6

Networked Embedded Systems – Example Applications in the Educational Environment

Fernando Lopes[1,2] and Inácio Fonseca[1]
[1]*Instituto Superior de Engenharia de Coimbra*
[2]*Telecommunication Institute*
Portugal

1. Introduction

In this chapter we present a network architecture and two field application examples involving the deployment of embedded systems for distributed applications in the educational environment. The main contribution intended for this chapter is the proposed network architecture, with emphasis in the specific distributed implementation of networked embedded systems. The contribution also includes the detailed description of the technology options, the choices for the hardware and the development tools, as well as the implementation steps required to achieve a successful fully working system.

A very wide group of automation applications can benefit from embedded networked technology for more flexible and efficient implementations [1]. This technology allows for better integrated communications, integrated local and global control, supervision and maintenance. The educational environment has a set of management requirements that can be fulfilled using networked embedded systems and a well defined communication concept [1,2,3,4].

In this chapter we will present an applied research work using embedded systems. The work targets applications in the educational environment, specifically in the automated access control, access management and course management areas. Two field applications are presented, that allow demonstrating the suitability of the proposed networked embedded system concept for applications in these areas. These applications are fully deployed and in service in a higher education institution. The developed system concept can find applications in other fields such the residential, infrastructure and industrial areas.

A first example application, to be presented in this chapter, is in the automated access control and access management area, where distributed and networked intelligent systems interface with remote sensors to validate user credentials, granting or denying individual access to premises. The presented system uses a microcontroller with a standard Ethernet interface, to validate users presenting a compatible *Ibutton* or *RFID* Card. The user credentials are validated locally or in a central database. Authorized users are given access to reserved areas. By using a powerful database, very flexible and complex high level administration and management functions can be implemented. By sharing the local IP network infrastructure, maximum commonality and flexibility are achieved. Both microcontroller and database server software are fully implemented using Open Source developing tools.

A second example application to be described in this chapter deals with a Linux-based Embedded Identification Module (EIM), which is integrated in an Information System specifically tailored for course management. The Embedded Identification System allows for networked fingerprint and RFID reading and identification. The Information System permits ubiquitous network access to teachers and course directors, while tasks such as calendar and timetable setting and correction, student class attendance and class progress information are integrated and efficiently managed. Student and teacher identification is achieved through the networked Embedded Identification Systems with fingerprint and RFID sensors.

This chapter is organized in five sections. The first section consists of this introduction, presenting the global objectives and shortly describing the two example applications to be detailed. The second section will present an application and technology review in the embedded systems area with especial emphasis on development tools and operating systems. The third section will detail the first application example, featuring the architecture and a field implementation for a flexible and scalable distributed access control and access management system. The fourth section of the chapter will present the second application example, describing the hardware and software developed to create an information system specifically tailored to be used as a distributed course management application. The last section concludes the chapter and includes a short discussion on the creation of specific firmware and the future use of real-time operating systems.

2. Applications and technology review

Embedded systems can find applications in a very wide group of automation environments. These include, in addition to the ones in the educational management area presented in Section 1, a whole set in the industrial, residential, medical, economic, consumer and sensor networks areas.

In the industrial field, embedded systems can be widely used for fault monitoring and diagnosis, with automated reports, scheduling, triggering and registering specialized maintenance actions [5].

In the residential area, many future applications are envisaged such as intelligent appliances with automated management capabilities as for example the signalling of missing or out-of-date items while creating and automatically emailing or SMS-forwarding shopping lists and other required tasks. A very active research topic in the joint medical and residential areas is the concept of Ambient Assisted Living [6]. This research topic targets the home assistance to the elderly or autonomy impaired patients. The remote monitoring and interaction possibilities, allow for specialized individual care. The goal is to dramatically improve the autonomy and quality of life of these patients, by not requiring long and frequent hospital attendance.

The international tracking and quality control of goods can also be efficiently managed by using RFID Tags supported by a communication and information infrastructure. Border control and product taxing are examples of tasks that can highly benefit from the implementation of such technologies. Another area of special interest is animal farming, where sub-dermal transponders are used for identification, health monitoring and management, registering and transport control.

Embedded systems and embedded operating systems are now pervasive in consumer and network equipment such as mobile phones, PDAs, tablets, Webpads, VoIP phones, robots, audio and video appliances such as set-top-boxes, media players and televisions, thin client devices, switches and routers, among many other consumer examples.

Another specific application area, which has received special interest recently, is the ubiquitous deploying of networked embedded systems for the global monitoring of large civil strategic buildings and infrastructures. These include forests, sensitive ecosystems, power plants and power grids, government buildings and communication infrastructures such as bridges, motorways, sea ports, railways and airports.

Integrated with Identification, Wireless, Localization (GPS), and Internet technologies, networked embedded systems can be used to create true global monitoring, control and management solutions. A related and very prominent area of research in this field is the advanced concept of Sensor Networks [7].

According to [8] "Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. Embedded systems in many cases must be optimized for life-cycle and business-driven factors rather than for maximum computing throughput".

Other important issues to be faced by the embedded developer are: real-time components, adaptive real-time requirements, compilers and timing analysis, development and execution platforms, control for embedded systems, testing and verification tools and strategies [9]. In what concerns the real-time components, which are one of the greatest implementation challenges, there are design tools in which systems are designed by putting together pieces that might be termed components. Examples are *MetaH*, *Ptolemy* and *Metropolis* tools [9].

In [10] and [11] the authors present the state of the art in Real-Time Operating Systems (RTOS) for embedded platforms. Definition of Hard *versus* Soft Real-Time Systems is presented, together with the scheduling policies – preemptive, non preemptive – and the typical design workflow of a real-time system. Some examples of RTOS are described such as QNX (POSIX compatibility), VxWorks, Kurt-Linux, Red-Linux and Windows. In these types of Operating Systems (OS), virtualization can also be used to allow different OS running in the same hardware: typically one RTOS and Windows. The characteristics regarding the handling of highly demanding deadlines can classify the types of RTOS to be used in: hard real-time system, critical real-time system and safety-critical real-time system.

The steps required to build a Linux Embedded System from scratch are presented in [12]. The document shows how to build *toolchains*, how to use embedded *bootloaders*, how to perform setup and configuration among other required tasks. An important issue discussed in this reference is the recent advances in free and Open Source software for embedded systems. The author presents a comparison between the number of installed systems with

Windows Embedded and Linux Embedded for different device categories such as PDAs, mobile phones, VoIP phones, robots, audio and video devices, thin client devices, gateways, tablets and Webpads. A slightly higher score is achieved by Linux, with about 51% against 49% for the number of devices at the time of analysis. This comparison is based on reports published by online sites *linuxdevices.com* and *windowsfordevices.com*.

In [13] and [14] the authors present the steps towards the installation and use of a *toolchain* compiler for ARM LPC based processors. The ARM/MIPS architecture has been steadily growing in the community for microcontroller firmware development and is now widely used. Other architectures for microcontrollers used in embedded systems are mostly based on RISC or CISC concepts. Example manufactures are ST, Infineon, Intel, Zilog, Texas, NXP and Microchip among others.

In another segment of tools are those related to HDL and VHDL circuit synthesizing. In these cases the real-time project is typically programmed in C and the tool will synthesize the HDL code to be uploaded to a FPGA circuit. This kind of systems uses RTOS that are highly hardware based and that make intensive use of hardware resources [15].

An important class within embedded systems groups includes those categorized as "little embedded systems" which exhibit small amount of memory usage. In this group are 8bit, 16 bit and 32 bit microcontrollers with memory sizes ranging from less than 25 kBytes up to 32 kBytes and low clock frequencies, with a typical value of 100 MHz. In this category small and light RTOS are necessary such as FreeRTOS, Contiki and SalvoRTOS. Other examples outside this segment are FreeOSEK, Toppers and Trampoline [15].

The development of complete embedded hardware and firmware solutions for specific applications is a very demanding task with many aspects to be considered. Generally speaking an embedded system should respect [15]:

- *Functional specifications*: observing the physical values and collecting them in real-time. Since the state of the controlled object is a function of real-time, the observed values are only temporarily present within a time-limited window. The update can be triggered in two possible ways: periodically - time-triggered observation; events - change of state observation or event-triggered;
- *Temporal specifications*: guaranteeing that all tasks are performed within their defined relative time sequence and individual time constraints;
- *Reliability specifications*: the most important attributes for the measurement of reliability are: fault tolerance, maintenance, availability and security.

When developing the hardware, firmware and software for the networked systems to be presented in Section 3 and Section 4, these global and other more specific aspects were thoroughly considered in the choices that were made.

3. Embedded access control and access management

Different technologies can be used when designing an automated access control and management system. See [16,17,18] and for commercial systems [19]. A starting point is the decision on how individual users are identified. This identification can be based on the presentation of unique credentials such as manual codes or *RFID* Tags. The credentials can also be based on biometric patterns. The most common are fingerprints and iris prints.

Other forms of biometric data can be used for identification, such as gait, voice and face recognition. These latter forms, however, are much more processing intensive.

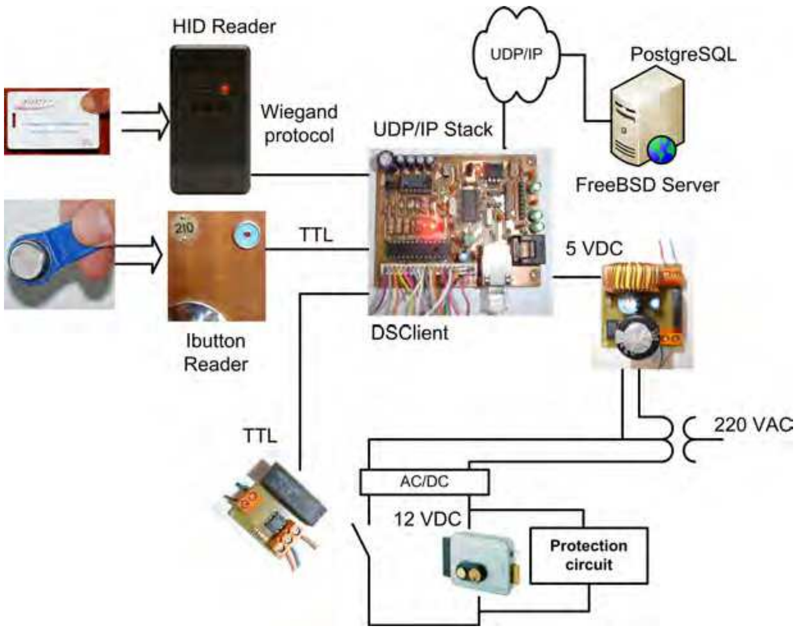


Fig. 1. Architecture of the developed access control system.

