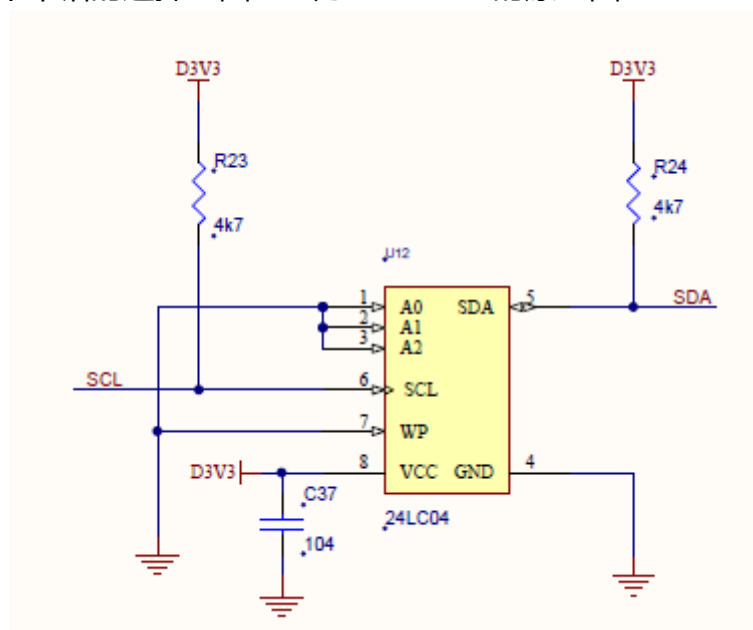


图 7.1 EEPROM 原理图部分

图 7.2 为 EEPROM 实物图

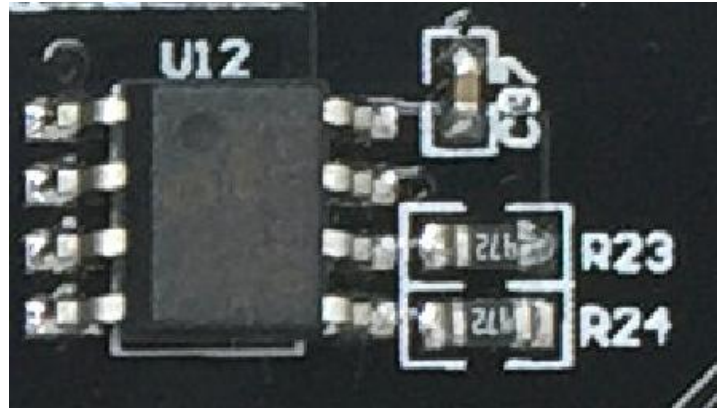


图 7.2EEPROM 实物图

EEPROM 引脚分配：

引脚名称	FPGA 引脚
SDA	P12
SCL	N12

八、实时时钟 DS1302

开发板板载了一片实时时钟 RTC 芯片，型号 DS1302，他的功能是提供到 2099 年内的日历功能，年月日时分秒还有星期。如果系统中需要时间的话，那么 RTC 就需要涉及到产品中。他外部需要接一个 32.768KHz 的无源时钟，提供精确的时钟源给时钟芯片，这样才能让 RTC 可以准确的提供时钟信息给产品。同时为了产品掉电以后，实时时钟还可以正常运行，一般需要另外配一个电池给时钟芯片供电 图 8.1 中为 U7 为电池座 我们将纽扣电池 (型号 CR1220，电压为 3V) 放入以后，当系统掉电，纽扣电池还可以给 DS1302 供电，这样，不管产品是否供电，DS1302 都会正常运行，不会间断，可以提供持续不断的时间信息。图 8.1 为 DS1302 原理图

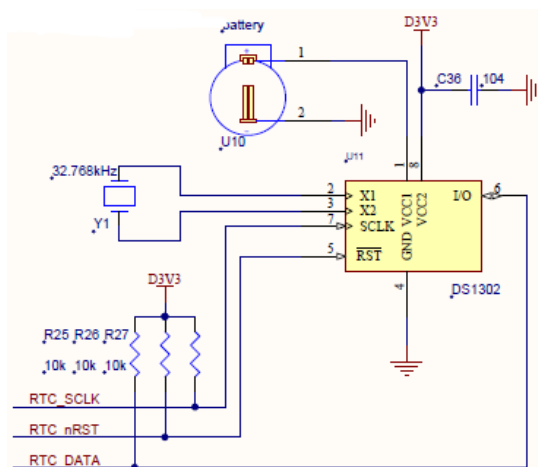


图 8.1 DS1302 原理图

图 8.2 为 DS1302 实物图

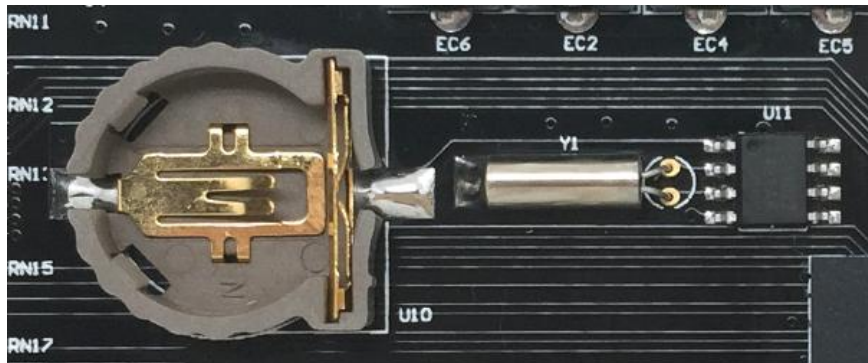


图 8.2 DS1302 实物图

DS1302 接口引脚分配：

引脚名称	FPGA 引脚
RTC_SCIK	E13
RTC_nRST	C13
RTC_DATA	D14

九、USB 转串口

开发板包含了Silicon Labs CP2102GM的USB-UAR芯片, USB接口采用MINI USB接口, 这个USB接口即实现了供电功能, 有可以实现USB转串口功能, 可以用一根USB线将它连接到上PC的USB口进行串口数据通信。

