

개발환경 구축: Cross compiler, bootp, tftp, NFS 등

단국대학교

컴퓨터학과

2009

백승재

baeksj@dankook.ac.kr

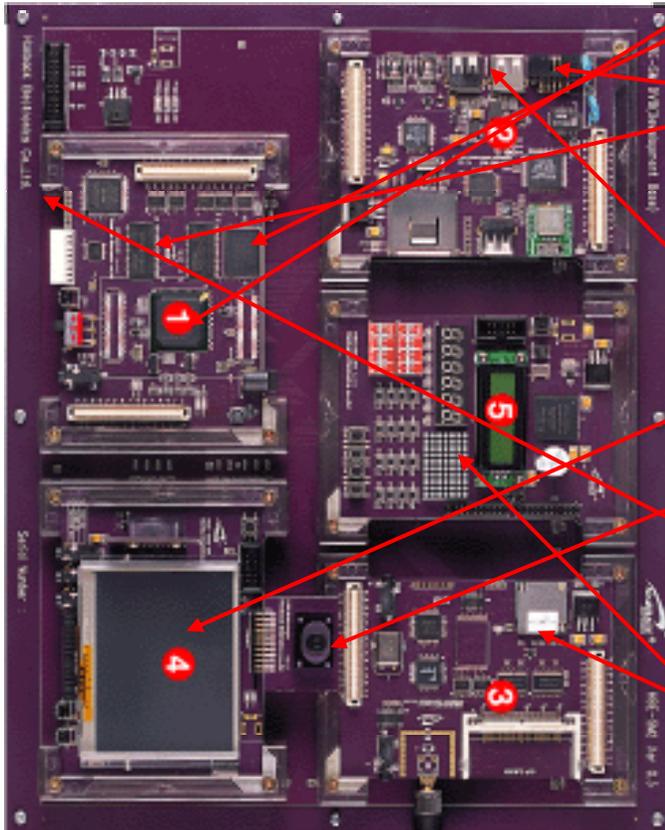
<http://embedded.dankook.ac.kr/~baeksj>

강의 목표

- Target board에 대한 기본적인 이해
- 개발 과정 이해 및 환경 구축
- 개발 사례 파악

Target board

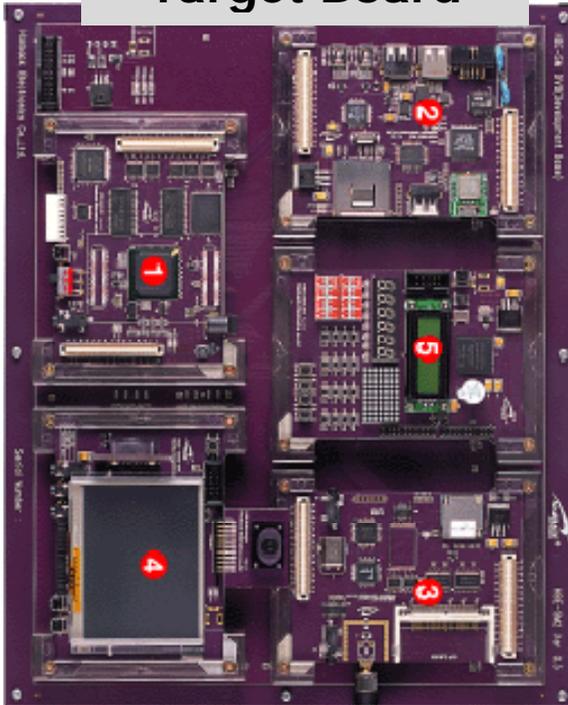
■ HBE-SM II-P320



Items	Hardware Specifications
CPU	Marvell PXA320B1C806 (806MHz)
Memory	NAND Flash Memory : Samsung K9F1G08U0B (128M x 8bit) Mobile-DDR RAM : 128MByte (512Mbit x 2ea)
FPGA	Altera Cyclone-2 (EP2C8F256C8ES) x 2ea
CPLD	Altera MAXII (EPM1270GF256C5) x 1ea
SRAM	Samsung K6R4016V1C (256Kx16 Bit) x 3ea
Ethernet	1 Port 10/100Base-T Ethernet
Audio	AC'97 Audio Codec (Built-in Touch Screen I /F) 1 Port Stereo Out (Stereo Jack), 1 Port Line IN (Stereo Jack), 1 Port MIC IN (Stereo Jack)
LCD	3.5" TFT LCD 1ea
Touch Screen	3.5 Touch Screen x 1ea
Camera	1.3M Pixel CMOS Camera x 2ea
USB	USB 1.1 Host : A type Connector x 1ea USB 1.1 Client : Mini type Connector x 1ea USB 2.0 Host : A type Connector x 1ea USB 2.0 OTG : Mini type Connector x 1ea
UART	RS232 Level : Stereo type Connector x 3ea 3.3V TTL Level : 2 Port for GPS & Bluetooth : 1 Port
I ² C	1 Port
IrDA	1 Port (Compliant to IrDA 1.1)
Storage	1ea CF Connector 1ea SD / MMC Connector
Wireless LAN	1 Port Wireless LAN (IEEE802.11b/g)
I / O Application	Text LCD, 7-Segment, Dot Matrix (7 x 5 x 2ea), Key Pad (4 x 4), DIP Switch (8point x 2ea), LED (8ea), Buzzer, Tact Switch (4ea)
I / F Module	1ea Bluetooth Module (On-Board)

Step1: Serial port & minicom setup

Target Board

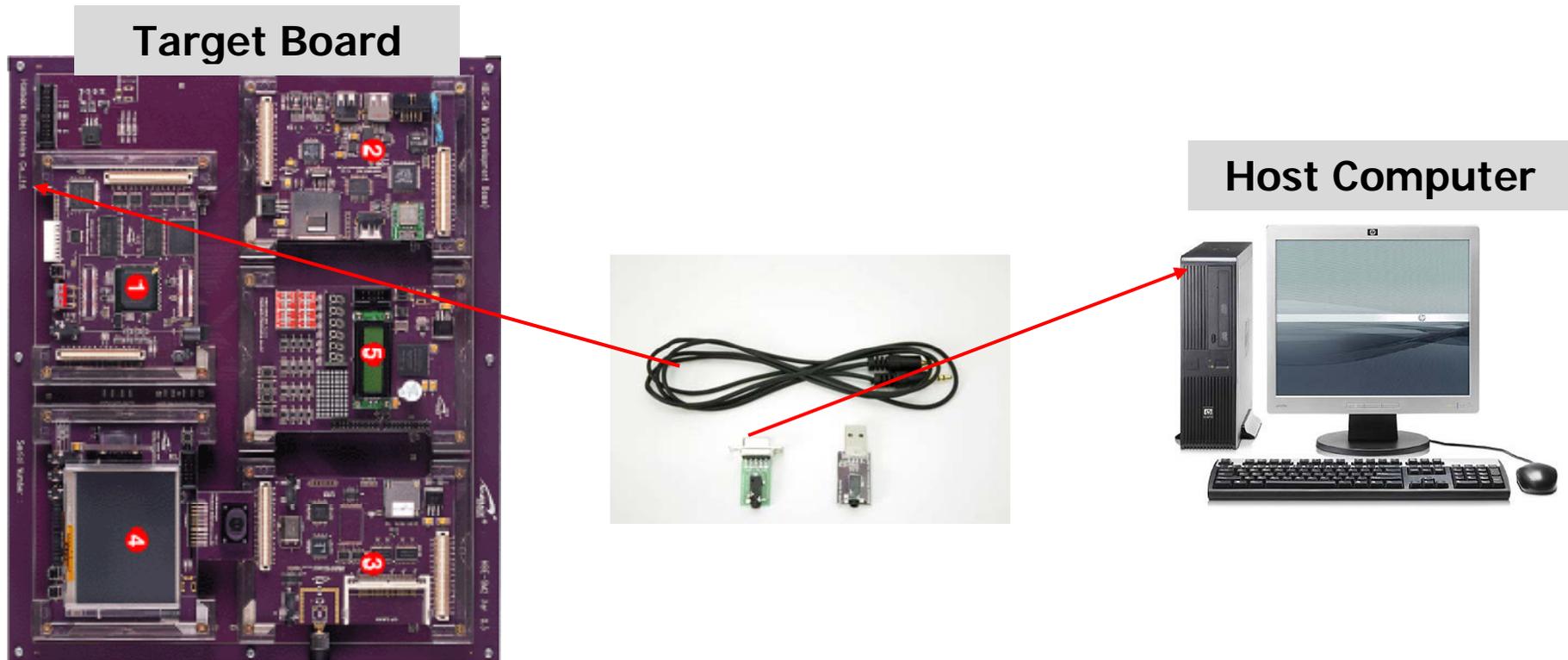


Host Computer



- Target board have no 'monitor'
- Serial port와 minicom 프로그램을 이용해 보드에 가상적인 모니터를 만들어 주자

