



# S800板第一次实验

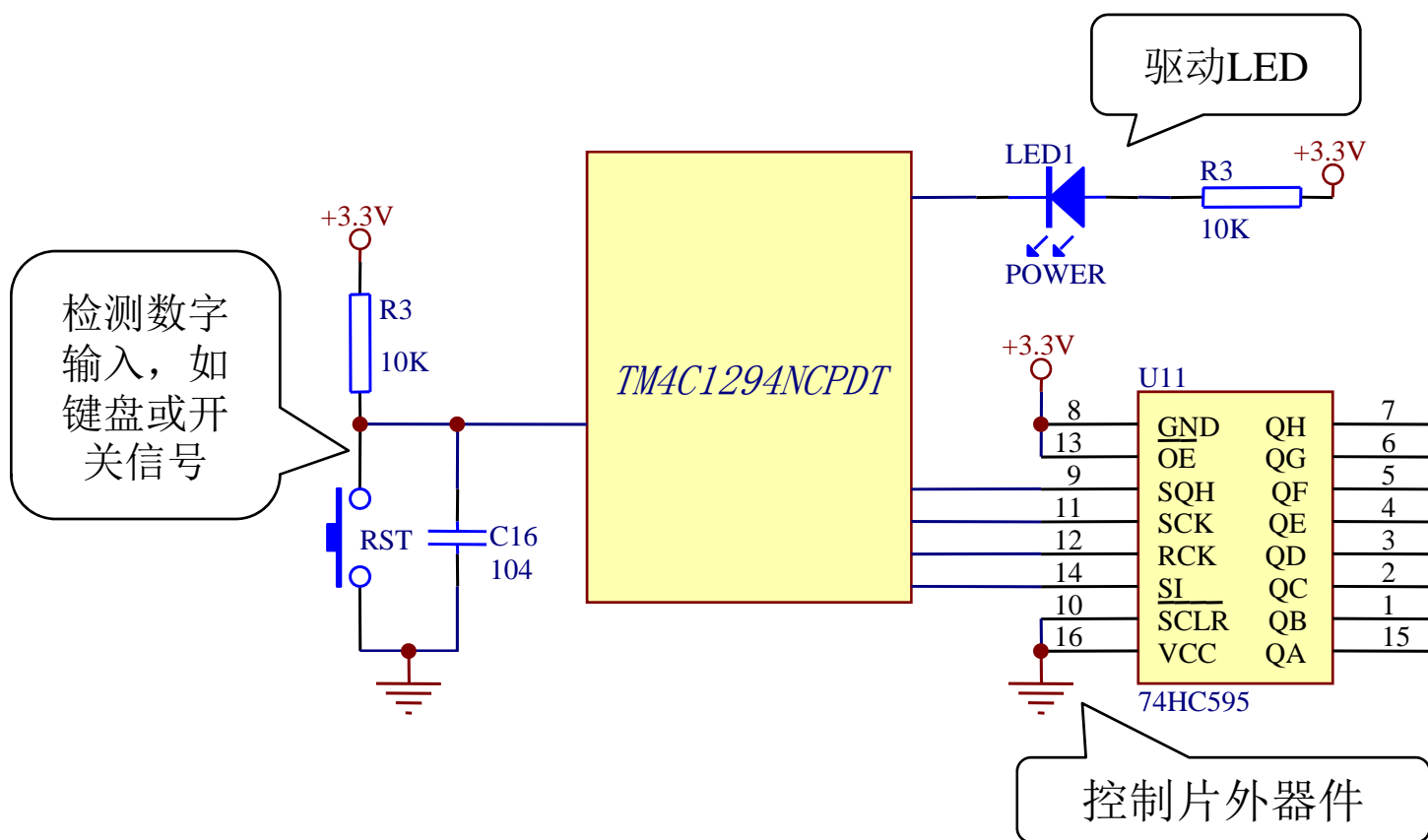


# GPIOs

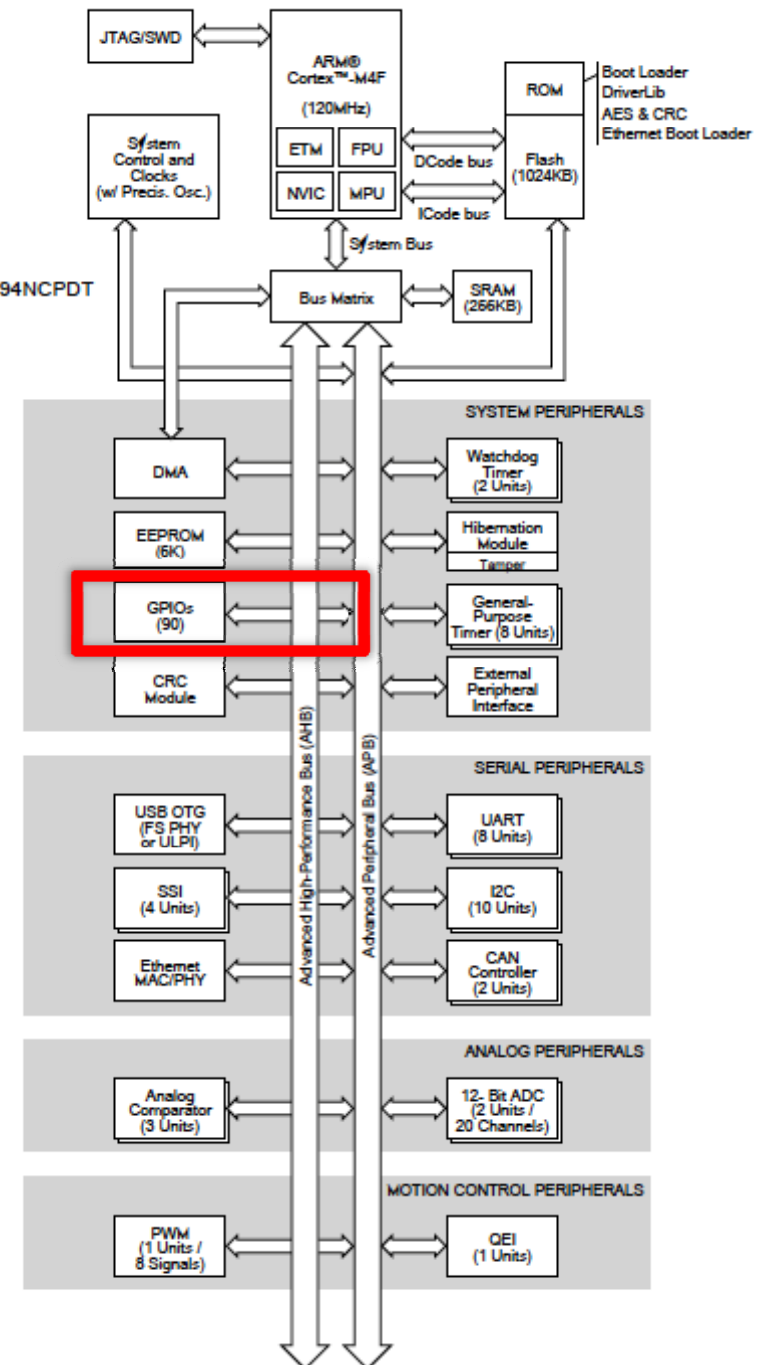
---

General-Purpose Input/Outputs

# GPIO应用



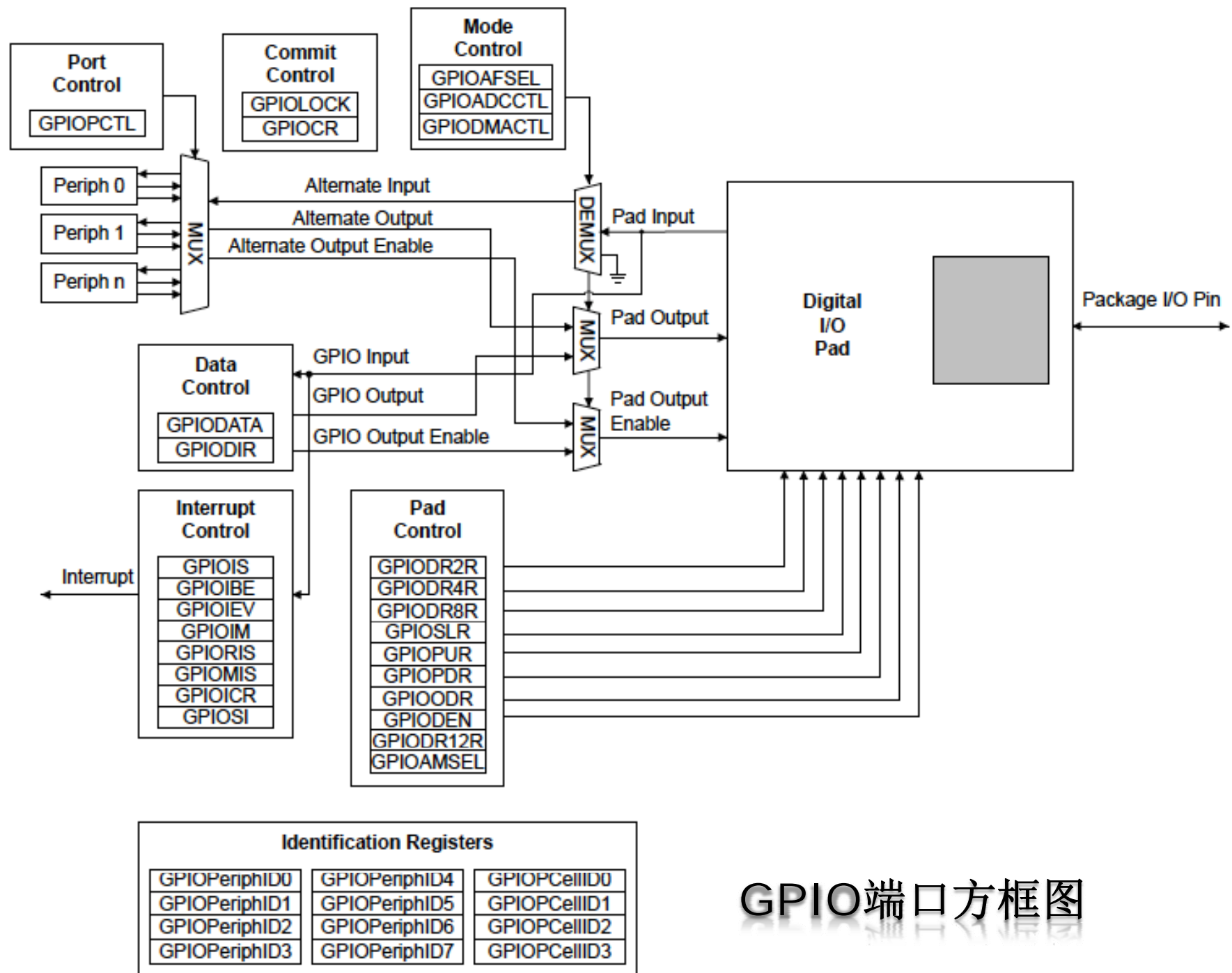
\_\_\_\_\_



# GPIO模组

---

- ⌘ GPIO模组最多由15个模块组成，每个模块对应一个GPIO端口
  - ⌘ 端口 A-H ，端口J， 端口K-N， 和端口P-Q
- ⌘ 每个端口最多8个管脚
- ⌘ 通过AHB总线(Advanced High Performance Bus)访问端口
- ⌘ 高度灵活的复用引脚，可以用作GPIO ， 或是一种或多种的外设功能
- ⌘ 最多支持90个可编程输入输出GPIO。每个管脚均支持：
  - ⌘ 可屏蔽中断生成
  - ⌘ 中断可以上升沿、下降沿或双边沿触发
  - ⌘ 按施密特触发的输入管脚
  - ⌘ 弱上拉或弱下拉电阻可配置
  - ⌘ 2-、4-、8-、10-、12-mA的可配置驱动电流
  - ⌘ 输出斜率可控
  - ⌘ OD（Open drain）门支持