串口的原理图如图 9.1 所示

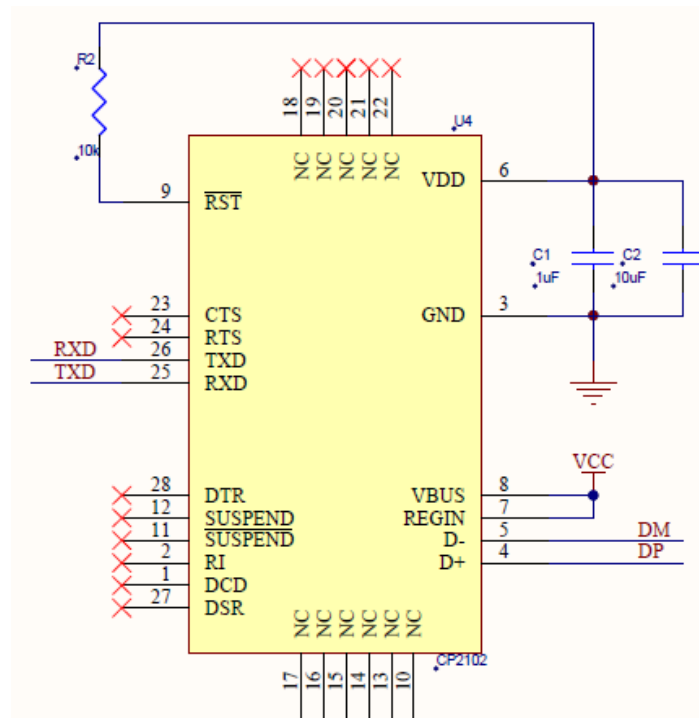


图 9.1 USB 转串口原理图

图 9.2 为 USB 转串口的实物图

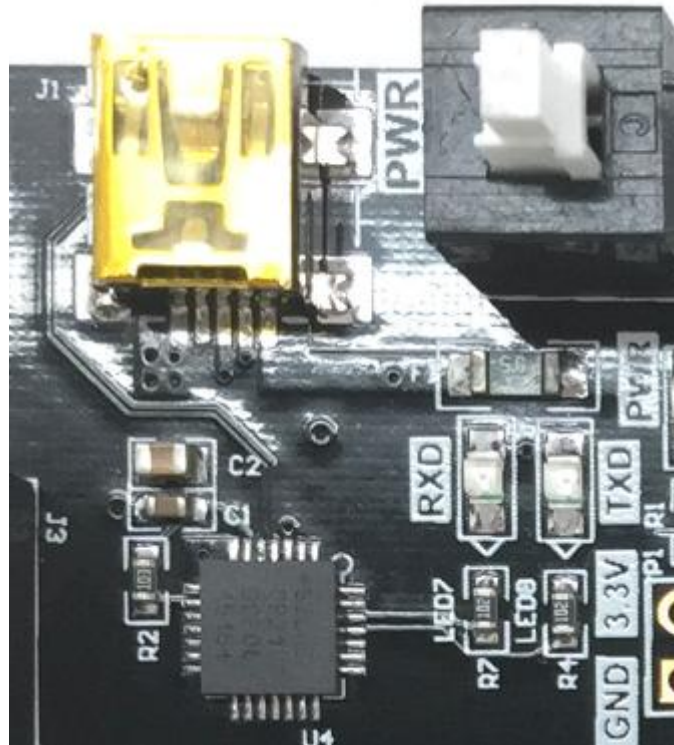


图 9.2 USB 转串口实物图

同时对串口信号设置了 2 个 led 指示灯(LED7,LED8)，LED7 和 LED8 会指示串口是否有数据发出或者是否有数据接受，如图 9.3 所示，

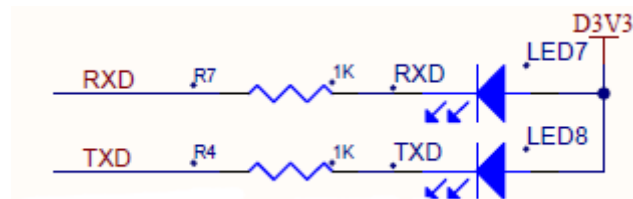


图 9.3 USB 转串口信号指示灯

串口引脚分配：

引脚名称	FPGA 引脚
RXD	C11
TXD	D12

十、VGA 接口

说到 VGA 接口，相信很多朋友都不会陌生，因为这种接口是电脑显示器上最主要的接口，从块头巨大的 CRT 显示器时代开始，VGA 接口就被使用，并且一直沿用至今，另外 VGA 接口还被称为 D-Sub 接口。

VGA 接口是一种 D 型接口，上面共有 15 针孔，分成三排，每排五个。比较重要的是 3 根 RGB 彩色分量信号和 2 根扫描同步信号 HSYNC 和 VSYNC 针。

引脚 1、2、3 分别为红绿蓝三基色模拟电压，为 0~0.714V peak-peak (峰-峰值)，0V 代表无色，0.714V 代表满色。一些非标准显示器使用的是 1Vpp 的满色电平。