Step1: Serial port & minicom setup



- Serial이나 USB 둘 중 하나만 사용하여 target board와 host computer를 연결

Step1: Serial port & minicom setup

- minicom : 리눅스용 터미널 에뮬레이터
 2. Minicom 환경 설정

Boudrate	115200bps
Date Size	8
Parity	None
Stop bits	1
H/W flow control	NO
S/W flow control	NO

```

root's X desktop (localhost,localdomain:1)
root@localhost:~
[root@localhost root]# setserial -a /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
Baud_base: 115200, close_delay: 50, divisor: 0
closing_wait: 3000
Flags: spd_normal skip_test
[root@localhost root]# minicom -s

```

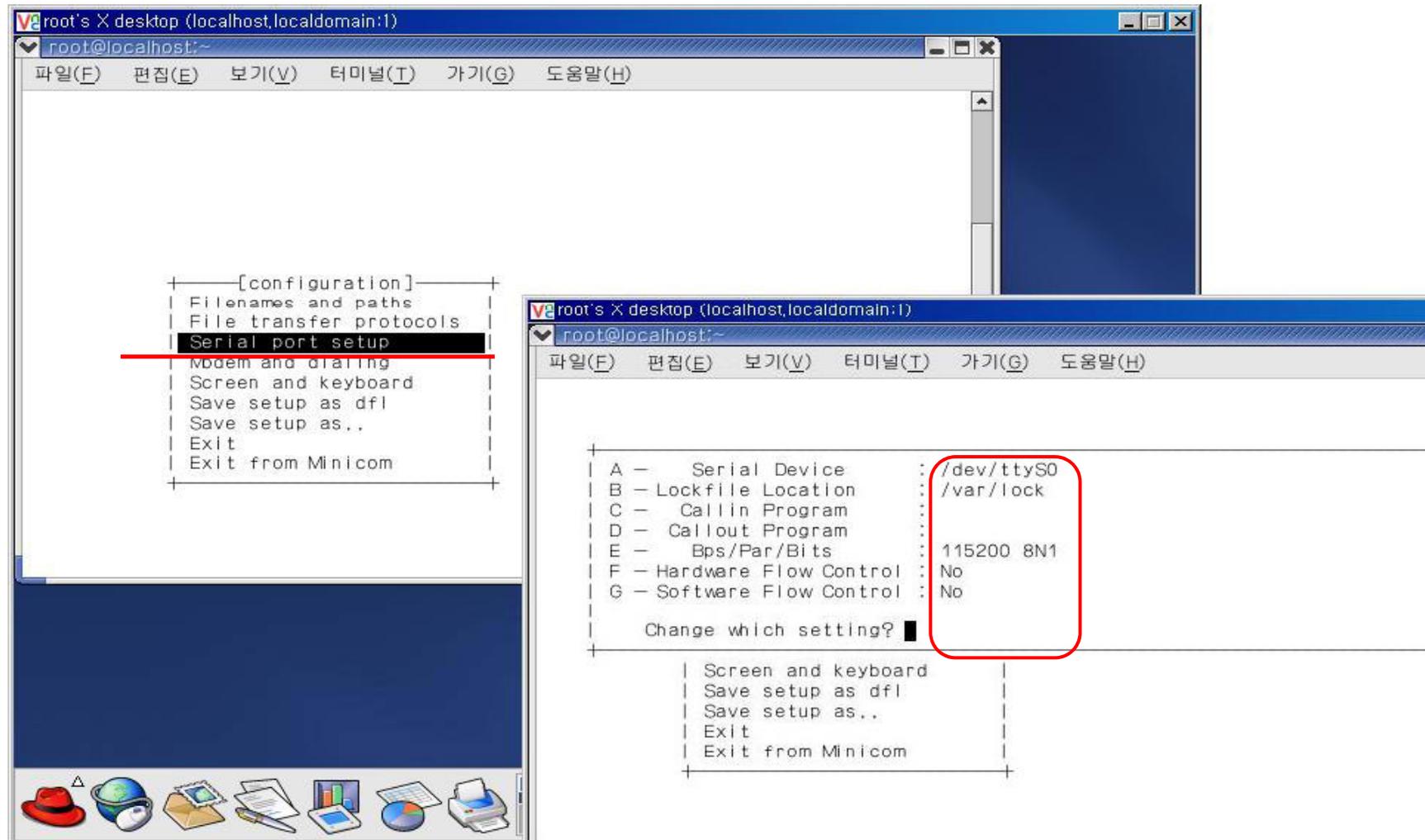
```

[configuration]
Filenames and paths
File transfer protocols
Serial port setup
Modem and dialing
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom

```

Step1: Serial port & minicom setup

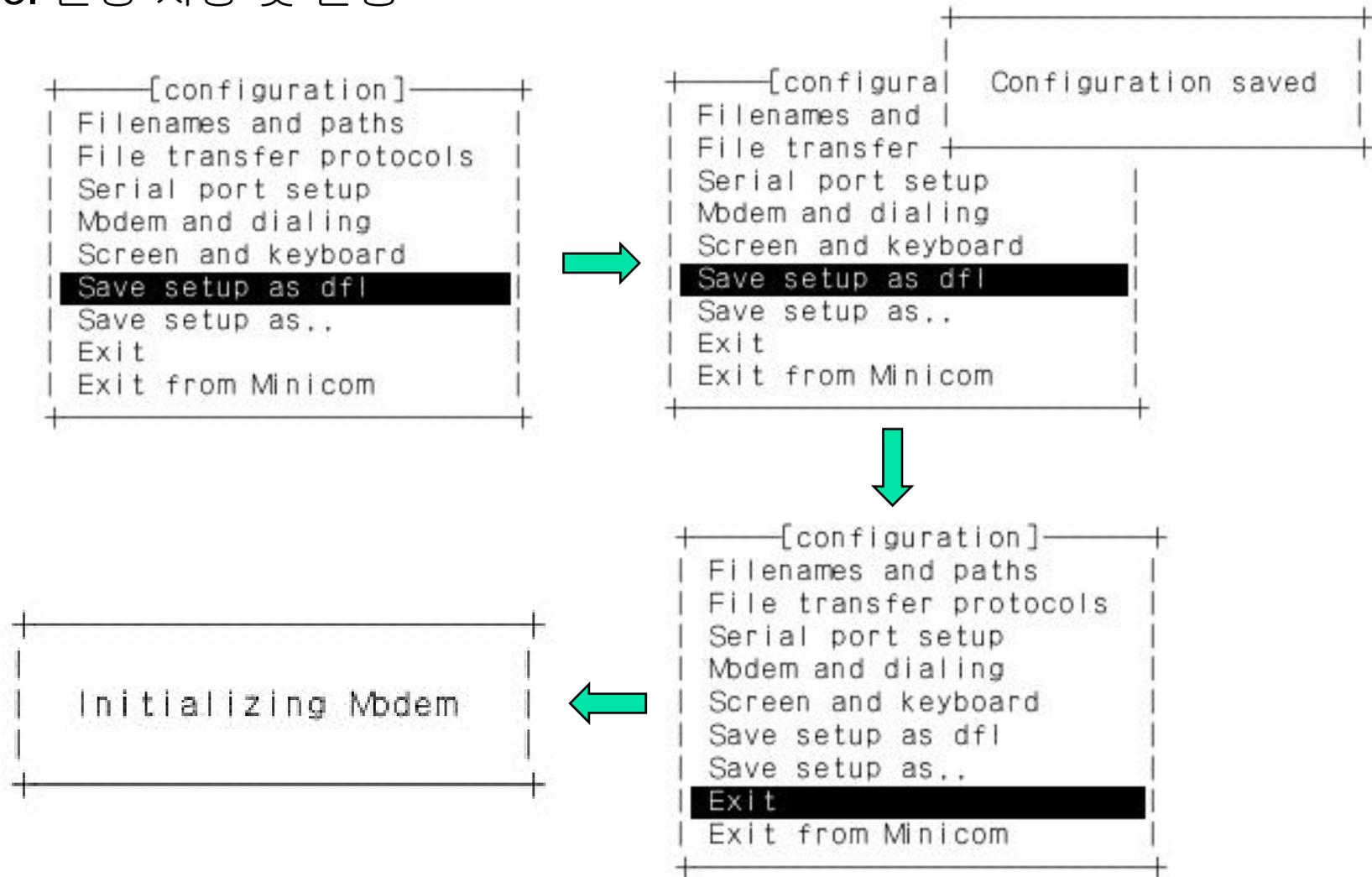
- minicom : 리눅스용 터미널 에뮬레이터
- ## 2. Minicom 환경 설정



