

L05 - How to use ZigBee applications in LinPAC?

Applies to:		
Platform	OS version	Kernel version
LinPAC series	LP-51xx: V1.8.3 or later LP-8x3x: V1.2 or later LP-8x4x: V1.16 or later Others: All versions	All versions

ZigBee is a specification based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). ZigBee operates in the ISM radio bands and uses a basic master-slave configuration that is suited to the dynamic mesh networks of many infrequently used devices that talk via small data packets. Up to 256 nodes are allowed.

The LinPAC series supports the ZigBee protocol. We can use the ZT-USBC module as a ZigBee converter that allows the USB interface to be converted to a personal area ZigBee wireless network. Use the LP-8441 with the ZT-USBC module to connect to the ZT-2043 module as an example to demonstrate the application of ZigBee, as illustrated in Figure 1.

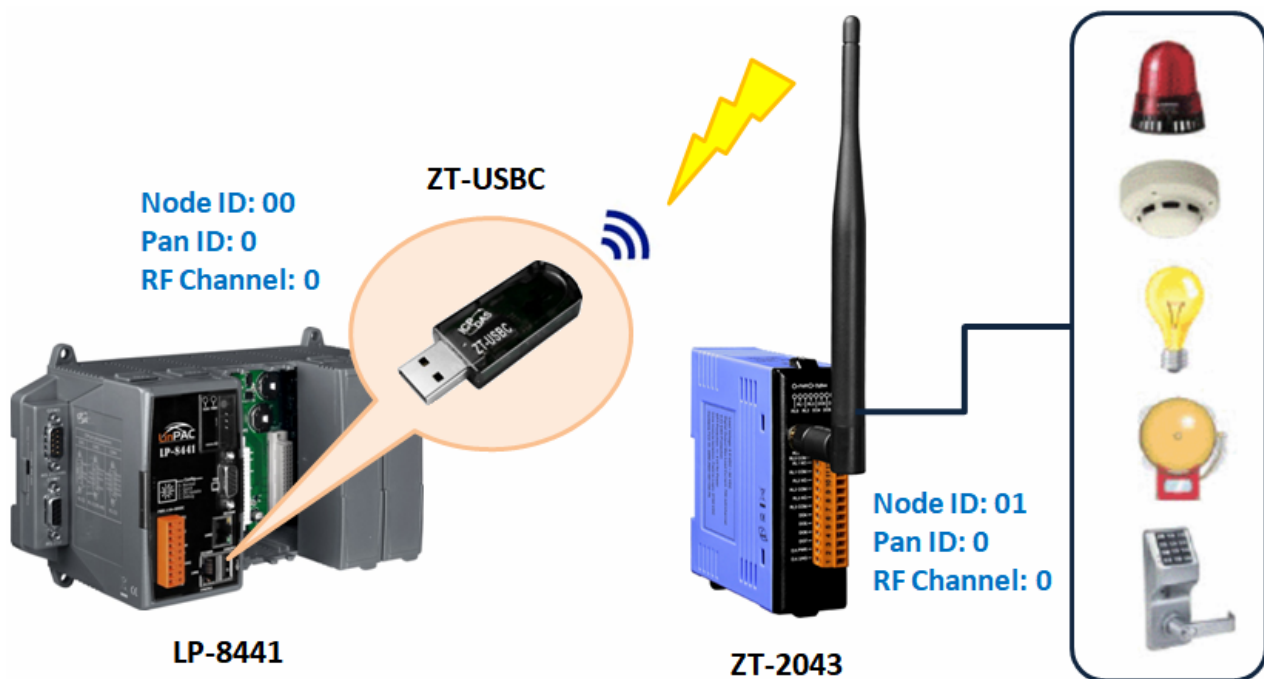


Figure 1. Application of Zigbee Technology using a LinPAC device

Refer to the following websites for more detailed information.

❑ ZT-USBC module:

http://www.icpdas.com/root/product/solutions/industrial_wireless_communication/wireless_solutions/zst-usb.html

❑ ZT-2043 module:

http://www.icpdas.com/root/product/solutions/industrial_wireless_communication/wireless_solutions/zst-2043.html

User can deploy ZigBee on the LinPAC series and configure it with the iztconfig.exe program, as illustrated in Figure 2.

