A Learning Environment for Embedded Systems

Lain-Jinn Hwang*, Tang-Hsun Tu†, Tzu-Hao Lin‡, and I-Ting Kuo§
Department of Computer Science and Information Engineering
Tamkang University, Tamshui, 251, Taipei, Taiwan
**E-mail: micro@mail.tku.edu.tw*
†*E-mail: war3_515@hotmail.com*
‡*E-mail: singy000@msn.com*
§*E-mail: yiting2002@hotmail.com*

Abstract

We build an environment that provides many people connect to and use it to practice how to operate the system. While the server on, anyone could through the way of remote login to connect to server and know the system environment. It does not need to spend a lot of time on trifle such as walking or driving to learning center or school for one-by-one class. We can learn free with all resource we have because we are home. It not only solves the question of high system cost, but also increases the efficiency on learning.

Keyword – embedded system, learning free, learning enviroment, remote login.

1 Introduction

Recently, the application of embedded system is more and more widely used in a lot of field. Examples of embedded systems range from electronic thermometer, automobile automatic navigation to real-time control for subsystems in the space shuttle. Therefore, it is important that study embedded system as an information technology researcher.

For every university, the development and application of embedded system field has emphasized by degrees. There are more and more lessons about embedded system have taught in many universities. The environment of learning embedded system is one of the important conditions of training information researchers. Nevertheless, with the expensive embedded system hardware, it is a handicap to many students, also it is unknown to general people that how to get the hardware. Beside, it is impossible to give every student a device in the course concerning embedded system because of the expensive cost (Figure 1). Even if an embedded system course, students usually use the machine pool, and they must take turns to operate a machine. In addition, most students may not afford it if those device malfunction during class. Even they would not do any test out of class. To those students who want to learn embedded system, they must study with high cost. In conclusion, the learning time is limited. For embedded system hardware use, it is extravagant that a device allot to a student. Thus, how to improve the work efficiency of hardware is this project focus on.

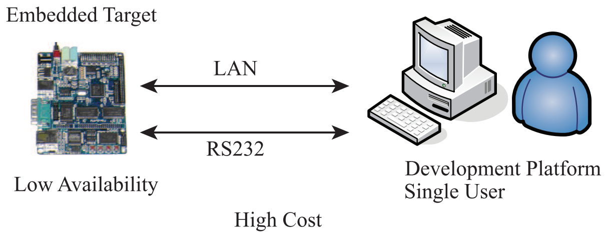


Figure 1: Traditional Learning Environment of Embedded System

2 Building Embedded System Learning Enviroment

We can make the maximum utilization if we could implement a platform [1], [2] (Figure 2) that supply the function such that multi-user could connect at the same time and share the same system resource [3]. In this way, we could not only use remote login to the machine anytime and anywhere, but also allow many people login in at the same time. For learner, it certainly reduces the learning threshold, and

