Int. J. Engng Ed. Vol. 26, No. 4, pp. 950–962, 2010
Printed in Great Britain.

0949-149X/91 $3.00+0.00

An Open Source and Network-based Remote Laboratory for Embedded Systems*

ZHOU RUI[1,2,3], ZHOU QINGGUO[1] †, CHENG GUANGHUI[1,3], WANG BAOJUN[1], LI LIAN[1] and NICHOLAS McGUIRE[1]

[1] *Engineering Research Center of Open Source Software and Real-Time System, Ministry of Education, Lanzhou University, P. R. China, 730000. E-mail: zhouqg@lzu.edu.cn*

[2] *Department of Computer Engineering, Polytechnic University of Valencia, Valencia, Spain, 46022*

[3] *School of Mathematics and Statistics, Lanzhou University, China, 730000*

The engineering education for embedded systems requires a suitable environment for practices and experiments. This kind of environment usually consists of a lot of hardware and software, so there is a pressing need to organize and manage all resources in a proper way to enable convenient access. In this paper, we present the design and implementation of a remote laboratory, aiming at the engineering education for embedded systems. This remote laboratory is built based on open source software and network technologies. It enables teachers to organize theoretical and practical contents in an easy way and students to communicate with teachers and take practical operations conveniently. The core of this remote laboratory is for students to learn in groups or by individual with the related resources as the interactive medium to guide, prompt and assist the learning and training procedure. With this environment, students can be located remotely while access real hardware devices easily. Meanwhile, it keeps the embedded development setup as a whole in good maintenance.

Keywords: remote laboratory; embedded systems; open source; network-based

1. INTRODUCTION

IN THE LAST DECADES, the development of Information Technologies and network has prompted new possibilities and new challenges for designing a network-based environment for engineering education for embedded systems. The design elements should facilitate instruction delivery, interaction, quality of learning, and user support, especially the support for remote experimentation with multiple users (students, researchers, and developers) of embedded systems. To satisfy these features, we have built up a remote laboratory in our institute for both of the students taking courses in embedded systems and researchers working on embedded systems. From the point of view of technology-based education, the engineering education for embedded systems is the combined area of engineering, computer science, integrated circuitry, communication technology, control systems, and software etc. It not only requires students to have adequate theory accumulation and comprehension, but also a great deal of experience of hands-on practice by exercising with real hardware [1].

The ability to combine practical applications with control of real hardware and manipulation of software is the advantage of technology-based education [2]. To build an educational environment for embedded systems, both hardware and software are indispensable. The environment includes not only typical embedded developing boards, e.g. PowerPC and MIPS etc., but also various kinds of other devices, e.g. PCs, switches, oscilloscopes, and signal generators etc., and necessary related software configurations. How to manage and maintain all of these resources as well as possible is the key issue to guarantee that this educational environment will be comfortable and convenient [3]. Remote laboratories have emerged as helpful educating tools and been prompted by the rapid development of network technology. In remote laboratories, equipment can be made available via the network environment and serve either as cost-effective replacements for or as complements to traditional laboratory measurements [4]. The remote laboratory presented in this paper enables users to achieve enough practical experience and theory learning of embedded systems. This paper describes the hardware and software organization of the remote laboratory and the diverse working flow and processing components. The design and implementation are based on entirely open source solutions. For one thing, open source solutions reduce the

* Accepted 19 February 2010.

† Corresponding author.

total costs of the remote laboratory because the software adopted can be retrieved freely. Also, open source provides more extensibility and flexibility for the remote laboratory than commercially constrained software. This enables more potential innovations to improve the remote laboratory.

Nowadays, open source software combined with network technologies has been rapidly progressing. It is now extensively used as a connectivity and reference tool for educational, commercial, and personal purposes. In education, it reveals various new methodologies for increasing the experience of learning as well as educational opportunities for a great number of students. Especially, remote education and nontraditional classrooms have the capability to contain more students while satisfying all functionalities of conventional classrooms and laboratory sessions [5].

To create an educational environment for embedded systems, one conventional way is to locate things distributedly: each student sits in front of a desk on which there is a host (e.g. a PC) connected with a target (e.g. an embedded developing board) [6]. In this scenario, it makes sense for everyone to do his/her own business. But considering the commonly high price of embedded devices, the more students there are, the more devices are needed, and the higher cost there will be for the laboratory. What is more serious, with the increasing number of students and limited budget, it is difficult to guarantee each student practice with one set of devices. This is a common situation that could degrade the hands-on learning experience significantly. Meanwhile, there is no way of accessing devices once people are out of laboratory. Moreover, the maintenance of a great many devices is also troublesome. So it is helpful to propose some solutions to make the environment more comfortable and convenient for both users and administrators. To build a remote laboratory could be an alternative to overcome the disadvantages of the traditional way.

In more detail, we implemented the remote laboratory on top of Gentoo Linux distribution with powerful Linux tools, shell scripts, and C programs. For practical work, via the network, users can remotely and conveniently access all of the hardware involved in this environment at any time. While for theory learning and interpersonal communication, there are also functional components. Users can customize their own plans or configurations in this environment to fit their own schedules. At present, the remote laboratory guarantees efficient and maximum usage of limited resources, especially hardware, and shows good extensibility for future development. This remote laboratory has been continuously innovating embedded system courses and research with positive progress in our institute.

The other parts of this paper are organized as follows. Section 2 lists related work on existing implementations of other remote education environments. Section 3 describes the whole idea of our remote laboratory. Section 4 focuses on the implementation of the remote laboratory. Section 5 presents some application samples. Section 6 gives conclusions and states future work.

2. RELATED WORK

The design and implementation of this remote laboratory involve several research and application aspects, including embedded systems, remote access, open source software, online communication, and various network technologies etc. The main goals of such an environment are [1]:

1. The seamless combination of theory learning and practical experiments for users;
2. The valid integration of collaboration and individual learning;
3. The harmonic online communication between users;
4. The effective management and maintenance of embedded systems and other related hardware/software;
5. Co-existence of in-situ laboratory and remote education or distributed learning;
6. Offering remote access and manipulation for real embedded laboratory equipment and instrumentation.