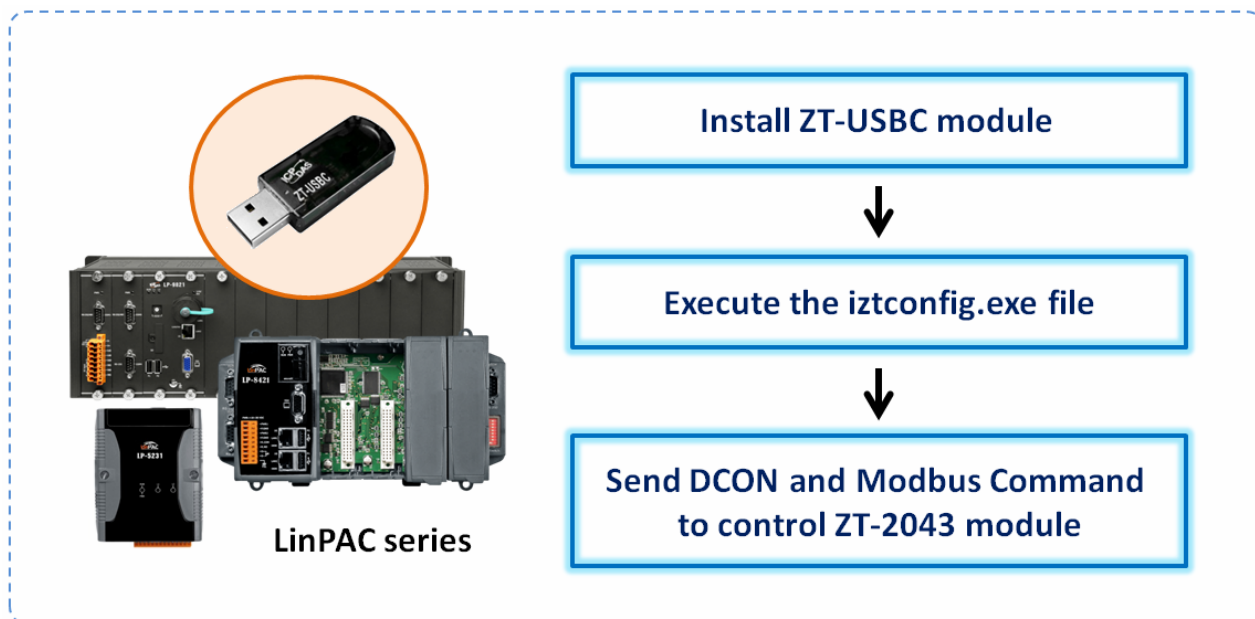


Figure 2. Flowchart illustrating how to configure ZigBee on a LinPAC device

Section 1: Install ZT-USBC module

The ZT-USB module is small-sized wireless ZigBee device that allows you to convert from a USB interface to a ZigBee wireless network. Install the ZT-USBC module on the LP-8441 module following the procedure described below:

Note: Before installing the ZT-USBC module, it is recommended that you first install the DCON Utility and the ZT2000 Configuration Utility on a Windows PC to configure both the ZT-USBC module and set the ZT-2043 module. For more detailed information, refer to ZT-USB series user manual, which you can download from:

http://ftp.icpdas.com.tw/pub/cd/usbcd/napdos/zigbee/zt_series/document/zt-usb_series/zt-usb_series_user_manual_0101_en.pdf

❑ Install the ZT-USBC driver

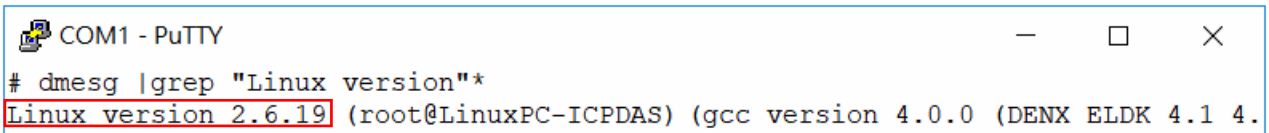
Before installing any drivers, you need to decide what version to use, as indicated in the table below:

Linux PAC(CPU/Linux kernel)		Install Driver	
		usbserial.ko	ftdi_sio.ko
PXA270	Linux Kernel 2.6.19 (LP-51xx/8x4x)	√	√
	Linux Kernel 2.6.29 (LP-51xx/8x3x)	X	√
Am335x	Linux Kernel 3.2 (LP-22xx/52xx/8x2x/9x2x)	√	√
Others	Linux Kernel 3.2	X	√
(X86...)	Linux Kernel 4.14	X	√

Note: √= manually install; X= automatically installed at boot time

Step 1: Use the following command to check the kernel version, as illustrated in Figure 3.

Command: `dmesg |grep "Linux version"`



```
COM1 - PuTTY
# dmesg |grep "Linux version"
Linux version 2.6.19 (root@LinuxPC-ICPDAS) (gcc version 4.0.0 (DENX ELDK 4.1 4.
```

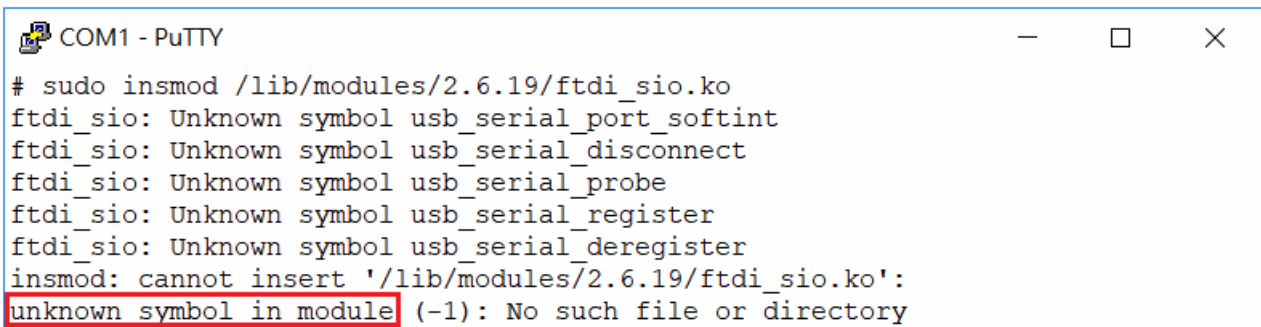
Figure 3. Identifying the Linux Kernel version number

Step 2: Use the following command to load the ZT-USBC driver into the kernel.

Command: `sudo insmod /lib/modules/2.6.19/ftdi_sio.ko`

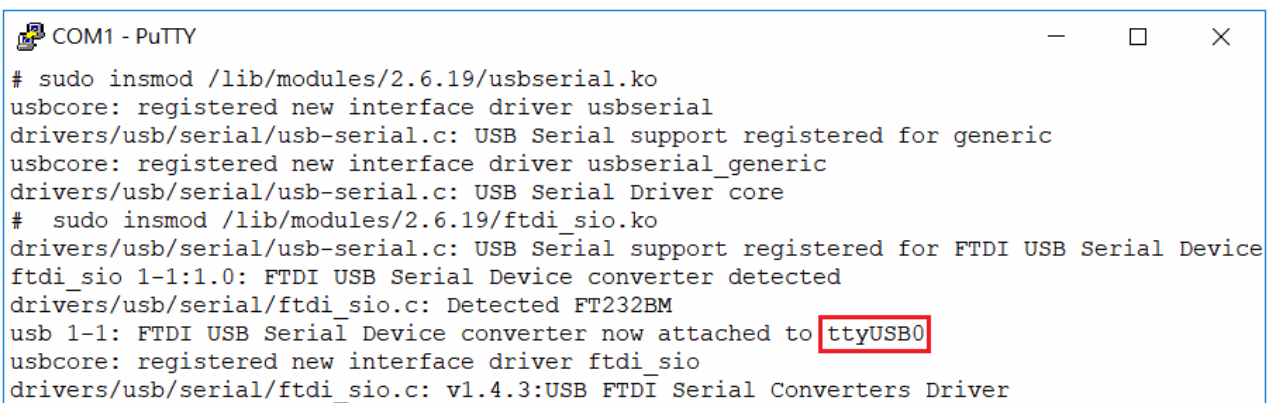
Note: If the message '**Unknown symbol in module**' appears after installing `ftdi_sio.ko` (as illustrated in Figure 4), you need to install both `usbserial.ko` and `ftdi_sio.ko` into the kernel and then locate the USB serial port for the ZT-USBC module, as illustrated in Figure 5.