三基色源端及终端匹配电阻均为 75 欧姆。如图 10.1

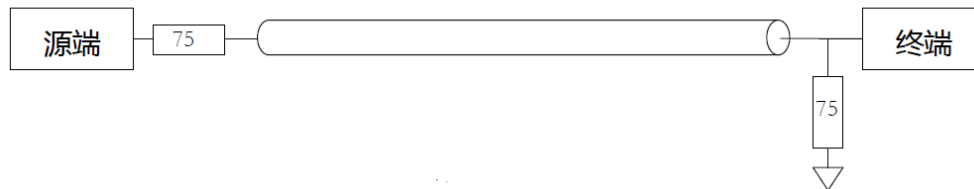


图 10.1 VGA 视频信号传输示意图

HSYNC 和 VSYNC 分别为行数据同步和帧数据同步，为 TTL 电平。FPGA 只能输出数字信号，而 VGA 需要的 R、G、B 是模拟信号，VGA 的数字转模拟信号是通过一个简单的电阻电路来实现。这个电阻电路可以产生 32 个梯度等级的红色和蓝色信号和 64 个梯度等级的绿色信号 (RGB 5-6-5)，VGA 接口部分电路如下图 10.2 所示

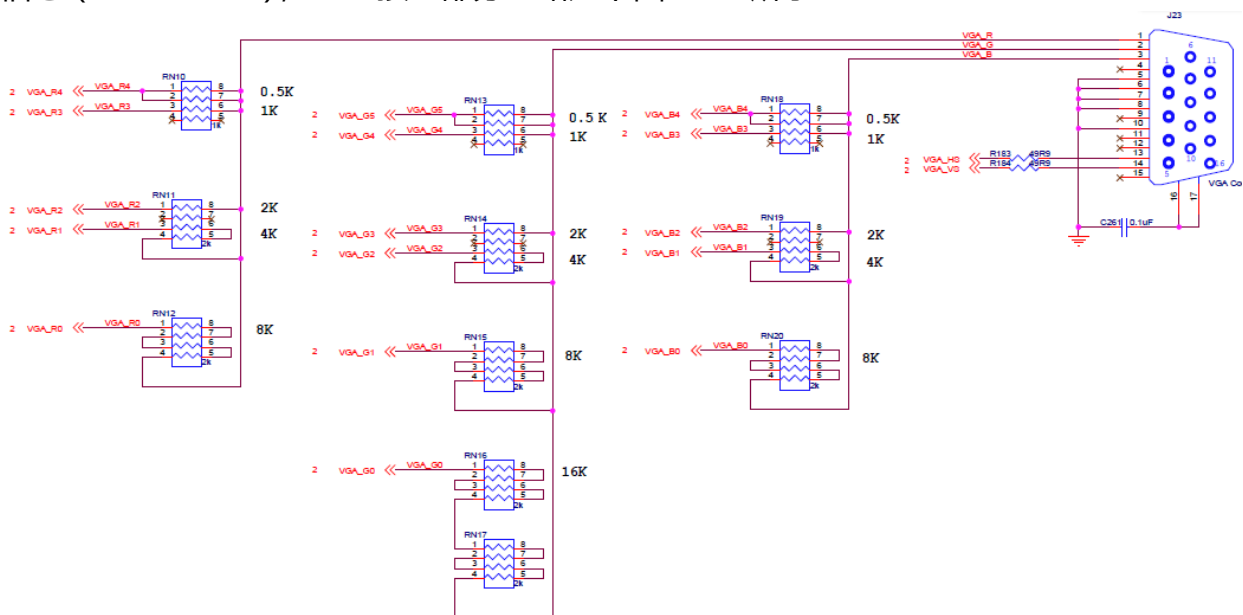


图 10.2 VGA 接口部分原理图

图 10.3 为 VGA 接口实物图



10.3 VGA 接口实物图

VGA 接口引脚分配

引脚名称	FPGA 引脚	备注
VGA_D<0>	P7	BLUE<0>
VGA_D<1>	M7	BLUE<1>
VGA_D<2>	P8	BLUE<2>
VGA_D<3>	N8	BLUE<3>
VGA_D<4>	L7	BLUE<4>
VGA_D<5>	M9	GREEN<0>
VGA_D<6>	N9	GREEN<1>
VGA_D<7>	P9	GREEN<2>
VGA_D<8>	L10	GREEN<3>
VGA_D<9>	M10	GREEN<4>
VGA_D<10>	P11	GREEN<5>
VGA_D<11>	M11	RED<0>
VGA_D<12>	M12	RED<1>
VGA_D<13>	L12	RED<2>
VGA_D<14>	N14	RED<3>
VGA_D<15>	M13	RED<4>
VGA_HS	M14	行同步信号
VGA_VS	L13	场同步信号

十一、SD 卡槽

SD 卡(Secure Digital Memory Card)是一种基于半导体闪存工艺的存储卡，1999 年由日本松下主导概念，参与者东芝和美国 SanDisk 公司进行实质研发而完成。2000 年这几家公司发起成立了 SD 协会(Secure Digital Association 简称 SDA)，阵容强大，吸引了大量厂商参加。其中包括 IBM，Microsoft，Motorola，NEC、Samsung 等。在这些领导厂商的推动下，SD 卡已成为目前消费数码设备中应用最广泛的一种存储卡。