There have been various researches on remote education environments. For engineering education, this kind of environment is used to break the limit of access to laboratory equipment and gain hands-on experience. A typical example is the ReLAX project, which aims to study the feasibility of making remote experimentation available as a component in distance learning [7]. The Monash University of Australia has built a flexible distance learning system allowing students to access an automation laboratory in real-time from a remote location [8]. There is a virtual laboratory system for frequency modulation experiments [5], and a system combined with remotely controlled instrumentation and digital design experiments named SelfLab@Home [9]. CoWeb is a distance learning technology used in over 100 classes at Georgia Institute of Technology for engineering, mathematics and some computer science courses [10]. DYNACORE, an EU funded project, provides users with a powerful tool for remote collaboration in experiments or observations requiring one or more remote facilities [11]. A 3D online learning system has been designed for operating virtual instruments with circuit-measuring function of embedded systems [12]. The S.J College of Engineering in India provides support for remote experimentation on an embedded system based on the graphical programming interface tool of LabVIEW with an embedded development board [2]. A remote and virtual laboratory system for embedded development based on Windows has been implemented at Feng Chia University [13]. There are also remote laboratories for control engineering in [14] [15] [16], and [17], and for robotics in [18] [19] [20] [21], and [22]. The Cybernetics Laboratory is a commercial system of processing remote experimentation and control [23]. The University of Ulster in UK has developed a remote-access laboratory for an electronics and software system course [24]. And MPCRL (Mechatronics/Process Control Remote Laboratory) is funded by the United States National Science Foundation [25]. There is a remote access laboratory for electrical circuit experiments [26], a remotely accessed HVAC (Heating, Ventilating, and Air Conditioning) laboratory for distance education [27], and a virtual course system [28] based on the WebCT e-Learning platform including interactive tools and a remote laboratory for greenhouse climate control and fertirrigation teaching/learning. The University of Maryland has developed a remote laboratory demonstration system for control and communication [29]. The Christian Brothers University has created an integrated laboratory environment combining the traditional CAD (Computer-Aided Design) with parametric design and 3D modeling [30]. A virtual learning environment based on web technology has been developed at Trinity College Dublin and applied to the development of a course in database software engineering [31]. [32] presents some developments of interactive web-based teaching for computing in an engineering degree. The PUNCH infrastructure of Purdue University has allowed educators to share and disseminate specialized tools and educational materials to offer classes in distance education environments [33]. The S.m.i.L.E project of Germany aims to provide a web-based and interactive course program for teaching electrical engineering [34]. [35], and [36] present two virtual laboratories focusing on the performance of robotics. WILEDS is a web-based teaching and learning environment for digital systems [37]. [38] and [39] describe virtual laboratories for analog electronics and electrical machinery. MicroLab is a web-based multi-user remote microcontroller laboratory designed for electric-electronic engineering education for experiments over CAN (Control Area Network) [40]. [41] describes the development and evaluation of a virtual laboratory in material science. An industrial communication networks laboratory provides a distributed industrial environment for laboratory sessions in [42]. A two-degrees-of-freedom helicopter can be manipulated by a web-based remote laboratory in Singapore [43]. [44] proposes an Internet-based mechatronics laboratory platform for distance learning in engineering education. [45] develops a web-based laboratory for the exercise of an electrical simulator of typical physical process in the field of automatic control. A portal of e-Learning environment has been utilized in a process synthesis course for chemical engineering education [46]. The University of Pisa makes use of LabVIEW to develop a virtual educational laboratory allowing the simulation of communication systems, equipment, devices, and measuring instruments, and remote control of the physical instruments via a standard IEEE 488 bus [47]. The Solar e-lab provides remote laboratories for the learning and training of a solar energy system over the Internet [48]. The eLaboratory is a convergence of remote access technologies and collaboration-based e-Learning for aerospace engineering [49].

These projects or systems have contributed a lot to remote training or learning in engineering education. But in consideration of embedded systems, most of these implementations are not dedicated. They commonly focus on other aspects of engineering education, e.g. control, mechatronics, robotics, and electronics etc. Although some projects aim at embedded systems in engineering education, such as [2], [12], [13], and [24]. The problem is that they tend to be crude in nature and fail to fully utilize existing design, test and debug software and equipment for embedded systems. Therefore, they are not designed and implemented flexibly or compatibly enough to contain more hardware and software in the future. A common issue is that the related software for these existing implementations is usually based on commercial packages, which leads to high costs and also less flexibility for customized configuration. In this paper, we present a remote education environment specific for learning and training, and researching and developing for embedded systems. The implementation follows a pure open source method, which guarantees the flexibility, extensibility, and compatibility of the whole environment.

3. SYSTEM ARCHITECTURE

The development of network and communication technologies has the potentiality to transform remote education into professional systems. The engineering education for embedded systems contains not only theory learning, but also a lot of hands-on experiments. Therefore, the fundamental function of this remote laboratory is to guarantee that, in this environment, students can conveniently learn from online courses and experiment with remotely accessed real hardware, and researchers can rapidly develop related projects with enough referencing resources at hand. Moreover, this remote laboratory makes the learning and experimenting activities beyond temporal and spatial limits, and enhances the effect of learning, developing, and researching to the maximum extent with limited hardware resources.

The remote laboratory consists of four major components (Fig. 1):

1. Remote Lab Portal: this component is built upon Gentoo Linux and developed by Moodle (Modular Object-Oriented Dynamic Learning Environment). Moodle is a free and open source CMS (Course Management System). It is designed with sound pedagogical principles,

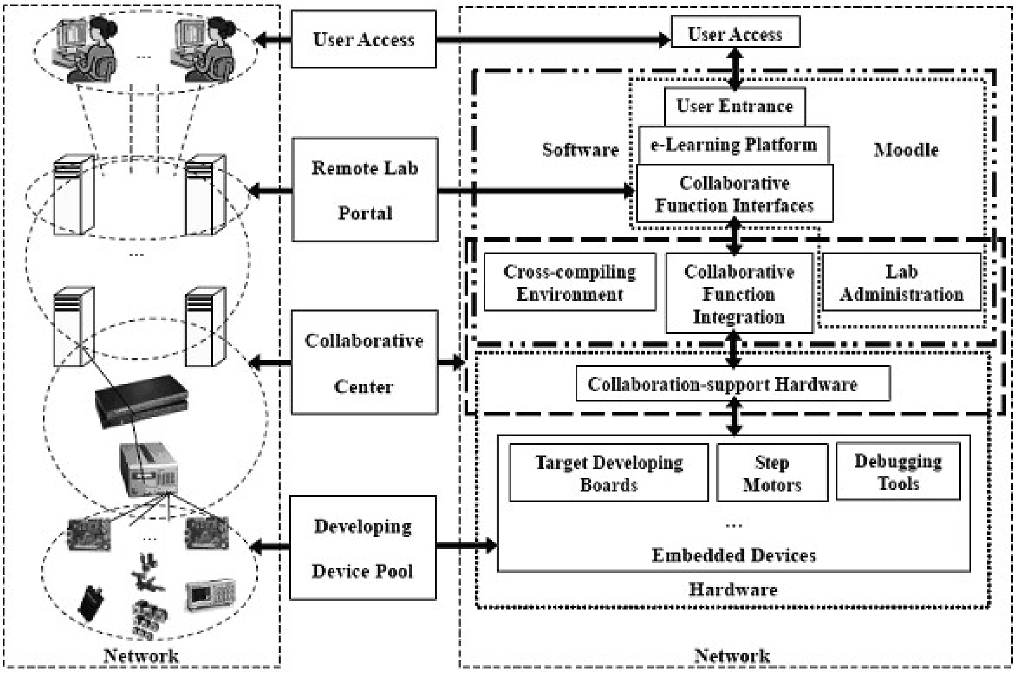

Fig. 1. Remote laboratory architecture.