Command: `sudo insmod /lib/modules/2.6.19/usbserial.ko`
`sudo insmod /lib/modules/2.6.19/ftdi_sio.ko`



```
COM1 - PuTTY
# sudo insmod /lib/modules/2.6.19/ftdi_sio.ko
ftdi_sio: Unknown symbol usb_serial_port_softint
ftdi_sio: Unknown symbol usb_serial_disconnect
ftdi_sio: Unknown symbol usb_serial_probe
ftdi_sio: Unknown symbol usb_serial_register
ftdi_sio: Unknown symbol usb_serial_deregister
insmod: cannot insert '/lib/modules/2.6.19/ftdi_sio.ko':
unknown symbol in module (-1): No such file or directory
```

Figure 4. Failed to install `ftdi_sio.ko`

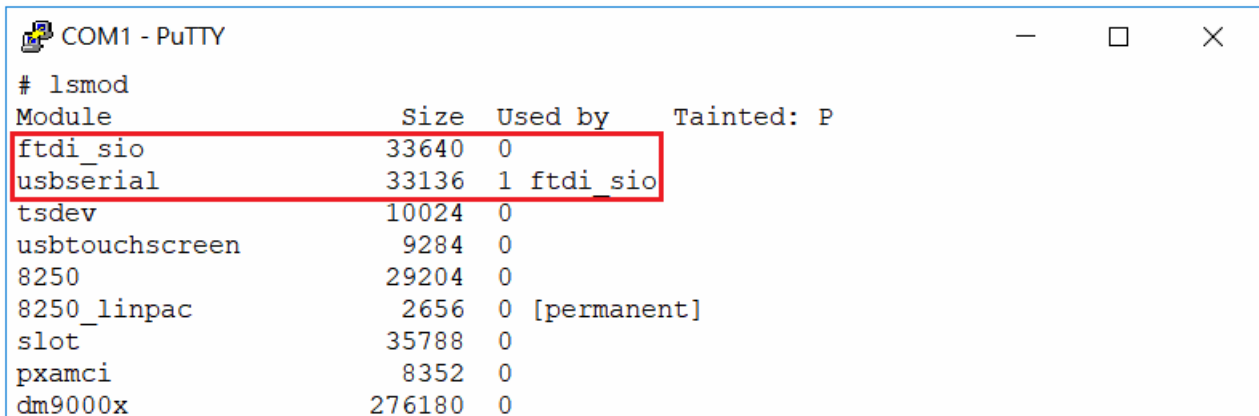


```
COM1 - PuTTY
# sudo insmod /lib/modules/2.6.19/usbserial.ko
usbcore: registered new interface driver usbserial
drivers/usb/serial/usb-serial.c: USB Serial support registered for generic
usbcore: registered new interface driver usbserial_generic
drivers/usb/serial/usb-serial.c: USB Serial Driver core
# sudo insmod /lib/modules/2.6.19/ftdi_sio.ko
drivers/usb/serial/usb-serial.c: USB Serial support registered for FTDI USB Serial Device
ftdi_sio 1-1:1.0: FTDI USB Serial Device converter detected
drivers/usb/serial/ftdi_sio.c: Detected FT232BM
usb 1-1: FTDI USB Serial Device converter now attached to ttyUSB0
usbcore: registered new interface driver ftdi_sio
drivers/usb/serial/ftdi_sio.c: v1.4.3:USB FTDI Serial Converters Driver
```

Figure 5. The default device port for the ZT-USBC module is '`ttyUSB0`'

Step 3: Use the following command to verify that check kernel modules are loaded the ZT-USBC driver, as illustrated in Figure 6.