SD 卡是现在非常常用的存储设备，我们扩展出来的 SD 卡，支持 SPI 模式，使用的 SD 卡为 MicroSD 卡。原理图如图 11.1 所示。

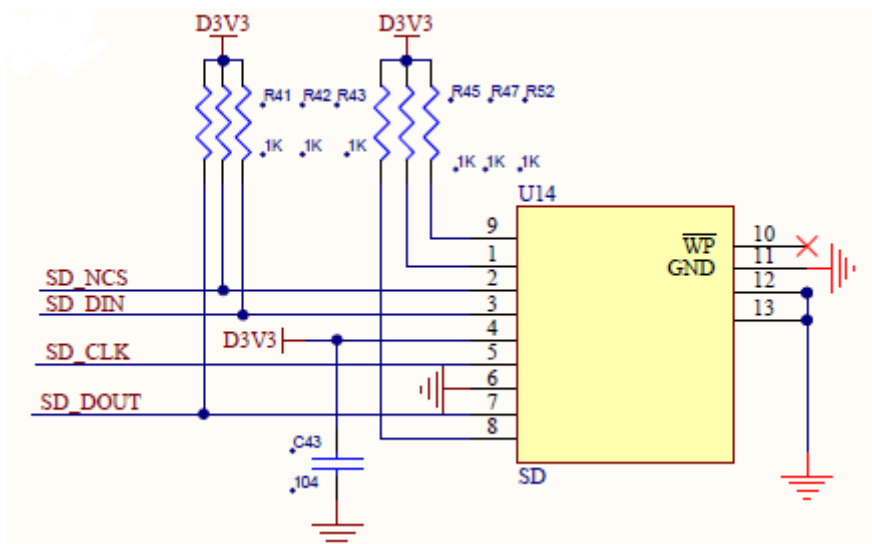


图 11.1 SD 卡槽原理图

图 11.2 SD 卡槽实物图

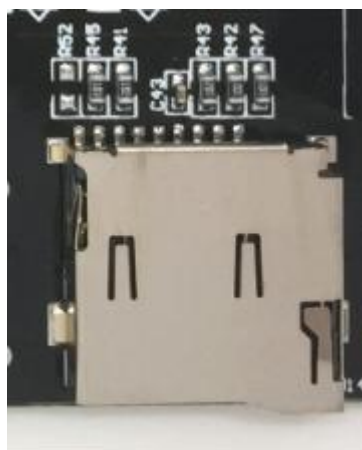


图 11.2 SD 卡槽实物图

SD 卡槽引脚分配

SD 模式	
引脚名称	FPGA 引脚
SD_NCS	N3
SD_DIN	L5
SD_CLK	M3
SD_DOUT	L4

十二、LED

开发板板载了 4 个用户 LED 发光二极管。4 个用户 LED 部分的原理图如图 12.1 当 FPGA 的引脚输出为逻辑 0 时，LED 会熄灭。输出为逻辑 1 时，LED 被点亮。

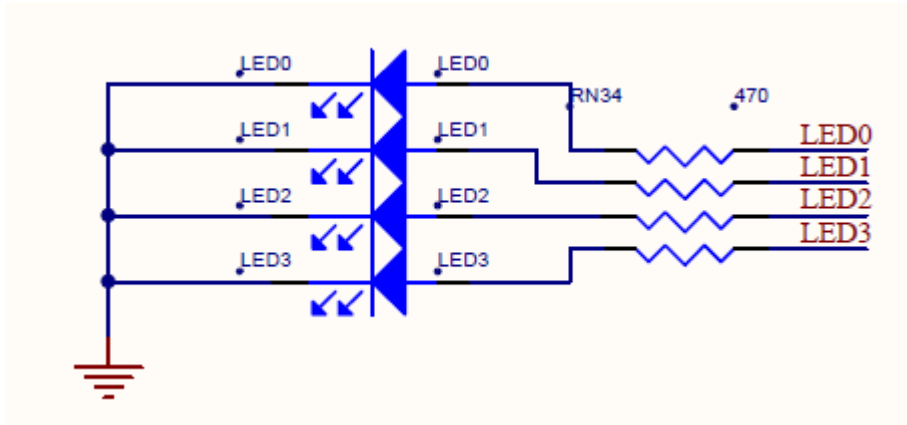


图 12.1 用户 LED 原理图

图 12.2 为 LED 实物图

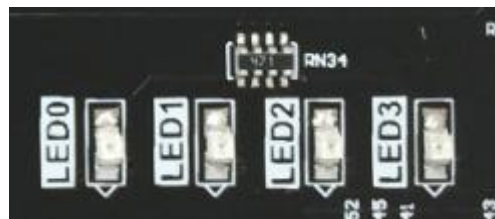


图 12.2 4 个 LED 实物图

LED 引脚分配：

引脚名称	FPGA 引脚
LED<0>	P4
LED<1>	N5
LED<2>	P5
LED<3>	M6

十三、 按键

开发板板载了 6 个独立按键 4 个用户按键(KEY1~KEY4), 2 个功能按键(PROG 和 RESET)。

按键都为低电平有效，4 个用户按键的原理图如图 13.1 所示

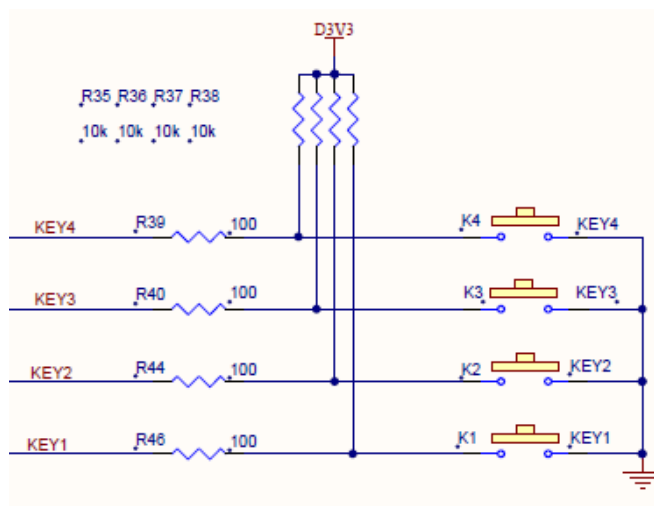


图 13.1 4 个用户按键原理图

2 个功能按键的原理图如图 13.2 所示, 其中 Reset 按键连接到 FPGA 的普通 IO 用于 FPGA 的程序复位 ,CONFIG 按键连接到 FPGA 的专用 FPGA_PROGRAM_B 引脚 ,用于 FPGA 程序的重新配置。