to help educators create effective online learning communities. Users can download and use it on any computer they have handy (including web hosts), yet it can scale from a single-teacher site to a university with 200,000 students [50]. Moodle promotes a social constructionist pedagogy (collaboration, activities, critical reflection, etc). It is suitable for 100% online classes as well as supplementing face-to-face learning. It has a simple, lightweight, efficient, compatible, and low-tech browser interface [51]. Here we enhance Moodle as the core software environment for remote education in the laboratory. This portal includes the following functions:

(a) User Entrance, allows users to access the remote laboratory via a uniformed interface. Users should login to the laboratory from here.
(b) Collaborative Function Interfaces, combine Moodle with Gentoo Linux to enable synchronized manipulations by Linux commands and software invocations. These interfaces are linked directly to Collaboration Center, which is another important component of the laboratory.
(c) E-Learning Platform, is the main part for theory learning and human-computer/human collaboration in the remote laboratory. In this platform, there are plenty of embedded systems-related resources including ebooks, slides, papers, video clips, audio clips etc. They are categorized and tagged in Moodle for convenient search and selection. What is more important is the interpersonal collaboration among users, such as between students and teachers. For this platform, Moodle implements the functions of collaboration with communication services (chats, workshops), community services (forums, wiki, glossaries), teaching services (assignments, quizzes, surveys), management services (courses, resources), and individual services (calendars, plans), etc. Meanwhile, users of this environment can access the real devices through Collaborative Function Interfaces. This platform is the core part in Remote Lab Portal.

2. Collaborative Center: this center is directly linked with Remote Lab Portal through software interfaces. It contains the software configurations that extend a channel of human-computer collaboration to real embedded devices. There is also fundamental hardware support to implement remote manipulations with real embedded devices. The cooperation of hardware and software in this center guarantees collaborative learning and manipulation between different users and real embedded devices. This component is transparent for common users, but critical for administrators of the laboratory.
   (a) Lab Administration, is the privileged function for administrators of the remote laboratory. It is in charge of the configuration and maintenance of the whole environment. It is implemented as the combination of Moodle functions and Linux system administration. It features functions such as:
      i. Device management, checks status of embedded devices, including both hardware and software. Once abnor-

mal condition happens, administrators will be informed immediately.

ii. User management, includes the functions of user registration, user authentication and user authorization, granting the permission to online resources in e-Learning Platform and real embedded devices.

iii. Service management, organizes and maintains diverse services provided by e-Learning Platform.

iv. Configuration management, takes charge of system-level software configuration. It includes not only the status of Moodle, but also the software functions of the following Collaborative Function Integration and Cross-compiling Environment

(b) Collaborative Function Integration integrates together the necessary Linux system commands, and frequently used tools and services in embedded systems, such as shell, minicom, nfs, telnet or ssh. Different users in e-Learning Platform can invoke these functions via Collaborative Function Interfaces to access and manipulate real embedded devices, with simultaneous online communication and cooperation. This portion also guarantees software extensibility for the remote laboratory. The pure open source implementation enables additional software to be integrated and configured into the environment conveniently. For example, EMC (Enhanced Machine Controller) [52], an open source CNC (Computer Numerical Control) software for our CNC machine tool control, has been integrated into the remote laboratory.

(c) Cross-compiling Environment, is indispensable for embedded devices of different hardware architectures. To guarantee the correctness and stability, this part is configured and maintained by administrators, but it is available for all users to utilize. Once there is something wrong with Cross-compiling Environment, administrators can be informed by users via communication services of e-Learning Platform, or by Lab Administration.

(d) Collaboration-supported Hardware, is responsible for communicating with the embedded devices for users' requirements. Only software is not enough to guarantee the collaboration of the remote laboratory, and the mechanism adopted is the cooperation of both software and hardware. The function of such hardware needs software configuration from Collaborative Function Integration to enable remote access.

3. Developing Device Pool: this component is composed of various types of embedded developing devices. The real purpose of human-computer collaboration of the remote laboratory is to implement multi-users' manipulations in this pool. This pool is connected with Collaboration-supported Hardware, and also parceled by the network environment of the laboratory. If budget permits, this pool can be expanded with more devices, such as developing boards, debugging tools, and something like oscilloscopes, signal generators, etc.
4. User Access: this component, utilizing web browser, is the first step to enter the remote laboratory. It is connected with User Entrance directly.

The whole organization of the remote laboratory is in an entirely open way—not only open source software is used, but also open and extensible architecture and mechanism. E.g., new devices can be added to the environment in Developing Device Pool and connected to Collaborative Center. Also, different services of Remote Lab Portal and Collaborative Center can be distributed into several specific machines with Developing Device Pool expanding to a large scale. The key point for remote education in this environment is the implementation of human-computer/human collaboration, and dedicated interaction with the real embedded devices. Meanwhile, multi users' manipulations are combined with synchronized communication, instruction, guidance and correction, which can improve the performance of the whole learning and training procedure.

5. System Implementation: the implementation can be divided into Hardware Organization and Software Configuration.

Hardware Organization

1. It is very common to connect hosts and target-embedded devices via UART when working with embedded systems. Also, for many other tools such as BDI2000 hardware debugger, UART is necessary. Collaboration-supported hardware must maintain the connections of devices in Developing Device Pool, provide extensibility for the pool in case of the quantity of devices increasing, and enable orderly user access to devices in the pool. Currently, a USB-to-Serial device (MOXA UPort 1650-16 16-port RS-232/422/485 USB-to-Serial Hub) is playing the role of this part. The USB plug and play allows easy serial port expansion without requiring IRQ, DMA, or I/O address resources. Users no longer need to open the chassis or power down the system with this device, saving on setup time and cost [53]. With the support of different serial port standards, it also enables the remote laboratory compatible with diverse embedded devices. This device is connected to Remote Lab Portal server via USB and then each serial port of it is linked to one embedded device.

In the remote laboratory, access to target-embedded devices is decided by the permission

