使用swig连接python与c

注：仅示例python3

通过python发一组数据到C，C调用底层硬件完成数据收尾操作，数据主要是char *的形式到c，python这边需要使用bytes对其相连。

```
from binascii import b2a_hex, a2b_hex

b = b'\x78\x79\x86\x88\xfa\x33\x45'    #以byte表示的bytes, binascii.a2b_hex之后的
print(f'b.hex() = {b.hex()}')

print(f'binascii.b2a_hex(b) = {binascii.b2a_hex(b)}')

b2a = b'78798688fa3345'    #以字符串形式表示的bytes, binascii.b2a_hex之后的
a2b = binascii.a2b_hex(b2a)
print(f'binascii.a2b_hex(b2a) = {binascii.a2b_hex(b2a)}')


io.py2c_test(a2b,7)    #以c的交互需要使用以byte(a2b)表示的bytes，不能是binascii.b2a_hex之后的。
```

在swig的.i文件中需要加入以下内容，表示c char*之间与python bytes相对而不是采用unicode，这一点相当重要，在python中不要使用字符串与C之间传递，而应该用bytes来传递，简单可行，不会发生字符转换规则以及check等等问题，latin-1转换>0x80的字符时有问题，utf8本身也不是属于byte0-255之间的字符定义）。

```
%begin %{
#define SWIG_PYTHON_STRICT_BYTE_CHAR
%}
```

c侧程序示例如下： 注：参数只能写成 char * ，不能写成unsigned char*, 但是可以在c内部通过强制转换将其转换成unsigned char *，

```
void py2c_test(char * tdi_bytes, int bytes_size)
{
    int i;
    printf("mpsse.c:: python to c bytes_size = %0d\n", bytes_size);
    for (i=0;i<bytes_size; i++) {
        printf("mpsse.c:: python to c (unsigned char ) = %0x tdi_bytes = %0x\n", (unsigned char )tdi_bytes[i], tdi_bytes[i]);
    }
}
```