

Zynq 7000裸机的lwip 样例程序echo server 实验

原创

leon_zeng0 于 2017-10-12 00:15:14 发布 21740 收藏 57

分类专栏: [c++](#) [fpga](#) [arm](#) 文章标签: [lwip](#) [zynq](#) [7000](#) [echo](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/leon_zeng0/article/details/78210022

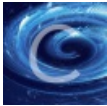
版权



[c++](#) 同时被 3 个专栏收录

74 篇文章 4 订阅

订阅专栏



[fpga](#)

49 篇文章 13 订阅

订阅专栏



[arm](#)

20 篇文章 0 订阅

订阅专栏

本文介绍开源tcp/ip协议栈lwip在 Zynq 7000裸机(无操作系统)上的应用。分为4部分, 硬件的设计, 软件的设计, 测试, 简单代码解释。

本文是在Alinx黑金Zynq 7000开发平台配套教程的学习过程中产生的。硬件平台是黑金的AX7010/ax7020,核心板AC7010/AC7020, 其他类似平台应该也可以。

预备知识是你先学会了helloWorld实验。如果你还没有学会helloworld实验, 那你看我的另一篇文章helloworld 实验, 或者你学习学习板提供的helloworld实验, 那应该更接近你的实际硬件。

工程文件下载处: <http://pan.baidu.com/s/1pKEqsS7>

如果想学习怎么pc端编程, 可以看我的另一个文章: [MFC 下的网络编程socket编程](#)