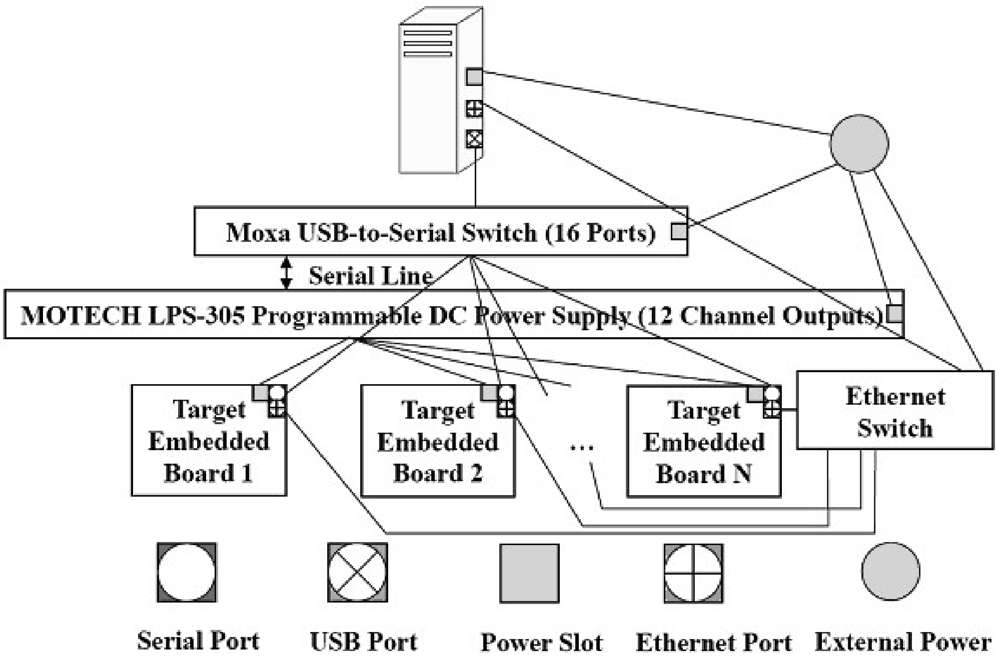

Fig. 2. Collaboration-supported hardware organization.

to serial ports of the USB-to-Serial Hub. The recently released Linux kernels support the hub to add 16 serial ports as system devices under /dev in the form of /dev/ttyMXUSB0 to /dev/ttyMXUSB15.

Another issue is that it is very common to reset target-embedded devices in Developing Device Pool during practical experiment, but in the remote laboratory, the embedded devices are not in the hands of users, so a power control program (Fig. 3) is created for the collaborative manipulation. This program can control a programmable power switch (MOTECH LPS-305 Programmable DC Power Supply), which contains 12-bit D/A converter to enable 12 channels of output.

In the remote laboratory, /dev/ttyMXUSB0 is specified to turn on/off and reset the selected devices. The power switch connects to the serial port of the hub corresponding to /dev/ttyMXUSB0. For power control, users only need to connect /dev/ttyMXUSB0 via minicom or other serial port accessing tools, and then execute the power control program to choose the embedded devices that need turning on/off from the 12 channels (Fig. 4). While, to access and manipulate different embedded devices, users must connect other ports (/dev/ttyMXUSB1—/dev/ttyMXUSB12) to the corresponding devices (Table 1).

```
static struct port_cmd_dispatcher port_cmds[] = {
        {{"on", 1, }, {handle_cmd_on, NULL, NULL,},},
        {{"off", 1, }, {handle_cmd_off, NULL, NULL,},},
        {{"reset", 1, }, {handle_cmd_reset, NULL, NULL,},},
        {{"stat", 0, }, {handle_cmd_stat, NULL, NULL,},},
        {{"help", 0, }, {handle_cmd_help, NULL, NULL,},},
};

static int tty_io(int fd, char* in)
static int send_cmd(const char* cmd, int port)
int handle_cmd_on(void* param)
int handle_cmd_off(void* param)
int handle_cmd_reset(void* param)
int handle_cmd_stat(void* param)
int handle_cmd_help(void* param)
static int parse_args(int argc, char* argv[])
static int do_main(int argc, char* argv[])
static void usage(void)
```

Fig. 3. Power control program.

The server for Remote Lab Portal is directly linked with Collaboration-supported Hardware via USB, and all of the devices in Developing Device Pool are connected together by Ethernet switch. So once users login the laboratory, they can access the embedded devices in a common serial port way, and also by telnet or ssh if the systems on the embedded devices provide such services.

2. Developing Device Pool, is a collection of various types of embedded devices. Organization and management of these devices are the key functions for this pool. This pool is organized in an open and extensible mechanism. At present, the embedded devices are connected directly with Collaboration-

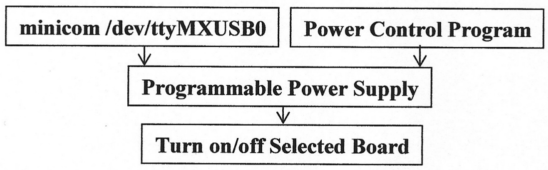

Fig. 4. Power control flow.

Table 1. USB-to-serial and power switch configuration

Port	Status	Baud Rate	Power Switch Channel
/dev/ttyMXUSB0	*Power Control*	*19200*	N/A
/dev/ttyMXUSB1	*PowerPC Board #1*	*115200*	*1*
/dev/ttyMXUSB2	*PowerPC Board #2*	*115200*	*2*
/dev/ttyMXUSB3	*PowerPC Board #3*	*115200*	*3*
/dev/ttyMXUSB4	*PowerPC Board #4*	*115200*	*4*
/dev/ttyMXUSB5	*VIA Board #1*	*115200*	*5*
/dev/ttyMXUSB6	*VIA Board #2*	*115200*	*6*
/dev/ttyMXUSB7	*VIA Board #3*	*115200*	*7*
/dev/ttyMXUSB8	*VIA Board #4*	*115200*	*8*
/dev/ttyMXUSB9	*MIPS Board #1*	*115200*	*9*
/dev/ttyMXUSB10	*MIPS Board #2*	*115200*	*10*
/dev/ttyMXUSB11	*MIPS Board #3*	*115200*	*11*
/dev/ttyMXUSB12	*MIPS Board #4*	*115200*	*12*
/dev/ttyMXUSB13	Not Configured	N/A	N/A
/dev/ttyMXUSB14	Not Configured	N/A	N/A
/dev/ttyMXUSB15	Not Configured	N/A	N/A

supported hardware via UART. So if Collaboration-supported hardware can provide more connections, this pool can be enlarged. Meanwhile, embedded devices themselves usually provide various types of interfaces and pins, e.g. USB, parallel port, wireless, and GPIO etc., with which we also can exploit another way to expand this pool.

Software configuration

1. Moodle is the core part of Software Configuration in the remote laboratory. It entirely covers Remote Lab Portal, and is responsible for Lab Administration. It is the most fundamental component for remote education in the environment. Users will take their first step with the mighty functions of Moodle. We have configured Moodle as the bridge for users to touch real embedded devices. In such a remote education environment, different users play different roles: administrators, teachers, students and developers etc., with different authorities. System administration is achieved via the combination of Moodle administration and Gentoo Linux system administration. Moodle user management synchronizes with Gentoo Linux's, to guarantee registered users to be permitted to remotely access necessary resources properly, e.g. USB port or serial port. This is part of user management involving user registration, user authentication and user authorization. The privileged functions for administrators are responsible for system configuration and maintenance, including device maintenance, user management, categorizing e-Learning resources and real hardware, etc. In Moodle, Collaborative Function Interfaces are respnsible for software integration and convenient usage of the functions in Collaborative Function Integration (Fig. 5, Fig. 6 and Fig. 7). Meanwhile the traditional stand-alone console prompt is still welcome by advanced Linux users. Therefore, in the e-Learning Platform, users can enjoy Linux features for embedded systems directly.
2. Software parts of Collaborative Center enable users to access and manipulate embedded devices in Developing Device Pool. It also offers privileged authorities for administrators:
   (1) A tool for accessing embedded device via serial port is indispensable. In the remote laboratory, Gentoo Linux contains minicom (Fig. 4), which is widely used to provide a console for embedded system. It provides an interactive communication between the host and target embedded devices via serial ports, and performs functions such as loading compiled kernel image, testing file system, configuring kernel command line for booting, and accessing the shell (if there is) in the operating system of embedded devices. Also, other similar tools compatible with Linux can be introduced and utilized.