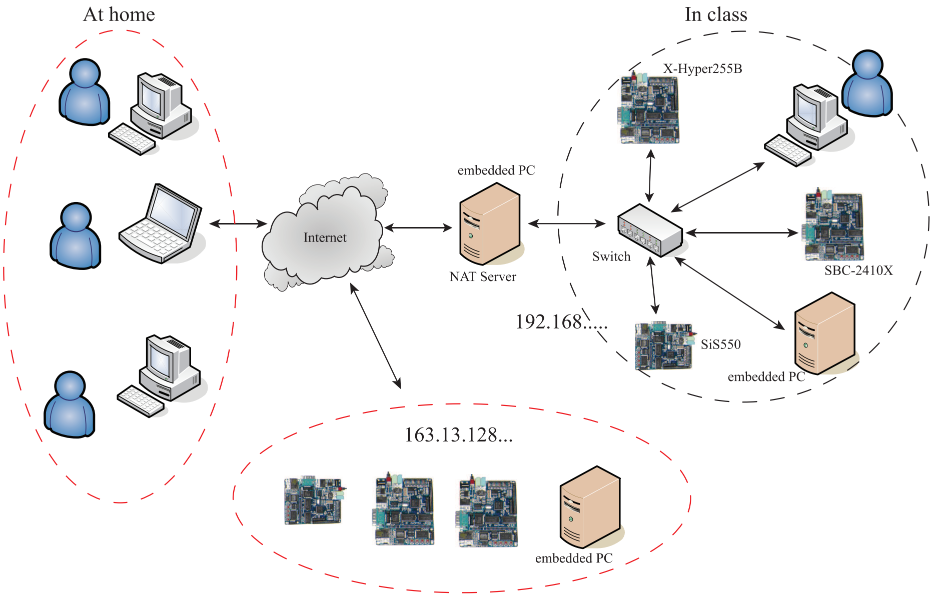


Figure 2: Our Learning Environment for Embedded Systems

the system utility rate is improved. What this plan does is providing a platform in which can remote login via Internet for many students, and could execute, test and modify the source code after compiling, and so on.

2.1 Hardware Platform

The hardware we are used to build open learning environment including NAT server, ARM9, ARM10 [4] and X86 embedded system. We choose K6-700MHz as NAT server because it only transmits packet to other machine, and there is no need to use powerful PC as NAT server. Nevertheless, NAT server still occupies an important position in the open learning system. Under the conditions of limited resource via NAT server translate virtual IP address that overlap with outside address, it does justice to what the hardware could work on.

2.1.1 SBC-2410x [5], [6]

SBC-2410x is section inserts the type system platform based on ARM9, the use processor is continues the ARM7 overhead construction processor after the Samsung, also develops ARM9 inserts the type system microprocessor, uses six PCB design, the series divides into ARM920T and the ARM940T two kinds. S3C2410X based on the core ARM920T, the interior has entire performance MMU (Memory Management Unit).

2.1.2 X-Hyper [7], [8], [9]

X-Hyper is an embedded system that is based on ARM10, and made by the Hybus Korea Technology Co., Ltd. The processor X-Hyper uses is the Micro Processor Intel Xscale PXA255. In ARM10 series, there are ARM1020E, ARM1022E and ARM1026EJ-S three kinds of cores. X-Hyper is used ARM1020E and the interior has entire performance MMU, DSP. Some PDA core is based on this kind of processor. In addition, there has Bluetooth and touch screen support. By the way, X-Hyper has a selling point in high-performance with low power in cell phone market.

2.1.3 SiS550 [10], [11]

SiS550 is designed by Taiwan Silicon Integrated Systems Technology Co., Ltd. SiS550 provides a high performance and low cost x86 compatible processor. Not same as normal x86 PC, it could not run any program that can run on desktop computer. We are using cross-compiler to learn how to work on it.

2.1.4 Desktop PC K6-700MHz

This machine is be used to be a NAT server. In our open learning embedded system environment plan, NAT server is an important rule if you want to login every different embedded system, it is for sure that we have to connect to this server first. Without this server, all we have done will plough the sand.

2.2 Software Platform

On the software, using open source Linux to build the work system, serve as contrast Linux the merit, such as kernel is smaller, and boot up speed is quick. Beside, there are some useful software offered on different SoC (System on Chip) platform [12], just like Busybox [13], Binutils[14], uClibc [15].. etc.

The software we use as follow:

1. Operating System: Linux Kernel 2.4.18 [16]

   Linux is a Unix-like computer operating system. Originally, Linux refers to Linux kernel, not means any one of the operating systems, it is only a part that can be the system kernel. Thanks to there are many programmers maintain and modify Linux such that it is becoming better and approaching perfection day by day. Now, Linux distribution has many software package bundled and it is more and more easy to use for anyone. Here, the kernel we use is version 2.4.18.
2. Library: uClibc 0.9.28 [15]

   We do not use Glibc, because of the space it takes is too much, there are some library are unused. So we use uClibc instead not only the space is quite small, but it also has libraries we need.
3. Cross Compiler: GCC 3.4.5 [17]

   The GNU Compiler Collection (shortened to GCC) is a set of programming language compilers produced by the GNU Project. Because of its high portability and compatibility, it has been spread and used widely. The size of GCC component is quite small. It is an advantage for that programming on embedded system.
4. BusyBox

   BusyBox is a computer program combining many standard UNIX utilities into a single small executable. It is typically used in a single-floppy or embedded Linux system because of its small size. And it is free software, licensed under the terms of the GNU General Public License.
5. VSftpd [18]

   Full name is "Very secure FTP daemon". It is a very secure, fast and stable FTP server. It has licensed under the GNU General Public License. There are some famous FTP site serve users and work under this software, such as ftp.redhat.com. Vsftpd server response speed is purported that it is two times of wuftpd server.