The second step is the decision on how individual identification credentials are validated. This action aims to answer the question: "Does this user have the permission to access this location?" The developed system infers identity through the presentation of an *RFID* Card or an *Ibutton*. Both technologies have in chip, hard stored, unique serial numbers, which can be used as identification credentials. Validation is performed using IP connectivity for the tentative matching of presented credentials with those stored in a local table or in a central *PostgreSQL* database server [20]. The complete architecture is shown in Figure 1. Noteworthy is the fact that the access system shares the local IP network infrastructure with existing services. This allows high hardware and software commonality, high flexibility, minimum deployment complexity and minimum cost.

In addition to the specific design of the *DSClient* hardware module, to be detailed in Section 3.1, the most significant decisions on the design of the presented distributed system are related to the communications infrastructure. Figure 1 includes a representation of the information flow between the various system components. The considered options for the IP communication architecture were the following:

- TCP/IP networking with HTTP 1.0 support. In this option the embedded system represented by the *DSClient* would remotely execute a PHP [21] script on the FreeBSD server [22]. This script would validate the user presented credentials. This simple setup has some advantages. In terms of the information flow only a TCP connection must be established. A disadvantage is the need to implement the TCP protocol. The HTTP protocol is not a difficulty in this approach, since it consists in the simple action of sending a read command, similar to a browser HTTP read;

- Use of TCP/IP networking with an active service running in the *FreeBSD* server. The running service would receive the user data from the *DSClient*, query the database for a matching and would answer the request. A clear advantage would be the simple protocol implementation, only requiring a socket definition and simple read and write operations. Disadvantages are the need to implement the TCP protocol and the additional need to establish a running service on the server;
- Use of UDP networking to send a validation request and wait for a probable answer in a defined timeframe. The main advantage of this solution is the simple UDP implementation and the fact that there is no need for the TCP implementation in firmware. A disadvantage is the absence of flow control. A service must also be setup on the server.

Given the above described options, a decision was made on the implementation of the simpler and faster UDP packet transmission when compared to TCP. Data encryption was also implemented with this approach. In addition, a simpler and faster *DSClient* can be built, without the need for the TCP firmware.

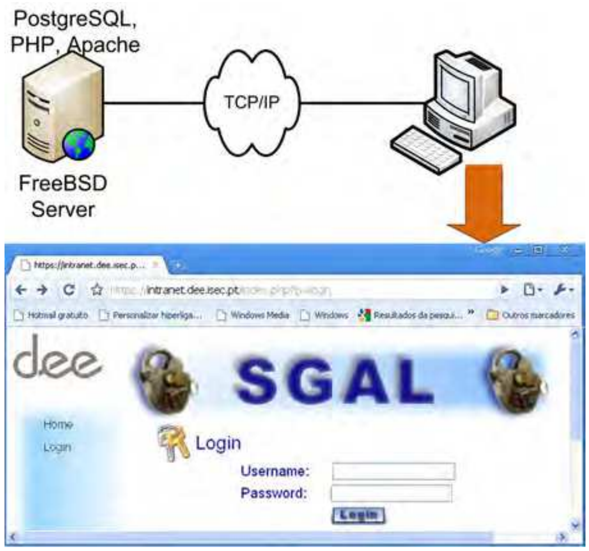


Fig. 2. Browser interface for the administration system.

In order to increase the security of the access control system, an additional packet identification technique was implemented, based on the positive matching between *DSClient* requests and server generated answers at the Layer 3 packet level. The request packet includes a security field computed using an elapsed time value and a generator polynomial. This field allows the positive identification of the server answer, guarantying the correspondence between the received packet and the sent UDP *DSClient* request. This technique avoids the non-authorized future use of the same answer packet, which may represent a positive validation, intended only for the ongoing query.

The high level administration of the system is accomplished using a Browser interface (Figure 2) (https://intranet.dee.isec.pt/SGAL_WEB_DEE). The interface consists of a set of WEB pages accessed through a secure connection. The pages are implemented using PHP language and an Apache server. The interface allows for the configuration and updating of the access database. With global administrator permissions the interface has the following main features:

- Add, Edit and Delete room administrators and associated fields;
- Add, Edit and Delete general users and associated fields;
- Add, Edit and Delete user permissions for individual rooms;
- Browsing and Exporting of registered accesses if enabled.

The main administration interface page is shown in Figure 2. It allows room administrator creation and editing; general user creation and editing; permission editing and general access management. SGAL relates to the initial name for the Web administration application which is, in Portuguese: "*Sistema de Gestão de Acesso a Laboratórios*".

It should be noted the intentional choice of Open Source software in this application, demonstrating the high potential of the available tools also in this field. The aforementioned embedded systems and many other small intelligent systems are also increasingly Linux based, with several examples previously given in Section 2.

3.1 Hardware and software implementation

The detailed implementation of the hardware considered a balance between the cost and dependability of the system. In a first prototype the Microchip *18F2685* microcontroller [23] was chosen, and was associated with the *ENC28J60* Ethernet Controller chip, also from Microchip. The *ENC28J60* implements the Ethernet physical layer at 10 Mbps in half or full-duplex modes. It has however some limitations negotiating with active network nodes such as some types of switches. The limitations reside on the speed and half/full duplex negotiation. It has a SPI control interface, available in most current microcontrollers, allowing very simple and cost effective solutions, which do not require high performance communications. The choice of the *18F2685* microcontroller was based on the availability of an on-chip CAN controller, which adds the flexibility to further associate hardware using the CAN interface. The simple hardware architecture for the *DSClient* module, presented in Figure 1, is shown in Figure 3.

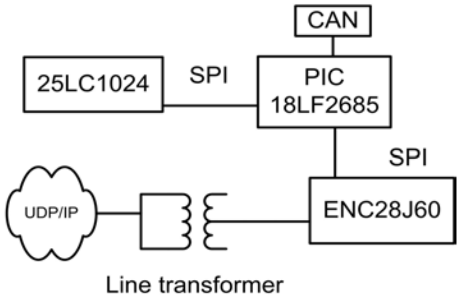


Fig. 3. Hardware architecture for the *DSClient* module.

A second prototype implemented the *DSClient* module using the PIC18F67J60 microcontroller, which includes an internal 10Mbps Ethernet controller compatible with the external ENC28J60 controller. Both prototypes include an external memory, using the 25LC1024 chip. This memory can be used to store encrypted credentials for the local room, providing redundancy for a temporary non-operational network. A third prototype uses an ARM Cortex-M3 [24] *Stellaris* microcontroller with integrated 100Mbps Ethernet, fully compatible in terms of behaviour with the two previous versions (Figure 4).

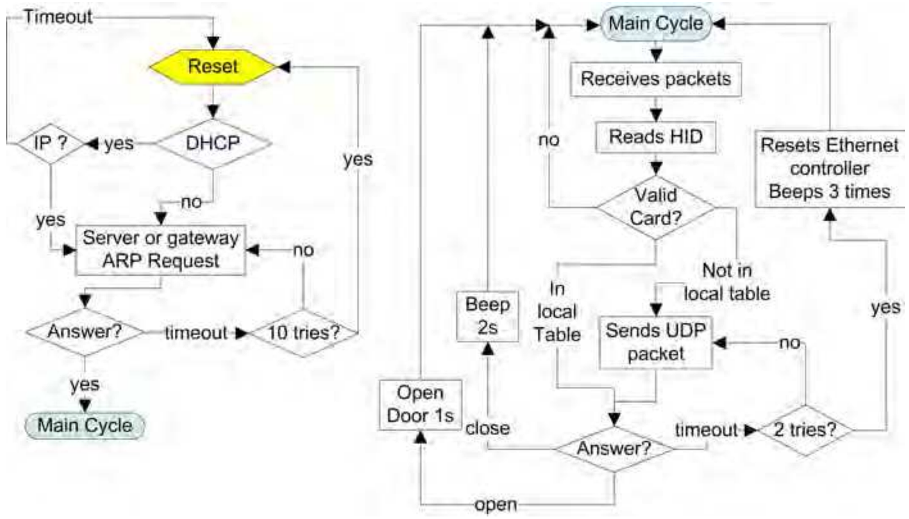


Fig. 4. Simplified flowchart illustrating the *DSClient* software. Left: initialization phase showing an ARP request that can be answered by the server or gateway, depending on the *DSClient* being on the same or on a different network. Right: main operation cycle for access control.

A flowchart representation, describing the main software decision flow implemented in the *DSClient* programmed firmware, is presented in Figure 4. It illustrates the *DSClient* software, with the initialization phase showing an ARP request that can be answered by the server or gateway, depending on the *DSClient* being on the same or on a different network. It also shows the main operation cycle for access control.

A detail that should be highlighted in the flowcharts of Figure 4 is the intentional reset of the Ethernet chip. This action is required since the Ethernet interface can stop answering network packets. This situation can arise if there is a faulty non answering *FreeBSD* server, but can also result from excessive IP traffic - for instance, due to an intentional denial of service attack. This non-answering state is detected by noticing a non-answering server, during repetitive *DSClient* validation requests. Combining the Ethernet resetting action with a watchdog, full system functionality can be always restored, without noticeable service reduction. User credentials are first checked for tentative local validation. If the current credentials are not present in the local table, a tentative central server validation follows. The *FreeBSD* server implements a UDP service in a system communication port, receiving the validation requests from the set of all distributed *DSClient* modules. The presented credentials are matched with those on the database and an appropriate answer is generated, through an UDP packet sent to the same *DSClient* system port.

The information provided by the *DSClient* modules is the following: *Ibutton* or *RFID* Card number, local time counter (nanosecond precision), number of valid HID or *Ibutton* readings, number of UDP requests without server answer, number of times the door was open, number of times access was not permitted, actual IP address of the module and some information associated with the security techniques. If local validation was not successful, the same information is sent to the server for registering and administration purposes.

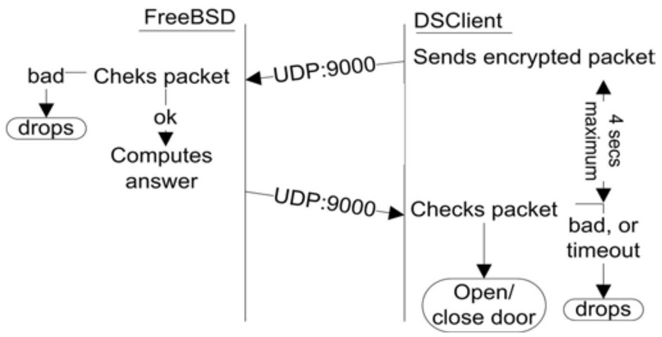


Fig. 5. Network flow for UDP packets, between a *DSClient* module and the database server, for central server validation.