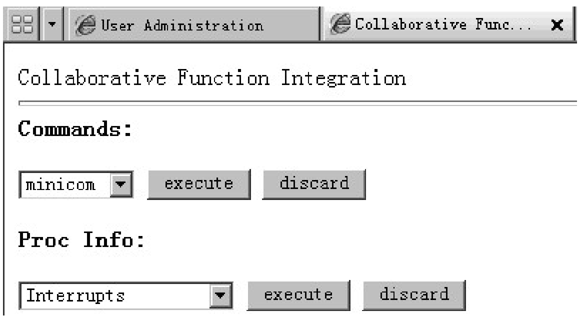

Fig. 5. Collaborative Function Integration extended in Moodle combined with Linux.

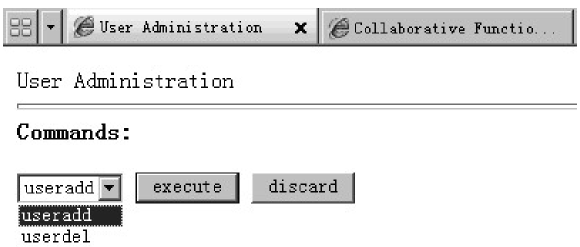

Fig. 6. User administration extended in Moodle combined with Linux.

```
if [ $CMD -eq 1 ] ; then
    echo "minicom:"
    /user/bin/minicom -swm -C log
fi

if [ $CMD -eq 2 ] ; then
    echo "Disk Usage:"
    /bin/df
fi

if [ $CMD -eq 3 ] ; then
    echo "Network Device Setting:"
    /sbin/ifconfig
fi

if [ $CMD -eq 4 ] ; then
    echo "Basic System Information:"
    /bin/uname -a
fi

if [ $CMD -eq 5 ] ; then
    echo "Process Running:"
    /bin/ps
fi

if [ $CMD -eq 6 ] ; then
    echo "Add User:"
    /usr/sbin/useradd
fi

if [ $CMD -eq 7 ] ; then
    echo "Delete User:"
    /usr/sbin/userdel
fi
```

Fig. 7. CGI implementation for Collaborative Function Integration and user administration.

(2) Cross-compiling Environment of the remote laboratory can be built directly from Gentoo Linux sources, which is one of the shining features of Gentoo. The non-x86 architectures (e.g. PowerPC, ARM and MIPS etc.) are common for embedded devices, so appropriately built cross-compiling environments are indispensable for compiling matching with these various architectures. Other open source solutions, such as ELDK (Embedded Linux Development Kit) [54] and crosstool [55] also can build this environment. Different Cross-compiling Environments for different architectures can be utilized by different users at the same time for compiling different objects.

(3) NFS (Network File System) is another frequently used tool in the learning and training of embedded systems. It maps directories of the hosts to the target embedded devices over networks, especially for the root file systems. Remote Lab Portal enables the NFS server to share directories, at least mapping one directory containing a basic file system to per target board as the root file system. These shared embedded file systems are originally built from Gentoo Linux sources according to the targets' architectures. These embedded file systems also can be built in other ways, such as the combination of busybox [56] and ELDK. The systems in embedded devices can be accessed by multi users via Ethernet, which provides the collaborative environment for a group to develop one project in one embedded device.

(4) Telnet/ssh is highly necessary for the remote laboratory. These tools enable convenient ways for human-computer collaboration, which means users can access and manipulate embedded devices conveniently via Ethernet.

(5) Software for advanced development/research, e.g. EMC for CNC machine tool control, can be added into Collaborative Center.

4. LEARNING AND TEACHING IMPACTS

Our institute has been holding embedded control courses and annual summer practical training for the undergraduates of our university since 2008. The remote laboratory is being used by students enrolled in these courses and also some researchers in our institute (Table 2).

For registered users, they can login via User

Table 2. Service condition of remote laboratory

User Type	Number of Users	Groups	Device Collections	Working Time
Students in 2008 summer practical training	18	6	4 PowerPC Boards 4 VIA Boards 4 MIPS Boards 1 USB-to-Serial Switch 1 Power Switch 1 BDI2000 Debugger 1 CNC Machine Tool 4 Stepper Motors 1 Oscilloscope 1 Signal Generator 1 Logic Detector	3.3 Hours/Week/Person
Students in 2009 embedded control courses	28	7		10.5 Hours/Week/Person
Students in 2009 summer practical training	36	9		8.4 Hours/Week/Person
Teachers	4	N/A		12.5 Hours/Week/Person
Institute researchers	6	N/A		16.8 Hours/Week/Person

Entrance of Remote Lab Portal. In the remote laboratory, they can benefit in the embedded control courses from remote learning and training in the following ways:

1. For common users:
   (1) Online classroom: online learning, video and audio;
   (2) Interactive communication: online chats, internal mails, offline messages and online workshops;
   (3) Community: group forums (different groups have different projects to do in these courses), project wikis (to organize and manage materials of group projects) and glossaries;
   (4) Resources: course resource (slides, papers and device manuals); acquisition/exchange/sharing, and personal learning resource (self-collecting course related materials) distribution;
   (5) Experiments: remote access (enter the laboratory remotely), collaboration (collaboration among members in one group to finish group project, collaboration among different groups for discussion and communication, teachers' instructions or guidance for students and students' consultation to teachers) and device sharing (members of one group using the same device and different groups sharing some number limited devices or large-scaled devices);
   (6) Individual: personal calendars, personal plans, evaluation results, online reports and scheduling for the entire learning procedure in these courses.
2. Additional for teachers:
   Teaching: online/offline assignments, online quizzes/tests/examinations, surveys, course resource distribution, student evaluation and responsibility for the entire teaching procedure in these courses.
3. Additional for administrators:
   Administration: user management, resource categorizing, device maintenance and responsibility for the whole environment.

Students involved in embedded control courses begin the experiments with theoretical knowledge, which can be achieved in online classroom of e-Learning Platform. Afterwards, the students are assigned with experiments in two parts:

1. General experiments: in Developing Device Pool, there are various devices like embedded developing boards, stepper motors, driver control modules and CNC machine tools etc. During the general experiments, the students will be taught to get familiar with the hardware, and the related software issue. The contents of the experiments include to build a file system for a PowerPC board and to compose EMC programs for the machine tools, etc.