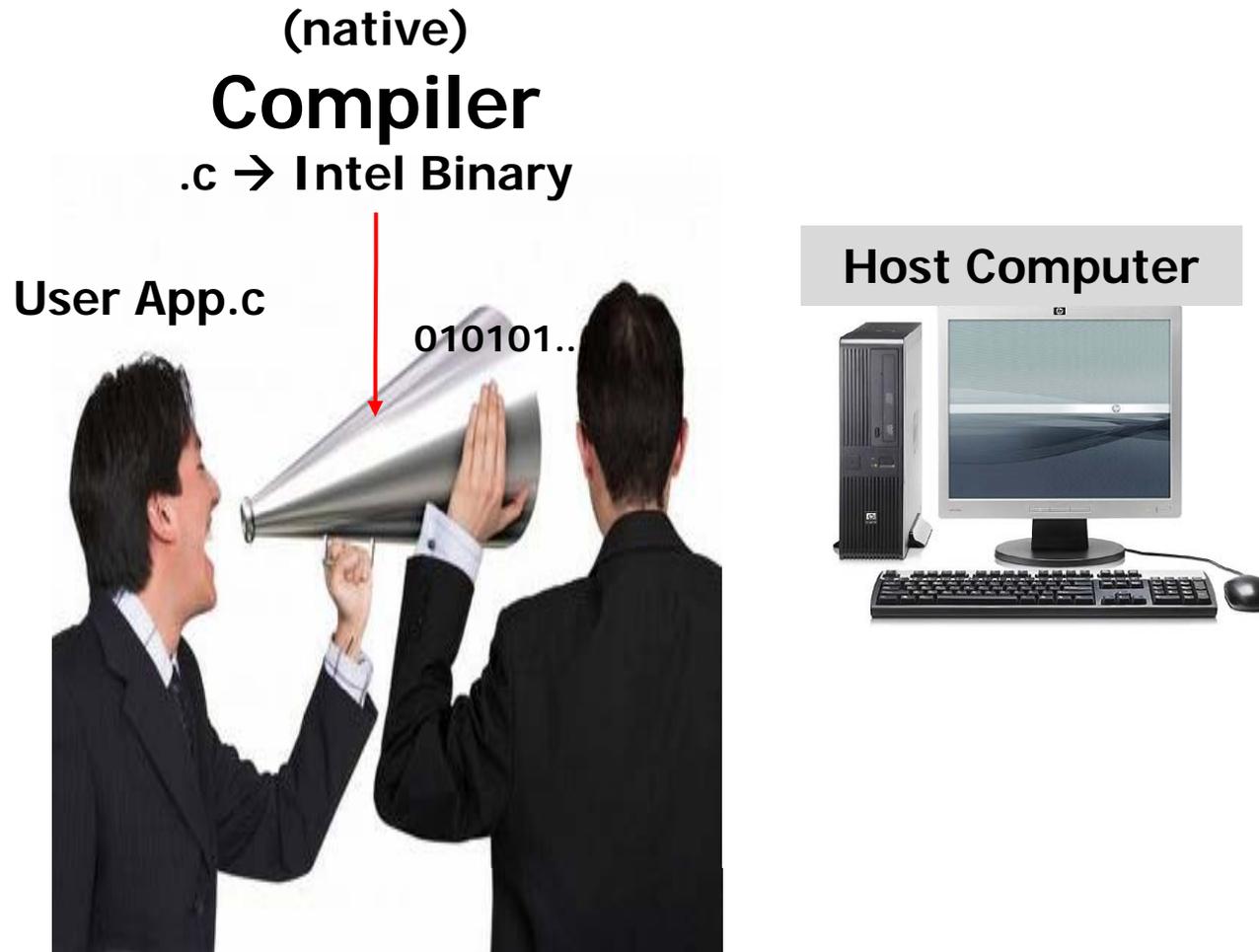
Step1: Serial port & minicom setup

■ minicom : 리눅스용 터미널 에뮬레이터

3. 설정 저장 및 실행

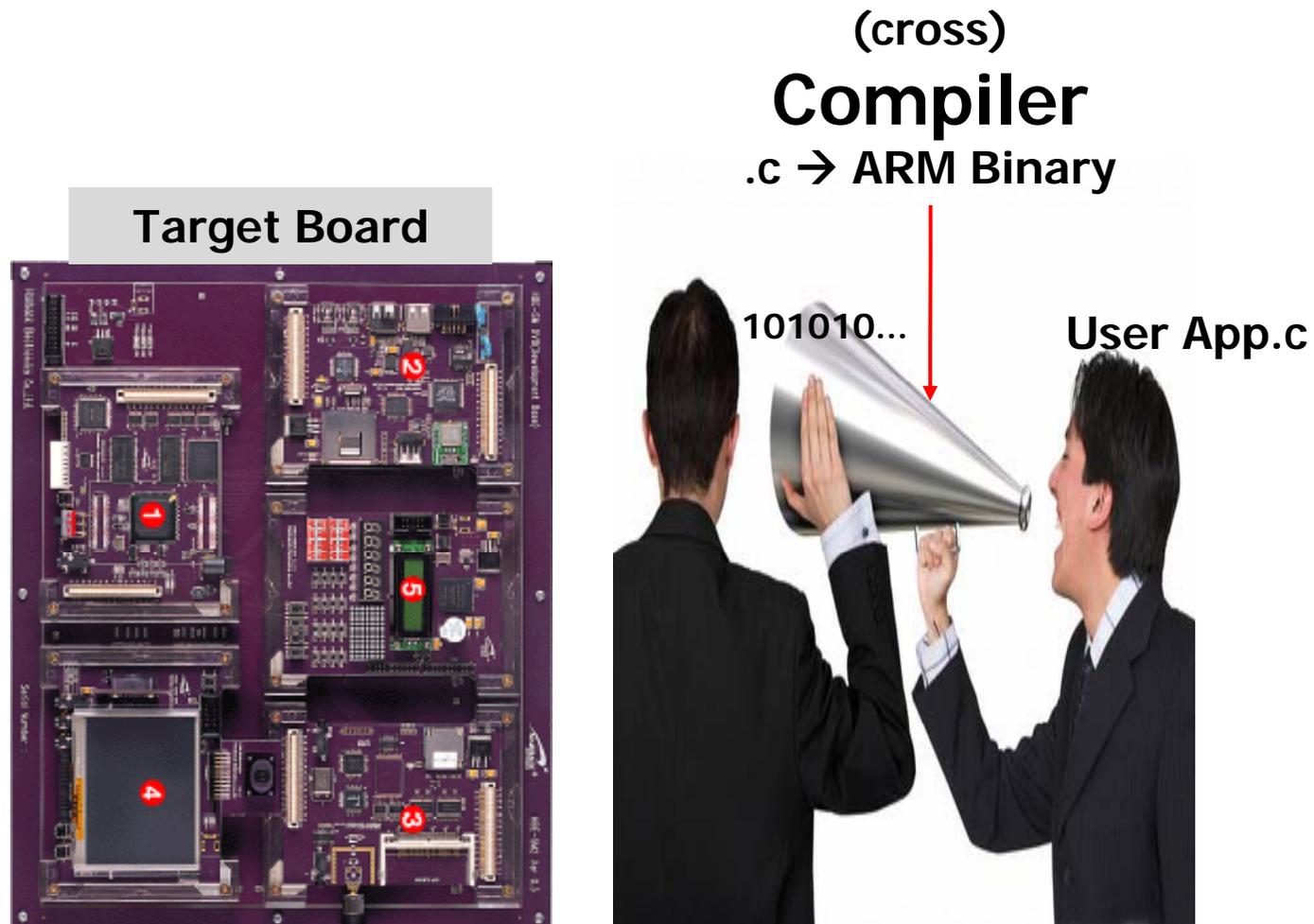


Step2: Cross compiler



- Native compiler
 - ✓ generates code for its own execution environment

Step2: Cross compiler

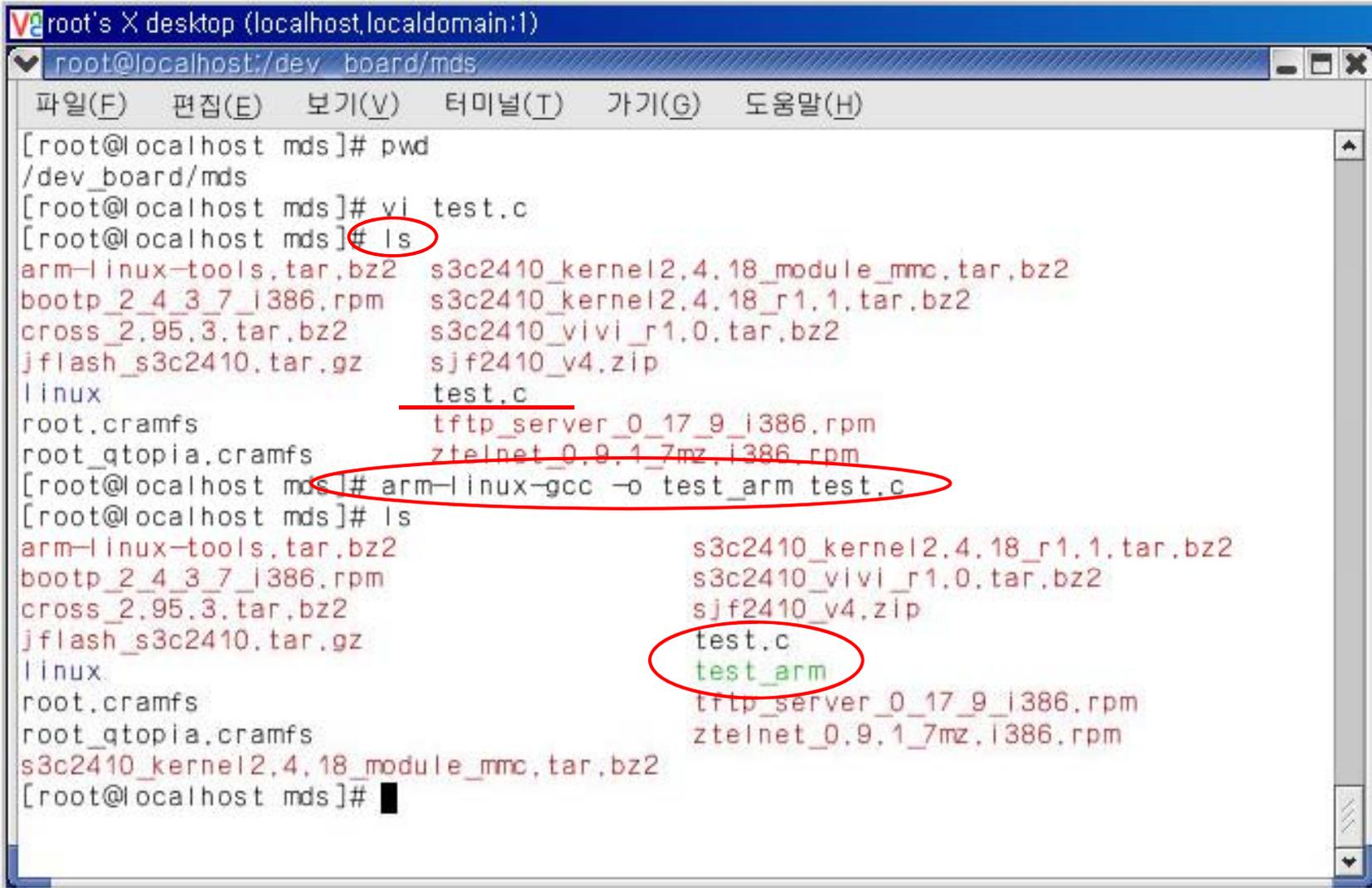


- Cross compiler
 - ✓ executes in one environment and generates code for another

Step2: Cross compiler

- Cross compiler
 - ✓ Modify the compiler, compile the compiler and use it
 - ✓ Download the compiler(already modified) for your architecture
 - tar.gz
 - rpm, ...
- HBE-SMII-P320 보드의 경우 CD에 cross compiler를 제공한다
 - ✓ /usr/local 디렉토리에 압축해제 한 후
 - ✓ PATH 설정 하여 사용하면 된다

Step2: Cross compiler



```
root's X desktop (localhost,localhost:1)
root@localhost:/dev_board/mds
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
[root@localhost mds]# pwd
/dev_board/mds
[root@localhost mds]# vi test.c
[root@localhost mds]# ls
arm-linux-tools.tar.bz2  s3c2410_kernel2.4.18_module_mmc.tar.bz2
bootp_2_4_3_7_i386.rpm  s3c2410_kernel2.4.18_r1.1.tar.bz2
cross_2.95.3.tar.bz2   s3c2410_vivi_r1.0.tar.bz2
jflash_s3c2410.tar.gz  sjf2410_v4.zip
linux                   test.c
root.cramfs             tftp_server_0_17_9_i386.rpm
root_qtopia.cramfs     ztelnet_0.9.1_7mz_i386.rpm
[root@localhost mds]# arm-linux-gcc -o test_arm test.c
[root@localhost mds]# ls
arm-linux-tools.tar.bz2  s3c2410_kernel2.4.18_r1.1.tar.bz2
bootp_2_4_3_7_i386.rpm  s3c2410_vivi_r1.0.tar.bz2