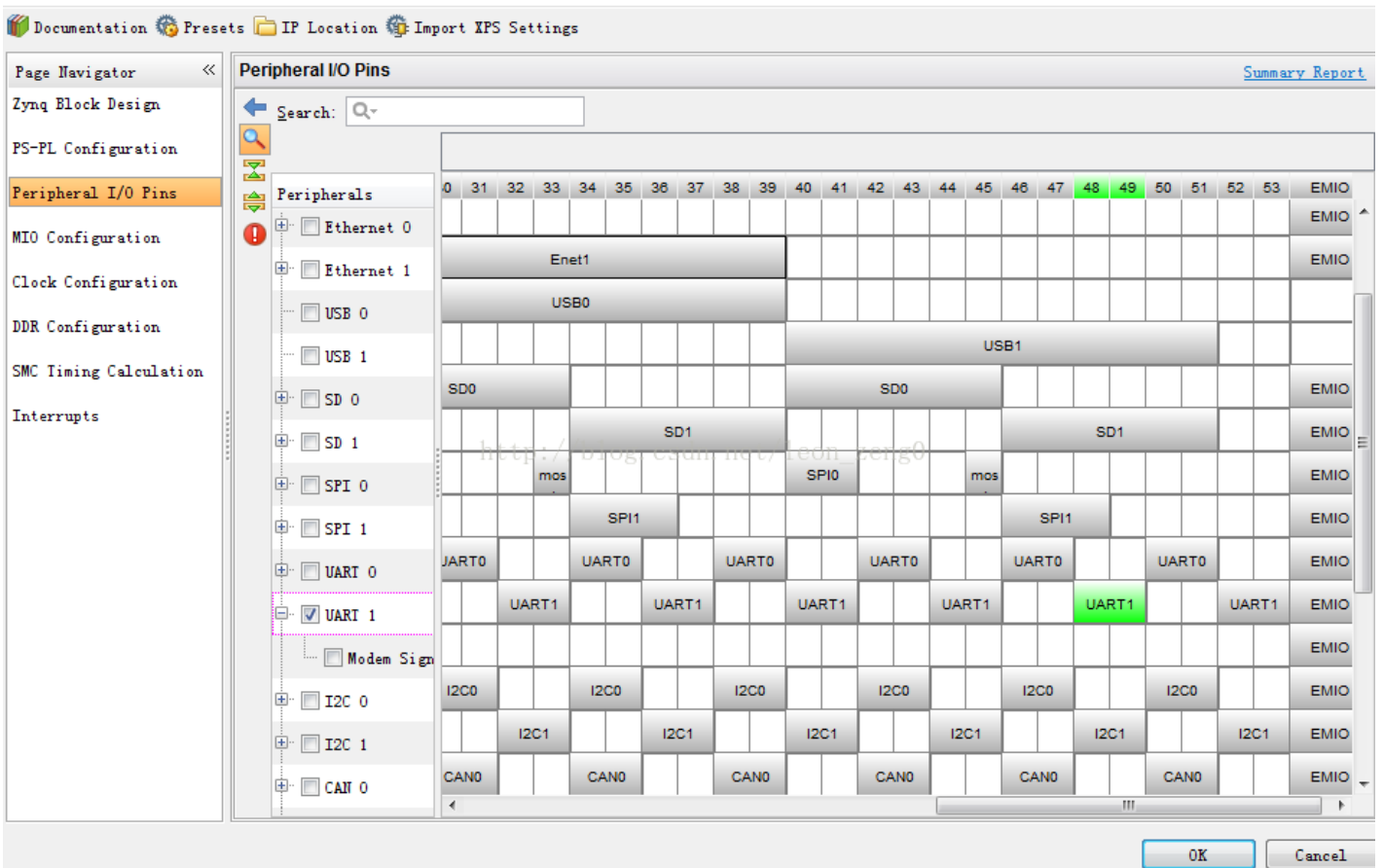
1: 硬件设计

Vivado工程创建: 与helloworld 工程的创建一样, 只是工程名为lwip.

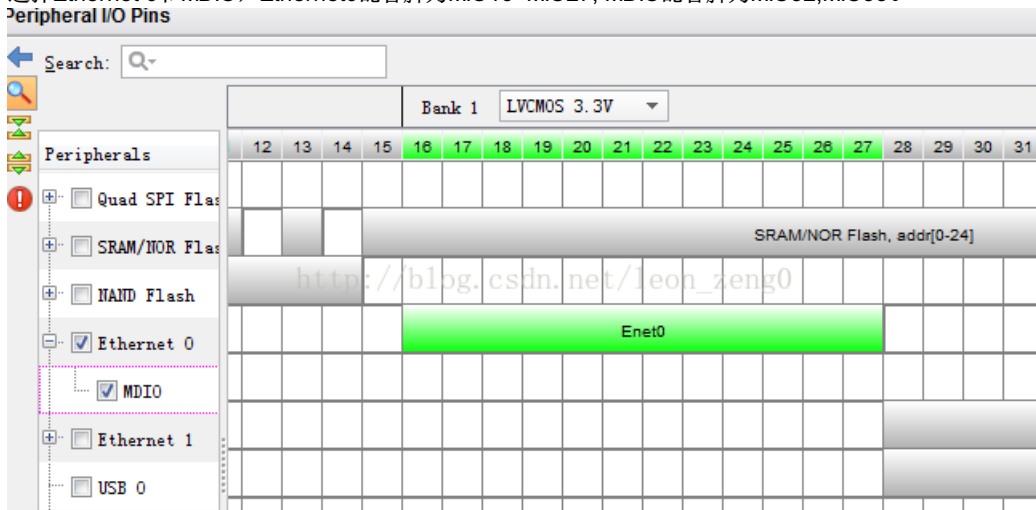
Zynq 的配置: 与helloworld 工程的配置基本一样, 唯一不同的是需要选择Ethernet.

