Mango24R2 Auto test_ethernet

http://www.mangoboard.com/
http://cafe.naver.com/embeddedcrazyboys
Crazy Embedded Laboratory

Document History

Revision	Date	Change note

1. Auto test_ethernet

1.1. 이더넷 테스트

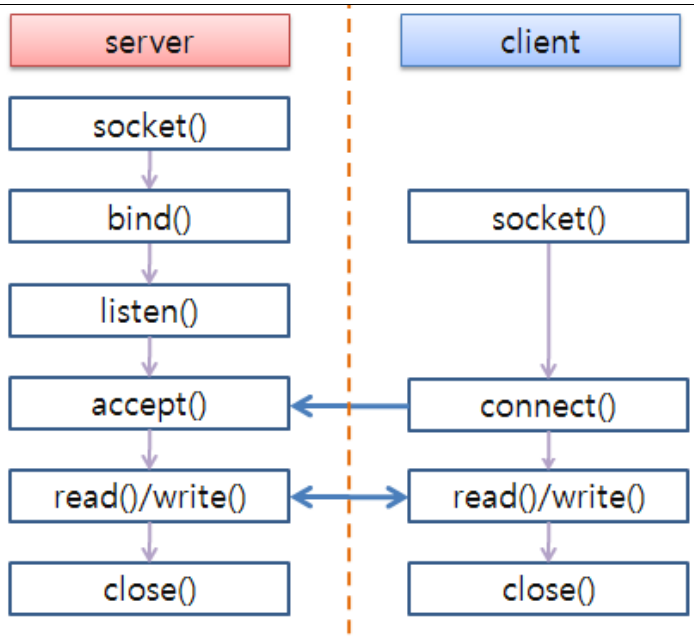
테스트용 지그를 만들 것 입니다.

정상보드 한대와 테스트 할 보드 하나를 준비합니다.

정상보드는 서버가 되고 테스트 할 보드는 클라이언트가 됩니다.

서버_클라이언트 환경은 아래와 같은 과정을 거치게 됩니다.

```
Socket 생성 -> Socket 에 이름연결 (bind)
    -> 클라이언트의 연결을 기다림(listen)
    -> 클라이언트를 받아들임 (Accept)
    -> 클라이언트의 명령을 받아서 적절한 서비스를 수행
```



이더넷 테스트 용 소스를 간단하게 보도록 하겠습니다.

vi socket-server.c

```
18 #include <stdlib.h>
 19 #include <sys/socket.h>
 20 #include <netinet/in.h>
 21 #include <arpa/inet.h>
 22 #include <stdio.h>
 23 #include <stdlib.h>
 24 #include <unistd.h>
 25 #include <errno.h>
```

```
26 #include <string.h>
27 #include <sys/types.h>
28 #include <time.h>
29
30 int main(int argc, char *argv[])
31 {
32         int listenfd = 0, connfd = 0;
33         struct sockaddr_in serv_addr;
34
35         char sendBuff[1025];
36         time_t ticks;
37
38         listenfd = socket(AF_INET, SOCK_STREAM, 0);
39         memset(&serv_addr, '0', sizeof(serv_addr));
40         memset(sendBuff, '0', sizeof(sendBuff));
41
42         serv_addr.sin_family = AF_INET;
43         serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
44         serv_addr.sin_port = htons(5000);
45
46         bind(listenfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr));
47
48         listen(listenfd, 10);
49
50         while(1)
51         {
52                 connfd = accept(listenfd, (struct sockaddr*)NULL, NULL);
53
54                 ticks = time(NULL);
55                 sprintf(sendBuff,"OK");
56                 write(connfd, sendBuff, strlen(sendBuff));
57
58                 close(connfd);
59                 sleep(1);
60          }
61 }
62
```

서버의 소스입니다.
클라이언트가 접속을 하면 "OK"를 클라이언트에 Send하고 접속을 끊습니다.

