

5.20 与串行端口的数据通信

问题

你想通过串行端口读写数据，典型场景就是和一些硬件设备打交道(比如一个机器人或传感器)。

解决方案

尽管你可以通过使用Python内置的I/O模块来完成这个任务，但对于串行通信最好的选择是使用 [pySerial包](#)。这个包的使用非常简单，先安装pySerial，使用类似下面这样的代码就能很容易的打开一个串行端口：

```
import serial
ser = serial.Serial('/dev/tty.usbmodem641', # Device name varies
                    baudrate=9600,
                    bytesize=8,
                    parity='N',
                    stopbits=1)
```

设备名对于不同的设备和操作系统是不一样的。比如，在Windows系统上，你可以使用0,1等表示的一个设备来打开通信端口“COM0”和“COM1”。一旦端口打开，那就可以使用 `read()`，`readline()` 和 `write()` 函数读写数据了。例如：

```
ser.write(b'G1 X50 Y50\r\n')
resp = ser.readline()
```

大多数情况下，简单的串口通信从此变得十分简单。

讨论

尽管表面上看起来很简单，其实串口通信有时候也是挺麻烦的。推荐你使用第三方包如 `pySerial` 的一个原因是它提供了对高级特性的支持(比如超时，控制流，缓冲区刷新，握手协议等等)。举个例子，如果你想启用 `RTS-CTS` 握手协议，你只需要给 `Serial()` 传递一个 `rtscts=True` 的参数即可。其官方文档非常完善，因此我在这里极力推荐这个包。

时刻记住所有涉及到串口的I/O都是二进制模式的。因此，确保你的代码使用的是字节而不是文本(或有时候执行文本的编码/解码操作)。另外当你需要创建二进制编码的指令或数据包的时候，`struct` 模块也是非常有用的。