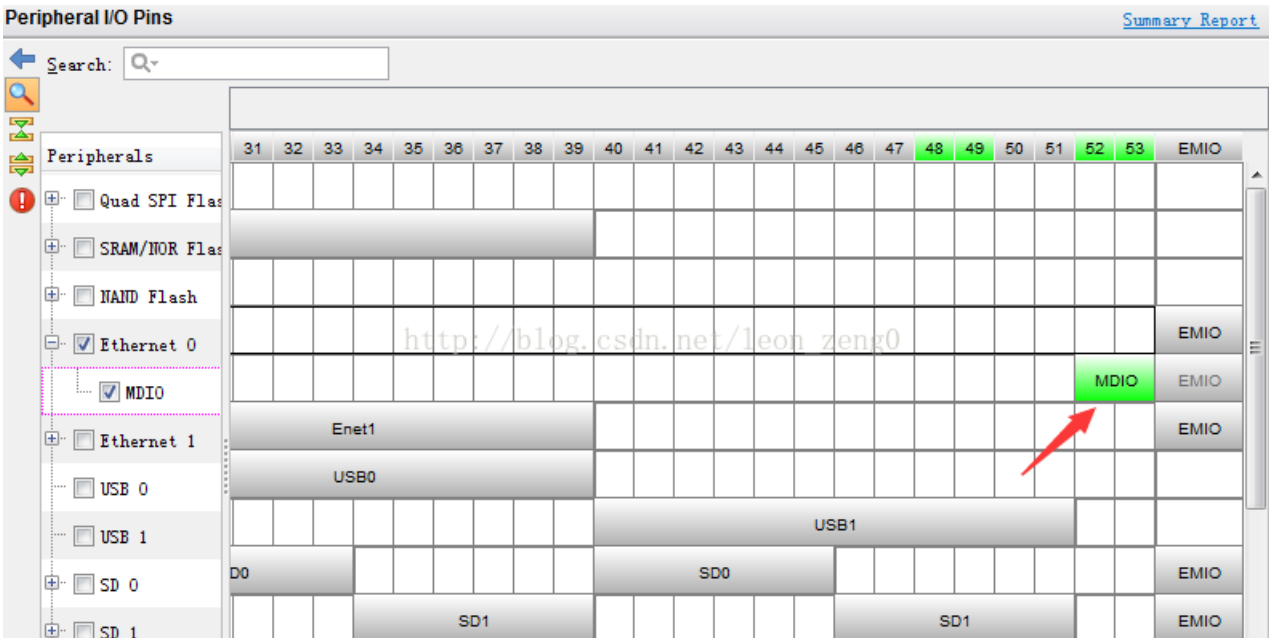
下面的配置是针对黑金的学习板AX7010/AX7020 或者核心板AC7010/AC7020, 如果你的硬件网络口与此不一样, 那你就需要查看并做相应改变。

串口的配置是一样的:

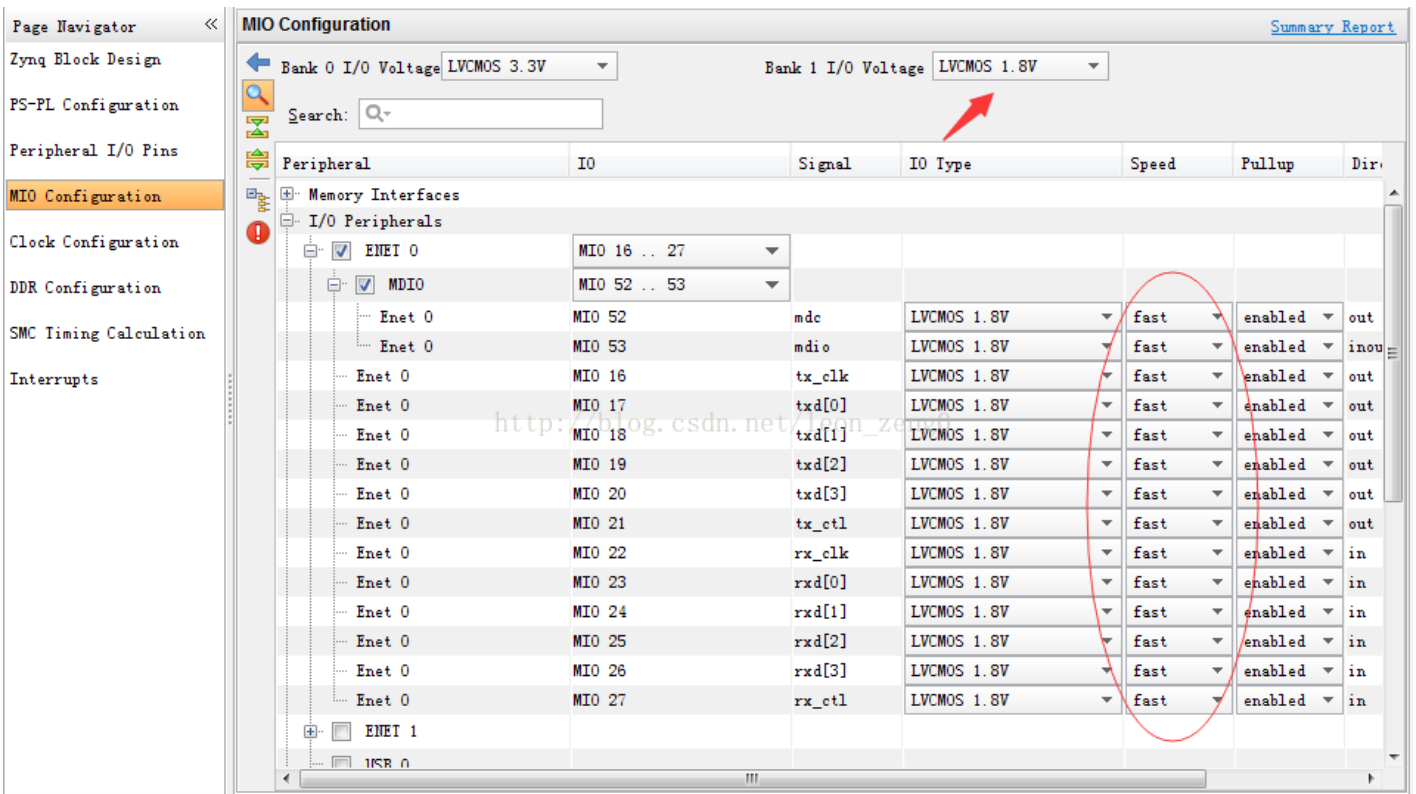


选择Ethernet 0和MDIO，Ethernet0的管脚为MIO16~MIO27，MDIO的管脚为MIO52,MIO53。





选择Ethernet 0的接口信号管脚的速度都为"fast"，电压为LVCMOS1.8V。参见黑金学习板教程1的P287



这就是所有的硬件设计要点，与helloworld一样，选择菜单File->Export->Export Hardware，你就完成硬件导入SDK。

我第一次操作的时候，没有完全照helloworld实验做，结果中间报错了，注意教程1的P92写了：

因为本实验中没有用到PL部分外设，所以就没有PL的外设需要挂到AXI的总线上，也就不需要AXI GP0的Master总线了，这里我们把FCLK_RESET0_N和M AXI GP0 interface的选项去掉。

2: 软件设计

我开始做实验时，也许测试方法不对，结果老不通，黑金说：

黑金修改后的lwip库，做lwip实验时可以替换掉，解决无法识别连接速度的问题。

怎么替换呢？

就是用他提供的lwip141_v1_3\src

C:\Xilinx\SDK\2015.4\data\embeddedsw\ThirdParty\sw_services\lwip141_v1_3\src

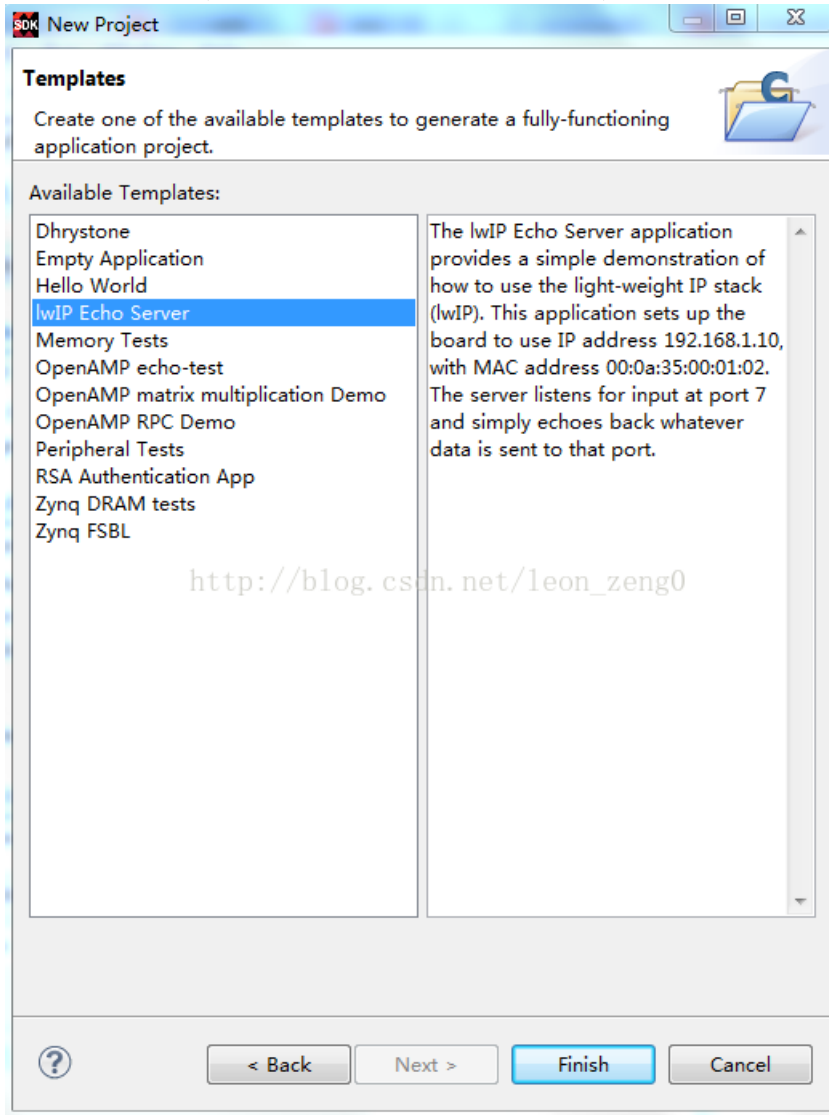
如果你没有这个，也可以用我提供的项目里的lwip\lwip.sdk\lwip_test_bsp\ps7_cortexa9_0\libsrc\lwip141_v1_3\src