3 Building Network Enviroment

To building the multi-user environment, we could not use the way that past people used. Early days, the device name not be called "embedded system", people not transferred data to embedded system on Internet. They usually programmed on PC then used COM port to transfer data. After programming, they just burned the file to flash ram to save their data. Because Internet was undeveloped then, they had no choice even if the way was troublesome. Today, we can finish it without this so troublesome way. After following step, we just need connect with RJ45 port, login embedded system when connect to the Internet.

3.1 Build NAT Server

We need NAT server as a bridge between user and every different embedded system. Of course, we can use build the environment without NAT server if there are enough resource, such that every machine has its own public IP. Under a limited resource condition, if there is only an IP address but we have many embedded systems, NAT server will be helpful to us. Besides, NAT server can be a firewall for Local Area Network. It is also safe more than direct connect to other machine (Figure 2).

On server phase, we select Linux as operating system. Linux had been historically using mainly as a server operating system, but its low cost, flexibility, and UNIX background make it suitable for a wide range of applications. Therefore, we use Linux as an operating system of NAT server.

3.2 Install Hardware Device

Certainly, hardware has to install then just could work correctly. There are some ways could build our environment, we usually consider the way as Figure 2. Here, we assume that is only a public IP address, no matter it is fixed IP or dynamic IP. As Figure 2, we use Linux as NAT server, and it has two network cards, one card for Internet, the other one for Local Area Network. Other machine are shared the same public IP to the outer network and have their own private IP address to inner network. In other words, Linux is like a bridge between different platforms. If we want to login those machine through Internet, it is must pass Linux server.

The advantage of this placement is that we can only operate the NAT server for network management. It is no need to worry about where the machine is and how to configure it because all preference setting has done when beginning.

3.3 Configure Network Setting

After hardware has installed, we begin to set the network configuration. When entering the command mode, we can input "route" to show the IP routing table message. If there are two network interface cards, we could set eth0 a private IP such as 192.168.1.2 for inner network, and set eth1 that used to connect to outer network such as ADSL. Additionally, if we want to open some service on another machine, we have to close corresponding port. Besides, the client must set gateway in addition, we do not mention how to set the details here.

4 What Can A Learning Environment Do

Of course, the purpose why we try to build open learning environment is based on the reason for learning embedded system any place and any time. In view of this point, we consider that learning through Internet is the best way to make. In particular, network has been an important and developed technology, there are too many information about technology or study will concern with what we use Internet to do. Therefore, we can learn many unlike techniques through Internet.

Another, open learning platform can offer a multi-user environment, and users can login to do their job at same time. The best advantage is that we could login at free time such as night, or midnight even. We do not stop our job because of there is no time at work, or stop our learning due to out of class. In other words, we can use this open learning platform to do the job that is needed to take a long time.

4.1 Learning Linux System

Recently, many people begin to learn Linux system because there are more and more company concerning computer science or information technology ask their employee to learn how to use Linux system. Those employers usually hire the man who has known Linux operation. For some company without sufficient fund, they may choose Linux as computer system instead of windows, or maybe they need a stable and dependable server. Anyway, Linux operating system has become another popular choice. The more Linux is popular, the more people want to learn Linux. Therefore, this platform can offer many people who are dying to learn Linux to be familiar with the operating environment. For students, they do not need to spend money especially to buy a machine then install Linux and try to find out how to make it work out of class, they can know all the basic instruction they need on embedded system.