In a first step, the FreeBSD server decrypts the UDP packet in order to validate its origin and credentials. If the packet is not valid it is immediately discarded and no answer is generated. A validated packet generates an UDP answer packet with the following information: pulse width to activate the electric locker (milliseconds), pulse width to activate the audio tone or BEEP (milliseconds), access authorization or access denial, date and time of authorization, and some other information associated with the security techniques.

The *DSClient* module, when receiving a server generated answer, verifies if it corresponds to its last request through the packet identification technique described in the in Section 3. If it does, the answer is validated if it was received in a time window of ***n*** seconds. Otherwise the answer is discarded. The information flow for packet exchange between *DSClient* modules and the database server is illustrated in Figure 5.

A total of three hardware prototypes for the *DSClient* module were built and tested. Two are the ones based on the Microchip microcontrollers and described above. The third is based on the ARM architecture. The Microchip versions were programmed using the MPLAB-C18 Tool [23]. The ARM based version was programmed using the GCC compiler targeting the ARM architecture [24]. List 1 shows the encrypted packet reception in the *FreeBSD* system.

3.2 An application to the educational environment

The described system, thanks to its distributed characteristics and modular implementation, has the flexibility for a large number of possible applications using *RFID* Tags or *Ibutton* technologies. Furthermore, it is highly hardware and software scalable, while exhibiting

```
int main(void)
{   struct sockaddr_in si_me,
si_other,si_pic;
   int s, i,
slen=sizeof(si_other),sz;
   unsigned char buf[BUFLEN];
   unsigned char lv_ip[50];
   unsigned char
lv_rfid[80],lv_mac[50];
   int indice_crypt,lv_abre;
   int last_ind_checksum;
   unsigned long lv_ncard;
   struct timeb timep;
   struct timespec lv_timep;
   time_t *ptime_t;
   unsigned char
lv_byte_pck_id;
   init_crypt_default();
   reset_default_passwd();
   indice_crypt=save_indice();
   if ((s=socket(AF_INET,
SOCK_DGRAM,
IPPROTO_UDP))==-1)
diep("socket");
   memset((char *) &si_me, 0,
sizeof(si_me));
   si_me.sin_family =
AF_INET;
   si_me.sin_port =
htons(PORT);
   si_me.sin_addr.s_addr =
htonl(INADDR_ANY);
   if (bind(s, &si_me,
sizeof(si_me))==-1)
diep("bind");
   if (connect_bd()==0)
printf("Error DB System\n");
   sz=28;
   if (!pic_decrypt(teste,sz)) {
         printf("Crypt Error:
%d bytes\n",sz);
   }
```

```
  while (1) {
      slen=sizeof(si_other);
      if ((sz=recvfrom(s, buf, BUFLEN, 0, &si_other, &slen))==-1)
diep("recvfrom()");
      if (!pic_decrypt(&buf[0],sz)) {
            printf("Crypt Error: %d bytes\n",sz);
            continue;
      }
      show_packet(buf,44,1,',',1);
      lv_ncard = *((unsigned long *) &buf[7+14]);
      snprintf(lv_rfid,80,"%d",lv_ncard);
      printf("RFID %s\n",lv_rfid);
snprintf(lv_ip,50,"%d.%d.%d.%d",buf[7],buf[8],buf[9],buf[10]);
      lv_abre=ask_DBsystem(lv_rfid,lv_ip,1);
      reset_default_passwd();
      buf[4]=indice_crypt;
      buf[18]=lv_abre;
      buf[19]=lv_abre;
      my_crypt(&buf[0],5,&buf[0]);
      restore_indice(indice_crypt);
      my_crypt(&buf[5],last_ind_checksum+2-5,&buf[5]);
give_me_checksums_crypt(&buf[last_ind_checksum],&buf[last_ind_checksum +
1]);
      indice_crypt=save_indice();
      show_packet(buf,32,1,',',0);

      if (sendto(s, buf, last_ind_checksum +2, 0, &si_pic, slen)==-1) diep("sendto()");
} // While loop
```

List 1. FreeBSD daemon receiving encrypted packets from Embedded *DSClient*.

very robust security characteristics [25]. Its robust security implementation allows for the use in sensitive applications. Based on the advanced configuration and administration of the system database, efficient and powerful information systems can be envisaged, with enough flexibility for a high number of secure distributed automation applications.

A field implementation of the described system is currently in service in the Electrical Engineering Department – "*Instituto Superior de Engenharia de Coimbra*" (Figure 6). A total of sixteen rooms are monitored and have access control installed. These include a general use printing room, a student project room, one general purpose classroom and ten specialized laboratories. Three experimental systems are also installed in other school departments, one in the Mechanical Engineering Department and two in the Physics and Mathematics Department. Common room access is the responsibility of the Head of Department while laboratories have dedicated administrators which are also responsible for the general use of the specific laboratory. Automated access control allows for high flexibility, maximizing resource availability and usage, while keeping a high degree of security and individual responsibility. A potential installation of 150 distributed modules exists.

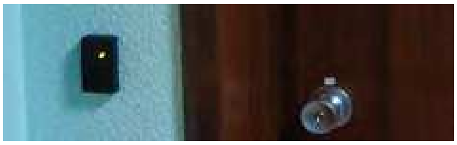


Fig. 6. Laboratory application example: Electrical Engineering Department - *Instituto Superior de Engenharia de Coimbra*.

The current implementation is being expanded to all department rooms to allow global management applications that can support most of the academic activities. Work in progress addresses fully registration of student attendance, permitting precise and dynamic adjustment in the number and time-allocation of laboratory classes. Early identification and active signalling of students with specific attendance difficulties can trigger timely and effective pedagogic correction actions by the course director.

Other functionalities can be added to the system such as the local management of Campus transactions. These can include student restaurants and bars, bookstore, academic fees, library registering and fine management, post-office, etc. This is a more security sensitive application area [25]. Some of these services are already deployed using other platforms.

Since there is a working Campus Access System, installed at *Instituto Superior de Engenharia de Coimbra*, which is managed by a private contractor (Figure 7), it is of most interest to interconnect both access networks and information processing applications, in order to maximize capabilities and comply with common security policies.

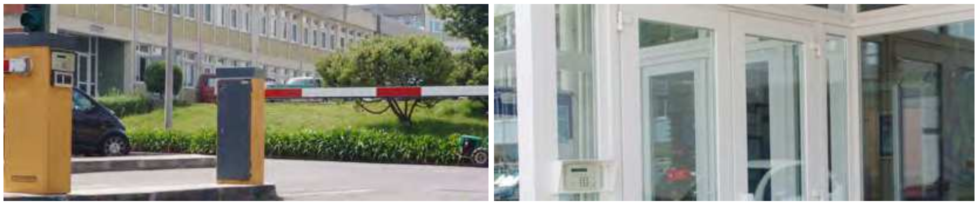


Fig. 7. Campus Access control - *Instituto Superior de Engenharia de Coimbra*. Installed car park and building access, by a security contractor using HID RFID technology.

3.3 Quality of service

Preliminary simple testing was performed, targeting the evaluation of the quality of service that can be achieved. The main concerns addressed initially were those related to sharing the network infrastructure with existing network services in a UDP transmission scenario.

The Electrical Engineering Department is a three store building with a two-level tree, 100 Mbps network infrastructure, based on 20 main switches (24 port) and a 1 Gbps fiber connection to the School backbone. About 240 permanently connected personal computers and 2 network servers are installed in offices, classrooms and laboratories. A total of 650 class hours are allocated each week for the duration of 15 week semesters. A set of 16 devices is fully operational at the present time. Two preliminary tests were conducted: rate of lost packets and denial of service through severe network congestion.

One semester of access statistics was registered for the installed devices. A rate of 1% of lost packets was measured in normal working conditions. This corresponds to extreme congestion situations and Ethernet chip limitations. In these cases the identification card has to be presented twice for the validation of credentials.

To assess the performance of the service in continuous high traffic environments, a second test was performed using the transmission of multicast video with controlled bitrate. A video server and a set of 8 video clients were connected, one in the same switch as the video server, and 7 on different switches. The video was sent in UDP multicast at 2, 4, 6, 8, 10 and 12 Mbps constant bitrates. Multicast traffic was not filtered for the switches where the HID readers were installed, reaching equally all ports.

It was observed that the rate of lost packets remained very low (in the order of 1%) for all 10Mbps *DSClient* modules, until it dropped drastically for some of them. This drop corresponded to the video bitrate reaching 10 Mbps at the same switch and port where the HID reader was connected. This is because 10 Mbps is the upper limit operating bitrate for the *DSClient* Ethernet chip. The 100Mbps *DSClient* Modules were unaffected under 12 Mbps constant bitrate. Further testing is planned for different traffic profiles, network setup and a more comprehensive set of statistics.

4. Integrated information system for course management

This section presents an Integrated Information System for the management of Technology Specialization Courses (TSC). These courses are subject to a stringent set of legal requirements that regulate the scientific, pedagogic and functioning components. To meet these requirements the availability of an organizational support structure is essential.

The developed information system accepts high-level definitions introduced by the Course Director, such as the teaching service allocation, the course calendar and schedule, as well as the individual student profile. Information relating to each class such as summary, individual student attendance, punctuality, participation and progress is dynamically managed by the teacher in each Training Unit.

Global information is continuously available online. Weekly reports are generated that allow the coordination of the course as well as the automatic sending of information required to generate all legal documentation.

According to Portuguese Law - Dec.-Lei n.º 88/2006, May 23 [26], TSC are post-secondary, non-higher education training courses, that can be taught at Higher Education Institutions or with their collaboration. According to Dec.-Lei n.º 782/2009, July 23 [26], since the 1st of October 2010, TSC are classified with the skill level 5 on a {1-8} scale, according to the European Qualifications Framework for life long learning [27]. On the other hand, in the registration process they are as qualification level 4, on a {1 -5} scale, according to the Decision n.º 85/368/EEC, published in the Official Journal of the European Communities, n.º L199, 31 July 1985. In both cases, TSC are described as a Qualification at a post-secondary non-higher education level, where credits can be gathered for further higher studies.

In accordance with the applicable law, TSC are organized by time editions with very different rules from those normally applicable to BSc and Master Courses in terms of organization, assessment and attendance. This requires the teaching Institutions to strongly adapt their steady mechanisms commonly used for course management. Thus, in addition to the registration of the final classifications and updating of student individual records, Dec.-Lei Law n.º 88/2006 also establishes the need to create, update and register on file:

- Records for the presence and punctuality of the students;
- The summary for each lesson;
- Justifications for not being present, validated by the TSC Director;
- Information about each assessment element, including individual marks.