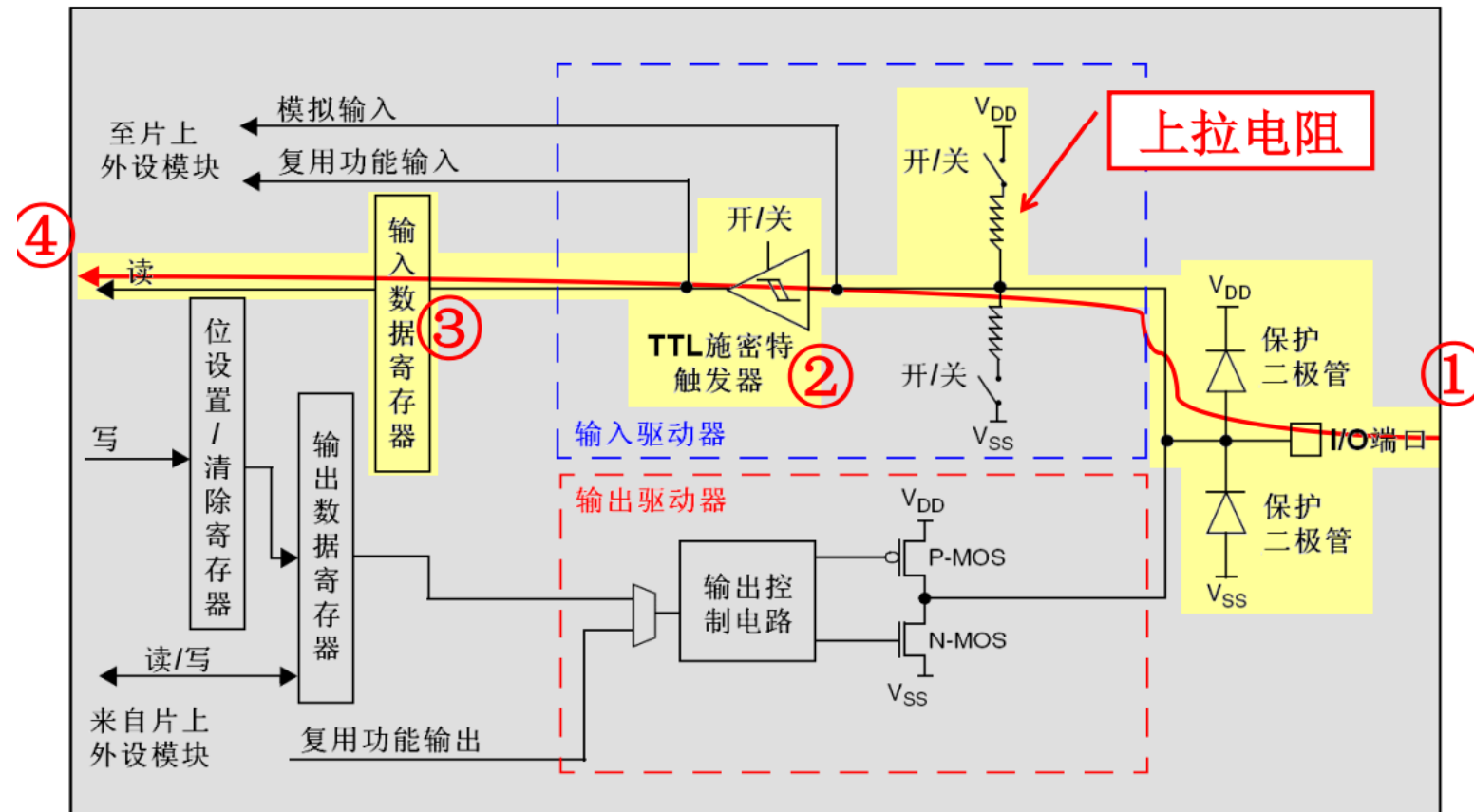


GPIO端口方框图

# GPIO模式

GPIO	TM4C1290 Series
Input Mode	Floating Pull-Up (GPIO_PIN_TYPE_STD_WPU ) Pull-Down (GPIO_PIN_TYPE_STD_WPD ) WAKE (GPIO_PIN_TYPE_WAKE_HIGH) (GPIO_PIN_TYPE_WAKE_LOW)
General Purpose Output	Push-Pull (GPIO_PIN_TYPE_STD ) Open-Drain (GPIO_PIN_TYPE_OD ) Push-Pull+Pull-Up (GPIO_PIN_TYPE_STD_WPU ) Push-Pull+Pull-Down (GPIO_PIN_TYPE_STD_WPD )
Analog	Analog (GPIO_PIN_TYPE_Analog )
Alternate Function Output	Push-Pull (GPIO_PIN_TYPE_STD ) Open-Drain(GPIO_PIN_TYPE_OD ) Push-Pull+Pull-Up (GPIO_PIN_TYPE_STD_WPU ) Push-Pull+Pull-Down (GPIO_PIN_TYPE_STD_WPD )

