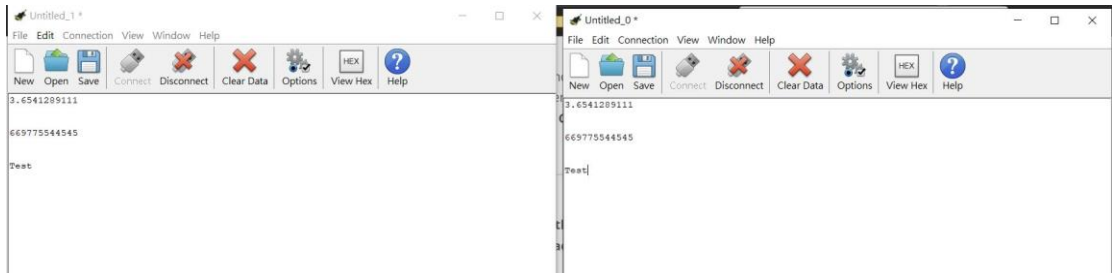


In this lab, we learn how to use XBee.

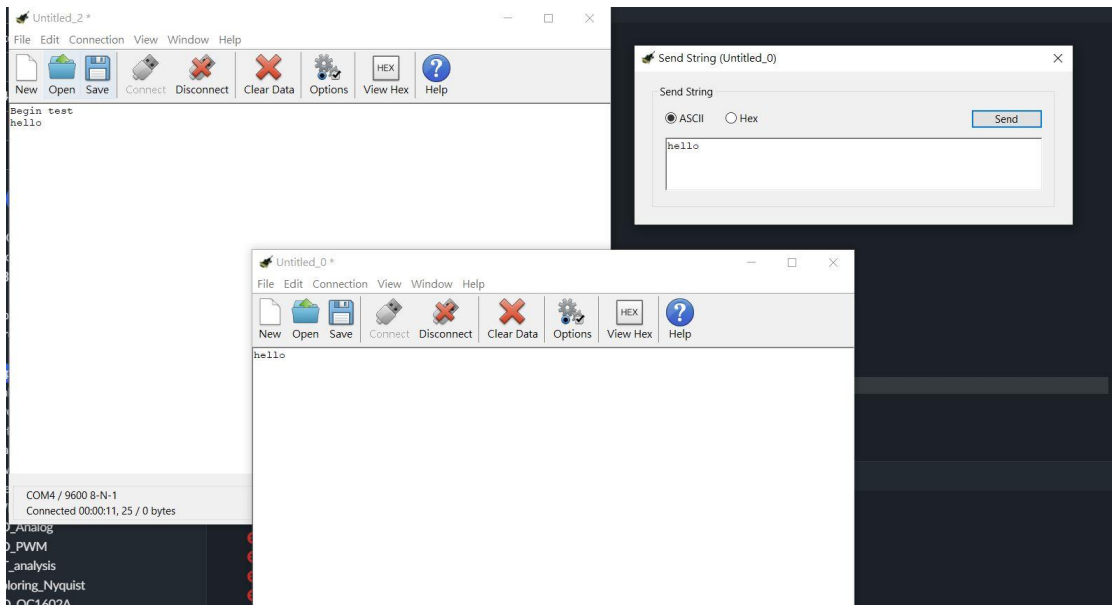
(1) Configure XBee and test

In this part, use coolterm to set 2 Xbee chip, then use mbed to test whether two 2 chips are set properly.

Connected chip A and chip B through usb adapter, test them.



Chip A with USB adapter and chip B connected to mbed.



(2) XBee programming

Next, we send message and get the feedback through both Mbed and Python.

```
Begin test
Send message: 0
Echo back: string=Send message: 0 with strlen=15.
Send message: 1
Echo back: string=Send message: 1 with strlen=15.
Send message: 2
Echo back: string=Send message: 2 with strlen=15.
Send message: 3
Echo back: string=Send message: 3 with strlen=15.
Send message: 4
Echo back: string=Send message: 4 with strlen=15.
```

```

PS D:\embedded system\11_4_XBee_remote> py Xbee_host.py COM10
message: 0
b'Send message: 0\n'
message: 1
b'Send message: 1\n'
message: 2
b'Send message: 2\n'
message: 3
b'Send message: 3\n'
message: 4
b'Send message: 4\n'
PS D:\embedded system\11_4_XBee_remote>

```

(3) With erpc service

In this section, we can see that erpc_service work fine through XBee.

```

Call led_off 2
Call led_off 3      LED 1 is Off.
Call led_on 1       LED 2 is Off.
Call led_on 2       LED 3 is Off.
Call led_on 3       LED 1 is On.
Call led_off 1      LED 2 is On.
Call led_off 2      LED 3 is On.
Call led_on 1       LED 1 is Off.
Call led_on 2       LED 2 is Off.
Call led_on 3       LED 3 is Off.
Call led_off 1      LED 1 is On.
Call led_off 2      LED 2 is On.
Call led_off 3
Call led_on 1
Call led_on 2

```

(4) Discussion

I use Bluetooth on Arduino before, and the configuration looked quite similar to XBee. But XBee is much easier to know how the interface do. With RX, TX I think it is quite similar to UART(Or maybe it is.) Take it as a remote version of UART(?).