Command: lsmod

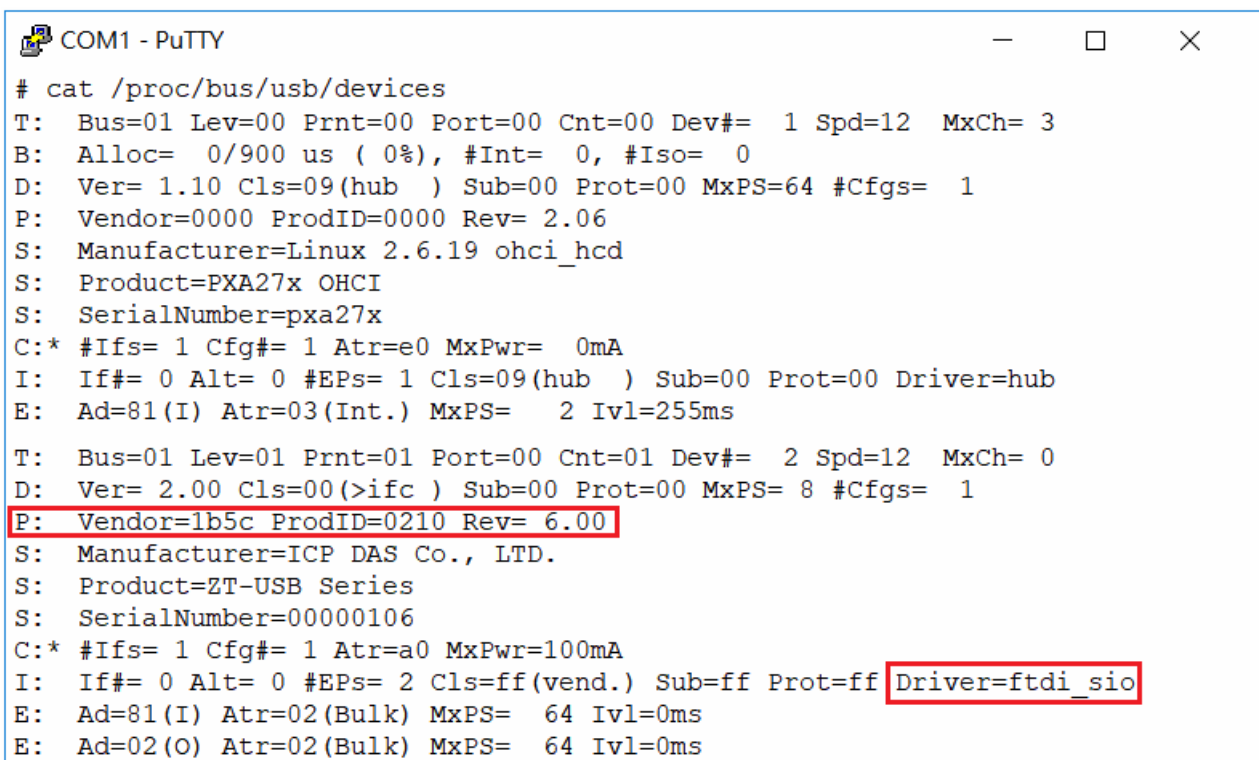


```
COM1 - PuTTY
# lsmod
Module              Size  Used by  Tainted: P
ftdi_sio            33640  0
usbserial           33136  1 ftdi_sio
tsdev               10024  0
usbtouchscreen      9284   0
8250                29204  0
8250_linpac         2656   0 [permanent]
slot                35788  0
pxamci              8352   0
dm9000x            276180 0
```

Figure 6. Display a list of loaded modules in the kernel

Step 4: Use the following command to verify that the Product ID and Vendor ID for the ZT-USBC device are written into the driver module, as illustrated in Figure 7.