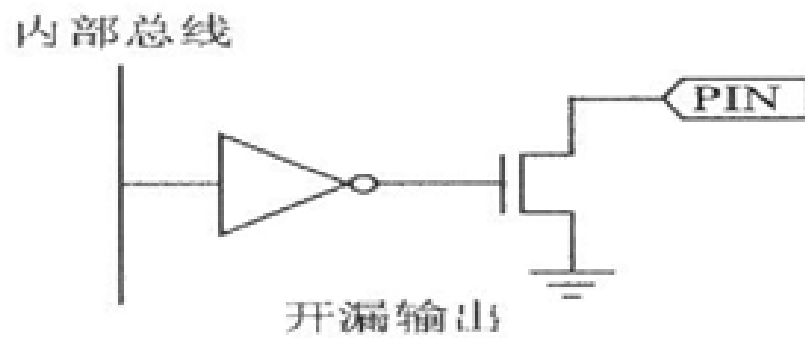
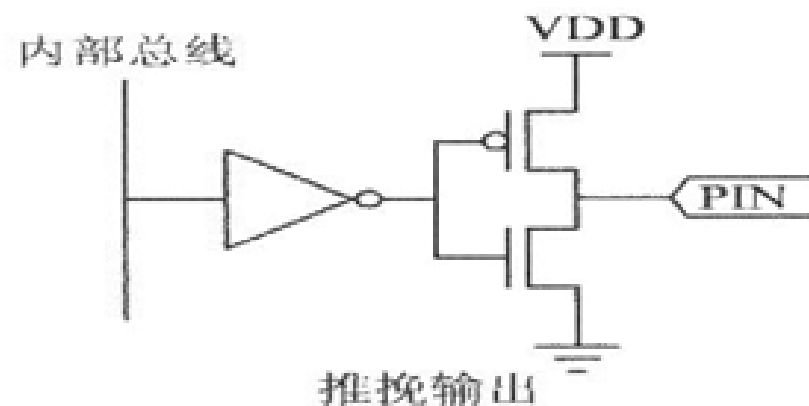
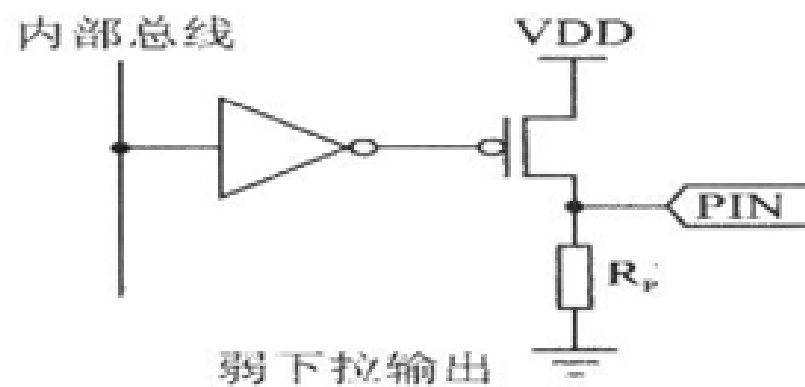
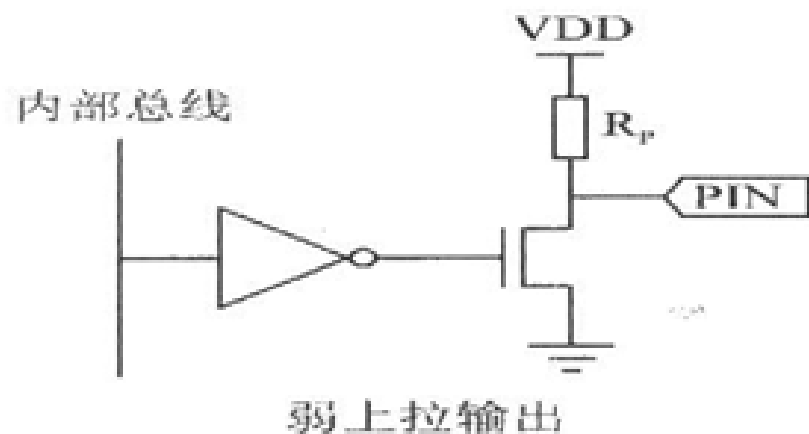
# 输入上拉模式





# GPIO不同模式下的等效电路

---



# 第一次实验

---

# 实验目的

---

- ⌘ 熟悉集成开发环境KEIL uVision5，学习新建一个工程项目
- ⌘ 掌握GPIO的工作原理，能够结合GPIO的输入与输出功能进行实验

# 实验相关硬件

---

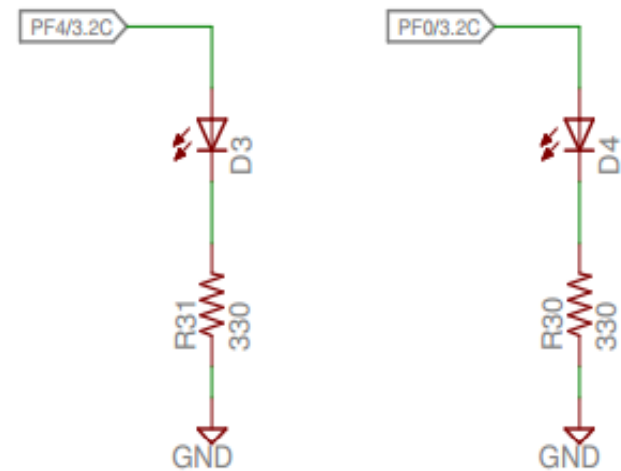
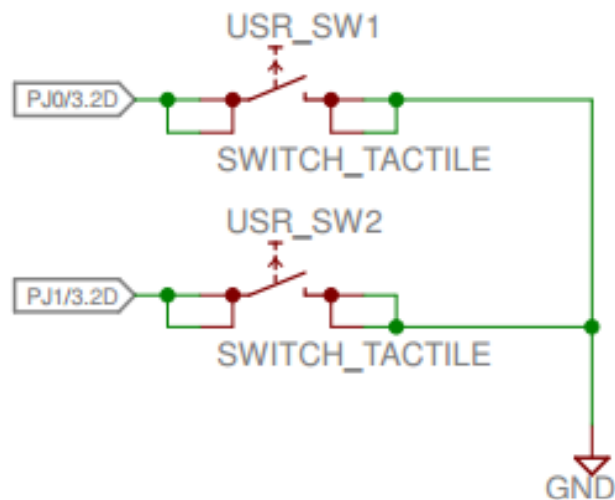
## ⌘ 红板