做了这个替换后，Launch SDK, 就进入软件设计界面。

和helloWorld 实验一样，File ->New -> Application Project

这里我取的工程名是lwip_test

重要的是下面的Next, 在Next界面里，选择的不是Hello World, 而是 lwIP Echo Server



像hello World 实验一样，你不用做任何修改，软件设计就完成了。

Build Project 然后 Debug As->Launch on hardware 或者 Run As->

就进入了测试阶段

3: 测试

其实硬件设计，软件设计都是很简单，但我在测试上花了好多天时间。因为测试不出结果，所以自然会认为软硬件设计哪里有问题。不断地去修改，查找软硬件的问题。

和hello world 实验一样，程序运行就会在终端上显示很多信息：

```
Serial-COM3 - SecureCRT
File Edit View Options Transfer Script Tools Help
Serial-COM3
-----lwIP TCP echo server -----
TCP packets sent to port 6001 will be echoed back
WARNING: Not a Marvell or TI Ethernet PHY. Please verify the initialization sequence
Start PHY autonegotiation
waiting for PHY to complete autonegotiation.
autonegotiation complete
link speed for phy address 3:/100log.csdn.net/leon_zeng0
DHCP Timeout
Configuring default IP of 192.168.1.10
Board IP: 192.168.1.10
Netmask : 255.255.255.0
Gateway : 192.168.1.1
TCP echo server started @ port 7
Ready Serial: COM3 17, 1 24 Rows, 80 Cols VT100 NUM
```

我的程序第一次运行就是这个结果，也许第一次的时候，软硬件设计就是对的。但是本实验的核心不是显示这些简单信息，而是要echo, 要看到回声。

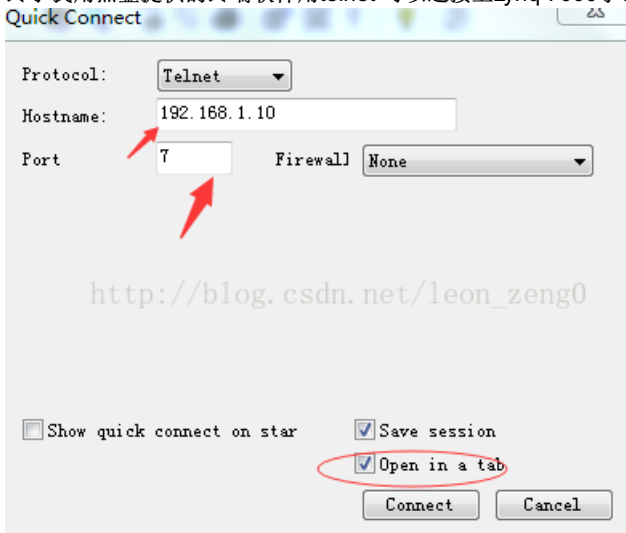
很自然，我开始就用黑金提供的网络助手做连接测试，可总是连接不上。这个不成功的问题太多，网线，网卡设置，软硬件，测试软件。当然网卡需要设置ip4 为静态网址，我用的是直连线，我当时认为比较简单，不需要hub。