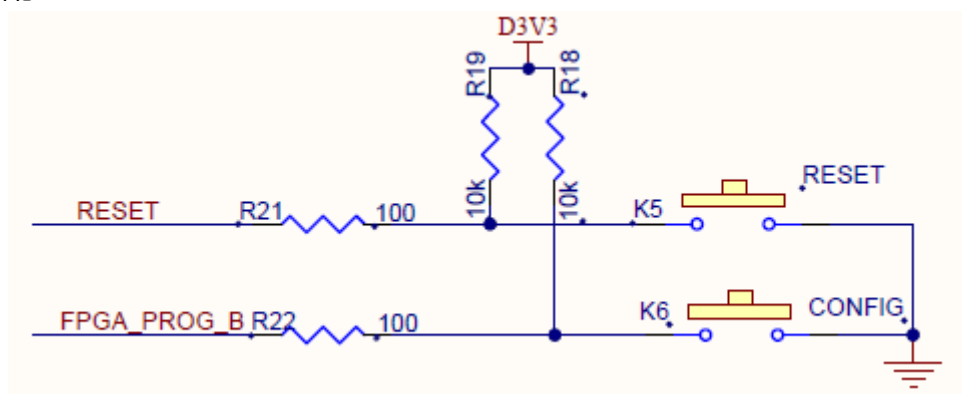


图 13.2 2 个功能按键原理图

图 13.3 为 6 个独立按键实物图

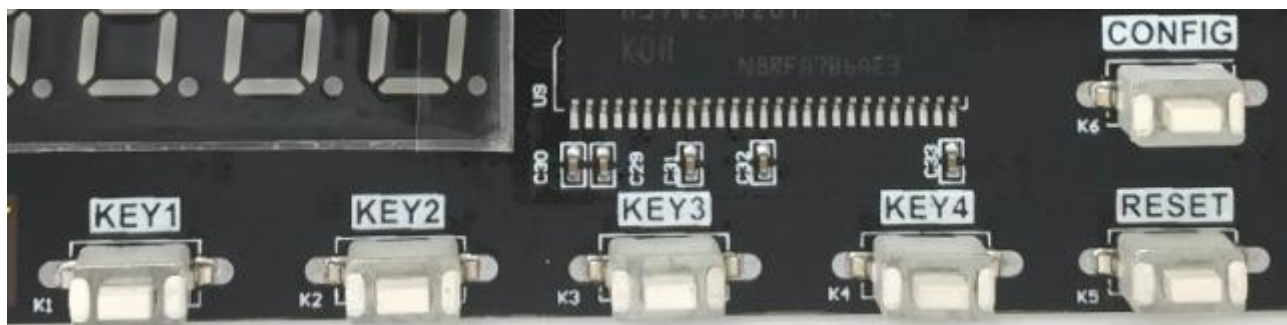


图 13.3 6 个独立按键实物图

按键引脚分配：

按键名称	FPGA 引脚	按键标号
KEY1	C3	KEY 1

KEY2	D3	KEY 2
KEY3	E4	KEY 3
KEY4	E3	KEY 4
RESET	L3	KEY6
PROG	T2	KEY5

十四、摄像头接口

开发板包含了一个 18 针的 CMOS 摄像头接口,可以连接 OV7670 摄像头模块和 OV5640 摄像头模块,可以实现视频采集功能,采集以后,可以通过 TFT 液晶屏或者 VGA 接口连接显示器进行显示。OV7670, 30W 像素,输出分辨率为 640*480; OV5640, 500W 像素,输出分辨率高达为 2592*1944。关于摄像头选择,用户可以根据自己实际需要进行选购。

CMOS 摄像头接口原理图如图 14.1 所示

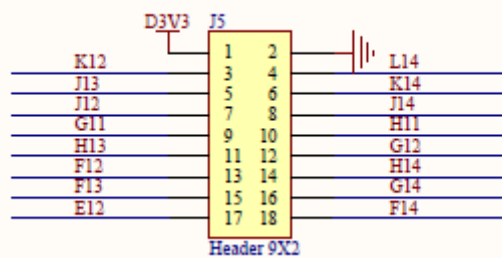


图 14.1 摄像头接口原理图

实物图如图 14.2 所示 (**摄像头模块为选配件**)

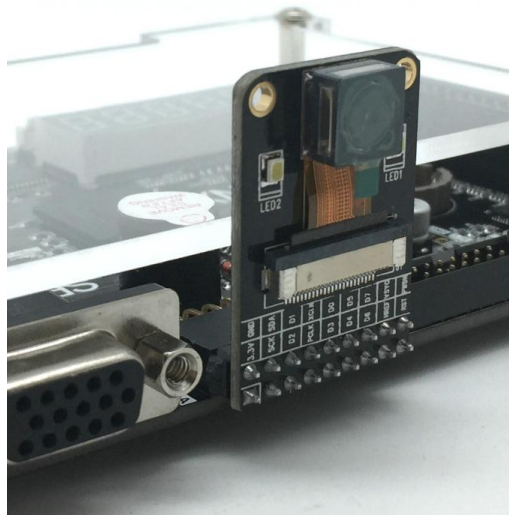


图 14.2 摄像头接口实物图

摄像头接口引脚分配：

引脚名称	FPGA 引脚
CMOS_SCLK	K12

CMOS_SDAT	L14
CMOS_VSYNC	J13
CMOS_HREF	K14
CMOS_PCLK	J12
CMOS_XCLK	J14
CMOS_D<7>	G11
CMOS_D<6>	H11
CMOS_D<5>	H13
CMOS_D<4>	G12
CMOS_D<3>	F12
CMOS_D<2>	H14
CMOS_D<1>	F13
CMOS_D<0>	G14
CMOS_RESET	E12
CMOS_PWDN	F14

十五、数码管

数码管是很常见的一种显示设备，一般分为七段数码管和八段数码管，两者区别就在于八段数码管比七段数码管多了一个“点”。我们采用的数码管为6位一体的八段数码管，数码管的段结构图 15.1 所示