名称	对应管脚	说明
USR_SW1	PJ0	用户输入按键，低有效
USR_SW2	PJ1	用户输入按键，低有效
D4	PF0	用户控制LED，绿，高有效

## ⌘ 蓝板

名称	对应管脚	说明
LED_M0	PF0	用户控制LED，低有效
LED_M1	PF1	用户控制LED，低有效

# 红板上的按键&LED灯



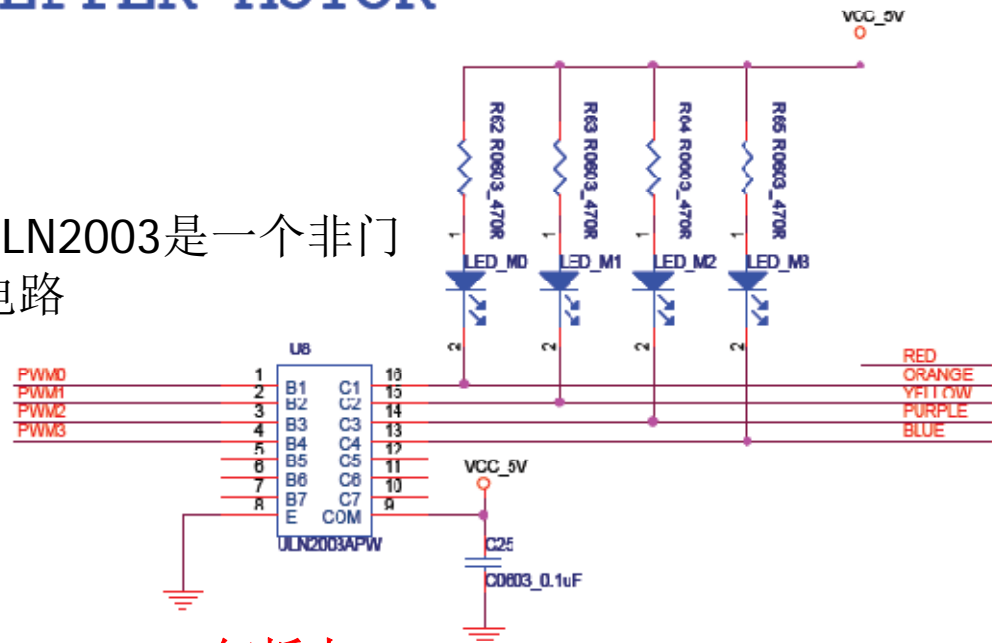
For Ethernet example Applications:  
LED4 is default configured as Ethernet Link OK  
LED3 is default configured as Ethernet TX/RX activity

User may re-configure these pins / LED's for any application usage.

# 蓝板上的LED灯

## STEPPER MOTOR

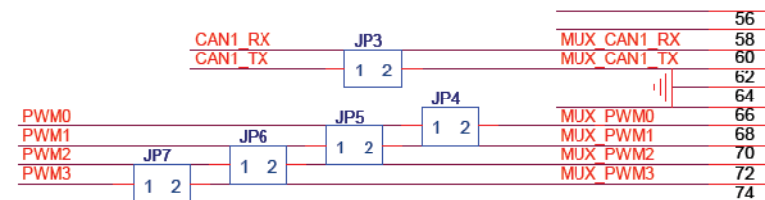
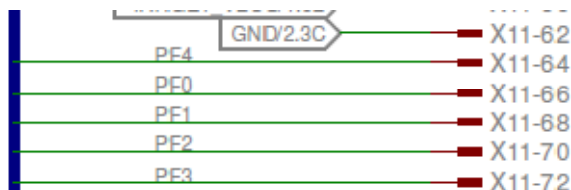
ULN2003是一个非门电路