cross_2.95.3.tar.bz2   sjf2410_v4.zip
jflash_s3c2410.tar.gz  test.c
linux                   test_arm
root.cramfs             tftp_server_0_17_9_i386.rpm
root_qtopia.cramfs     ztelnet_0.9.1_7mz_i386.rpm
s3c2410_kernel2.4.18_module_mmc.tar.bz2
[root@localhost mds]#
```

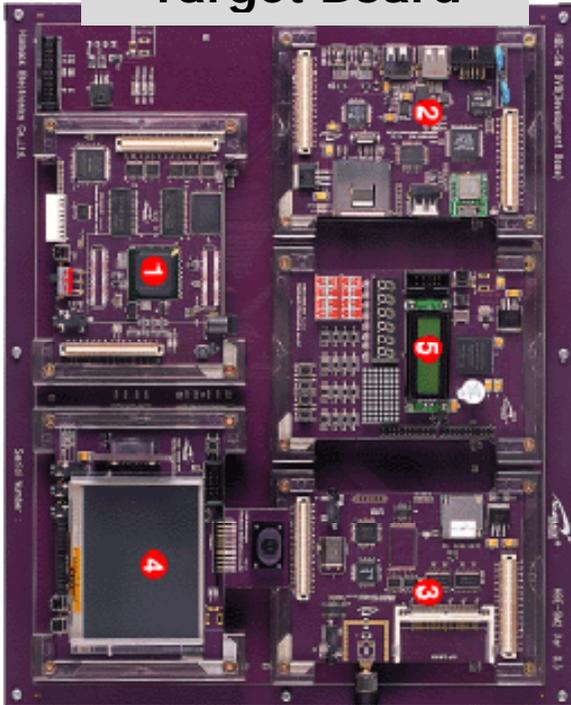
Step2: Cross compiler

```
root's X desktop (localhost,localhost:1)
root@localhost:/dev board/mds
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
root.cramfs tftp_server_0_17_9_i386.rpm
root_qtopia.cramfs ztelnnet_0.9.1_7mz.i386.rpm
s3c2410_kernel2.4.18_module mmc.tar.bz2
[root@localhost mds]# gcc -o test_ia test.c
[root@localhost mds]# ls
arm-linux-tools.tar.bz2 s3c2410_kernel2.4.18_r1.1.tar.bz2
bootp_2_4_3_7_i386.rpm s3c2410_vivi_r1.0.tar.bz2
cross_2.95.3.tar.bz2 sif2410_v4.zip
jflash_s3c2410.tar.gz test.c
linux test_arm
root.cramfs test_ia
root_qtopia.cramfs tftp_server_0_17_9_i386.rpm
s3c2410_kernel2.4.18_module mmc.tar.bz2 ztelnnet_0.9.1_7mz.i386.rpm
[root@localhost mds]# file test_arm
test_arm: ELF 32-bit LSB executable, ARM, version 1 (ARM), for GNU/Linux 2.0.0,
dynamically linked (uses shared libs), not stripped
[root@localhost mds]# file test_ia
test_ia: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux
2.2.5, dynamically linked (uses shared libs), not stripped
[root@localhost mds]# ./test_ia
HH~ Cross Compile is Very EASY~~~ ^^;
[root@localhost mds]# ./test_arm
bash: ./test arm: cannot execute binary file
[root@localhost mds]#
```

각 파일이 어느 아키텍처에서 실행가능한지 확인할 수 있다.