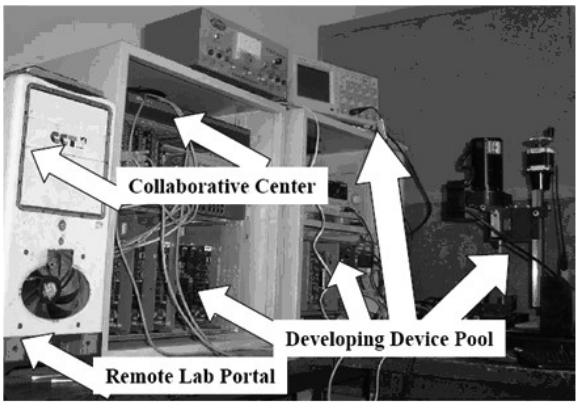

Fig. 8. Physical devices of remote laboratory.

2. Group projects: this part provides comprehensive practice for the students in groups. In the remote laboratory, embedded developing boards are linked to driver control modules via GPIO pins, which connected with stepper motors or CNC machine tools (Fig. 8). Meanwhile, oscilloscopes and signal generators are linked with different pins on the boards to monitor control signals. For debugging, BDI2000 is connected to boards via JTAG. The grouped students are required to perform control actions on the stepper motors or machine tools with embedded devices. The students should connect to embedded devices via serial port by minicom or via network by telnet/ssh. For stepper motors, they should program GPIO to send instructions to driver control modules that will manipulate stepper motors directly. For machine tools, the case is more complicated. They should compose codes in EMC. The EMC program controls the actions of the machine tool. Once the program is executed, EMC sends it through network to the selected embedded devices, which supply necessary real-time performance and scheduling for the whole control procedure (e.g. PowerPC440EP Yosemite with Xenomai RTOS running). Then the embedded board will activate the driver control module that generates signals to the machine tool to work as the EMC program design (Fig. 9). This part of experiment is based on the general part. It aims to enable students to solve practical problems. The main idea is to use embedded devices to control stepper motors or the machine tool properly. It requires the combined knowledge of embedded systems, RTOS and control theory, etc.

After the courses and training, the students have acquired adequate theoretical knowledge and practical experience to know the fundamental work for development with embedded systems. For the teachers, they are willing to use the new technological methods to improve classroom effects. We have collected user feedback for the remote laboratory since 2008 (Figs 10–13). More than 70% of

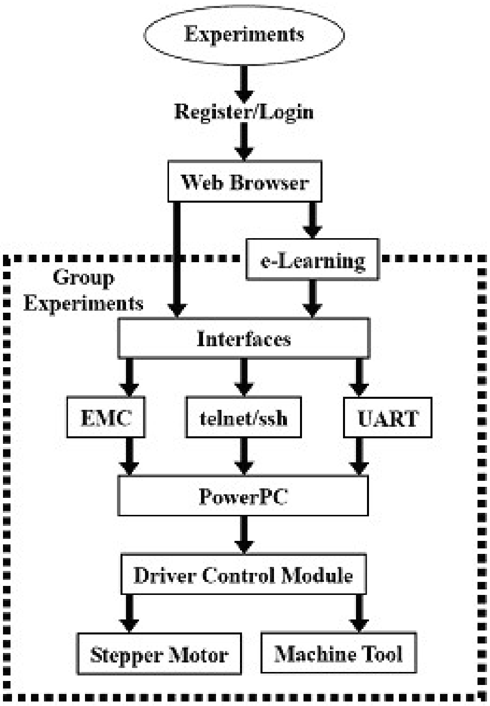

Fig. 9. Group experiments.

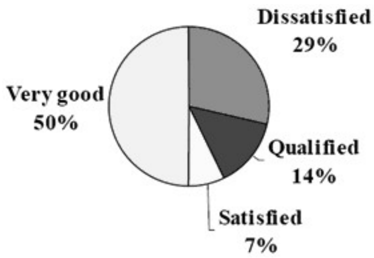

Fig. 10. User rating of students in 2008 summer practical training.

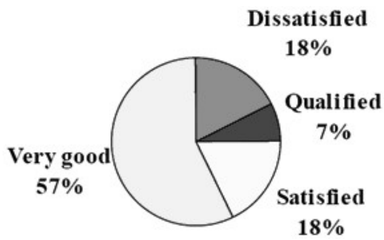

Fig. 11. User rating of students in 2009 embedded control courses.

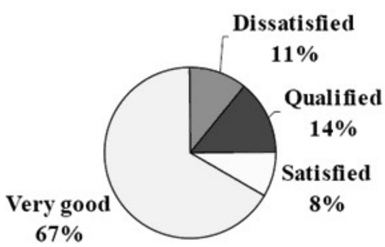

Fig. 12. User rating of students in 2009 summer practical training.

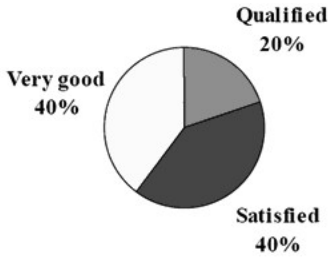

Fig. 13. User rating of teachers and institute researchers.

users, including students, teachers and researchers, have rated the environment positively. The disadvantages according to the feedback should be considered to include the possible time lags when accessing the environment out of the university via the Internet, the current limited quantity of some necessary hardware such as the debugger and machine tool, etc.

5. CONCLUSIONS

In this paper we present our remote laboratory for engineering education of embedded systems based on open source solutions and network technologies. We design the environment originally for students, teachers and researchers involved in embedded systems, with better collaboration, organization and management. After preliminary implementation and utilization, we have received satisfactory feedbacks from different users. The current implementation of the remote laboratory enables related students, teachers and researchers to work well together. They can conveniently arrange their own learning and working progress in the environment, experiment on their new creative ideas about embedded systems, interact with each other for some common goals. Especially, all of these users are now able to experiment and gain hands-on experience of embedded devices according to their own schedule and progress on an individual basis with minimum device occupation conflicts. And experiments can be done via the Internet connection even when they are out of the physical laboratory.

Meanwhile, future work is necessary to improve the remote laboratory as far as it goes. Updating functions of Collaborative Function Integration in Moodle based on newly issued requirements needs to continue. In case of enlargement of the environment, it is necessary to dispatch functions and services of the Remote Lab Portal and Collaborative Center, e.g. Cross-compiling Environment, NFS server, etc., to different servers for more stability. Also Moodle Network is a new feature added in the latest release of Moodle [50]. The network feature allows a Moodle administrator to establish a link with another Moodle node and to share resources with users of both Moodle nodes. This is helpful to enhance content and knowledge sharing, and enable the remote laboratory to extend into a large scale distributed collaborative education environment. Also, it is better to increase the number of devices such as the hardware debugger and machine tool.

Acknowledgement—This work was supported by National Natural Science Foundation of China under Grant No. 60973137, National High Technology Research and Development Program of China (863 Program) under Grant No. 2009AA01A138 and Gansu International Sci.&Tech. Cooperation under Grant No. 090WCGA891.