我估计用hub，而不是直连线可能容易成功一点。程序里如果能够自动获取，那么zynq 7000会优先选用自动获取的ip地址，当然会显示处理。这样你的本机就容易和他在一个网络里。

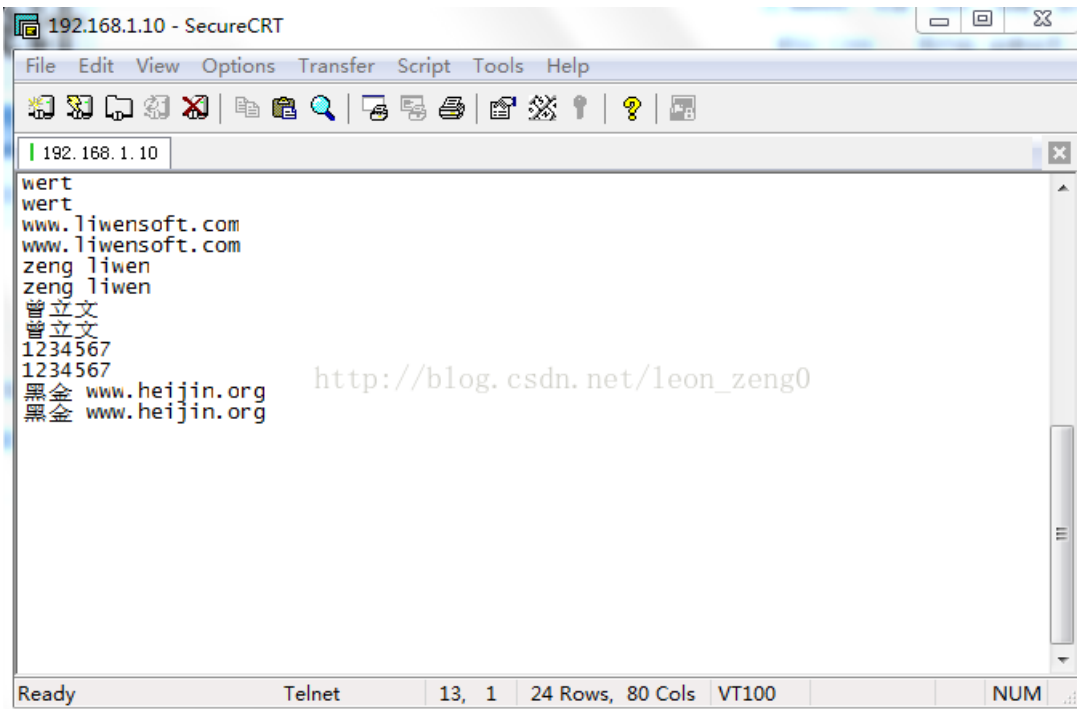
设置好后，我用ping 测试是成功的，但就是不能连接上。

我不再相信网络助手，在网上找了tcp socket 通讯的例子做测试，有个例子只是server，要求我用telnet 连接做测试。

终于我用黑金提供的终端软件用telnet 可以连接上zynq 7000了。其连接测试是这样的：



连接上之后，我就看到回声了，压抑我十多天的痛苦终于过去了。



4: 简单的代码解释

这个解说大部分来自xilinx xapp1026.pdf

这里主要有2个程序文件main.c echo.c

main.c 的主循环程序是:

```
/* receive and process packets */
while (1) {
    if (TcpFastTmrFlag) {
        tcp_fasttmr();
        TcpFastTmrFlag = 0;
    }
    if (TcpSlowTmrFlag) {
        tcp_slowtmr();
        TcpSlowTmrFlag = 0;
    }
    xemacif_input(echo_netif);
    transfer_data();
}
```

TcpFastTmrFlag 和 TcpSlwTmrFlag 是TCP TX中断所必需的, 每隔250ms 和 500ms 定时器中断里设定的。
应用循环的功能是不断接受包(xemacif_input) 然后传送给lwip.