Command: cat /proc/bus/usb/devices



```
COM1 - PuTTY
# cat /proc/bus/usb/devices
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 3
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 2.06
S: Manufacturer=Linux 2.6.19 ohci_hcd
S: Product=PXA27x OHCI
S: SerialNumber=pxa27x
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 2 Iv1=255ms

T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 2 Spd=12 MxCh= 0
D: Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=1b5c ProdID=0210 Rev= 6.00
S: Manufacturer=ICP DAS Co., LTD.
S: Product=ZT-USB Series
S: SerialNumber=00000106
C:* #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr=100mA
I: If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=ftdi_sio
E: Ad=81(I) Atr=02(Bulk) MxPS= 64 Iv1=0ms
E: Ad=02(O) Atr=02(Bulk) MxPS= 64 Iv1=0ms
```

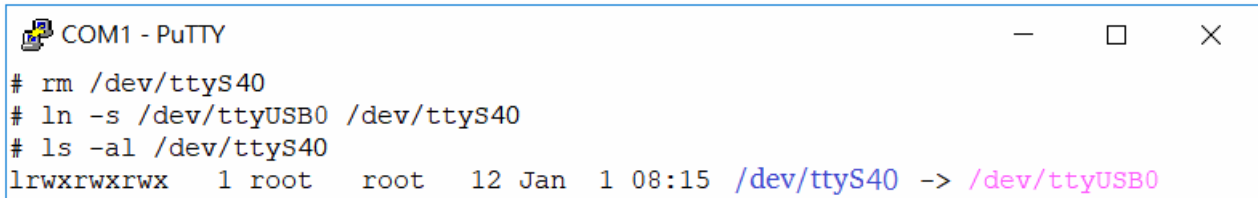
Figure 7. Verifying that a USB device is present

Note: If the version number for the LinPAC kernel is not **2.6.19**, you need to execute the command below to add the "Product ID" and "Vendor ID" once these steps are complete.

Command: echo "1b5c 0210" > /sys/bus/usb-serial/drivers/ftdi_sio/new_id

❑ Creating a symlink to the USB converter

To create a soft link to the device node of the ZT-USBC(/dev/ttyUSB0). For example, in order to communicate with the device via COM42 using the LinPAC SDK, you can remove /dev/ttyS40 and create a soft link for /dev/ttyUSB0 to /dev/ttyS40, as illustrated in Figure 8.



```
COM1 - PuTTY
# rm /dev/ttyS40
# ln -s /dev/ttyUSB0 /dev/ttyS40
# ls -al /dev/ttyS40
lrwxrwxrwx 1 root root 12 Jan 1 08:15 /dev/ttyS40 -> /dev/ttyUSB0
```

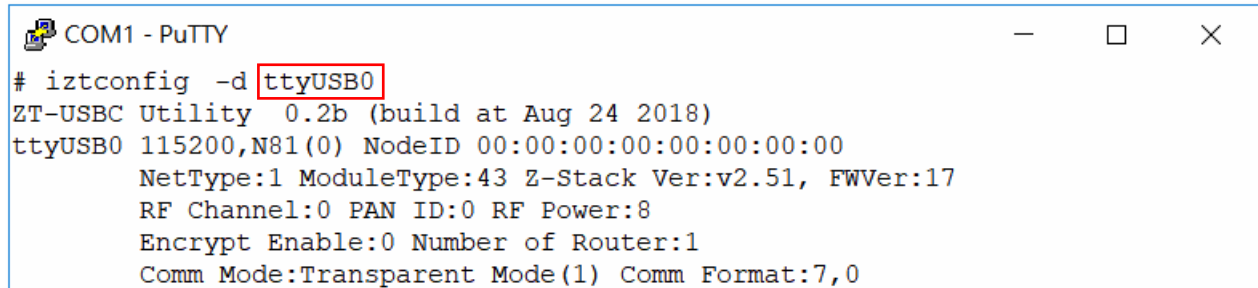
Figure 8. Creating a symlink between the ttyS40 and ttyUSB0

Section 2: Execute the iztconfig.exe

The iztconfig.exe demo program illustrates how to configure the ZigBee wireless network. In this demo program, you set up the ZigBee wireless network in the LinPAC, which is described below:

- ❑ Use the following command to list the ZigBee configuration for the ZT-USBC module, as illustrated in Figure 9.

