

第 13 课，学会 GPIO 输出并控制一个继电器闭合动作

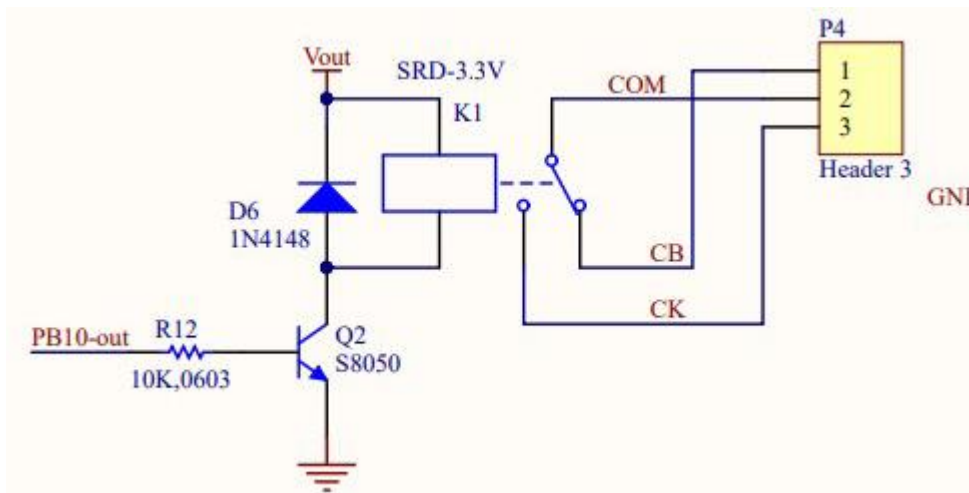
本课学习内容：

- 学会使用 GPIO 输出功能
- 利用 GPIO 输出控制一路继电器进行动作

在讲解什么是 GPIO 之前，我们先讲解一下 MCU 对外控制原理，在数字电路中，电压信号只有两种情况，高电平和低电平，用数字来记录就是 1 和 0。单片机内部的 CPU，寄存器，总线等等结构都是通过 1 和 0 两种信号来运作的，数据也是以 1 或者 0 来保存的。单片机的输入输出管脚，也就是 IO 口，也是只输出或识别 1 和 0 两种信号，也就是高电平和低电平。当单片机输出一个或一组电平信号到 IO 口后，外部的设备就可以“读”到这些信号，并进行相应操作，这就是单片机对外部的控制。当外部一个或一组电平信号送到单片机的 IO 口时，单片机也可以读到这些信号，并进行分析操作，这就是单片机对外部设备信号的读取。当然实际的操作中，这些信号可能十分复杂，必须严格地按照规定的时间顺序（时序）输入输出。每种设备也都规定了自己的时序（IIC,SPI,串口通信），只要都严格遵守，就可以控制任何设备，做出只要你想像得出的任何事情。我们将这种能对外输出高低电平，对内采集输入输入高低电平的硬件引脚叫做 GPIO,STM32 单片机的高电平为 3.3V,低电平为 0V,千万记住对于外围接入输入输出硬件设备做到电平匹配。