在进入这个循环前, echo Server 设置必要的回调函数。

这些设置就在echo.c 里的start_application函数里。

recv_callback是数据接收中断程序。

accept_callback是连接建立接收中断程序。

start_application 解释如下:

```
/* create new TCP PCB structure 建立新的TCP 的PCB结构*/
pcb = tcp_new();
if (!pcb) {
    xil_printf("Error creating PCB. Out of Memory\n\r");
    return -1;
}

/* bind to specified @port 绑定在端口port */
err = tcp_bind(pcb, IP_ADDR_ANY, port);
if (err != ERR_OK) {
```

```
xil_printf("Unable to bind to port %d: err = %d\n\r", port, err);
return -2;
}

/* we do not need any arguments to callback functions */
tcp_arg(pcb, NULL);

/* listen for connections 侦探连接 */
pcb = tcp_listen(pcb);
if (!pcb) {
    xil_printf("Out of memory while tcp_listen\n\r");
    return -3;
}

/* specify callback to use for incoming connections 设定输入连接中断函数 */
tcp_accept(pcb, accept_callback);
```

而接收数据的中断函数是在accept_callback里设定的。

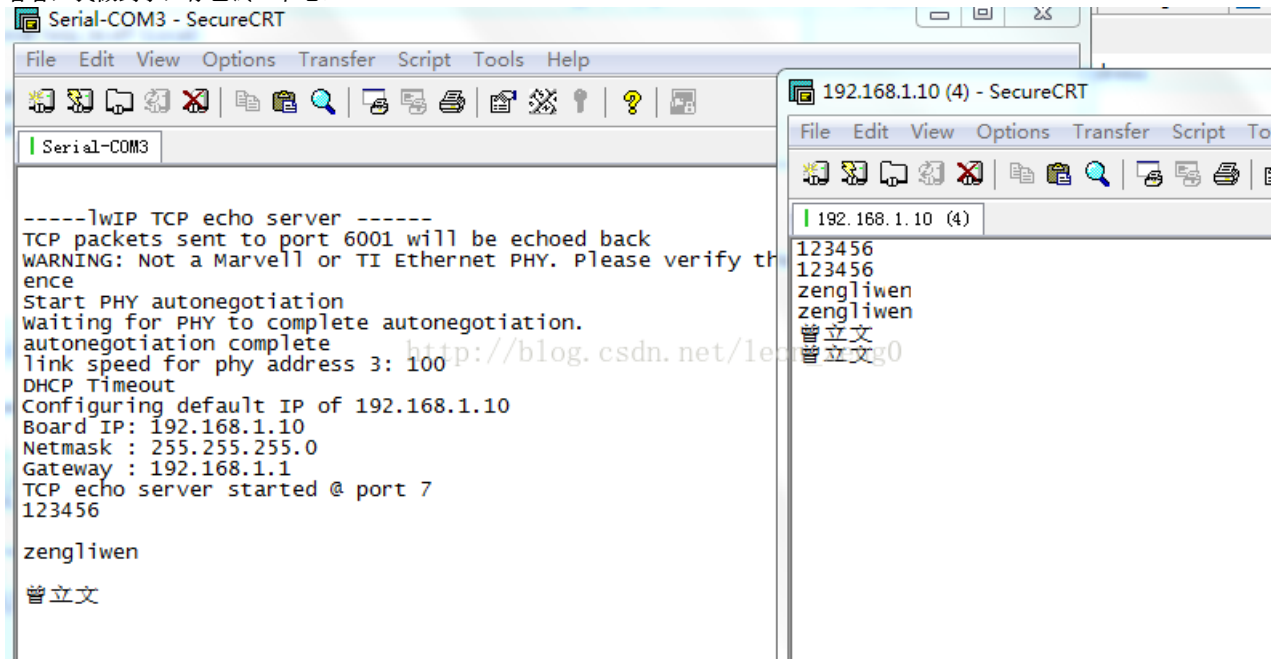
```
/* set the receive callback for this connection */
tcp_recv(newpcb, recv_callback);
```

5: 对软件做点修改

是不是应该对软件做点什么修改，什么也不做，有点不好意思。

到底zynq7000收到数据了吗？我有种怀疑态度，所以我想让监控的终端上显示我输入的字符串，而不仅仅是echo。

看看，我做到了，你也试一下吧。



6: 后续

echo server 只是lwip应用的一个开始，看看xapp1026, lwip实验还有

Web Server

TFTP Server

TCP RX 速度测试

TCP TX 速度测试

当然我们最大的目的是实际应用