In this context, for each training unit, it is necessary to maintain an updated file with all the mentioned items as well as educational support material. For each student, in addition to the final assessment results, it is necessary to organize a personal file containing the relevant classification information and his personal and professional profile. In addition, it is mandatory, on a monthly basis, to inform the relevant government agency on the actual number of students with effective class attendance.

Overall, course management is a very demanding, data-intensive distributed task. It must be automated in order to guarantee that valuable human resources are used in the most noble activity that is the teaching itself.

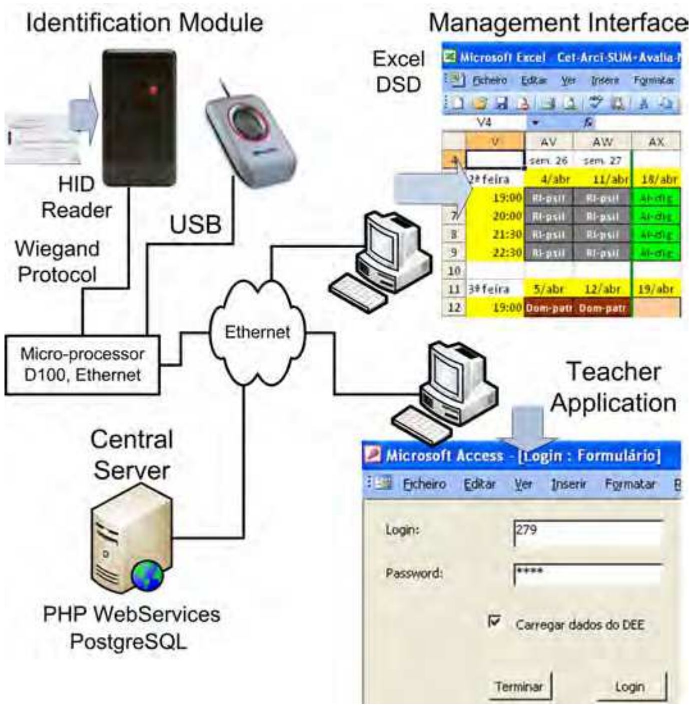
Information technologies are a very powerful tool for the efficient management of activities that require the processing of large amounts of data in an intensive and distributed manner. There are however security and confidentiality issues, including the management of the access to third parties and to individuals with administrative privileges.

4.1 Global architecture

An embedded system is used as the basis component of a distributed information system, specially developed to manage Technology Specialization Courses (TSC). In these courses, full control of student progress and student presence in classes is extremely important. The distributed information system for course management includes the functionality to record student attendance in class by using a specially developed module which is based on the Linux Embedded operating system [28] running on a ARM processor. This module permits to read fingerprints and RFID tags for student identification and further network transmission of relevant information – this module is hereafter named Embedded Identification Module (EIM).

The Embedded Identification Module is integrated in the whole course management information system as illustrated in Figure 8. The system consists of four hardware and software modules connected via TCP/IP over Ethernet: the Teacher Application, the Central Server, the Embedded Identification Module and the Management Interface.

The Teacher Application, developed in Microsoft Access (MS-Access) [29], is the teacher interface with the central server. It is based on a local database with forms allowing the management of information about class summaries, attendance, punctuality, participation and individual and collective progress. This application also permits the generation of local reports, by exporting in MS Excel format, information containing class syllabus, partial and



Login: 279

Password: ****

☑ Carregar dados do DEE

Fig. 8. Architecture of the integrated system: Teacher Application (MS-Access, Local); Central Server (Global Database); Management Interface and Embedded Identification Module (RFID, Fingerprint, Local).

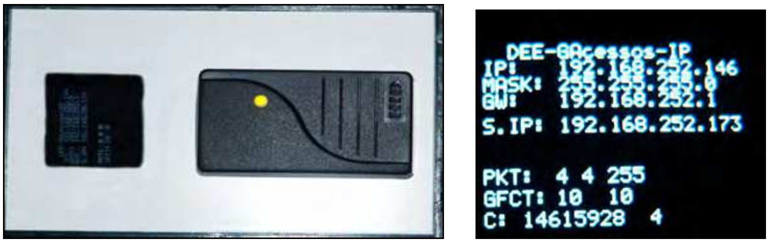
final classifications. It also allows the browsing of the planning and schedule for past and future classes. The application requires a *login* and *password* to identify the teacher.

The Teacher Application receives from the central server, managed by the TSC Director, the teaching service allocation, the course schedule and calendar as well as the student individual profile. Later, it sends back the information on class operation. This information will be the basis for the server generated global reports and to the final documentation about the course operation. The interconnection between the Teacher Application and the Central Server is achieved through a Website using WebServices technology [30].

The Embedded Identification Module communicates with the Teacher Application and with the Central Server. In the Teacher Application the user selects a search in the local network to find the Embedded Identification Module IP. The target test hardware for fingerprint validation uses a commercial router, the D100 from *Huwaei*, which is a very inexpensive hardware, with specific Linux-based firmware developed by the authors (see Figure 9). This router has an USB port allowing the connection of the fingerprint reader, a *Microsoft* model in this application. A custom Broadcom Chipset board is being developed for further use.

The fingerprint image acquisition is based on software derived from the *libfprint* library. The images are taken from the fingerprint reader, having been turned off the proprietary encryption mechanism of the device. The images are then processed to obtain the unique characteristics of each fingerprint. This unique features, consisting in a set of distances, are recorded in a USB-Pen attached to the D100, using an encryption mechanism. The data from each fingerprint is assigned an identification number. This number identifies the person in the central PostgreSQL database.

To build the Linux Embedded firmware, the *buildroot*, *busybox* and *u-boot* projects have contributed as a starting point - for more details see [12].



```
  DEE-GAcessos-IP
IP:     192.168.252.146
MASK:   255.255.255.0
GW:     192.168.252.1
S.IP: 192.168.252.173

PKT:  4 4 255
GFCT: 10  10
C: 14615928  4
```

Fig. 9. EIM - System to read HID-RFID Cards.

When using RFID authentication only, the system uses an ARM processor similar to the device explained in Section 3 - see Figure 1 for a hardware illustration and Figure 9 for an enclosure and test overview. The mechanism to send data to the central database is also similar to the method proposed for the RFID authentication method described in Section 3.

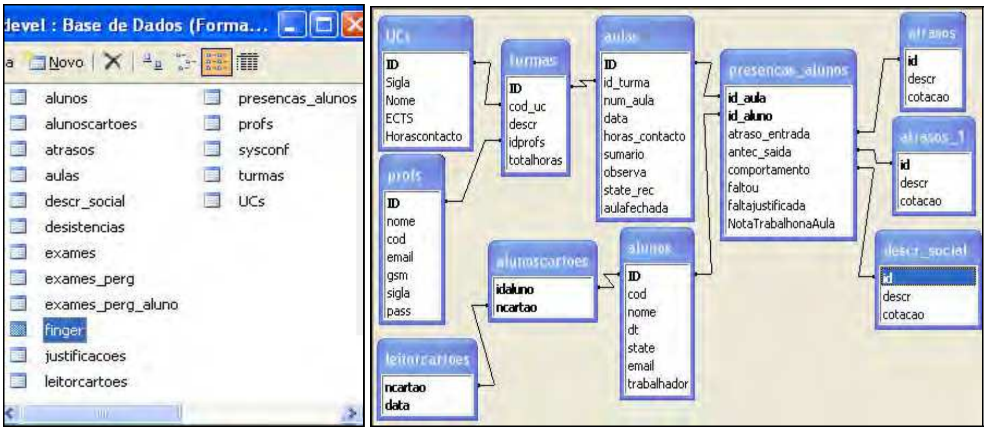
The Central Server is based on an Apache server [21] and a PostgreSQL database [20] installed on a FreeBSD operating system [21]. WebServices implemented in PHP are used as communication technology [21]. The teacher service allocation, the course calendar and course schedule, validation of missed class and the individual student profile, all are entered and updated by the TSC Director, using a Management Interface. This Management Interface uses MS-Excel, Visual Basic for Applications (VBA) [31] and WebServices to communicate with the Central Server.

The Central Server registers the information generated and validated locally by each teacher when using the Teacher Application, maintaining a global database. From this global database, the elements needed to the organization, operation and supervision of the course, in accordance with the legal requirements, are automatically generated and forwarded to the interested parties. Thus, the system ensures the data recording, edition and validation locally in the Teacher Application, with subsequent submission, registration and processing in the Central Server.

4.2 Teacher Application

The Teacher Application was developed in Microsoft Access allowing localized and autonomous management of information and, through a mechanism of communication and synchronization, the periodic transmission of teacher validated information to the Central Server. The local database in MS-Access consists of seventeen separate tables that are presented in Figure 10 - left. From the global set of tables, eleven are interconnected according to the structure in Figure 10 - right.

The Teacher Application communicates with the Central Server via WebServices. A set of functions in VBA and a set of forms were developed for this purpose. For interaction via


devel : Base de Dados (Forma...

alunos, alunoscartoes, atrasos, aulas, descr_social, desistencias, exames, exames_perg, exames_perg_aluno, finger, justificacoes, leitorcartoes, presencas_alunos, profs, sysconf, turmas, UCs

UCs: ID, Sigla, Nome, ECTS, Horascontacto

turmas: ID, cod_uc, descr, idprofs, totalhoras

aulas: ID, id_turma, num_aula, data, horas_contacto, sumario, observa, state_rec, aulafechada

presencas_alunos: id_aula, id_aluno, atraso_entrada, antec_saida, comportamento, faltou, faltajustificada, NotaTrabalhonaAula

atrasos: id, descr, cotacao

atrasos_1: id, descr, cotacao

descr_social: id, descr, cotacao

profs: ID, nome, cod, email, gsm, sigla, pass

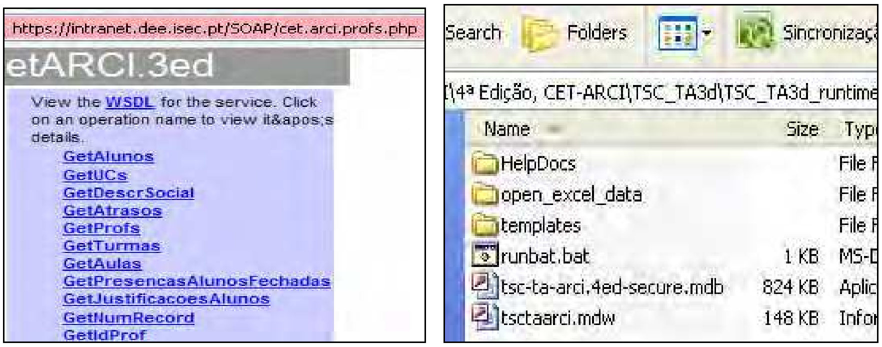
alunoscartoes: idaluno, ncartao

alunos: ID, cod, nome, dt, state, email, trabalhador

leitorcartoes: ncartao, data

Fig. 10. Left: Structure of the local database in MS-Access. Right: Interconnection between the main tables.