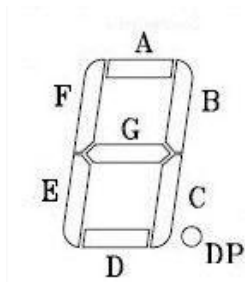


图 15.1 数码管的段结构

我们使用的是共阳极数码管，当某一字段对应的引脚为低电平时，相应字段就点亮，当某一字段对应的引脚为高电平时，相应字段就不亮。

说完上面的原理图，我们来看我们开发板上的设计。

六位一体数码管是属于动态显示，由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位数码管并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感。

六位一体数码管的相同的段都接在了一起，一共是 8 个引脚，然后加上 6 个控制信号引脚，一共是 14 个引脚，如图 15.2 所示，其中 DIG[0..7]是对应数码管的 A,B,C,D,E,F,G,H(即点 DP)；SEL[0..5]是六个数码管的六个控制引脚，也是低电平有效，当控制引脚为低电平时，对应的数码管有了供电电压，这样数码管才能点亮，否则无论数码管的段如何变化，也不能点亮对应的数码管。

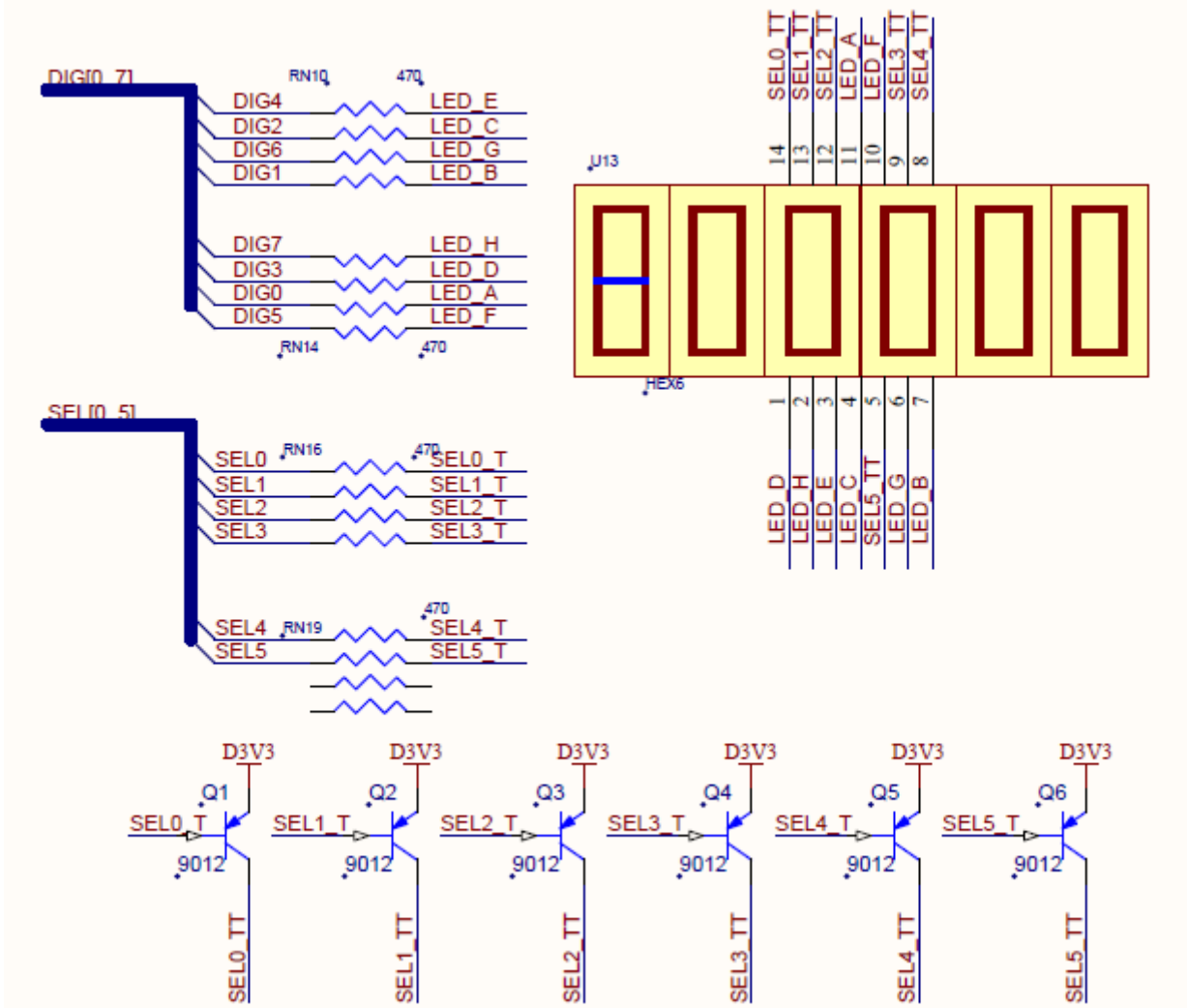


图 15.2 数码管原理图

图 15.3 为数码管实物图



图 15.3 数码管实物图

数码管引脚分配

引脚名称	FPGA 引脚	备注
DIG[0]	C7	对应段 A
DIG[1]	E6	对应段 B
DIG[2]	C5	对应段 C
DIG[3]	F7	对应段 D
DIG[4]	D6	对应段 E
DIG[5]	E7	对应段 F
DIG[6]	D5	对应段 G
DIG[7]	C6	对应点 DP
SEL[0]	D8	从右面数第一个数码管
SEL[1]	E8	从右面数第二个数码管
SEL[2]	F9	从右面数第三个数码管
SEL[3]	F10	从右面数第四个数码管
SEL[4]	E10	从右面数第五个数码管
SEL[5]	D9	从右面数第六个数码管