Step3: JTAG

Target Board

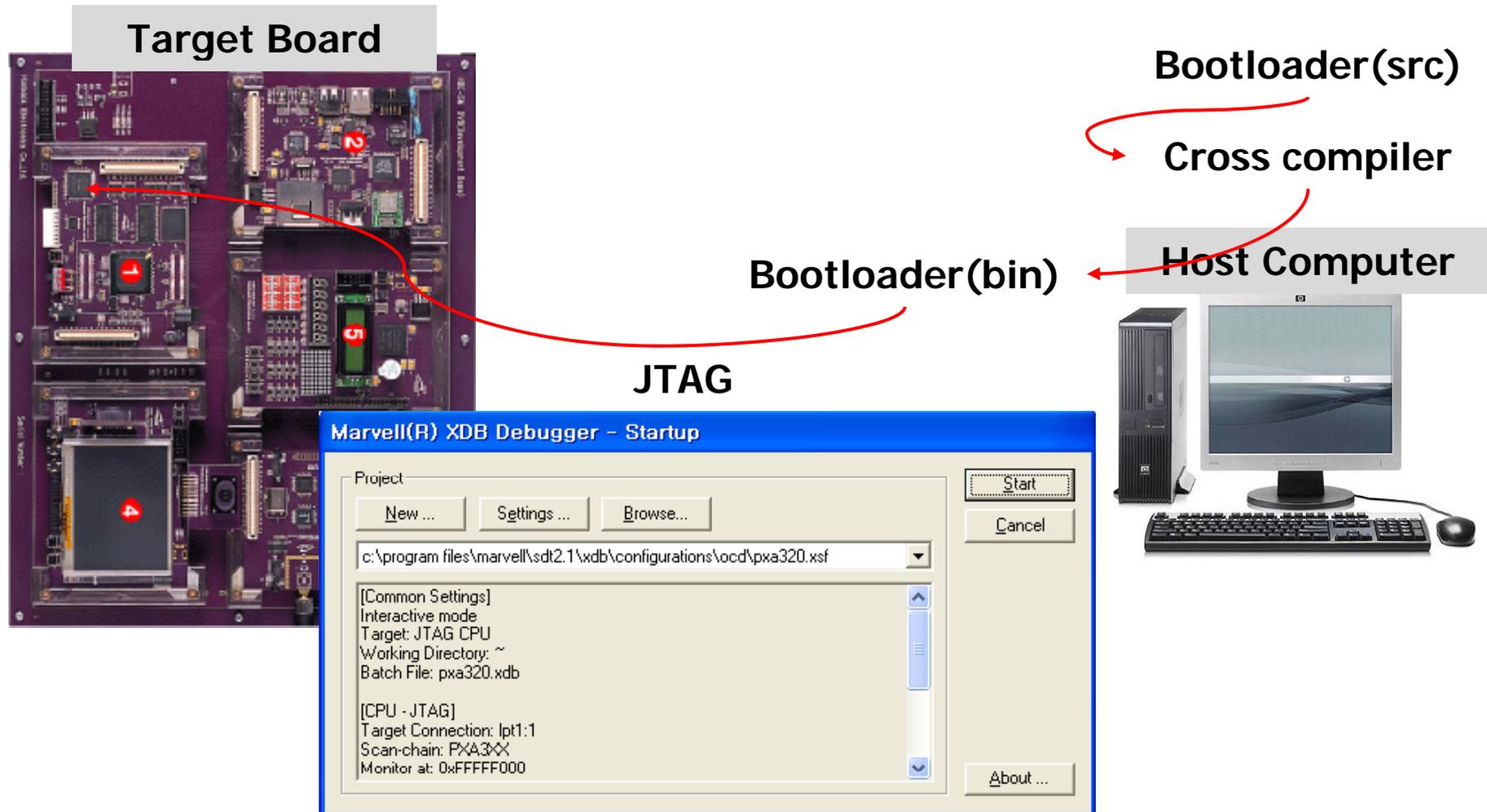


Host Computer



- PC of ARM cpu have the value 0x00000000 after power on
- Bootloader must be written in 0x00000000
- NOR flash memory located in 0x00000000
- How can you writing bootloader into NOR flash memory?

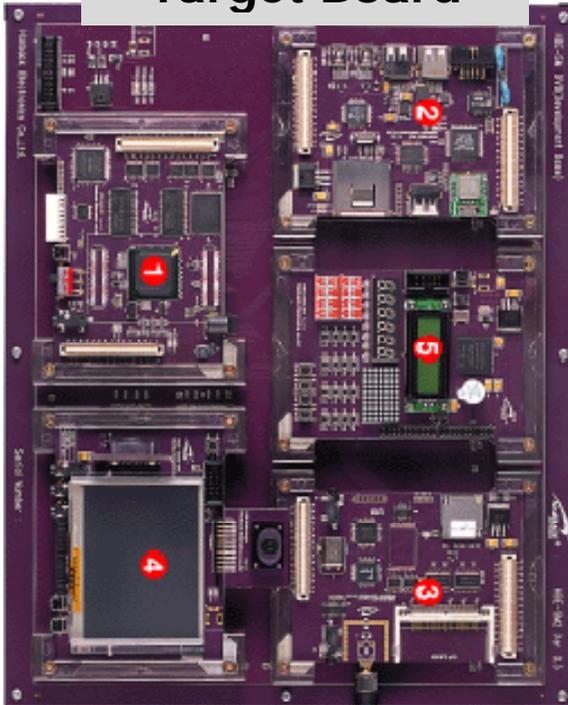
Step3: JTAG



- In the Linux?