WebServices with the PostgreSQL database on the central server, a developed VBA class *TSCARCIws* exchanges information through a specific address on the Intranet, whose functions are presented in Figure 11 - left.


https://intranet.dee.isec.pt/SOAP/cet.arci.profs.php

etARCI.3ed

View the WSDL for the service. Click on an operation name to view it&apos;s details.

- GetAlunos
- GetUCs
- GetDescrSocial
- GetAtrasos
- GetProfs
- GetTurmas
- GetAulas
- GetPresencasAlunosFechadas
- GetJustificacoesAlunos
- GetNumRecord
- GetIdProf

Search Folders Sincronizaçã

:\4ª Edição, CET-ARCI\TSC_TA3d\TSC_TA3d_runtime

Name	Size	Typ
HelpDocs		File F
open_excel_data		File F
templates		File F
runbat.bat	1 KB	MS-D
tsc-ta-arci.4ed-secure.mdb	824 KB	Aplic
tsctaarci.mdw	148 KB	Infor

Fig. 11. Left: WebServices developed for communication between MS-Access and the PostgreSQL. Right: Developed application structure in MS-Access.

In terms of the local organization of files that support the implementation of the Teacher Application, the structure is shown in Figure 11 - right. In the templates folder is a spreadsheet developed in Excel, for which the application exports the information. This spreadsheet allows managing all the assessments for each training unit.

Each teacher is assigned a *username* and a *password* to access the system. After login, the teacher may choose to update the local database from the Central Server, which is mandatory the first time, or in a regular access, can use the application only locally with previous downloaded data.

When updating the local database with new information from the Central Server, the local information that has already been published is not modified. In both modes, local use only or server update, teacher lessons will be available for browsing and summary writing. After teacher identity validation, the Main Menu illustrated in Figure 12 will be available. The options shown in Figure 12 allow managing all information concerning the specific TSC training unit, namely: Edit summaries, consulting old class data, send data to the Central Server, report generation and deleting of the local database.

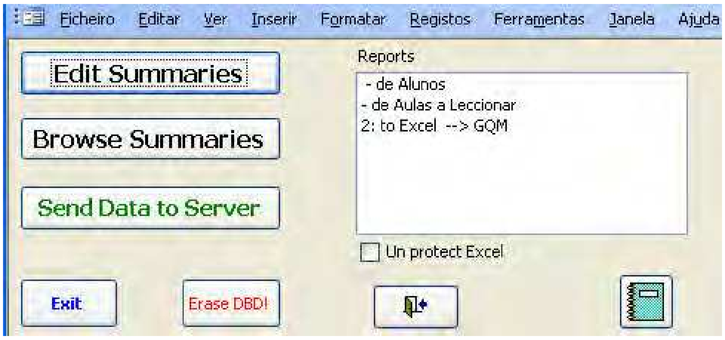
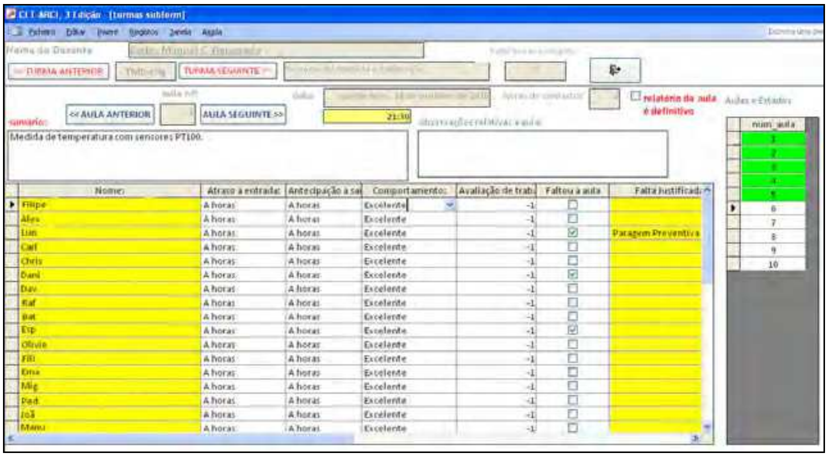


Fig. 12. Teacher Application Main menu. Editing and Browsing Summaries, Presence Management, Sending data to the Central Server and Reporting.

Figure 13 shows the interface for summaries, assiduity, punctuality, participation and for student assessment. This information is transmitted to the Central Server only after the teacher indicates that the class reports are final. After validation and transmission, classes are marked as delivered and are signalled with a green background in the summaries interface, as can be seen in Figure 13 - right.



sumario:

Medida de temperatura com sensores PT100.

Nome:	Atraso à entrada:	Antecipação à sa	Comportamento:	Avaliação de trab.	Faltou à aula	Falta Justificada
Filipe	A horas	A horas	Excelente	-1	☐	
Alex	A horas	A horas	Excelente	-1	☐	
Luis	A horas	A horas	Excelente	-1	☑	Paragem Preventiva
Carl	A horas	A horas	Excelente	-1	☐	
Chris	A horas	A horas	Excelente	-1	☐	
Dani	A horas	A horas	Excelente	-1	☑	
Dav	A horas	A horas	Excelente	-1	☐	
Raf	A horas	A horas	Excelente	-1	☐	
Pat	A horas	A horas	Excelente	-1	☐	
Esp	A horas	A horas	Excelente	-1	☑	
Olivie	A horas	A horas	Excelente	-1	☐	
Fili	A horas	A horas	Excelente	-1	☐	
Ema	A horas	A horas	Excelente	-1	☐	
Mig	A horas	A horas	Excelente	-1	☐	
Pad	A horas	A horas	Excelente	-1	☐	
Joã	A horas	A horas	Excelente	-1	☐	
Manu	A horas	A horas	Excelente	-1	☐	

num_aula
1
2
3
4
5
6
7
8
9
10

Fig. 13. Teacher Application. Summaries edition.

Data transmission to the Central Server includes data integrity check to detect transmission errors. Data transmitted without errors assumes a definitive state and can no longer be changed locally. Later edition is only possible for specific classes with the intervention of the TSC Director by setting an authorization in the Management Interface.

Export option for specific training units can be generated as a set of spreadsheets for assiduity, participation, exam and final classifications. Class summaries, absence justifications and student progress can also be exported.

The Teacher Application performs the main functionality of the system, since it is the liaison between the Embedded Identification Module, the Central Server and the teachers themselves. In addition to the functions already described, the Teacher Application allows to connect to the Embedded Identification Module. This identification module will be described in the next section and its job is to read RFID tags and fingerprints.

4.3 Embedded identification module operation

Each student in the class is identified by the presentation of an HID RFID card or by his fingerprint. The association between the student and his RFID identification number is performed in the PostgreSQL database using the Central Server Management Interface. The card reader system, illustrated in Figure 9, was programmed using C language and embedded in a microcontroller with ARM architecture [22, 23].

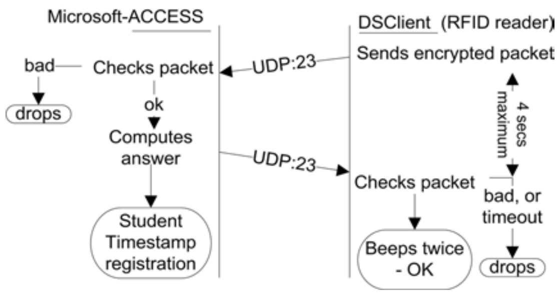
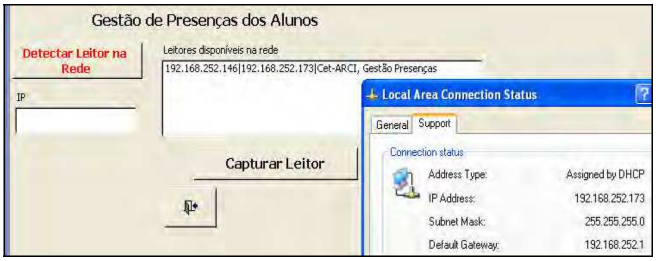


Fig. 14. Communication between the Teacher Application and the EIM.

The exchange of information between the Teacher Application and the Embedded Identification Module (EIM) is encrypted and carried through the UDP protocol. Figure 14 shows a diagram representing the bidirectional communication between the two systems.


Gestão de Presenças dos Alunos

Detectar Leitor na Rede

IP

Leitores disponíveis na rede

192.168.252.146|192.168.252.173|Cet-ARCI, Gestão Presenças

Capturar Leitor

Local Area Connection Status

General Support

Connection status

Address Type: Assigned by DHCP

IP Address: 192.168.252.173

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.252.1

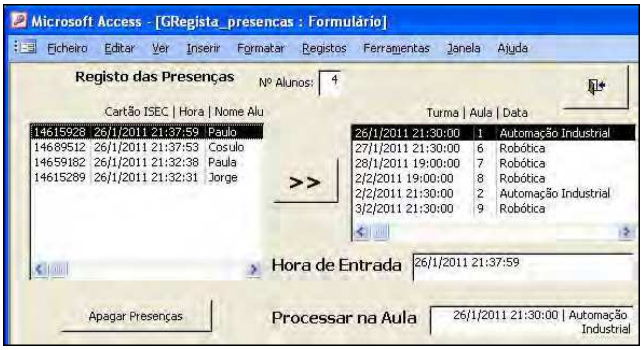
Fig. 15. Operation of the Teacher Application in capture mode. Searches in the local network for EIM devices and informs them where to forward packets.