4.2 Web Development

Although the resource is limited on embedded system, we can still build a web server on it. There are many people can build a web or FTP server to share their own photo or diary through Internet, and they may design a web page on their own to show how their web pages are funny and pretty. To some student, maybe they just want to learn web design such as PHP or new technique AJAX. With open learning platform, they can learn how to design the special effects and implement some practical function, and then maintain the web server together.

4.3 Application Porting

To most computers, machine code is usually compatible. For example, we can execute the program on CPU that is IA-32 such as Athlon or Pentium4, or we can run the same program on the same operating system. We cannot run the same program on two different OS or hardware. For instance, we can not run media player for desktop computer to other platform as ARM and play it, because the two platforms are different, their object code are not same.

For compatibility and portability, programmers usually use cross-compiler compile the source code again make it supported to other platform and make it executable, we call the process as “porting”. If we want to run program on embedded system, it is essential to learn how to port program between different platforms. Under this environment, we can port any process to each different hardware or software platform freely via NAT server, it is convenient to develop an application or test the compatibility on different platform or debug for some errors. Further, this environment offers an interactive platform to let students have the chance to know what teamwork means.

5 Experimental results

We build a leanring enviroment as Figure 3, users can login to the embedded system from the NAT server. They can do something which we talk about in section 4.

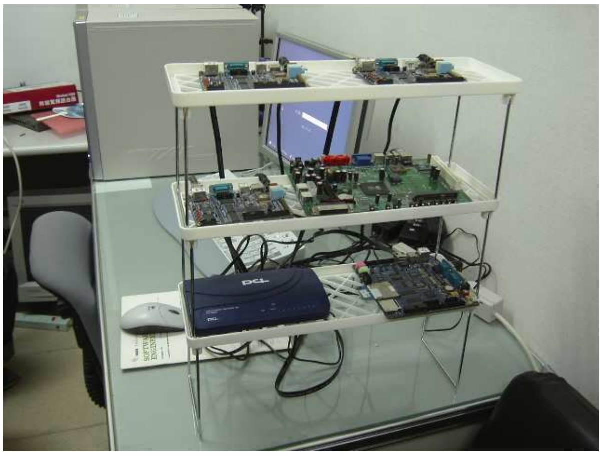


Figure 3: Our Embedded System Learning Environment

Time\Day	**Mon.**	**Tue.**	**Wes.**	**Thu.**	**Fri.**	**Sat.**	**Sun.**
0:00～8:00	Relax or Schedule					Relax or Schedule	
8:00～12:00	Teaching Time, Face to Face						
12:00～13:00	Relax or Schedule						
	Schedule						
13:00～24:00	Crilit Singy Hopeblue Yiting	Hopeblue Yiting Tom Frank	Tom Frank Jack Helen	Jack Helen John Sam	John Sam Crilit Singy		

Table 1: Learning-Time Schedule

In teaching,for instance, if there were ten students who want to learn embedded system in the class, we can plan a schedule as Table 1 for our learning enviroment.

There are ten students who are Crilit, Singy, Hopeblue, Yiting, Tom, Jack, John, Frank, Helen and Sam. Teacher can teach them face to face in the morning. After teaching, students can learn the embedded system at home in distance learning, teacher can schedule the time for learning like Table 1. For beginner, we can plan to group for six to ten people at the same time for learning (for example, we can see that multi-user login in Figure 5 and its home directory and kernel version). If users don't running high computing power program, the embedded system is accommodated. Besides, students also can login at saturday and sunday or any free time.