vi socket-client.c

```
18 #include <stdlib.h>
 19 #include <sys/socket.h>
 20 #include <sys/types.h>
 21 #include <netinet/in.h>
 22 #include <netdb.h>
 23 #include <stdio.h>
 24 #include <string.h>
 25 #include <stdlib.h>
 26 #include <unistd.h>
 27 #include <errno.h>
 28 #include <arpa/inet.h>
 29
 30 #define RESULT_PATH "/root/result"
 31
 32 int main(int argc, char *argv[])
 33 {
 34        int sockfd = 0, n = 0;
 35        char recvBuff[1024];
 36        char buf[255];
 37        struct sockaddr_in serv_addr;
 38
 39        if(argc != 2)
 40        {
 41              printf("₩n Usage: %s <ip of server> ₩n",argv[0]);
 42              return 1;
 43        }
 44
 45        memset(recvBuff, '0',sizeof(recvBuff));
 46        if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
 47        {
 48              printf("₩n Error : Could not create socket ₩n");
 49              return 1;
 50        }
```

```
51
52      memset(&serv_addr, '0', sizeof(serv_addr));
53
54      serv_addr.sin_family = AF_INET;
55      serv_addr.sin_port = htons(5000);
56
57      if(inet_pton(AF_INET, argv[1], &serv_addr.sin_addr)<=0)
58      {
59            printf("₩n inet_pton error occured₩n");
60            return 1;
61      }
62
63      if( connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
 64      {
65          printf("₩n Error : Connect Failed ₩n");
66          return 1;
67      }
68
69      while ( (n = read(sockfd, recvBuff, sizeof(recvBuff)-1)) > 0)
70      {
71            recvBuff[n] = 0;
72            memset(buf,'0',sizeof(buf));
73            sprintf(buf,"echo %s > %s/lan_result",recvBuff,RESULT_PATH);
74            printf("client : %s₩n",buf);
75            system(buf);
76            if(fputs(recvBuff, stdout) == EOF)
77            {
78                  printf("₩n Error : Fputs error₩n");
79            }
80      }
81
82
83
84
85
86      if(n < 0)
87      {
88            printf("₩n Read error ₩n");
```

```
 89        }
 90
 91        return 0;
 92 }
```

클라이언트는 서버에 접속하고 서버에서 send한 만큼 recv합니다.

그 후 지정된 경로에 서버에서 보내온 메시지를 저장합니다.

Makefile를 만들어서 컴파일 합니다.

```
vi Makefile
```

```
1
  2 #export PATH=/opt/arm-2009q3/bin:$PATH
  3
  4 CC = /opt/arm-2009q3/bin/arm-none-linux-gnueabi-gcc
  5
  6 all: server client
  7
  8 clean:
  9        rm -f *.o server client
 10
 11 server: socket-server.o
 12        $(CC) $^ -o $@
 13
 14 client: socket-client.o
 15        $(CC) $^ -o $@
```

make all 명령어를 사용하면 아래와 같이 출력됩니다.

```
/lan_test$ make all
/opt/arm-2009q3/bin/arm-none-linux-gnueabi-gcc       -c -o socket-server.o socket-server.c
/opt/arm-2009q3/bin/arm-none-linux-gnueabi-gcc socket-server.o -o server
/opt/arm-2009q3/bin/arm-none-linux-gnueabi-gcc       -c -o socket-client.o socket-client.c
/opt/arm-2009q3/bin/arm-none-linux-gnueabi-gcc socket-client.o -o client
```

make 안 만들고 컴파일 하여도 괜찮습니다.

1.2. 테스트용

rootfs.tar의 압축을 풀고 root안에 컴파일한 server와 client를 cp합니다.

```
mkdir rootfs
```

```
tar xf rootfs.tar -C rootfs
cp client server image/rootfs/root
```

아래와 같이 스크립트를 추가합니다.