我们先用一下 Pyboard-tool，进行一下简单的脚本控制，训练一下 GPIO 输出功能。

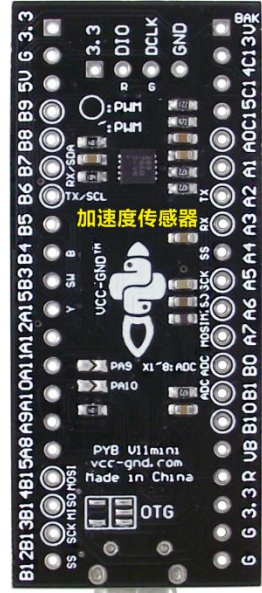
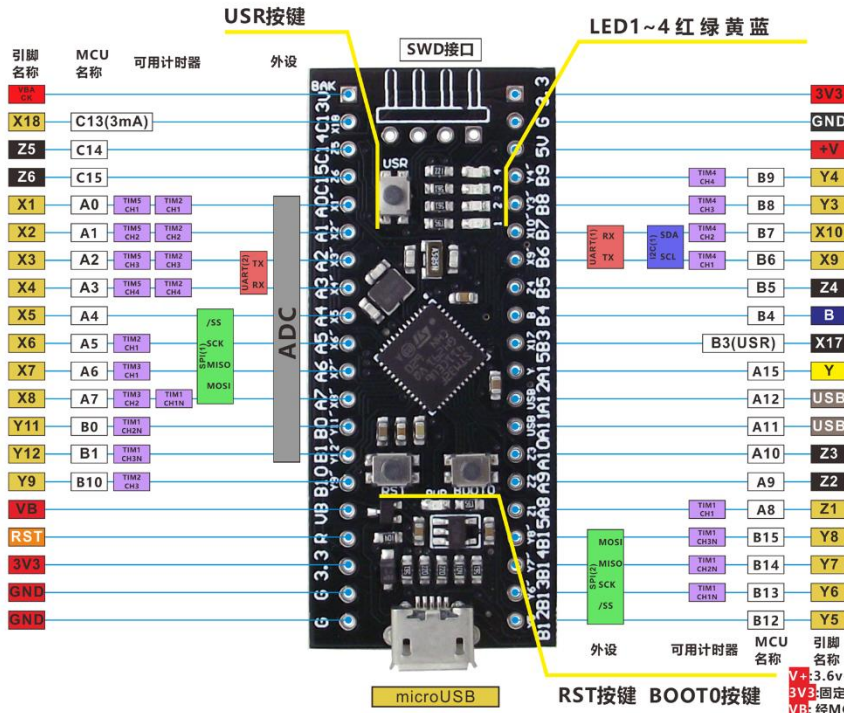
注意有时候使用 Pyboard-tool 调试器之前，确保 pyboard 生成的可移动硬盘中的脚本不会影响我们这次 REPL 语句验证。其次，我们得知道在我们 PYBminiDB 中，是哪一个端口连接到我们的继电器硬件，需要我们先看一下底板原理图。



我们看到控制继电器的引脚为 PB10 引脚，在我们的 Pyboard 板中根据 PYBmini 板子的引脚示意图，可以用“PB10”或“B10”或“Y9”名称进行相应的控制 GPIO 输出状态，根据原理图我们知道高电平可以使继电器产生动作，低电平释放动作。可以利用万用表进行继电器输出检

源地 PYBminiDB 学习板
嵌入式 STM32-python 学习教程
测。

VCC-GND.COM



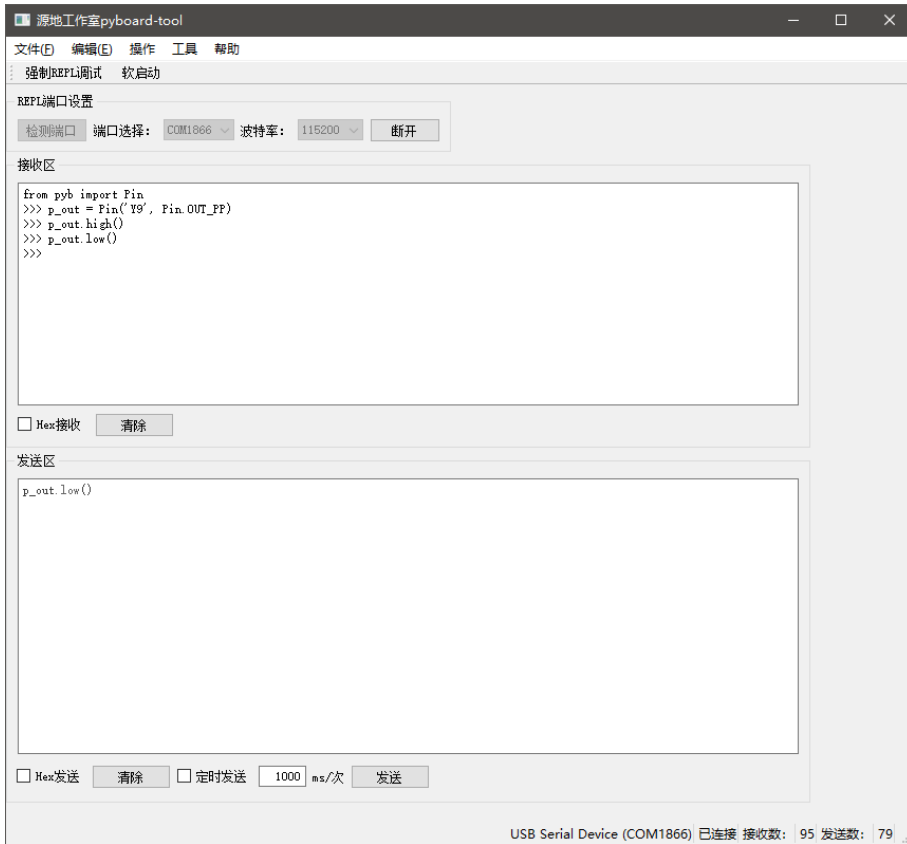
µPython™ uPython™ pyboard™
源地® VCC-GND®
PYBV1.1mini