REFERENCES

1. E. Damiani, F. Frati and D. Rebeccani, The Open Source Virtual Lab: a Case Study, *Proc. of the Workshop on Free and Open Source Learning Environments and Tools (FOSLET06)*, Lugano, Switzerland, 2006, pp. 5–12.
2. Jagadeesh Chandra A. P. and R. D. Sudhaker Samuel, Design of a Real-Time On-Line Web-Based Collaborative Learning Environment for Embedded Applications, *Proc. of the 2nd Workshop on Blended Learning (WBL2008)*, Jinhua, China, 2008, pp. 46–54.
3. J. Lee, J. H. Cho, K. M. Park and J. S. Kim, Organization of Embedded System Development Methodology for Developing Product Family, *Proc. of Software Engineering (SE2006)*, Innsbruck, Austria, 2006, pp. 154–159.
4. P. Lundgren, K. O. Jeppson and Å. Ingerman, Lab on the Web—Looking at Different Ways of Experiencing Electronic Experiments, *Int. J. Eng. Educ.* **22**(2), 2006, pp. 308–314.
5. C. C. Ko, B. M. Chen, S. Y. Hu, V. Ramakrishnan, C. D. Cheng, Y. Zhuang and J. P. Chen, A Web-Based Virtual Laboratory on a Frequency Modulation Experiment, *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews,* **31**(3), 2001, pp. 295–303.
6. Z. Nedic, J. Machotkd and A. Najhlsk, Remote Laboratories Versus Virtual and Real Laboratories, *Proc. of the 33rd ASEE/IEEE Frontiers in Education Conference*, Boulder, CO, USA, 2003, pp. T3E-1–T3E-6.
7. C. S. Tzafestas, N. Palaiologou, and M. Alifragis, Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation, *IEEE Transactions on Education,* **49**(3), 2006, pp. 360–369.
8. J. Z. Zhang, A. K. Ball, M. Clare and W. Extine, Design of a Real-Time Remote-Access Engineering Laboratory Using Integrated Web Service and Wireless Technology for Distance Learners, *World Transactions on Engineering and Technology Education*, **4**(2), 2005, pp. 231–234.
9. C. Ciubotariu, G. Hancock, Work In Progress—Virtual Laboratory with a Remote Control Instrumentation Component, *Proc. of the 34th ASEE/IEEE Frontiers in Education Conference*, Savannah, GA, USA, 2004, pp. T3C-18–T3C-19.
10. M. Guzdial, P. Ludovice, M. Realff, T. Morley, K. Carroll and A. Ladak, The Challenge of Collaborative Learning in Engineering and Math, *Proc. of IEEE/ASEE Frontiers in Education Conference (FIE)*, Reno, NV, USA, 2001, pp. T3B-24–T3B-29.
11. J. A. Rodríguez, E. García, J. Rejas, M. A. Durán, Web-based Distributed Systems for Collaborative Remote Experiments, *Proc. of the 7th International Conference on Accelerator and Large Experimental Physics Control Systems*, Trieste, Italy, 1999, pp. 483–485.
12. F. C. Kao, Y. L. Tung and W. Y. Chang, The Design of 3D Virtual Collaborative Learning System with Circuit-Measuring Function, *Proc. of the 16th International Conference on Computers in Education (ICCE2008)*, Taipei, Taiwan, China, 2008, pp. 183–184.
13. Y. N. Hsu, A Virtual Laboratory of Embedded Systems for Use in Multiuser Environments, *M.S Degree Thesis*, Feng Chia University, Taiwan, China, 2005.
14. C. Chiliculita and L. Frangu, A Web Based Remote Control Laboratory, *Proc. of the 6th Multiconference on Systemic, Cybernetics and Informatics*, Orlando, Florida, USA, 2002.
15. C. S. Peek, O. D. Crisalle, S. Dépraz and D. Gillet, The Virtual Control Laboratory Paradigm: Architectural Design Requirements and Realization Through a DC-Motor Example, *Int. J. Eng. Educ.*, **21**(6), 2005, pp. 1134–1147.
16. C. Fernándes, M. A. Vicente and L. M. Jiménez, Virtual Laboratories for Control Education: a Combined Methodology, *Int. J. Eng. Educ.* **21**(6), 2005, pp. 1059–1067.
17. B. Duan, K. V. Ling, H. Mir, M. Hosseini and R. K. L. Gay, An Online Laboratory Framework for Control Engineering Courses, *Int. J. Eng. Educ.* **21**(6), 2005, pp. 1068–1075.
18. A. Bicchi, A. Coppelli, F. Quarto, L. Rizzo, F. Turchi, and A. Balestrino, Breaking the Lab's Walls—Tele-Laboratories at the University of Pisa, *Proc. of 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp. 1903–1908.
19. H. Temeltas, M. Gokasan and S. Bogosyan, Hardware in the Loop Robot Simulators for On-site and Remote Education in Robotics, *Int. J. Eng. Educ.* **22**(4), 2006, pp. 815–828.
20. F. Torres, F. A. Candelas, S. T. Puente, J. Pomares, P. Gil and F. G. Ortiz, Experiences with Virtual Environment and Remote Laboratory for Teaching and Learning Robotics at the University of Alicante, *Int. J. Eng. Educ.* **22**(4), 2006, pp. 766–776.
21. J. Noguez and L. E. Sucar, Intelligent Virtual Laboratory and Project-Oriented Learning for Teaching Mobile Robotics, *Int. J. Eng. Educ.* **22**(4), 2006, pp. 743–757.
22. A. R. S. Castellanos, L. H. Santana, E. Rubio, I. S. Ching and R. A. Santonja, Virtual and Remote Laboratory for Robot Manipulator Control Study, *Int. J. Eng. Educ.* **22**(4), 2006, pp. 702–710.
23. Cyberlab, http://www.cyberlab.org
24. M. J. Callaghan, J. Harkin, T. M. McGinnity and L. P. Maguire, An Internet-based Methodology for Remotely Accessed Embedded Systems, Proc. of *IEEE Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia, 2002, pp. 157–162.
25. H. Wong, V. Kapila and A. Tzes, Mechatronics/Process Control Remote Laboratory, *Proc. of 2001 American Society for Engineering Education Annual Conference & Exposition*, Albuquerque, NM, USA, 2001, pp. 25–28.
26. I. Gustavsson, A Remote Access Laboratory for Electrical Circuit Experiments, *Int. J. Eng. Educ.* **19**(3), 2003, pp. 409–419.