For classroom operation, the teacher uses the Teacher Application to search for identification modules in the local room network using broadcast messages. After getting a response, the connected equipment is captured using the interface shown in Figure 15.

```
, Captures the EIM module
Private Sub EIM_Capture_command()
   Dim lv_i As Integer, cod_leitor As String
   Dim myItens, lip
   ...
   cod_leitor = Lista4.ItemData(lv_i)
   myItens = Split(cod_leitor, "|")
   lip = Winsock3.LocalIP
   gv_ip_leitor_HID = myItens(0)
   If lip <> myItens(1) Then
      Winsock3.RemoteHost = myItens(0)
      lip_parts = Split(lip, ".")
      Call clean_list
      Winsock3.SendData              Chr(253)              +
Chr(Val(lip_parts(0))) + Chr(Val(lip_parts(1))) + _
            Chr(Val(lip_parts(2))) + Chr(Val(lip_parts(3)))
      Winsock3.RemoteHost = "0.0.0.0"
      Exit Sub
   End If
   Call Change_Form_attendance_record
End Sub

Private Sub Winsock3_DataArrival(ByVal bytesTotal As
Long)
   Dim buff As Meus_DADOS
   Dim strData As String

   Winsock3.GetData strData, bytesTotal
   If (bytesTotal > 20) And (bytesTotal = Len(strData))
Then
      If bytesTotal = 84 Then
         If                          descodifica(strData,
CStr(Winsock3.RemoteHostIP)) = 3 Then
            'changes Screen
            Call Change_Form_attendance_record
         End If
      End If
   End If
End Sub
```

```
Private Function Decode_EIM_data(str As String, rip As
String) As Integer
   Dim ip As String, app As String, mac As String
   Dim lim As Integer, i As Integer, lip As String
   Dim anyDate() As Byte

   anyDate = StrConv(str, vbFromUnicode)
   Decode_EIM_data = 0
   If (anyDate(2) <> &H2) Then
      Exit Function
   End If
   If Not ((anyDate(0) = &HFE) Or (anyDate(0) = 252))
Then
      Exit Function
   End If
   ip = CStr(anyDate(5)) + "." + CStr(anyDate(6)) + "." +
CStr(anyDate(7)) + "." + CStr(anyDate(8))
   lim = 20
   While (anyDate(lim) And (lim < 82)) <> 0
         lim = lim + 1
   Wend
   app = Mid(str, 20, lim - 20 + 1)
   mac = Hex(anyDate(9)) + ":" + Hex(anyDate(10)) + ":"
+ Hex(anyDate(11)) + ":" + _
      Hex(anyDate(12)) + ":" + Hex(anyDate(13)) + ":" +
Hex(anyDate(14))
   Lista4.AddItem (rip + "|" + ip + "|" + app)
   If (anyDate(0) = &HFE) Then
         Decode_EIM_data = 2
   Else
         lip = Winsock3.LocalIP
         If lip <> ip Then
            MsgBox "Error Capturing EIM Device"
         Else
            Decode_EIM_data = 3
         End If
   End If
End Function
```

List 2. VBA code for EIM capture by sending a broadcast message announcing the Teacher Application network IP.


Microsoft Access - [GRegista_presencas : Formulário]
Ficheiro Editar Ver Inserir Formatar Registos Ferramentas Janela Ajuda
>>
Apagar Presenças

Registo das Presenças

Nº Alunos: 4

Cartão ISEC | Hora | Nome Alu

```
14615928  26/1/2011 21:37:59  Paulo
14689512  26/1/2011 21:37:53  Cosulo
14659182  26/1/2011 21:32:38  Paula
14615289  26/1/2011 21:32:31  Jorge
```

Turma | Aula | Data

```
26/1/2011 21:30:00  1  Automação Industrial
27/1/2011 21:30:00  6  Robótica
28/1/2011 19:00:00  7  Robótica
2/2/2011 19:00:00   8  Robótica
2/2/2011 21:30:00   2  Automação Industrial
3/2/2011 21:30:00   9  Robótica
```

Hora de Entrada: 26/1/2011 21:37:59

Processar na Aula: 26/1/2011 21:30:00 | Automação Industrial

Fig. 16. After the Teacher Application captures the EIM, students should approach their RFID card to the EIM to identify themselves and record time-of-entry.

After the EIM has been captured, students entering the classroom approach the HID card to the EIM, identifying them and recording the time-of-entry as illustrated in Figure 16. Time-of-entry is later used for presence related information in accordance with a start time reference defined by the TSC Director.

As an alternative to the HID reader, the identification from the fingerprint reader can be used. A Microsoft Fingerprint Reader with USB interface is used. The sensor is interfaced with a microcontroller containing embedded Linux, where the image processing algorithms leading to the identification are programmed.

For confidentiality reasons, associated with the collection and processing of biometric data, the entire processing software shall be contained and secured in the microcontroller. For identification, only a unique index is provided. Fingerprints are stored in-device, on an SD card, encrypted and associated to a student identification index. The widespread and final use of this sensor will be implemented only after all legal requirements associated with the protection of personal data are guaranteed.

4.4 Implementation details

For the RFID reader, the EIM is implemented with an ARM Cortex-M3 processor. All the programming tools for this microcontroller have been developed based on the GNU GCC *toolchain* for ARM Cortex-M3 version 4.3.3, *Binutils-2.19.1* and *newlib-1.18.0* under *Gygwin* with *lwip* TCP Stack. Information about setting up a free GCC *toolchain* is available from [13].

For the Fingerprint reader, as presented in Section 4.1, the D100 router from *Huwaei* with specific Linux-based firmware was used as test platform for the Broadcom Chipset.

By default, the EIM sends data to the Central Server (FreeBSD) but the Teacher Application can capture the device through a broadcast packet, informing the EIM where to send its data packets, that is to the Teacher Application. List 2 shows the VBA code that triggers the capture of the EIM device in the local network.

After the EIM has been informed where to send its UDP packets with RFID information, the Teacher Application waits for this information through a socket data reception event. List 3 shows the Teacher Application (MS-Access) code for encrypted packet reception. List 4 shows the firmware for the main state machine while a similar state machine is also used to control Ethernet communications as presented in Fig. 15.

Figure 17 shows the Common Firmware Environment (CFE) available from Broadcom. It is the first code that is executed when the router boots. It performs functions such as system initialization, setting up a basic environment, optionally providing a command line interface and loading and executing a kernel image. After this first code Linux will boot if it was initially programmed in the flash memory of the router.

Figure 18 illustrates the fingerprint test hardware setup, and results from the image processing algorithm, based on Linux Embedded. This system is compatible with the communication protocol used by the ARM RFID system. A custom board based on the Broadcom Chipset will be used for the long-term application.

```

Private Sub Winsock_readhid_DataArrival(ByVal bytesTotal
As Long)
  Dim strData As String, arrayOUT() As Byte
  Dim anyDate() As Byte, nrdif_card As Long
  Dim rslt As Boolean, last_ind_checksum As Long
  Dim TIME_BEEP_10MS As Integer
  Dim TIME_LED_ON As Integer, lv_open As Integer

  If bytesTotal <= 0 Exit Sub
  Winsock_lehid.GetData strData, bytesTotal
  anyDate = StrConv(strData, vbFromUnicode)
  If (bytesTotal > 0) And (bytesTotal = Len(strData)) Then
    rslt = gv_crypt_class.pic_decrypt(anyDate, bytesTotal)
    If rslt Then ' MSG ok, sends answer
      nrdif_card = anyDate(7 + 14 + 3)
      nrdif_card = nrdif_card * 256
      nrdif_card = nrdif_card + anyDate(7 + 14 + 2)
      nrdif_card = nrdif_card * 256
      nrdif_card = nrdif_card + anyDate(7 + 14 + 1)
      nrdif_card = nrdif_card * 256
      nrdif_card = nrdif_card + anyDate(7 + 14 + 0)
      ' --------- packet to answer EIM
      TIME_BEEP_10MS = 100
      TIME_LED_ON = 50
      lv_open = 0
      last_ind_checksum = 30
      anyDate(14) = anyDate(0)
      anyDate(0) = Int((6 * Rnd) + 1)
      anyDate(2) = TIME_BEEP_10MS
      anyDate(18) = lv_open
      anyDate(20) = TIME_LED_ON
      anyDate(21) = anyDate(20)
      last_ind_checksum = last_ind_checksum + 2
      ReDim arrayOUT(last_ind_checksum - 1)
      For lv_open = 0 To last_ind_checksum - 1
        arrayOUT(lv_open) = anyDate(lv_open)
      Next
      Call                  gv_crypt_class.pic_crypt(arrayOUT,
last_ind_checksum - 2)
      Winsock_readhid.RemoteHost = gv_ip_leitor_HID
      Winsock_readhid.SendData arrayOUT
      Call insert_in_bd_EIM_hid_RFID(nrdif_card)
```

```
    End If
  End If
End Sub
, VBA crypt_c class
, .... Other fuctions not listed
Public Sub reset_class()
    indice = 0 'Indice na password
    Call init_crypt_default
    Call reset_default_passwd
    indice_crypt = save_indice()
    gv_lv_ind2crypt = save_indice()
End Sub

Function pic_decrypt(ByRef buf() As Byte, sz As Long)
As Boolean
    Dim ind As Integer, ch0 As Byte, ch1 As Byte

    Call reset_default_passwd
    Call my_decrypt(buf, 5, buf, 0)
    ind = buf(4)
    Call restore_indice(ind)
    Call my_decrypt(buf, sz - 7, buf, 5)
    Call give_me_checksums(ch0, ch1)
    If ((ch0 = buf(sz - 2)) And (ch1 = buf(sz - 1))) Then
        pic_decrypt = True
    Else
        pic_decrypt = False
    End If
End Function

Function pic_crypt(ByRef buf() As Byte, sz As Long)
As Boolean
    Dim ind As Integer, ch0 As Byte, ch1 As Byte

    Call reset_default_passwd
    buf(4) = gv_lv_ind2crypt
    Call my_crypt(buf, 5, buf, 0)
    Call restore_indice(gv_lv_ind2crypt)
    Call my_crypt(buf, sz - 5, buf, 5)
    Call give_me_checksums_crypt(ch0, ch1)
    gv_lv_ind2crypt = save_indice()
    buf(sz) = ch0
    buf(sz + 1) = ch1
    pic_crypt = True
End Function
```