/rootfs/root$ vi ethernet_test.sh

```
#!/bin/sh
  export PATH=/usr/bin/:/sbin:$PATH

    RESULT_DIR=/root/result/
    LAN_RESULT=lan_result

    echo "rm -rf $RESULT_DIR"
    rm -rf $RESULT_DIR

  echo "mkdir $RESULT_DIR"
  mkdir $RESULT_DIR

  echo "cd    $RESULT_DIR"
  cd    $RESULT_DIR

  ifconfig eth0 192.168.10.2


  RTY=5
  echo "[TEST] LAN TEST ....."
  while [ $RTY != 0 ]
  do
  /root/client 192.168.10.1
  RTY=$(($RTY-1))
  done

  echo "[TEST_result] LAN Result ....."
  RESULT=`cat $LAN_RESULT`
  if [ -f $LAN_RESULT ]
  then
```

```
        if [ $RESULT = "OK" ]
        then
    /root/bmp_reader i /root/bmp/lan_ok.bmp
        else
    /root/bmp_reader i /root/bmp/lan_fail.bmp
        while [ 1 ]
        do
        echo "LAN eth0   Fail"
        done
        fi
else
    /root/bmp_reader i /root/bmp/lan_fail.bmp
        while [ 1 ]
        do
        echo "LAN eth0 Fail"
        done
fi

sleep 1
```

아래와 같이 "**/bin/sh /root/ethernet_test.sh** "을 S99_build_system.sh 스크립트에 추가합니다.

/etc/init.d$ vi S99_build_system.sh

```
1 #!/bin/sh
  2 echo "Start ethernet test   ..."
  3 /bin/sh /root/ethernet_test.sh
  4
  5 echo "Start nand boot ubifs filesystem ..."
  6 mkdir -p /mnt/nand
  7
  8 flash_erase /dev/mtd2 0 0
  9 echo "[Step 1] mtd2 flash_erase done ..."
 10
 11 ubiattach /dev/ubi_ctrl -m 2
 12 echo "[Step 2] mtd2 ubiattach done ..."
 13
 14 ubimkvol /dev/ubi0 -N rootfs -m
 15 echo "[Step 3] mtd2 ubimkvol done ..."
 16
```

```
17 mount -t ubifs ubi0:rootfs /mnt/nand
18 echo "[Step 4] mount done ..."
19
20 tar xvf /root/rootfs.tar -C /mnt/nand
21 echo "[Step 5] file copy done ..."
22
23 sleep 2
24
25 umount /mnt/nand
26 echo "[Step 6] umount done ..."
27
28 echo "Complete nand boot ubifs filesystem ..."
29
30 sleep 1
31
32 /bin/sh /root/bmp_output.sh
```

아래와 같이 "rootfs_test.tar"로 압축합니다.

```
cd ../..
tar cf ../rootfs_test.tar *
```

```
$cp sdwriter_sdhc   sdwriter_sdhc_test
$vi sdwriter_sdhc_test
```

기존

```
ROOTFS_NAME=rootfs.tar
```

변경 후

```
ROOTFS_NAME=rootf_test.tar
```

1.3. 지그용

지그용도 만들어야 합니다.

rootfs를 카피 후 이름 변경합니다.

```
cp –a rootfs rootfs_jig
```

```
vi etc/init.d/S99_build_system.sh
```

아래와 같이 변경합니다.

```
1 #!/bin/sh
  2
  3 /bin/sh /root/ethernet_jig.sh
```

파일명을 변경합니다.

```
$ mv S99_build_system.sh S99_ethernet_jig
```

파일을 생성합니다.

root$ vi ethernet_jig.sh

```
1 #!/bin/sh
  2 echo "ethernet test_server"
  3 echo "ifconfig eth0 192.168.10.1"
  4 ifconfig eth0 192.168.10.1
  5 echo "/root/server start"
  6 /root/server &
```

압축합니다.

```
rootfs_jig$ tar cf ../rootfs_jig.tar *
```

sdwriter을 변경합니다.

```
$cp sdwriter_sdhc sdwriter_sdhc_jig
$vi sdwriter_sdhc_jig
```

기존

```
 ROOTFS_NAME=rootfs.tar
```

변경 후

```
 ROOTFS_NAME=rootf_jig.tar
```

테스트용 sd카드에 테스트용 이미지를 라이트합니다.

```
sudo ./sdwriter_sdhc_test sdb 24
```

지그용 sd카드에 지그용 이미지를 라이트합니다.