红板上

红板与蓝板均有PF0控制的LED，两个LED，一绿（高有效）、一红（低有效）

蓝板上



# 预定义的说明

## ⌘ gpio.h

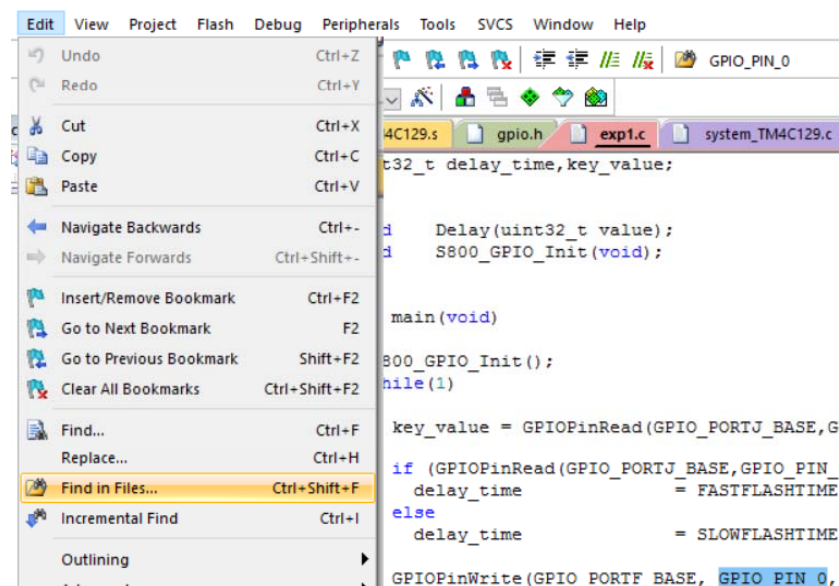
定义管脚对应的值

*#define GPIO\_PIN\_0           0x00000001 // GPIO pin 0*

*#define GPIO\_PIN\_7           0x00000080 // GPIO pin 7*

## ⌘ Hw\_memmap.h 定义每个模组的GPIO首地址

*#define GPIO\_PORTF\_BASE       0x40025000 // GPIO Port F*



# 实验中涉及到的主要函数

---

⌘ 模组使能，在使用GPIO前，需要对本组模组使能

☒ SysCtlPeripheralEnable

⌘ 配置GPIO为数字输出，或数字输入，或复用功能，或模拟量输入

☒ GPIOPinTypeGPIOOutput

☒ GPIOPinTypeGPIOInput

⌘ 配置PAD模式GPIOPadConfigSet，对输入来说输出电流选项无效



# 函数SysCtlPeripheralEnable

---

⌘ 驱动库手册：P505

功能	<p>使能一个外设。</p> <p>上电时，全部的外设都被禁止。为了使外设能工作或响应寄存器的读/写操作，它们必须被使能。使能的方法是调用头文件&lt;sysctl.h&gt;中的这个函数。</p>
原型	<code>Void SysCtlPeripheralEnable(uint32_t ui32Peripheral)</code>
参数	ui32Peripheral 是要使能的外设。
说明	<p>The ui32Peripheral参数必须取下面的其中一个值：</p> <p><b>SYSCTL_PERIPH_GPIOA, SYSCTL_PERIPH_GPIOB, ... ..</b></p> <p><b>SYSCTL_PERIPH_I2C0, SYSCTL_PERIPH_I2C1, .....</b></p>

# 函数SysCtlPeripheralReady

⌘ 驱动库手册：P508

功能	确定外设是否准备好
原型	Bool SysCtlPeripheralReady(uint32_t ui32Peripheral)
参数	ui32Peripheral 是要查询的外设
说明	<p>This function determines if a particular peripheral is ready to be accessed. The peripheral may be in a non-ready state if it is not enabled, is being held in reset, or is in the process of becoming ready after being enabled or taken out of reset.</p> <p>The ui32Peripheral parameter must be only one of the following values: <b>SYSCTL_PERIPH_ADC0</b>, .....</p> <p><b>SYSCTL_PERIPH_GPIOA</b>, ....., <b>SYSCTL_PERIPH_I2C0</b>, ....., <b>SYSCTL_PERIPH_TIMER0</b>, .....</p>
返回值	Returns <b>true</b> if the specified peripheral is ready and <b>false</b> if it is not.

# 函数GPIOPinTypeGPIOOutput

---

⌘ 驱动库手册：P273

功能	配置管脚用作输出引脚。
原型	<code>void GPIOPinTypeGPIOOutput (uint32_t ui32Port, uint8_t ui8Pins)</code>
参数	<code>ui32Port</code> 是GPIO端口的基址。 <code>ui8Pins</code> 是管脚的位组合（bit-packed）表示。
说明	<p>GPIO管脚必须正确配置，以便作为GPIO输出能正常工作。这个函数为管脚提供了正确的配置。</p> <p>管脚用一个位组合（bit-packed）的字节来指定，在这个字节中，置位的位用来识别被访问的管脚，字节的位0代表GPIO端口管脚0、位1代表GPIO端口管脚1，等等。</p>

# 函数GPIOPinTypeGPIOInput

---

⌘ 驱动库手册：P272

功能	配置管脚用作输入引脚。
原型	Void GPIOPinTypeGPIOInput(uint32_t ui32Port, uint8_t ui8Pins)
参数	ui32Port 是GPIO端口的基址。 ui8Pins是管脚的位组合（bit-packed）表示。
说明	<p>GPIO管脚必须正确配置，以便作为GPIO输入能正常工作。这个函数为管脚提供了正确的配置。</p> <p>管脚用一个位组合（bit-packed）的字节来指定，在这个字节中，置位的位用来识别被访问的管脚，字节的位0代表GPIO端口管脚0、位1代表GPIO端口管脚1，等等。</p>

# 函数GPIOPadConfigSet

⌘ 驱动库手册：P265

功能	设置指定管脚的配置。
原型	Void GPIOPadConfigSet(uint32_t ui32Port, uint8_t ui8Pins, uint32_t ui32Strength, uint32_t ui32PinType)
参数	ui32Port 是GPIO端口的基址。 ui8Pins 是特定管脚的位组合(bit-packed)表示。 ui32Strength指出输出驱动强度。 ui32PinType指定管脚类型
说明	<p>这个函数设置所选GPIO端口指定管脚的驱动强度和类型。对于配置用作输入端口的管脚，端口按照要求配置，但是对输入唯一真正的影响是上拉或下拉终端的配置。</p> <p>参数ui32Strength可以是下面的一个值： GPIO_STRENGTH_2MA, GPIO_STRENGTH_4MA GPIO_STRENGTH_8MA, GPIO_STRENGTH_8MA_SC GPIO_STRENGTH_6MA, GPIO_STRENGTH_10MA GPIO_STRENGTH_12MA</p>

# 函数GPIOPadConfigSet

⌘ 续上页

说明	<p>其中GPIO_STRENGTH_xMA指示2、4或8mA的输出驱动强度；而GPIO_OUT_STRENGTH_8MA_SC指定了带斜率控制（slew control）的8mA输出驱动。</p> <p>参数ui32PinType可以是下面的其中一个值：</p> <p>GPIO_PIN_TYPE_STD; GPIO_PIN_TYPE_STD_WPU GPIO_PIN_TYPE_STD_WPD; GPIO_PIN_TYPE_OD GPIO_PIN_TYPE_ANALOG; GPIO_PIN_TYPE_WAKE_HIGH GPIO_PIN_TYPE_WAKE_LOW</p> <p>其中，GPIO_PIN_TYPE_STD*指定一个推挽管脚，GPIO_PIN_TYPE_OD*指定一个开漏管脚，*_WPU指定一个弱上拉，*_WPD指定一个弱下拉，GPIO_PIN_TYPE_ANALOG 指定一个模拟输入。</p> <p>管脚用一个位组合（bit-packed）的字节来指定，在这个字节中，置位的位用来识别被访问的管脚，字节的位0代表GPIO端口管脚0、位1代表GPIO端口管脚1等等。</p>
----	---