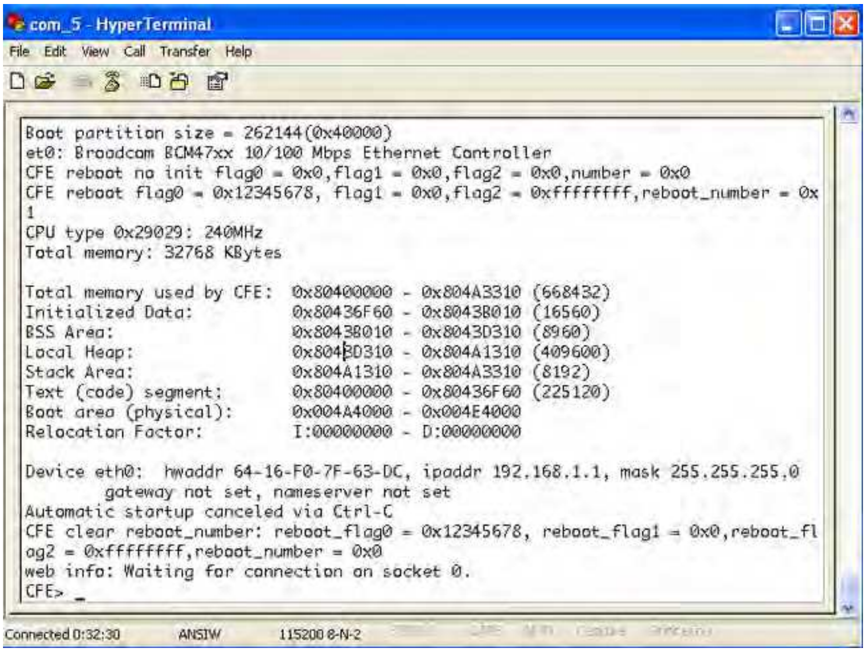
List 3. MS-Access receiving encrypted packets from the EIM.

```
// Loop forever.
gv_state_grafcet=0;
gv_udp_link.state=0;
gv_serverip=0x8B14A8C0;  // 192.168.20.139

show_my_ip();
gv_leitura_cartao=gv_gpio_out_on=0;
while(1) {
   clock = HID_Clock;
   data = HID_Data << 1;
   ulData= (clock) | ( data );
   If (Fingerprint_ID!=0) ulData= Fingerprint_ID;
   if (gv_state_grafcet>0) show_packets_out();
   switch (gv_state_grafcet) {
   case 0: // waiting DHCP for an IP
          ulIPAddress = lwIPLocalIPAddrGet();
```

```
      if (gv_udp_link.state==50) {
            gv_udp_link.state=10;
            if (gv_answer==0) {
                  gv_state_grafcet=40;
                  lv_t_secs =g_ulSystemTimeSeconds+1;
                  GPIOPinWrite(PgpioBASE, BEEPgpioPIN, 1);
                  lv_conta_iter=0;
            } else  {
                  lv_t_secs =g_ulSystemTimeSeconds+2;
                  gv_state_grafcet=30;
                  GPIOPinWrite(PgpioBASE, PgpioPIN, 1);
                  gv_gpio_out_on =0;
            }
      }
      break;
  case 30: // set GPIO out on for a while
```

```
        if (ulIPAddress!=0) {
            gv_state_grafcet=10;
            gv_udp_link.state=0;
            lv_t_secs=g_ulSystemTimeSeconds+2;
        }
        break;
    case 10: // RFID or fingerprint valid read?
        if (ulData&1!=3) if (Le_cartao()) {
            nleituras_cartao++;
            show_state(gv_ncartao,nleituras_cartao);
            gv_state_grafcet=20;
            gv_gpio_out_on =0;
            break;
        }
        if (gv_gpio_out_on!=0) {
            gv_gpio_out_on =0;
            GPIOPinWrite(PgpioBASE,PgpioPIN, 1);
            gv_state_grafcet=30;
        }
        break;
    case 20: // sends RFID card or fingerprint id
        udp_protocol_ask((UDP_LINK *)&gv_udp_link);
        if (gulSystemTSecs > lv_time_seconds) {
            GPIOPinWrite(PgpioBASE, PgpioPIN, 0);
            gv_gpio_out_on =0;
            gv_state_grafcet=10;
        }
        break;
    case 40: // Beeps (another GPIO out line)
        if (gulSystemTSecs > lv_time_seconds) {
            lv_t_secs =g_ulSystemTimeSeconds+1;
            gv_state_grafcet=50;
            GPIOPinWrite(PgpioBASE, BEEPgpioPIN, 0);
            if (lv_conta_iter>1) gv_state_grafcet=10;
        }
    case 50: // Beeps (another GPIO out line)
        if (gulSystemTSecs > lv_time_seconds) {
            lv_conta_iter++;
            lv_t_secs = gulSystemTSecs +1;
            gv_state_grafcet=40;
            GPIOPinWrite(PgpioBASE, BEEPgpioPIN, 1);
        }
        break;
    }
```

List 4. ARM-embedded program or Linux embedded daemon – Principal state machine.



```
Boot partition size = 262144(0x40000)
et0: Broadcom BCM47xx 10/100 Mbps Ethernet Controller
CFE reboot no init flag0 = 0x0,flag1 = 0x0,flag2 = 0x0,number = 0x0
CFE reboot flag0 = 0x12345678, flag1 = 0x0,flag2 = 0xffffffff,reboot_number = 0x
1
CPU type 0x29029: 240MHz
Total memory: 32768 KBytes

Total memory used by CFE:  0x80400000 - 0x804A3310 (668432)
Initialized Data:          0x80436F60 - 0x8043B010 (16560)
BSS Area:                  0x8043B010 - 0x8043D310 (8960)
Local Heap:                0x8043D310 - 0x804A1310 (409600)
Stack Area:                0x804A1310 - 0x804A3310 (8192)
Text (code) segment:       0x80400000 - 0x80436F60 (225120)
Boot area (physical):      0x004A4000 - 0x004E4000
Relocation Factor:         I:00000000 - D:00000000

Device eth0:  hwaddr 64-16-F0-7F-63-DC, ipaddr 192.168.1.1, mask 255.255.255.0
        gateway not set, nameserver not set
Automatic startup canceled via Ctrl-C
CFE clear reboot_number: reboot_flag0 = 0x12345678, reboot_flag1 = 0x0,reboot_fl
ag2 = 0xffffffff,reboot_number = 0x0
web info: Waiting for connection on socket 0.
CFE> _
```

Fig. 17. The Common Firmware Environment (CFE) from Broadcom.

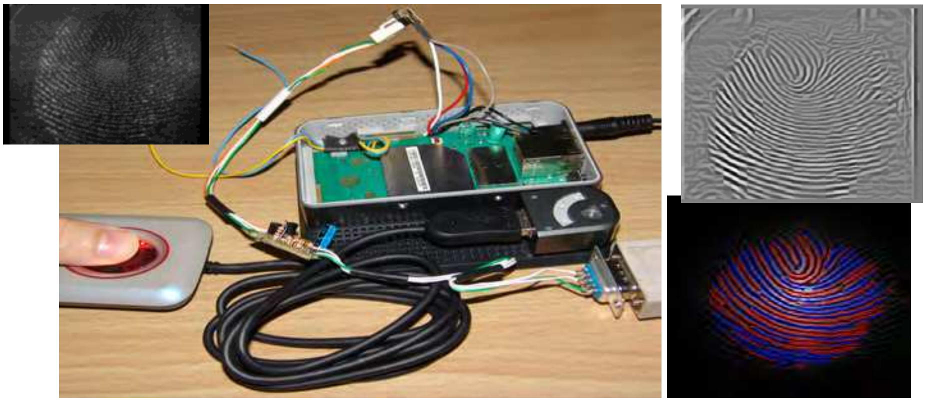


Fig. 18. Left: Test hardware - Huwaei D100 router with Linux Embedded for fingerprint acquisition. Right: An example of a fingerprint and its processing.

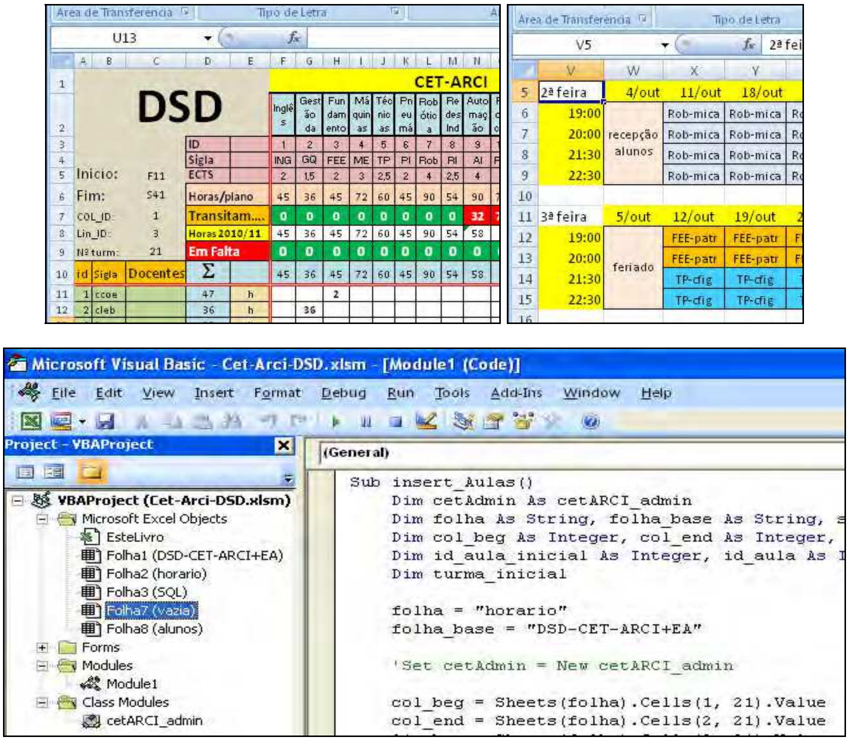
4.5 Central server and management interface

The Central Server main function is to register and maintain the final course information in a central database. This information, associated with the management of the TSC, includes high level general information and specific information for each training unit. The server software also allows to automatically generate all the legally required documents for the operation of the course. *E-mail* alerts are also created and addressed to teachers who have not timely updated data in the Teacher Application. A further weekly report is generated and *e-mailed* to the TSC Director with information on the presence, punctuality, participation and progress of the students. This information allows an efficient supervision of the course operation. In addition, it permits to fulfill the legal information communication requirements such as the exact and actual number of students attending the course.

In terms of implementation technology, the Central Server is based on an Apache Server and on a PostgreSQL database. These software packages are installed on a FreeBSD operating system and use WebServices implemented in PHP as communication technology. In order to ensure maximum consistency and simplify synchronization between the local and central databases, the PostgreSQL tables in the Central Server and the MS-Access tables in the Teacher Application are identical.

The TSC Director, through the Management Interface, defines in the Central Server the common elements and top-level requirements for the course operation. The interface can be accessed through the Intranet or Internet and allows the introduction of all relevant information as for example the Teacher Service Allocation. This interface uses Excel spreadsheets, VBA programming and WebServices to communicate with the Central Server.

Figure 19 presents an example of the Teacher Service Allocation and the course schedule for a set of training units. It also presents the VBA source code used to search in Excel spreadsheets for the Teacher Service Allocation, generating the SQL commands that are sent through WebServices to the PostgreSQL database.