Step4: bootp

Target Board



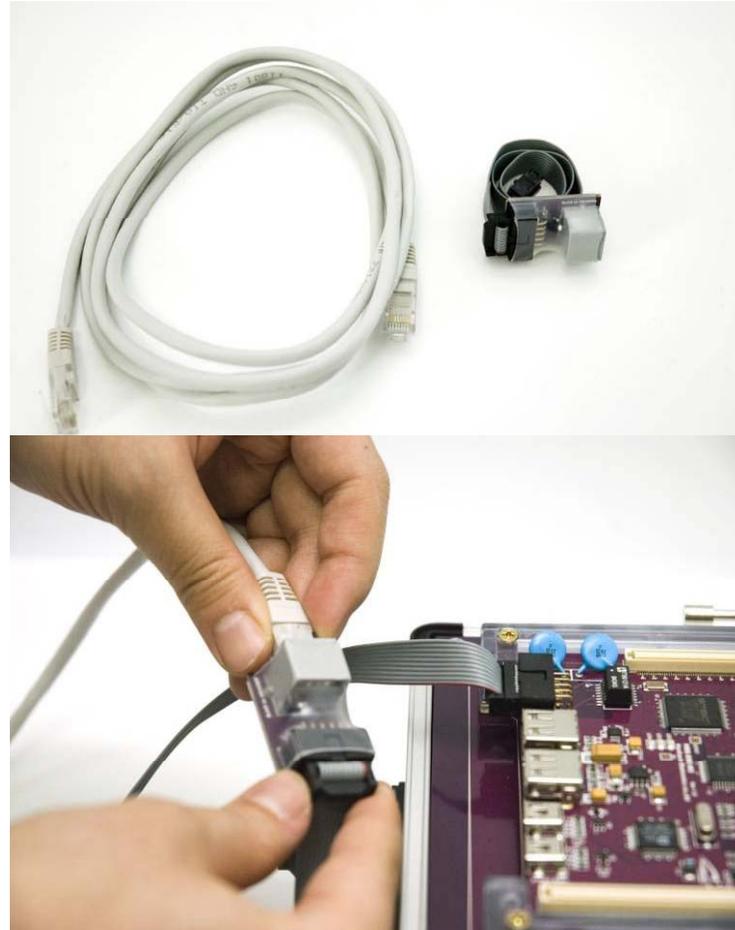
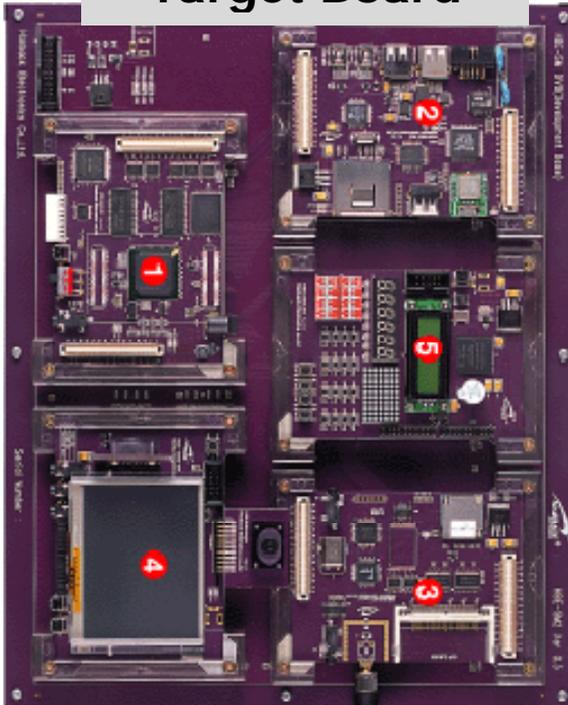
Host Computer



- Bootloader will running on the target board
 - ✓ You can use peripherals
- Kernel image and rootfilesystem also should be written to Flash memory on the board
- Use faster interface than JTAG

Step4: bootp

Target Board

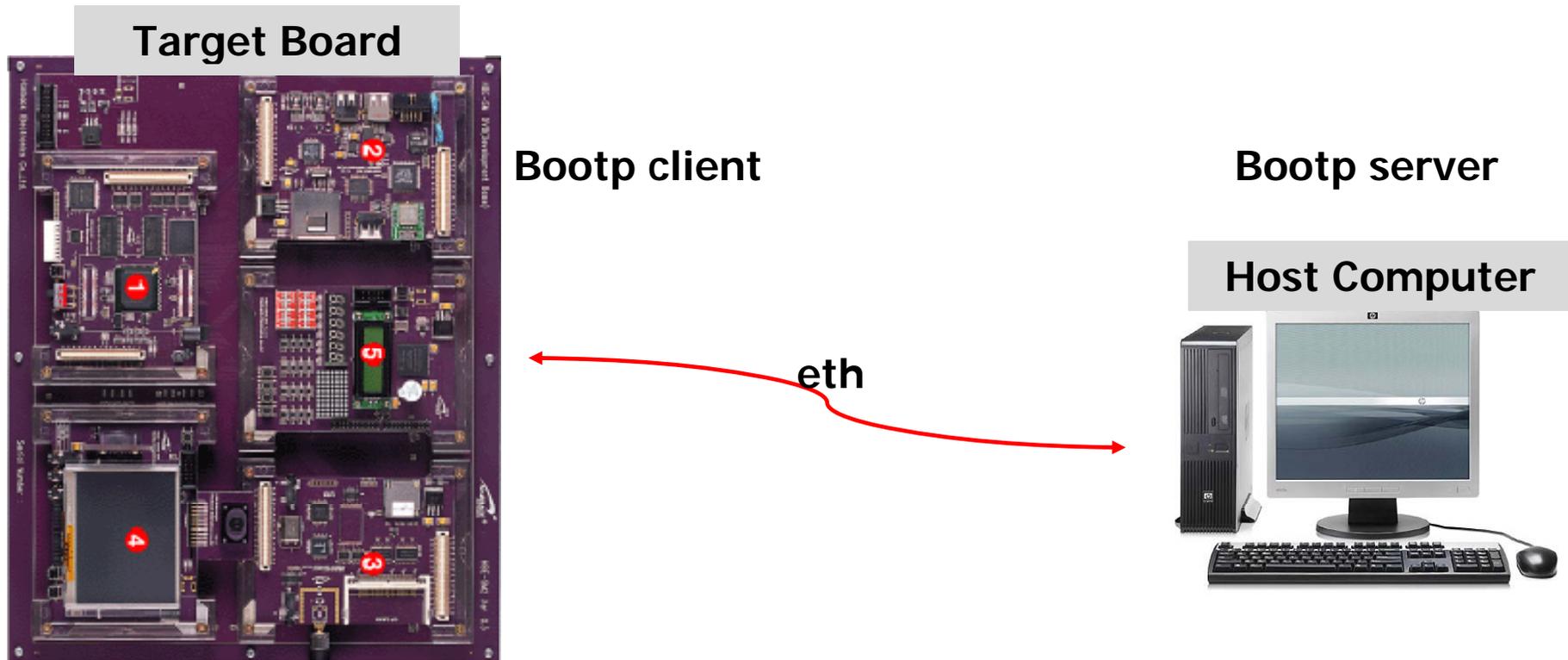


Host Computer



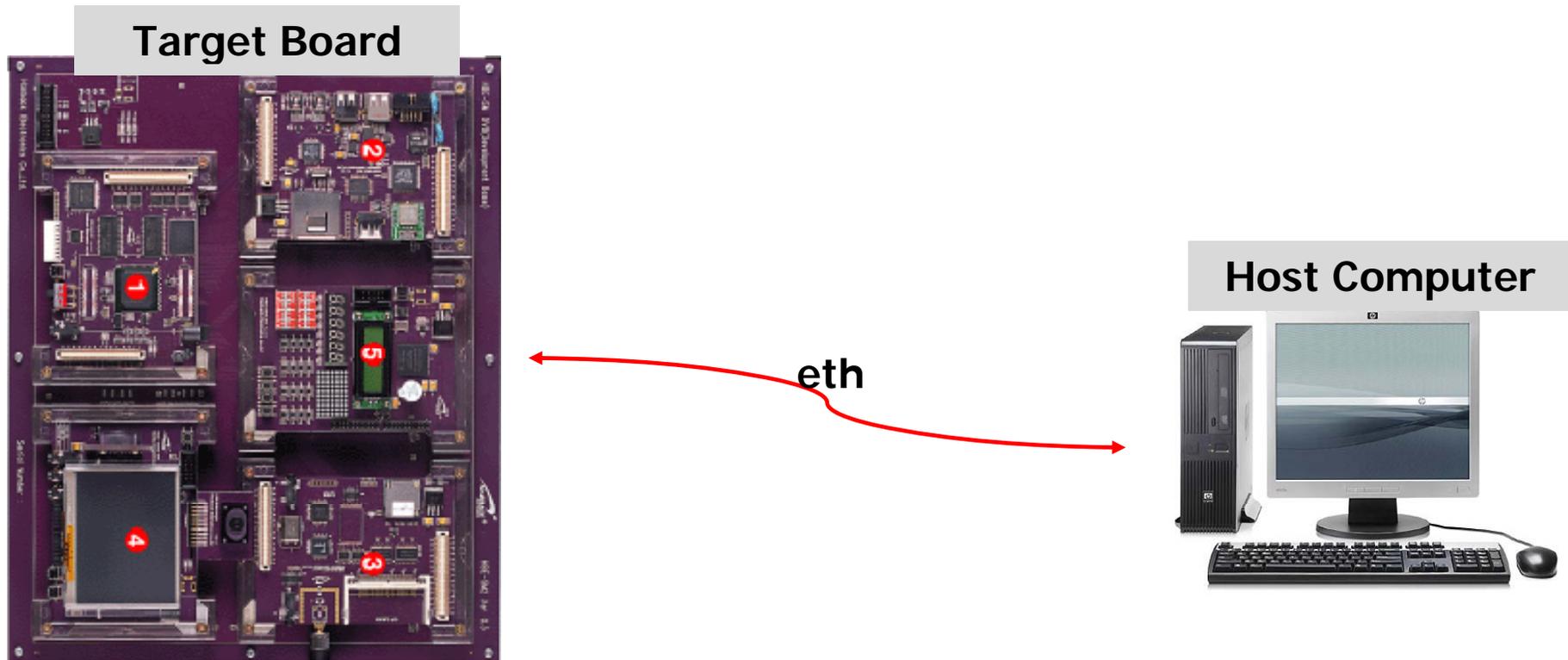
- (Cross)Ethernet cable 연결

Step4: bootp



- IP address required(for target board)
- You can get IP address automatically using DHCP if there is DHCP server
- Bootp (your host computer should be bootp server)