# 函数GPIOPinRead

---

⌘ 驱动库手册：P266

功能	读取指定管脚的当前值
原型	<code>int32_t GPIOPinRead(uint32_t ui32Port, uint8_t ui8Pins)</code>
参数	<code>ui32Port</code> 是GPIO端口的基址。 <code>ui8Pins</code> 是管脚的位组合（bit-packed）表示。
说明	The values at the specified pin(s) are read, as specified by <code>ui8Pins</code> . Values are returned for both input and output pin(s), and the value for pin(s) that are not specified by <code>ui8Pins</code> are set to 0.
返回值	Returns a bit-packed byte providing the state of the specified pin, where bit 0 of the byte represents GPIO port pin 0, bit 1 represents GPIO port pin 1, and so on. Any bit that is not specified by <code>ui8Pins</code> is returned as a 0. Bits 31:8 should be ignored.

# 函数GPIOPinWrite

---

⌘ 驱动库手册：P284

功能	往指定管脚写一个值
原型	Void GPIOPinWrite(uint32_t ui32Port, uint8_t ui8Pins, uint8_t ui8Val)
参数	ui32Port 是GPIO端口的基址。 ui8Pins是管脚的位组合（bit-packed）表示。 ui8Val是要写到管脚的值。
说明	Writes the corresponding bit values to the output pin(s) specified by ui8Pins. Writing to a pin configured as an input pin has no effect.



# 编程示例

---

```
//Enable PortF
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
//Wait for the GPIO moduleF ready
while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF));
// Set PF0 as Output pin
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_0);
//Set the PJ0,PJ1 as input pin
GPIOPinTypeGPIOInput(GPIO_PORTJ_BASE,GPIO_PIN_0 | GPIO_PIN_1);
//Set the PJ0,PJ1 has pull resistor up
GPIOPadConfigSet(GPIO_PORTJ_BASE,GPIO_PIN_0 |
    GPIO_PIN_1,GPIO_STRENGTH_2MA,GPIO_PIN_TYPE_STD_WPU);
```

# 实验1.1

---

- ⌘ 运行示例程序
- ⌘ 实验现象：正常运行时，慢闪PF0；当按下USR\_SW1时，快闪PF0
- ⌘ 读懂例程，观察现象
- ⌘ 进入Debug模式，增加Watch Window，观察key\_value的值

# 思考题

---

- ⌘ GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_0, GPIO\_PIN\_0)函数中，每个函数项的意义。第三个函数项为GPIO\_PIN\_0，如果改为1或改为2，或其他值，分别有什么现象？
- ⌘ 结合硬件说明GPIOPadConfigSet行的作用。如果此行注释，在WATCH窗口中观察key\_value值会有什么变化。

# 实验1.2

---

## ⌘ 实验内容:

当按下USR\_SW1键时，点亮LED\_M0，放开时，熄灭LED\_M0

当按下USR\_SW2键时，点亮LED\_M1，放开时，熄灭LED\_M1

# 实验1.3

---

## ⌘ 实验内容:

当第一次短按USR\_SW1键时，闪烁LED\_M0，  
当第二次短按USR\_SW1键时，熄灭LED\_M0，  
当第三次短按USR\_SW1键时，闪烁LED\_M1，  
当第四次短按USR\_SW1键时，熄灭LED\_M1，  
如此循环往复

# 实验中出现的問題

---

⌘ 开机，选择win7系统

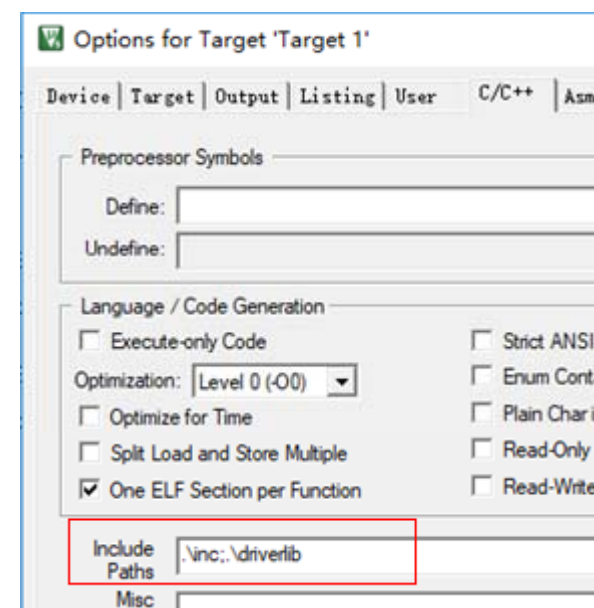
⌘ 找不到头文件

☑ 确保文件夹inc、driverlib已经拷贝到工程文件夹

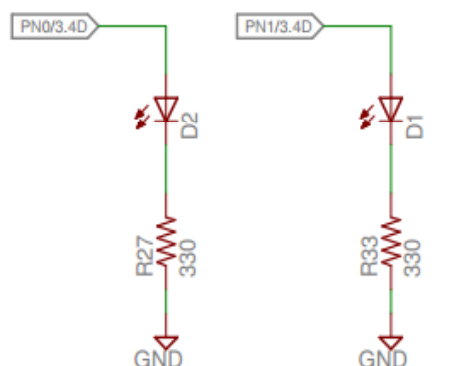
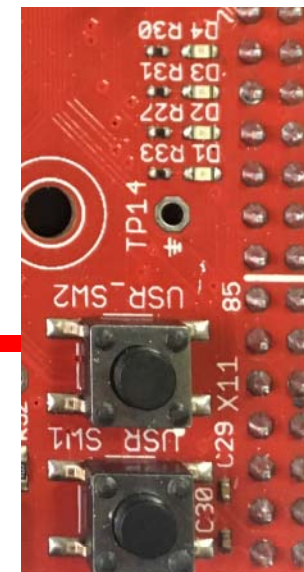
☑ 配置正确