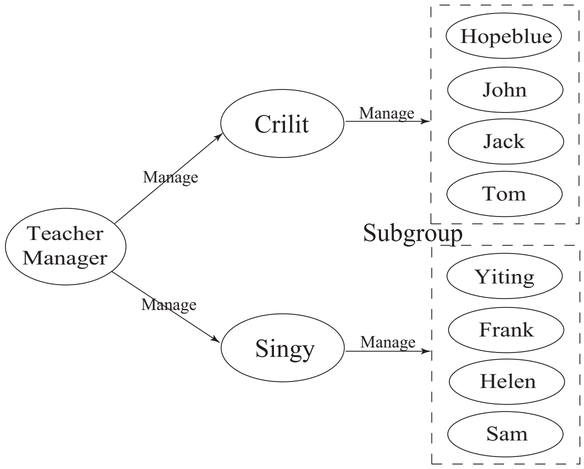


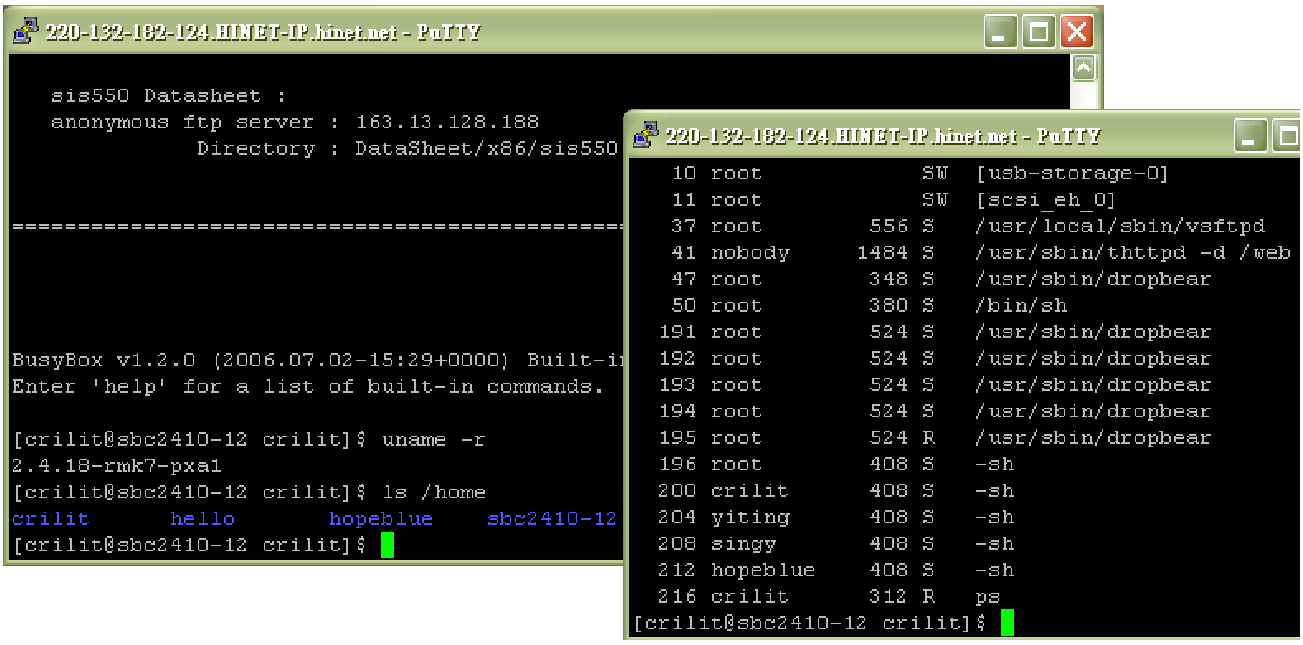
Figure 4: Account for Management

If a class which has more that fourty students, the management of accounts is take too much time for only one teacher or manager. To solve this problem, we can divide the groups to some small subgroups and choice a leader of subgroups for manage. For instance, we divide ten students of Table 1 for two subgroups in Figure 4 to manage. So, teacher or manager only need to manage the leaders of subgroups.

This learning enviroment is suitable for beginners or middle-level users. We also can reduce the students of group to learn, if need. Here we compare with traditional enviroment with our lenrning enviroment in Table 2:

Enviroment	Traditional	Our Enviroment
Learning Type	One PC to One Embedded	More at the same time
Learning Time	Fixed	Any Time (It can be schedule.)
Place for Learning	In Class Only	Distance learning (SSH, SFTP..etc)
Management	Need to manage place and machine	Group for some people and one for leader to manage
Space	Large	Small
Cost	High	Low
Suitable Users	Advance, Expert	Beginner, Middle-Level