Command: iztconfig.exe -d <USB device>

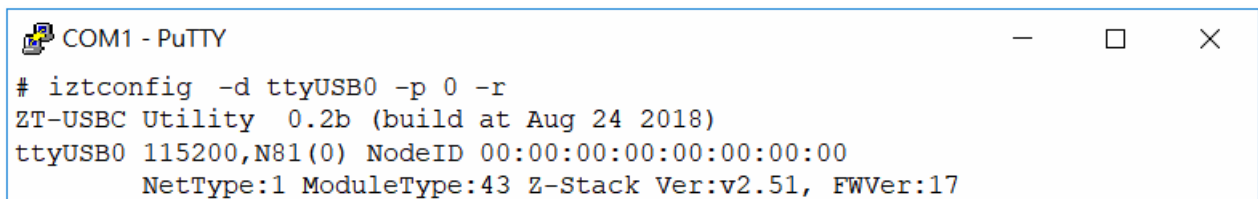


```
COM1 - PuTTY
# iztconfig -d ttyUSB0
ZT-USBC Utility 0.2b (build at Aug 24 2018)
ttyUSB0 115200,N81(0) NodeID 00:00:00:00:00:00:00:00
NetType:1 ModuleType:43 Z-Stack Ver:v2.51, FWVer:17
RF Channel:0 PAN ID:0 RF Power:8
Encrypt Enable:0 Number of Router:1
Comm Mode:Transparent Mode(1) Comm Format:7,0
```

Figure 9. List the configuration for the ZT-USBC module using the iztconfig.exe command

- ❑ Use the following command to modify the communication parameters for the ZigBee, as illustrated in Figure 10.

Command: iztconfig.exe -d <USB device> -p <PAN ID> -r



```
COM1 - PuTTY
# iztconfig -d ttyUSB0 -p 0 -r
ZT-USBC Utility 0.2b (build at Aug 24 2018)
ttyUSB0 115200,N81(0) NodeID 00:00:00:00:00:00:00:00
NetType:1 ModuleType:43 Z-Stack Ver:v2.51, FWVer:17
```

Figure 10. Setting the PAN ID to 0 for the ZT-USBC module

NOTE: You need to use the '-r' parameter to restart the ZT-USBC module to ensure that the module is set successfully.

Section 3: Send DCON and Modbus command to control ZT-2043 module

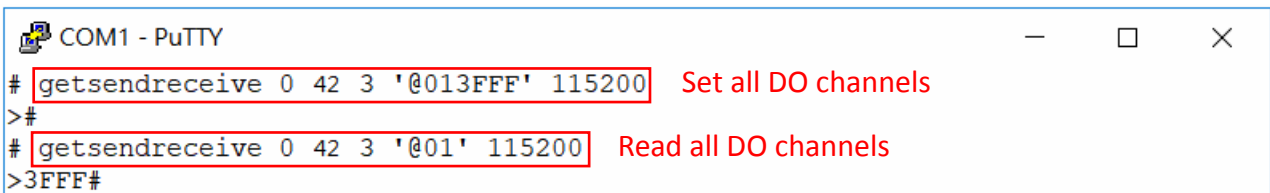
The ZT-2043 offers 14 sink type Digital Output channels. To use the ZT-USBC module communicate with the ZT-2043 module, the 'Pan ID' and 'RF Channel' must be the same if they are on the same ZigBee network. The ZT-2043 module will then send a request to join the ZigBee wireless network and start working.

NOTE: The configuration for the ZT-2043 module can be adjusted by using both the external rotary switch and the DIP switches. After completing the configuration, you must power off and power on the ZT-2043 module to ensure that the configuration is successful.

❑ Send a DCON command to the ZT-2043 module

Use the following command to send a DCON command to set or read the DO channel status on the ZT-2043 module, as illustrated in Figure 11.