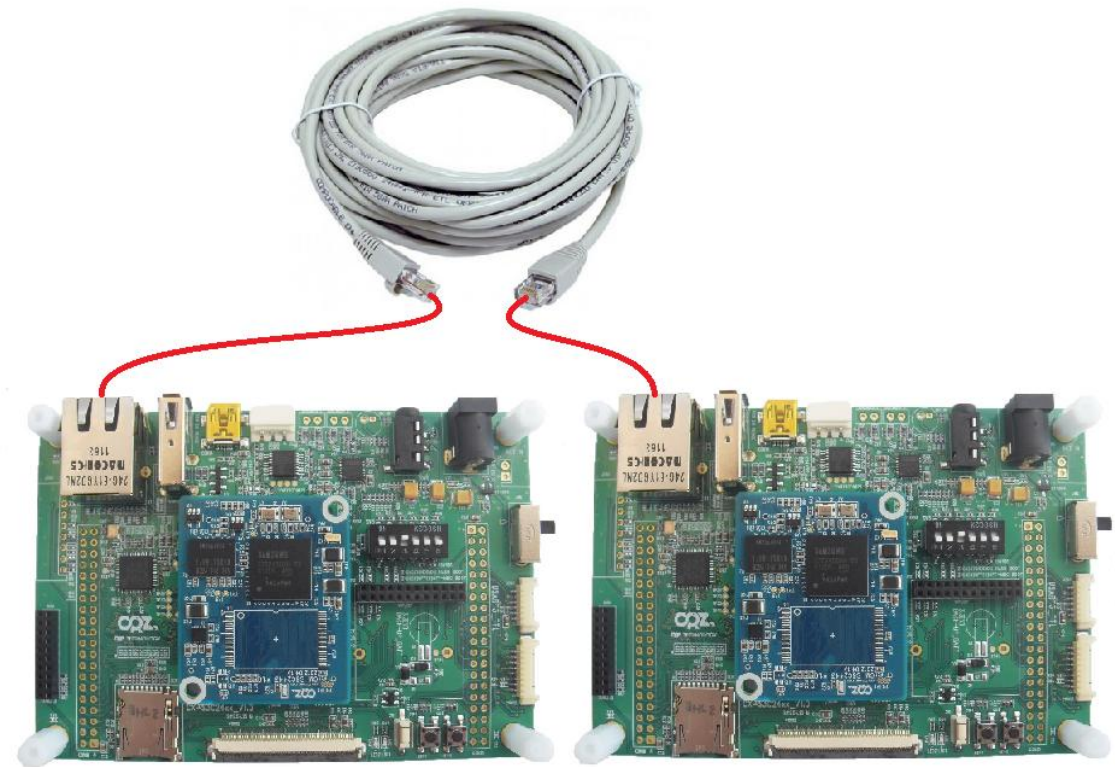
```
sudo ./sdwriter_sdhc_jig sdb 24
```

1.4. 테스트방법

보드 2개를 준비합니다.

위에서 준비한 테스트용 SD카드와 지그용 SD카드를 각각 보드에 삽입합니다.

LAN Cable을 보드에 연결합니다.



SD/MMC부팅모드(3번 ON)로 하고 전원을 인가하면 됩니다.

1.5. 테스트 결과

1.5.1. LAN TEST OK

LAN TEST OK 시 LCD화면에 아래 그림이 출력됩니다.

1.5.2. LAN TEST Fail

LAN TEST Fail 시 LCD화면에 아래 그림이 출력됩니다.