Table 2: Compare with Learning Environment



220-132-182-124.HINET-IP.hinet.net - PuTTY
220-132-182-124.HINET-IP.hinet.net - PuTTY

```
   sis550 Datasheet :
   anonymous ftp server : 163.13.128.188
              Directory : DataSheet/x86/sis550


================================================



BusyBox v1.2.0 (2006.07.02-15:29+0000) Built-i
Enter 'help' for a list of built-in commands.

[crilit@sbc2410-12 crilit]$ uname -r
2.4.18-rmk7-pxa1
[crilit@sbc2410-12 crilit]$ ls /home
crilit      hello       hopeblue    sbc2410-12
[crilit@sbc2410-12 crilit]$
```

```
   10 root              SW   [usb-storage-0]
   11 root              SW   [scsi_eh_0]
   37 root          556 S    /usr/local/sbin/vsftpd
   41 nobody       1484 S    /usr/sbin/thttpd -d /web
   47 root          348 S    /usr/sbin/dropbear
   50 root          380 S    /bin/sh
  191 root          524 S    /usr/sbin/dropbear
  192 root          524 S    /usr/sbin/dropbear
  193 root          524 S    /usr/sbin/dropbear
  194 root          524 S    /usr/sbin/dropbear
  195 root          524 R    /usr/sbin/dropbear
  196 root          408 S    -sh
  200 crilit        408 S    -sh
  204 yiting        408 S    -sh
  208 singy         408 S    -sh
  212 hopeblue      408 S    -sh
  216 crilit        312 R    ps
[crilit@sbc2410-12 crilit]$
```

Figure 5: Multi-User Login for Embedded System

6 Conclusion

The enviroment which we built only use a few resources but offer multi-users to operate instructions synchronously. For user, it will become easier to learn how to make the system work. Beside, the users only need to modify or test their program on PC for remoting. By using cross-compiler, a single build environment can be maintained that would build for each different target platform. It is not only to save the whole cost in building embedded system, but also increase much study time for some user, because they can login free any time. We believe that the efficiency of free study will better then study in a limited time.

References

[1] C. Chien and S. Chuen-Tsai, "A computer-network-supported cooperative distance learning system for technical communication education," *IEEE Trans. Professional Communication*, vol. 39, pp. 205–214, Dec 1996.

[2] H. Rego, T. Moreiro, F. Morales, and H. Barbosa, "Educational technological specifications to support distance education in an e-learning platform," in *Proc. IEEE Int. Conf. Information Technology Based Higher Education and Training*, pp. F2C7–F2C12, Jul 2005.

[3] W. Horton, *Designing Web-Based Training (Book style with paper title and editor)*. Robert-Ipsen, 1993.

[4] "ARM: Advanced RISC Machine." http://www.arm.com.

[5] "Samsung-electronics." http://www.samsung.com/tw.

[6] Samsung-Electronics, *S3C2410A 200MHz & 256MHz 32-bit RISC Microprocessor User's Manual*, revision 1.0 ed.

[7] "HyBus." http://www.hybus.net.

[8] HyBus Co., *Embedded Linux System, X-Hyper255A Developer's Manual*, revision 1.0 ed., Jun 2003.

[9] HyBus Co., *Embedded System Application using X-Hyper255B-TKUIII based on Intel PXA255*, revision 1.0 ed., Aug 2004.

[10] "SiS: Silicon Integrated Systems Corp." http://www.sis.com.

[11] Silicon Integrated Systems Corp, *SiS55x System Design Guide*, revision 0.9 ed., Mar 2001.

[12] K. Yaghmour, *Building Embedded Linux Systems*. O'Reilly and Associates, 2003.

[13] "Busybox." http://www.busybox.net.

[14] "Binutils: GNU Binary Utilities." http://www.gnu.org/software/binutils.

[15] "uClibc: A C library for embedded Linux." http://www.uclibc.org.

[16] "Linux Kernel." http://www.kernel.org.

[17] "GCC: the GNU Compiler Collection." http://gcc.gnu.org.

[18] "VSftpd." http://vsftpd.beasts.org.