十六、蜂鸣器

蜂鸣器不多解释了，我们在设计的时候，通过一个三极管进行控制，当低电平时，三极管导通，蜂鸣器响；当高电平，三极管截止，蜂鸣器不响；**为了方便起见，我们在蜂鸣器跟 FPGA 之间加入了一个跳帽 (CB1)，如果讨厌蜂鸣器响，可以把跳帽去掉即可。**原理图如图 16.1

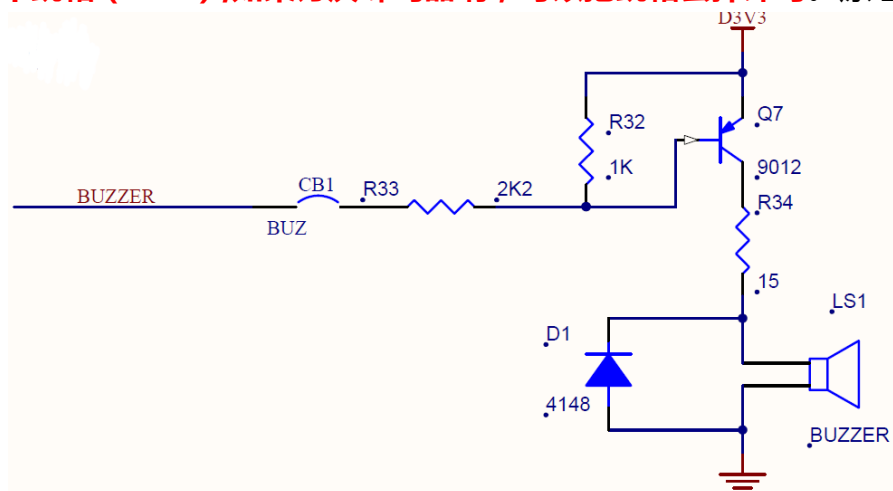


图 16.1 蜂鸣器原理图

图 16.2 为蜂鸣器实物图，黄色为连接蜂鸣器和 FPGA 引脚的跳帽，如果不希望蜂鸣器响，拔掉即可。



图 16.2 蜂鸣器原理图

蜂鸣器引脚分配：

引脚名称	FPGA 引脚
BUZZER	J11

十七、扩展口

开发板预留 2 个扩展口，扩展口有 40 个信号，其中，5V 电源 1 路，3.3V 电源 2 路，地 3 路，IO 口 34 路。这些 IO 口都是独立的 IO 口，没有跟其他设备复用。IO 口连接到 FPGA 引脚上，电平为 3.3V。**切勿直接跟 5V 设备直接连接，以免烧坏 FPGA。如果要接 5V 设备，需要接电平转换芯片。**

在扩展口和 FPGA 连接之间串联了 33 欧姆的排阻，用于保护 FPGA 以免外界电压或电流过高造成损坏，扩展口 J2,J3 的电路如图 17.1, 17.2 所示