⌘ GPIO\_PIN\_0的定义

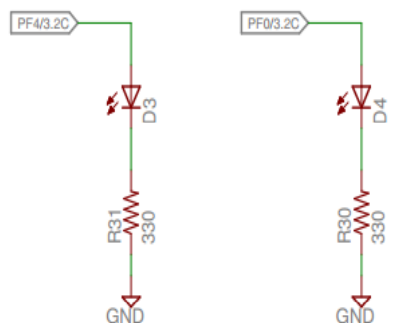
⌘ 增加Watch Window，观察变量



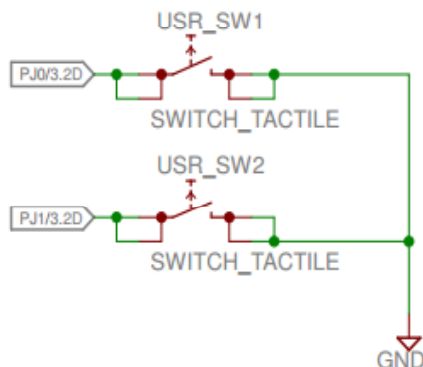
# 红板上：2键、4LED灯



See PF0 and PF4 for additional LED's used for Ethernet or user application

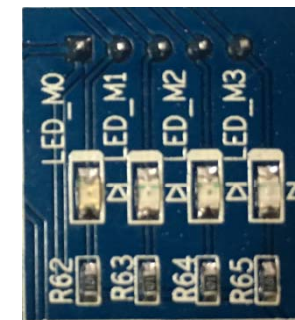


For Ethernet example Applications:  
LED4 is default configured as Ethernet Link OK  
LED3 is default configured as Ethernet TX/RX activity  
User may re-configure these pins / LED's for any application usage.

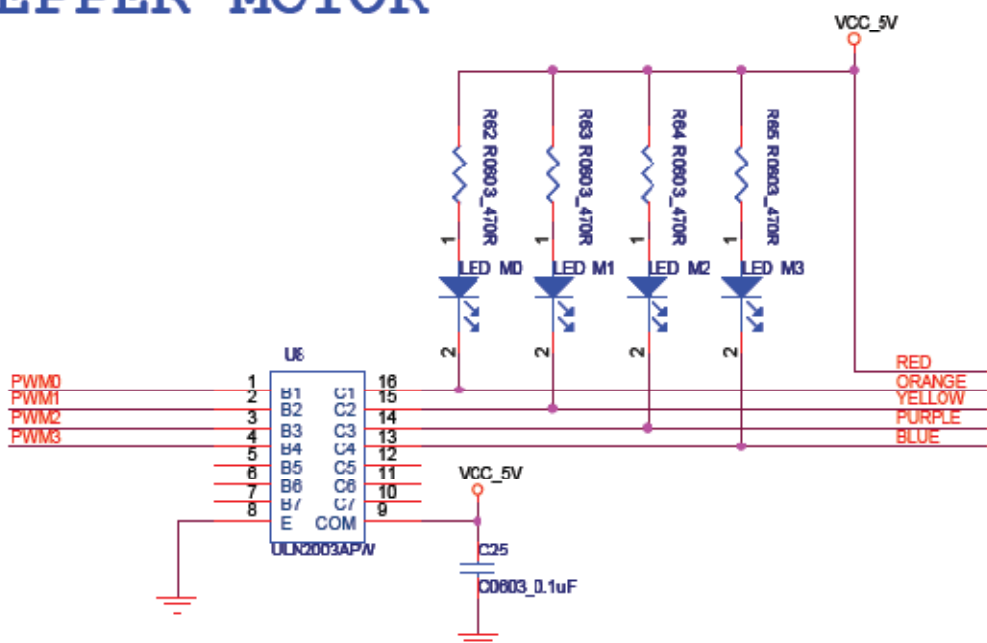


USR_SW1	PJ0	用户输入按键低有效
USR_SW2	PJ1	用户输入按键低有效
D0		3.3V电源指示，绿LED高有效
D1	PN1	用户控制绿LED高有效
D2	PN0	用户控制绿LED高有效
D3	PF4	用户控制绿LED高有效
D4	PF0	用户控制绿LED高有效

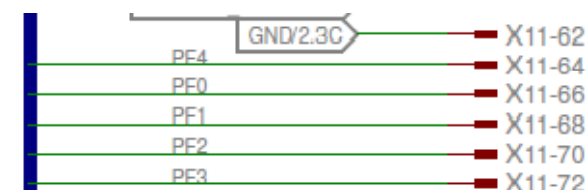
# 蓝板上:



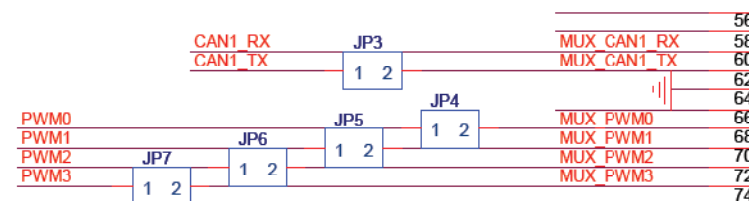
## STEPPER MOTOR



红板上



蓝板上



红板与蓝板均有PF0控制的LED，两个LED，一绿（高有效）、一红（低有效）

LED_M0	PF0	用户控制LED低有效
LED_M1	PF1	用户控制LED低有效
LED_M2	PF2	用户控制LED低有效
LED_M3	PF3	用户控制LED低有效