Step4: tftp



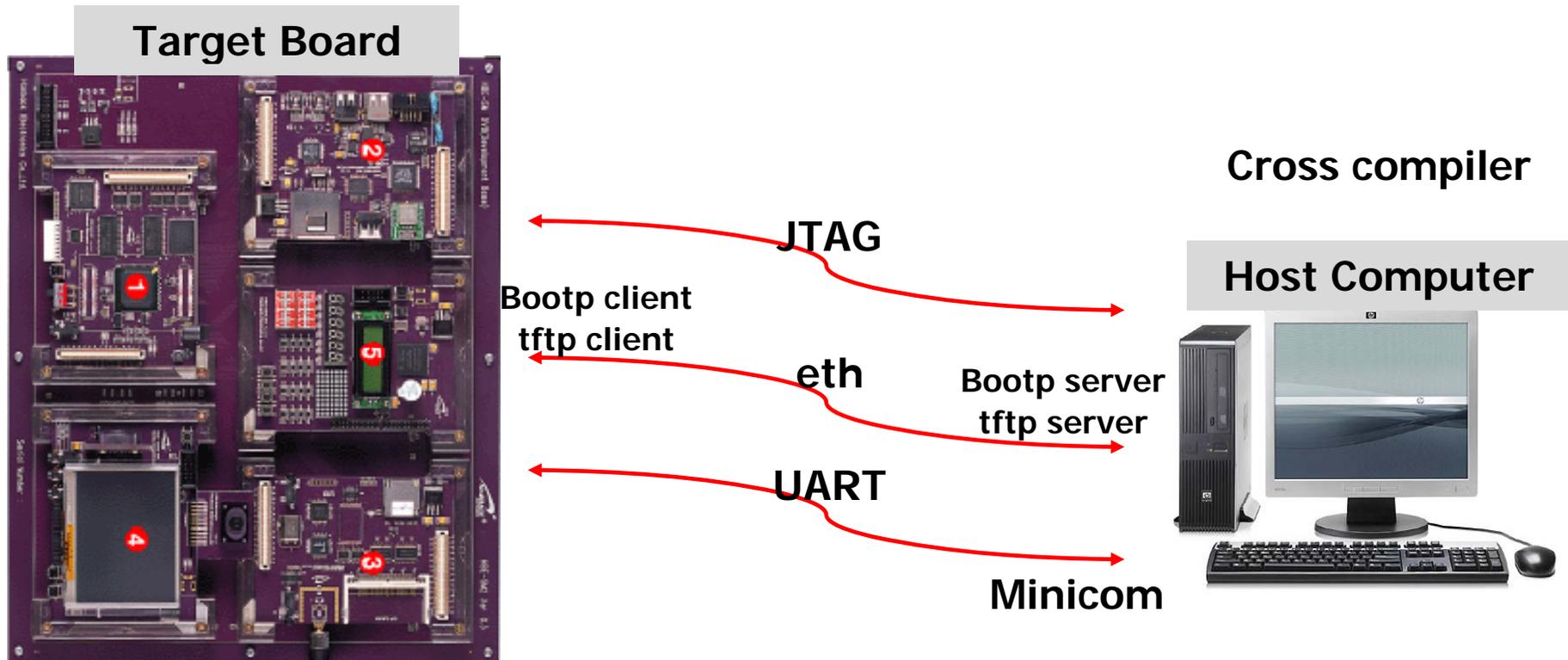
- What program do you use when you download some files from server?
- You can download file from FTP server using FTP client program
- TFTP(your host computer should be tftp server)

Step4: tftp

```
root's X desktop (localhost.localdomain:1)
root@localhost:/etc/xinetd.d
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
[root@localhost xinetd.d]# pwd
/etc/xinetd.d
[root@localhost xinetd.d]#
[root@localhost xinetd.d]# ls -al tftp
-rw-r--r-- 1 root root 509 9월 7 12:24 tftp
[root@localhost xinetd.d]#
[root@localhost xinetd.d]# vi tftp
root's X desktop (localhost.localdomain:1)
root@localhost:/etc/xinetd.d
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
# default: off
# description: The tftp server serves files using the trivial file transfer
# protocol. The tftp protocol is often used to boot diskless
# workstations, download configuration files to network-aware printers,
# and to start the installation process for some operating systems.
service tftp
{
    socket_type          = dgram
    protocol             = udp
    wait                 = yes
    user                 = root
    server               = /usr/sbin/in.tftpd
    server_args          = -s /tftpboot
    disable              = no
    per_source           = 11
    cps                  = 100 2
    flags                = I|V4
}
[root@localhost xinetd.d]# service xinetd restart
xinetd 를 정지함:
xinetd (을)를 시작합니다:
[root@localhost xinetd.d]#
[root@localhost xinetd.d]#
[root@localhost xinetd.d]# netstat -a | grep boot
udp        0      0 *:bootps          *:*
[root@localhost xinetd.d]# netstat -a | grep tftp
udp        0      0 *:tftp            *:*
[root@localhost xinetd.d]#
```

[확인]

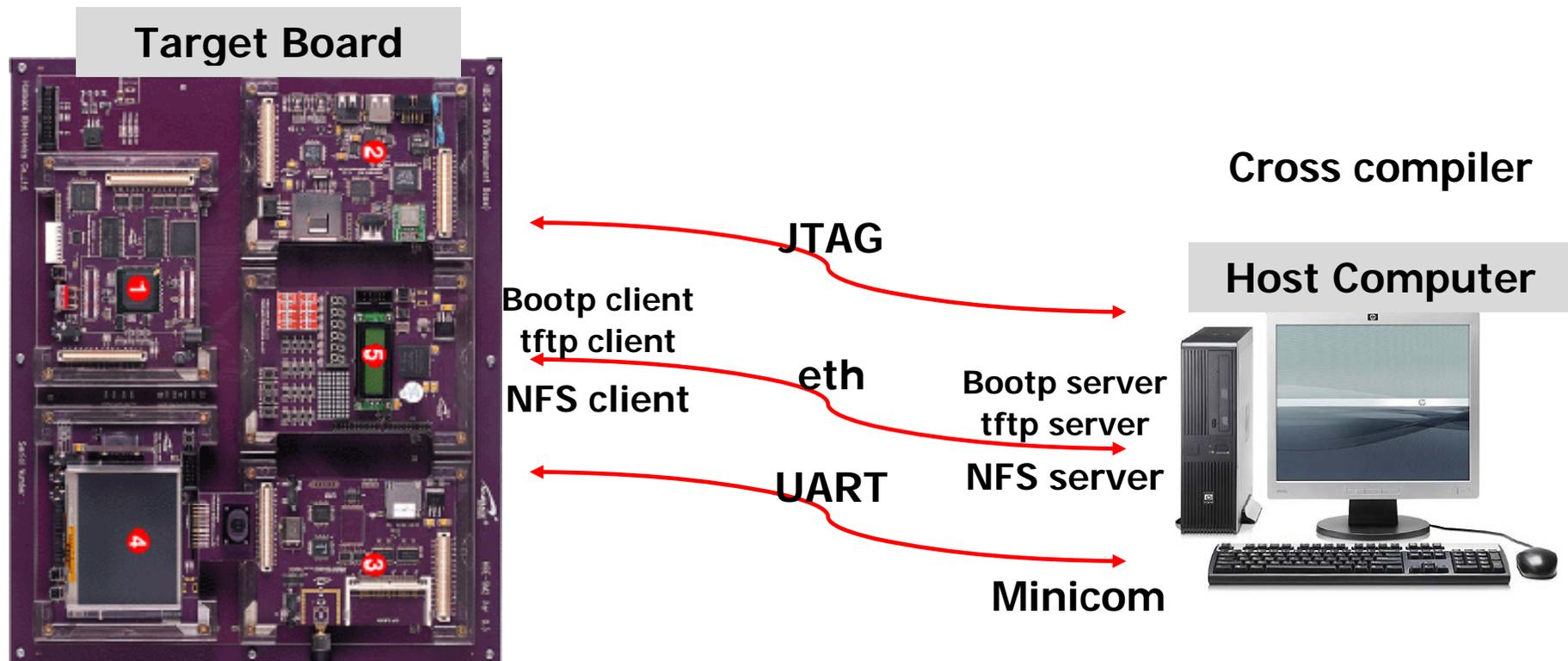
전체 구성도



Linux booting 확인



Step5: NFS



- Target board에서 Linux booting후
 - ✓ Device driver 및 application 개발
- 매번 tftp로 download?