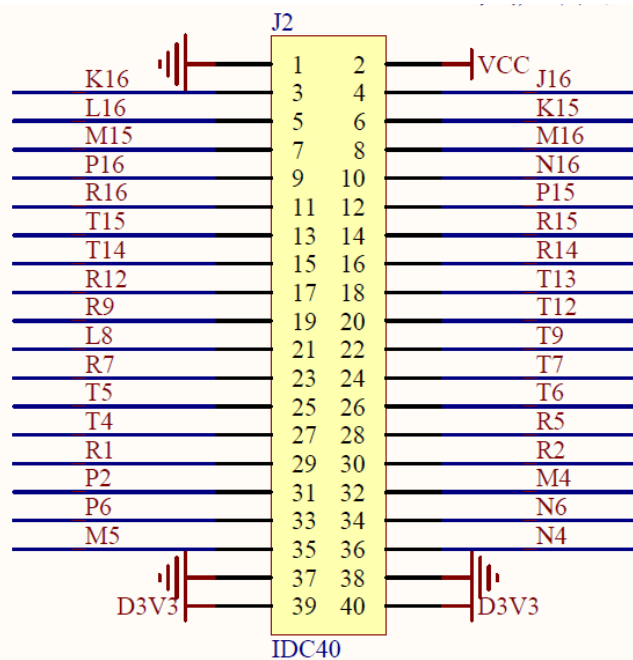


图 17.1 J2 扩展口原理图

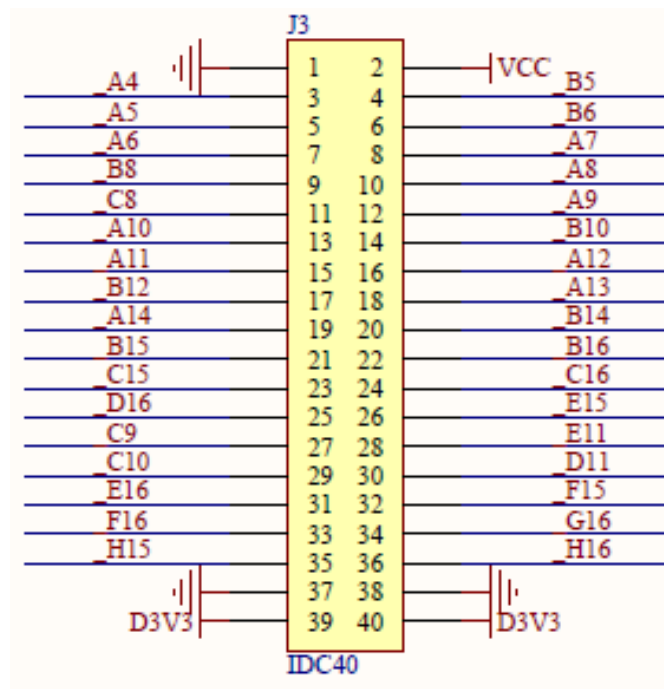


图 17.2 J3 扩展口原理图

图 17.3 为 J2,J3 扩展口实物图，扩展口的 Pin1，Pin2 和 Pin39，Pin40 已经在板上标示出。

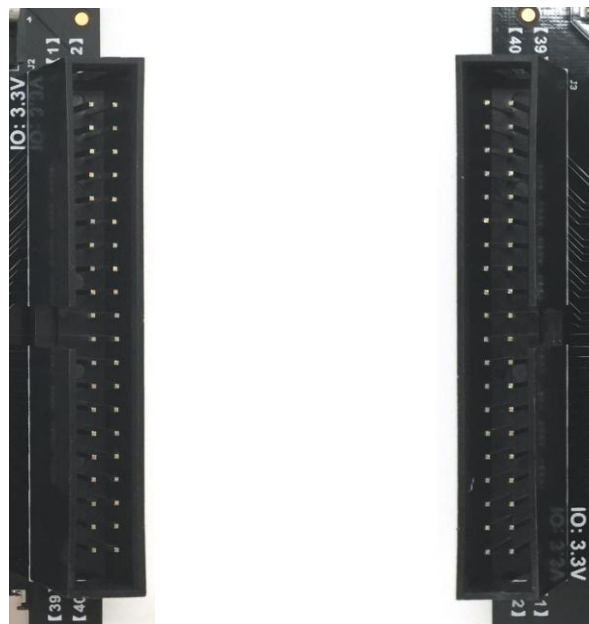


图 17.3 J2, J3 扩展口实物图

此扩展口在连接我们的扩展模块的时候，方向如图 17.4 所示，1,2 脚在接口的上方（注意 PCB 上的标识）。

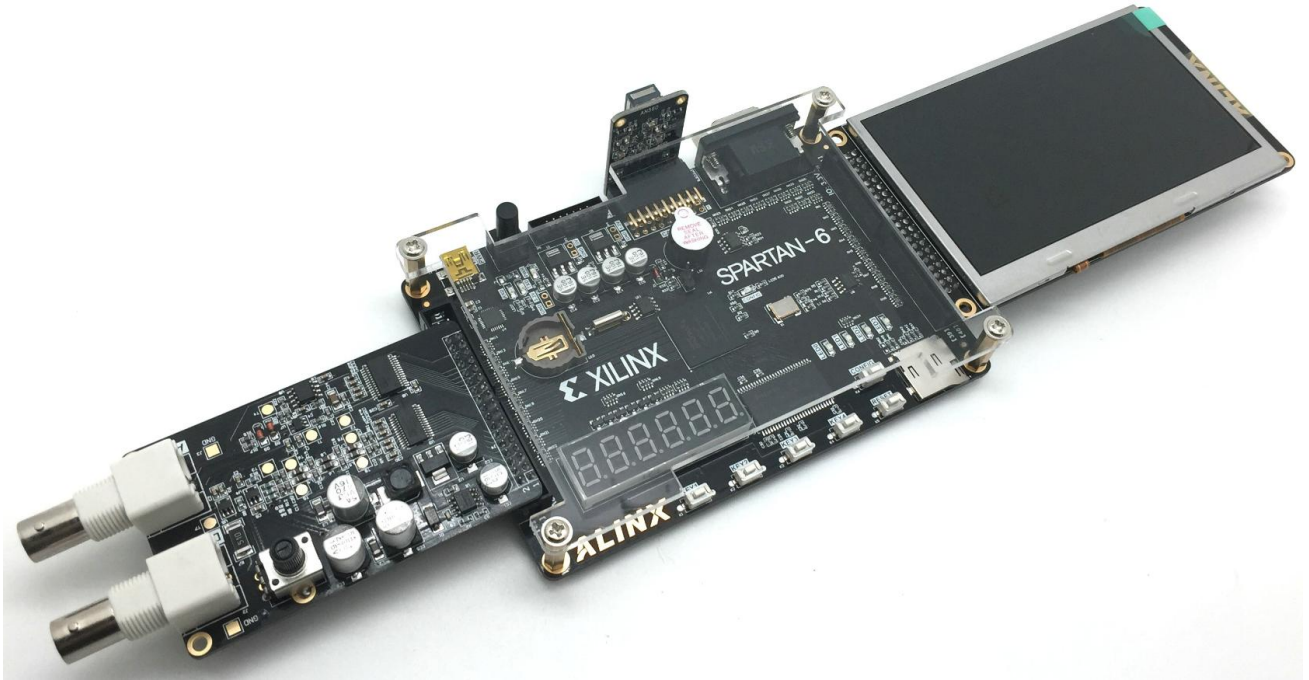


图 17.4 J3 扩展口实物图

J2 扩展口引脚分配

引脚编号	FPGA 引脚	引脚编号	FPGA 引脚
1	GND	2	VCC5V
3	K16	4	K15
5	L16	6	K15
7	M15	8	M16
9	P16	10	N16
11	R16	12	P15
13	T15	14	R15
15	T14	16	R14
17	R12	18	T13
19	R9	20	T12
21	L8	22	T9
23	R7	24	T7
25	T5	26	T6
27	T4	28	R5
29	R1	30	R2
31	P2	32	M4
33	P6	34	N6

35	M5	36	N4
37	GND	38	GND
39	D3V3	40	D3V3

J3 扩展口引脚分配

引脚编号	FPGA 引脚	引脚编号	FPGA 引脚
1	GND	2	VCC5V
3	A4	4	B5
5	A5	6	B6
7	A6	8	A7
9	B8	10	A8
11	C8	12	A9
13	A10	14	B10
15	A11	16	A12
17	B12	18	A13
19	A14	20	B14
21	B15	22	B16
23	C15	24	C16
25	D16	26	E15
27	C9	28	E11
29	C10	30	D11
31	E16	32	F15
33	F16	34	G16
35	H15	36	H16
37	GND	38	GND
39	D3V3	40	D3V3