Área de Transferência
Tipo de Letra
U13
Área de Transferência
Tipo de Letra
V5
2ª fei
Microsoft Visual Basic - Cet-Arci-DSD.xlsm - [Module1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - VBAProject
(General)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1						CET-ARCI								
2			DSD			Inglês	Gestão da	Fundamento	Máquinas	Técnicas	Pneumá	Robótica	Redes Ind	Automação
3				ID		1	2	3	4	5	6	7	8	9
4				Sigla		ING	GQ	FEE	ME	TP	PI	Rob	RI	AI
5	Inicio:		F11	ECTS		2	1,5	2	3	2,5	2	4	2,5	4
6	Fim:		S41	Horas/plano		45	36	45	72	60	45	90	54	90
7	COL_ID:		1	Transitam....		0	0	0	0	0	0	0	0	32
8	Lin_ID:		3	Horas 2010/11		45	36	45	72	60	45	90	54	58
9	Nº turm:		21	Em Falta		0	0	0	0	0	0	0	0	0
10	id	Sigla	Docentes	Σ		45	36	45	72	60	45	90	54	58
11	1	ccoe		47	h			2						
12	2	cleb		36	h		36							

	V	W	X	Y
5	2ª feira	4/out	11/out	18/out
6	19:00	recepção alunos	Rob-mica	Rob-mica
7	20:00		Rob-mica	Rob-mica
8	21:30		Rob-mica	Rob-mica
9	22:30		Rob-mica	Rob-mica
10				
11	3ª feira	5/out	12/out	19/out
12	19:00	feriado	FEE-patr	FEE-patr
13	20:00		FEE-patr	FEE-patr
14	21:30		TP-cfig	TP-cfig
15	22:30		TP-cfig	TP-cfig

```
VBAProject (Cet-Arci-DSD.xlsm)
  Microsoft Excel Objects
    EsteLivro
    Folha1 (DSD-CET-ARCI+EA)
    Folha2 (horario)
    Folha3 (SQL)
    Folha7 (vazia)
    Folha8 (alunos)
  Forms
  Modules
    Module1
  Class Modules
    cetARCI_admin
```

```
Sub insert_Aulas()
    Dim cetAdmin As cetARCI_admin
    Dim folha As String, folha_base As String, s
    Dim col_beg As Integer, col_end As Integer,
    Dim id_aula_inicial As Integer, id_aula As I
    Dim turma_inicial

    folha = "horario"
    folha_base = "DSD-CET-ARCI+EA"

    'Set cetAdmin = New cetARCI_admin

    col_beg = Sheets(folha).Cells(1, 21).Value
    col_end = Sheets(folha).Cells(2, 21).Value
```

Fig. 19. Excel spreadsheet: Time and the Teacher Service Allocation for the course together with VBA programming using WebServices.

5. Discussion and conclusions

In this chapter we intended to present successful applications using networked embedded systems. For this purpose we described a network architecture and two application examples in the educational area, which are full implemented and successfully deployed. The first application consists in a campus automated access control and access management system while the second implements a full-featured distributed course management tool, tailored for a specific scientific environment. Both applications are in operation at the Electrical Engineering Department of the *Instituto Superior de Engenharia de Coimbra*.

Distributed control modules exhibit high programming flexibility, with the possibility of local and remote programming. The modules are very reliable and have powerful processing and interfacing capabilities. Furthermore, they are highly hardware and software scalable, while exhibiting very robust security characteristics [25]. Open Source Software, both in the distributed modules and in the central database, guarantees flexibility and high evolution potential.

Real-Time Operating Systems (RTOS) are candidates for both applications described in this chapter and work is in progress to use FreeRTOS as a basic development environment for the ARM processor. Despite the fact that these are not real-time intensive applications, RTOS can greatly facilitate system development because multiple tasks can be programmed as if they were unique.

In this chapter we presented and detailed a set of engineering choices that can be used to develop successful networked embedded systems applications. These choices include network and communication architecture, hardware, development and programming tools for the embedded modules as well as communication technology and server operating system and basis software.

As a general conclusion we can state that networked embedded systems exhibit a great potential to serve as remote intelligent modules in high performance distributed applications such as those that can be developed for the educational and industrial fields. Furthermore, it was demonstrated that it is possible to create advanced networked distributed applications using Open Source Software for the system infrastructure.

6. References

[1] G. Pottie, W. Kaiser, *Principles of Embedded Networked Systems Design*, Cambridge University Press, 2005.

[2] Embedded Design, Available from www.embedded.com and www.linuxdevices.com, last visit on 24th September 2011.

[3] Fonseca, I.; Lopes, F.; "*An embedded system and an architecture for access control and access management: An application to the educational environment*", Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on , vol., no., pp.1-5, 16-19 June 2010, Available from ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5556614&isnumber=555659, last visit on 2nd October 2011.

[4] Fonseca, I; Coelho Teixeira, C.; Lopes, F.; "Sistema de Informação Integrado para a Gestão de Cursos de Especialização Tecnológica", Information Systems and Technologies (CISTI), 2011 6th Iberian Conference.

[5] Hongtao Zeng, Jiang Guo, Zhihuai Xiao, *Real time embedded maintenance system of hydro generator excitation system*, International Conference on Condition Monitoring and Diagnosis, CMD 2008, Beijing, China, April 21-24.

[6] Ambient Assisted Living, Available from www.aal-europe.eu, last visit on 24th September 2011.

[7] S. S. Iyengar and R.. R. Brooks, Eds, *Distributed Sensor Networks*, Chapman & Hall, 2005

[8] Philip J. Koopman, Jr, *Embedded System Design Issues*, Proceedings of the International Conference on Computer Design (ICCD 96), Available from www.ece.cmu.edu/~koopman/iccd96/iccd96.html, last visit on 2nd October, 2011.

[9] Seventh Framework Programme, European Network of Excellence on Embedded Systems Design, Available from www.artist-embedded.org/artist/State-of-the-Art,715.html, last visit on 2nd October, 2011.

[10] Krishna Kavi, Robert Akl, Ali Hurson, *Real-Time Systems: An Introduction And The State-Of-The-Art*,Wiley Encyclopedia of Computer Science and Engineering, 2008, Available from csrl.unt.edu/~kavi/Research/encyclopedia-realtime.pdf, last visit on 2nd October 2011.

[11] Ramesh Yerraballi, University of Texas, Real-Time Operating Systems: An Ongoing Review, Available from citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.5799&rep=rep1&type=pdf, last visit on 2nd October 2011.

[12] Michael Opdenacker, *Embedded Linux From Scratch… in 40 minutes!*, Available from free-electrons.com.

[13] *GCC toolchain for ARM*, Available from www.microbuilder.eu/projects/LPC1343ReferenceDesign/LPC1343Toolchain.asx and www.microbuilder.eu/Tutorials/SoftwareDevelopment/BuildingGCCToolchain.aspx, last visit on 24th September 2011.

[14] Karim Yaghmour, Jon Masters,Gilad BenYossef, Philippe Gerum, Michael Opdenacker, *Building Embedded Linux Systems*, O'Reilly, August 2008.

[15] Hiroyuki Tomiyama, Shinya Honda, Hiroaki Takada, *System-Level Design Tools and RTOS for Multiprocessor SoCs*, MPSoC 2004, Available from www.yxi.com/applications/Tomiyama.pdf, last visit on 2nd October 2011.

[16] Fernanda Coutinho, Inácio Sousa Fonseca, *Local Security System Using CAN*, CAN Newsletter, December 2001, EUA.

[17] Evan Welbourne, Magdalena Balazinska, Gaetano Borriello, Waylon Brunette, *Challenges for Pervasive RFID-Based Infrastructures*, Fifth IEEE International Conference on Pervasive Computing and Communications, New York, USA, March 2007.

[18] Hugo Oliveira, Tiago Oliveira, *Controlador de Acessos Modular Baseado em Tecnologia de Cartões de Proximidade*, In Portuguese, Relatório de fim de curso, Ano 2004, Escola Superior de Tecnologia de Setúbal.

[19] Commercial Security Systems, Available from www.idonic.com and www.bioglobal.pt, last visit on 24th September 2011.

[20] PostgreSQL Global Development Group, Available from www.postgresql.org, last visit on 24th September 2011.

[21] Apache Web Server and PHP language, Available from www.apache.org and www.php.net, last visit on 24th September 2011.

[22] FreeBSD Unix System, Available from www.freebsd.org, last visit on 24th September 2011.

[23] Microchip Technology, Available from www.microchip.com, last visit on 24th September.

[24] Gnu Arm ToolChain, Available from www.gnuarm.com, last visit on 24th September 2011.

[25] Chen X., Makki K., Yen K., Pissinou N., *Sensor network security: a survey*, IEEE Communications Surveys & Tutorials, Volume 11, Issue 2, Pages:52 – 73, 2009.

[26] Portugal, "*Diário da República*", Available from www.dre.pt, last visit on 4th September 2010.

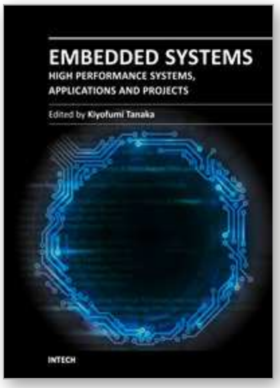
[27] Recomendação 2008/C111/01/CE do Parlamento Europeu e do Conselho, "*Quadro Europeu de Qualificações para a aprendizagem ao longo da vida*", 23 de Abril de 2008.

[28] Building a Linux Embedded system, Available from http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Tutorial-Improving-an-embedded-Linux-system, last visit on 24th September 2011.

[29] Microsoft, *Access Developer Center*, Available from msdn.microsoft.com/en-us/office/aa905400, last visit on 4th Sep. 2011.

[30] W3C Web of Services, *WebServices*, Available from www.w3.org/standards/webofservices, last visit on 24th September 2011.

[31] Teresa Hennig, Rob Cooper, Geoffrey L. Griffith and Armen Stein, *Access 2007 VBA Programmer's Reference*, Wiley Publishing 2007.


Embedded Systems - High Performance Systems, Applications and Projects
Edited by Dr. Kiyofumi Tanaka

ISBN 978-953-51-0350-9
Hard cover, 278 pages
Publisher InTech
Published online 16, March, 2012
Published in print edition March, 2012

Nowadays, embedded systems - computer systems that are embedded in various kinds of devices and play an important role of specific control functions, have permeated various scenes of industry. Therefore, we can hardly discuss our life or society from now onwards without referring to embedded systems. For wide-ranging embedded systems to continue their growth, a number of high-quality fundamental and applied researches are indispensable. This book contains 13 excellent chapters and addresses a wide spectrum of research topics of embedded systems, including parallel computing, communication architecture, application-specific systems, and embedded systems projects. Embedded systems can be made only after fusing miscellaneous technologies together. Various technologies condensed in this book as well as in the complementary book "Embedded Systems - Theory and Design Methodology", will be helpful to researchers and engineers around the world.