Command: `getsendreceive <slot> <comport> <timeout> <command> <baudrate>`



```
COM1 - PuTTY
# getsendreceive 0 42 3 '@013FFF' 115200 Set all DO channels
>#
# getsendreceive 0 42 3 '@01' 115200 Read all DO channels
>3FFF#
```

Figure 11. Sending a DCON command to control the ZT-2043 module

❑ Send Modbus command to ZT-2043 module

(1) Modify the 'setmodbus.c' and 'getmodbus.c' demo files, which can be found in the examples folder provided with the LinPAC SDK.

There two parts of the program will need to be modified and recompiled, as illustrated below:

➤ setmodbus.c

```
switch(function) {
    case FC05ForceSingleCoil:
    case FC15ForceMultipleCoil:
        szBuf[0] = wValue & 0xff;
        szBuf[1] = (wValue >> 8) & 0xff;
        break;
    case FC06PresetSingleRegister:
    case FC16PresetMultipleRegister:
        szBuf[0] = (wValue >> 8) & 0xff;
        szBuf[1] = wValue & 0xff;
        break;
    default:
        usage();
        return FAILURE;
}
```

➤ **getmodbus.c**

```
if (RetVal == 0) {
    if((function==1) || (function==2)){
        //printf("%d", szBuf[0]);
        RetVal = (szBuf[0]|szBuf[1]<<8);
        printf("%ld", RetVal);
    }
    else {
        RetVal = (szBuf[0]<<8) |szBuf[1];
        printf("%ld", RetVal);
    }
} else if (RetVal == CmdError) {
    printf("Module at COM%d return error code %d !!!\n",comport,szBuf[0]);
    return FAILURE;
} else {
    printf("Module at COM%d error %d !!!\n",comport,RetVal);
    return FAILURE;
}
```

(2) Execute the demo programs to control the ZT-2043 module, as illustrated in Figure 12.

Use the following command to send a Modbus command to set the status of the DO channel on the ZT-2043 module.

Command: ./setmodbus <comport> <baudrate> <netid> <command> <addr>
<count> <value> <timeout>

Use the following command to send a Modbus command to read the status of the DO channel on the ZT-2043 module.

Command: ./getmodbus <comport> <baudrate> <netid> <command> <addr>
<count> <timeout>

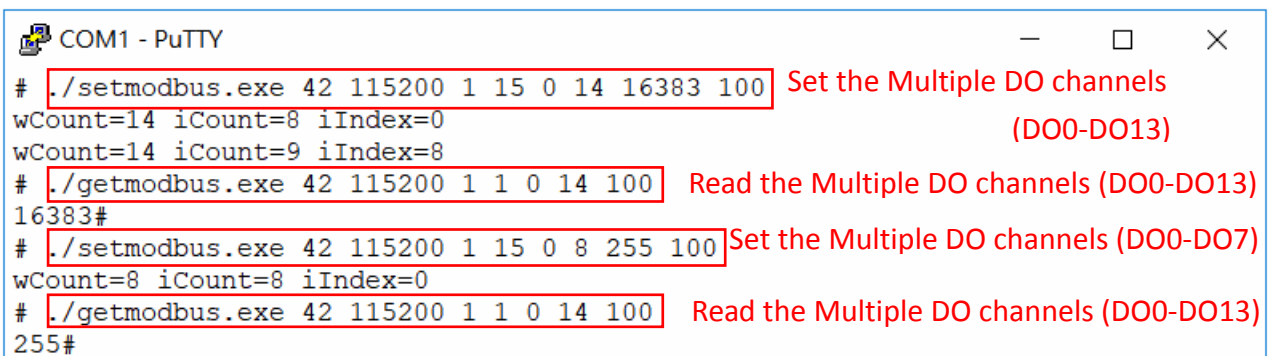


Figure 12. Sending Modbus command to control the ZT-2043 module

For more detailed information about the DCON and Modbus protocol, refer to Chapter 4 of the ZT-2000 DIO Series user manual, which you can download from:

http://ftp.icpdas.com.tw/pub/cd/usbcd/napdos/ZigBee/zt_series/document/zt_dio/zt-2000_dio_usermanual_0121_en.pdf