- V+: 3.6v-12v 直流电源输入端(当USB连接时由USB供电).
- 3V3: 固定输出3.3V电源端,最大输出电流不要超过300mA.
- VB: 经MOS管保护的电池电源输入端.
- VBACK: 备份电源输入端.B6(X9),B7(X10)连接到了加速度传感器I2C引脚上,慎用.B2连接到了加速度传感器的中断输出引脚,故没有引出.
- B: B4被板载LED4蓝灯使用,慎用.Y: A15被板载LED3黄灯使用,慎用.
- USB: A11,A12外接USB DM和DP,虽然引出引脚, 只能在不使用USB功能时使用.
- Z2: A9.被用作USB插入检查, 当不用系统时, 背后焊盘断开可用.
- Z3: A10.被用作USB的ID检查, 当不用系统时, 背后焊盘断开可用.
- Z4: B5为加速度传感器AVDD引脚供电使用, 慎用.
- Z5: C14连接到实时时钟晶振上, 平时慎用.Z6: C15连接到实时时钟晶振上, 平时慎用.
- X17: B3被USR按键功能使用, 慎用.
- PWR: 当板子通电后PWR电源指示灯亮起, 方便在电路故障(负载短路)后指示供电情况.
- BOOT0: 按住BOOT0按键再按下RST按键将可以方便进入DFU状态, 方便刷固件.
- SWD: 接口可以连接使用调试器例如J-link或ST-Link, 进行常规C语言开发.



```
from pyb import Pin #应用 PIN 库  
p_out = Pin('Y9', Pin.OUT_PP)#输出推挽模式  
p_out.high()#输出为高电平，进而控制电路进行继电器的闭合
```



`p_out.low()` #输出为低电平，释放继电器。同样可以利用万用表进行对应的引脚检测，看看是否动作。

在这我们补充一下，继电器的作用，继电器（英文名称：relay）是一种电控制器件，是当输入量（激励量）的变化达到规定要求时，在电气输出电路中使被控量发生预定的阶跃变化的一种电器。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系。通常应用于自动化的控制电路中，它实际上是用小电流去控制大电流运作的一种“自动开关”。故在电路中起着自动调节、安全保护、转换电路等作用。

最后总结一下常用的 GPIO 输出的语法用法：

#引脚和 GPIO 使用

#如果是正式的版本的 Pyboard 可以使用 X Y 开头的对应+数字标识控制对应的引脚

```
from pyb import Pin
```

源地 PYBminiDB 学习板

嵌入式 STM32-python 学习教程

VCC-GND.COM

```
p_out = Pin('X1', Pin.OUT_PP)
```

#X1 引脚赋值于 p_out 作为输出功能使用

```
p_out.high()
```

#输出高电平

```
p_out.low()
```

#输出低电平

#或者是控制引脚的高低电平

```
p_out.value(1)
```

```
p_out.value(0)
```

#或这样控制引脚的高低电平

```
p_out(1)
```

```
p_out(0)
```