27. W. J. Hutzel, A Remotely Accessed HVAC Laboratory for Distance Education, *Int. J. Eng. Educ.* **18**(6), 2002, pp. 711–716.
28. F. Rodríguez, M. Berenguel, J. L. Guzmán and S. Dormido, A Virtual Course on Automation of Agricultural Systems, *Int. J. Eng. Educ.* **22**(6), 2006, pp. 1197–1210.
29. D. J. Pines and P. A. Lovell, A Remote Demonstration System to Enhance Engineering Classroom Instruction and Student Learning, *Int. J. Eng. Educ.* **14**(4), 1998, pp. 257–264.
30. Y. S. Shiue, B. B. Beard, M. L. Santi and J. E. Beaini, Integrated Laboratory for Manufacturing Education, *Int. J. Eng. Educ.* **15**(1), 1999, pp. 51–57.
31. V. P. Wade, J. B. Grimson and C. Power, WWW-based Educational Environments for Software Engineers, *Int. J. Eng. Educ.* **15**(2), 1999, pp. 130–136.
32. G. G. Roy and P. L. Lee, Interactive Web-based Teaching for Computing in an Engineering Degree, *Int. J. Eng. Educ.* **15**(5), 1999, pp. 358–364.
33. N. H. Kapadia, J. A. B. Fortes, M. S. Lundstrom and D. R. Valles, PUNCH: A Computing Portal for the Virtual University, *Int. J. Eng. Educ.* **17**(2), 2001, pp. 207–219.
34. H. Siemund and H. Goebel, A Web-assisted Electronics Course using the S.m.i.L.E Program, *Int. J. Eng. Educ.* **18**(6), 2002, pp. 736–744.
35. F. Naghdy, P. Vial and N. Taylor, Embedded Internet Laboratory, *Int. J. Eng. Educ.* **19**(3), 2003, pp. 427–432.
36. F. A. Candelas, S. T. Puente, F. Torres, F. G. Ortiz, P. Gil, and J. Pomares, A Virtual Laboratory for Teaching Robotics, *Int. J. Eng. Educ.* **19**(3), 2003, pp. 363–370.
37. A. A. Kassim, S. A. Kazi and S. Ranganath, A Web-based Intelligent Learning Environment for Digital Systems, *Int. J. Eng. Educ.* **20**(1), 2004, pp. 13–23.
38. M. J. Moure, M. D. Valdés, A. Salaverría and E. Mandado, Virtual Laboratory as a Tool to Improve the Effectiveness of Actual Laboratories, *Int. J. Eng. Educ.* **20**(2), 2004, pp. 188–192.
39. C. Elmas and M. A. Akcayol, Virtual Electrical Machinery Laboratory: A Fuzzy Logic Controller for Induction Motor Drives, *Int. J. Eng. Educ.* **20**(2), 2004, pp. 226–233.
40. A. Kutlu, MicroLab: A Web-based Multi-user Remote Microcontroller Laboratory for Engineering Education, *Int. J. Eng. Educ.* **20**(5), 2004, pp. 879–885.
41. J. Hashemi, K. A. Austin-Stalcup, E. E. Anderson, N. Chandrashekar and A. Majkowski, Elements of a Realistic Virtual Laboratory Experience in Materials Science: Development and Evaluation, *Int. J. Eng. Educ.* **21**(3), 2005, pp. 534–545.
42. A. Rosado-Muñoz, J. Muñoz-María, J. Calpe-Maravilla, J. Guerola-Tortosa and W. Blay-Corcho, An Industrial Communication Networks Laboratory for Distributed Automation Systems, *Int. J. Eng. Educ.* **21**(5), 2005, pp. 964–972.
43. C. C. Ko, B. M. Chen, J. P. Chen, J. Zhang and K. C. Tan, A Web-Based Laboratory on Control of a Two-Degrees-of-Freedom Helicopter, *Int. J. Eng. Educ.* **21**(6), 2005, pp. 1017–1030.
44. W. J. Shyr, Internet-Based Laboratory Platform for Distance Learning in Engineering Education, *Int. J. Eng. Educ.* **25**(4), 2009, pp. 693–700.
45. M. Stefanovic, M. Matijevic and V. Cvijetkovic, Web-Based Laboratories for Distance Learning, *Int. J. Eng. Educ.* **25**(5), 2009, pp. 1005–1012.
46. M. Krajnc, E-learning Environment Integration in the Chemical Engineering Educational Process, *Int. J. Eng. Educ.* **25**(2), 2009, pp. 349–357.
47. A. N. D'andrea, F. Giannetti, G. Manara, M. Michelini and P. Nepa, A Virtual Educational Laboratory for Telecommunications Engineering, *Int. J. Eng. Educ.* **24**(1), 2008, pp. 144–152.
48. I. M. Michaelides and P. C. Eleftheriou, Using Engineering Remote Laboratories to Enhance Student Learning—a Distributed Learning Experience, *Int. J. Eng. Educ.* **25**(3), 2009, pp. 577–584.
49. M. G. Helander and M. R. Emami, Engineering eLaboratories: Integration of Remote Access and eCollaboration, *Int. J. Eng. Educ.* **24**(3), 2008, pp. 466–479.
50. Moodle: http://moodle.org/ (Accessed 29 January 2010).
51. P. G. Raj and P. R. Gupta, Development of Collaborative Learning Environment Using LMS, *Proc. of Annual Seminar of C-DAC Noida Technologies*, Noida, India, 2009, pp. 133–140.
52. EMC, http://www.linuxcnc.org/ (Accessed 29 January 2010).
53. MOXA, http://www.moxa.com/product/UPort_1610-16_1650-16.htm (Accessed 29 January 2010).
54. ELDK, http://www.denx.de/ (Accessed 29 January 2010).
55. crosstool, http://www.kegel.com/crosstool/ (Accessed 29 January 2010).
56. busybox, http://www.busybox.net/ (Accessed 29 January 2010).

Zhou Rui received the B.S. degree in Computer Science from School of Information Science and Engineering, Lanzhou University, China in 2004. During 2004–2006, he was major in Computer Software Theory for MS. Once he finished his study, he was selected as a Ph.D. candidate in Applied Mathematics, School of Mathematics and Statistics, Lanzhou University since September 2006. Since February 2008, he has beeen a joint educated Ph.D. student of Universidad Politécnica de Valencia, Spain and Lanzhou University. He is researching embedded and real-time systems in Open Source Software and Real-Time System Engineering Researching Center of Ministry of Education of the People's Republic of China and Distributed and Embedded System Lab of Lanzhou University.

Zhou Qingguo received the B.S. and MS degrees in Condensed Matter Physics from the School of Physical Science and Technology, Lanzhou University in 1996 and 2001 respectively, and Ph.D. degree in Theoretical Physics from School of Physical Science and Technology, Lanzhou University in 2005. He is currently an Associated Researcher of School of Information Science and Engineering, Lanzhou University, the Vice Director of Open Source Software and Real-Time System Engineering Researching Center of Ministry of Education of the People's Republic of China and the Director of Distributed and Embedded System Lab of Lanzhou University.

Cheng Guanghui received the B.S. degree from School of Life Science, Lanzhou University in 2003, and MS degree from School of Information Science, Lanzhou University in 2007. He has been a Ph.D. candidate in Applied Mathematics, School of Mathematics and Statistics, Lanzhou University since September 2007.

Wang Baojun received the B.S. degree from School of Chemistry, Lanzhou University in 2004. He is currently a research fellow of Distributed and Embedded System Lab, Lanzhou University.

Li Lian received the MS degree in Algebra from Lanzhou University in 1982. He is currently a professor of School of Information Science and Engineering, Lanzhou University.

Nicholas McGuire is currently a Cuiying Guest professor of Lanzhou University.