Step5: NFS

```
root's X desktop (localhost,localdomain:1)
root@localhost:/
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
[root@localhost /]# ls -la
.          .automount      boot          etc           lib           root
..         .fonts.cache-1  dev           home          local        opt          sbin         usr
.autofsck  bin             dev_board    initrd        misc         proc        tftpboot    var
[root@localhost /]# mkdir nfs
[root@localhost /]# chmod 777 nfs
[root@localhost /]# ls -al | grep nfs
drwxrwxrwx  2 root    root          4096  9월  11  16:43  nfs
[root@localhost /]# chown nobody nfs
[root@localhost /]# chgrp nobody nfs
[root@localhost /]# ls -al | grep nfs
drwxrwxrwx  2 nobody  nobody       4096  9월  11  16:43  nfs
[root@localhost /]#
```

클라이언트로 부터의
마운트 요청시 서비스
할 디렉토리 생성

drwxrwxrwx 2 root root 4096 9월 11 16:43 nfs

drwxrwxrwx 2 nobody nobody 4096 9월 11 16:43 nfs

접근 권한 문제 등으로 인해
마운트가 실패하는 것을
방지하기 위한 설정

Step5: NFS

■ NFS환경 파일 수정

- ✓ /etc/exports 의 내용을 읽어 어떤 호스트에게 마운트를 허가하고, 어떤 제한을 뒀야 하는지 판단

```

root's X desktop (localhost,localdomain:1)
root@localhost:~# cat /etc/exports
/nfs localhost(rw,insecure)
/nfs 192.168.184.0/24(rw,insecure)
  
```

NFS로 공유할 디렉토리

rw : r(읽기), w(쓰기)허용
Insecure : 암호 인증 하지 않음

192.168.184대역은
모두 허용

Step5: NFS

```
root's X desktop (localhost,localdomain:1)
root@localhost:~/
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
[root@localhost /]# ls -la
.          .automount      boot          etc           lib           mnt          root         tmp
..         .fonts.cache-1 dev           home         lost+found   opt        sbin         usr
.autofsck  bin             dev_board    initrd       misc         proc
[root@localhost /]# mkdir nfs
[root@localhost /]# chmod 777 nfs
[root@localhost /]# ls -al | grep nfs
drwxrwxrwx  2 root    root         4096   9월  11  16:43  nfs
[root@localhost /]# chown nobody nfs
[root@localhost /]# chgrp nobody nfs
[root@localhost /]# ls -al | grep nfs
drwxrwxrwx  2 nobody  nobody      4096   9월  11  16:43  nfs
[root@localhost /]# vi /etc/exports
[root@localhost /]# service nfs restart
NFS mountd를 종료 중입니다:
NFS 데몬을 종료 중입니다:
Shutting down NFS quotas:
NFS 서비스를 종료 중입니다:
NFS 서비스를 시작하고 있습니다:
Starting NFS quotas:
NFS 데몬을 시작함:
NFS mountd를 시작하고 있습니다:
[root@localhost /]#
```

편집한 내용을 반영하기 위해서 nfs 데몬을 다시시작한다.
#/etc/rc.d/init.d/nfs restart

```
[실패]
[실패]
[실패]
[확인]
[확인]
[확인]
[확인]
```

데몬이 잘 작동 되는지 확인

■ Board에서 nfs서버 접속

```
root's X desktop (localhost,localhost:1)
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
[root@2410AM1 mnt]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:C0:FF:EE:08
          inet addr:192.168.184.111  Bcast:192.168.184.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:0
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:100
          RX bytes:3644 (3.5 KiB)  TX bytes:0 (0.0 B)
          Interrupt:2 Base address:0x300

[root@2410AM1 mnt]$ ifconfig eth0 down
[root@2410AM1 mnt]$ ifconfig eth0 192.168.184.138 netmask 255.255.255.0
detected!! HALF
eth0: using half-duplex 10Base-T (RJ-45)
[root@2410AM1 mnt]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:C0:FF:EE:08
          inet addr:192.168.184.138  Bcast:192.168.184.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:0
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:100
          RX bytes:3644 (3.5 KiB)  TX bytes:0 (0.0 B)
          Interrupt:2 Base address:0x300

[root@2410AM1 /]$ ls
bin  dev  etc  lib  linuxrc  lost+found  mnt  proc  root  sbin  tmp  usr  var
[root@2410AM1 /]$ mkdir /mnt/nfs
[root@2410AM1 /]$ mount -t nfs -o nolock 192.168.184.1:/nfs /mnt/nfs
```

Target Board에서 Host에 NFS로 접속 하기 위하여 target의 IP를 설정. Host NIC의 IP에 맞게 적절히 설정해줘야 함

nfs에 연결될 마운트 디렉토리

Step5: NFS

The image shows two terminal windows. The top window is titled 'root's X desktop (localhost,localdomain:1)' and shows a terminal session on 'root@localhost:/nfs'. The user runs 'ls -a' and then 'vi CanYouSeeMe?'. After saving and exiting, 'ls -a' is run again, showing the newly created file 'CanYouSeeMe?' with permissions '-rw-r--r--' and size 53. The bottom window is titled 'root@localhost:~' and shows a terminal session on 'root@2410AMI /'. The user runs 'cd /mnt/nfs' and then 'ls -a', which shows the file 'CanYouSeeMe?' with permissions '-rw-r--r--' and size 53, confirming it is accessible from the target board.

Host에서 만든 파일이 target board에서도 보이는 것을 확인 할 수 있다

Step5: NFS

```
root's X desktop (localhost,localdomain:1)
root@localhost:/nfs
[root@localhost nfs]# ls -la
.
..
[root@localhost nfs]# vi CanYouSeeMe?
[root@localhost nfs]# ls -la
합계 12
drwxrwxrwx  2 root    root    4096  9월 11 18:47 .
drwxr-xr-x 23 root    root    4096  9월 11 18:18 ..
-rw-r--r--  1 root    root     53  9월 11 18:47 CanYouSeeMe?
[root@localhost nfs]#

root@localhost:~
[root@2410AM /]$ cd /mnt/nfs
[root@2410AM nfs]$ ls -la
total 9
drwxrwxrwx  2 root    root    4096  2004년 9월 11 18:47 .
drwxr-xr-x  4 root    root    4096  2004년 1월 1 00:02 ..
-rw-r--r--  1 root    root     53  2004년 9월 11 18:47 CanYouSeeMe?
[root@2410AM nfs]$ cd ..
[root@2410AM mnt]$ pwd
/mnt
[root@2410AM mnt]$ umount nfs
[root@2410AM mnt]$ cd nfs
[root@2410AM nfs]$ ls -la
total 2
drwxr-xr-x  2 root    root    4096  2004년 1월 1 00:02 .
drwxr-xr-x  4 root    root    4096  2004년 1월 1 00:02 ..
[root@2410AM nfs]$
```

반대로, **umount** 한 후에는 **host**의 파일을 볼수 없게 된다

[root@2410AM nfs]\$ umount nfs
[root@2410AM mnt]\$ cd nfs
[root@2410AM nfs]\$ ls -la

생각해 볼 사항

- Cross compiler가 없다면?
- 보드의 CD가 없다면?