

## A Remote Control System to Configure Network Devices

Sandra Sendra, Emilio Granell, Ismael Climent, Jaime Lloret

Universidad Politécnic de Valencia

Camino Vera s/n, 46022, Valencia, Spain

sansenco@posgrado.upv.es; emgraro@posgrado.upv.es; ismael@cspteleco.net; jlloret@com.upv.es

**Abstract**— Due to the emergence of new technologies and the proliferation of e-learning services and online courses, the need to provide remote access to real devices from any device, anywhere at any time is higher. There are several software applications that can simulate a laboratory virtually and others where you can configure the network devices remotely. But, but in order to provide more experience to the students, real devices should be used. In this paper, we present a remote control system that is able to manage the access to multiple console connections, in order to configure real devices. The advantage of our system is that it is an easy and low cost solution. This system provides ubiquitous access to the students, 24 hours a day, 7 days a week. Furthermore, it is also very easy to extend its system to other fields like Internet of Things.

**Keywords**— Remote control System, Internet of Things, Remote laboratory.

### I. INTRODUCTION

Nowadays, many people and enterprises that have Internet access are quite high, due to the growth of telecommunications infrastructure and the development of the next generation networks. It causes the need of well-trained professionals with good skills to configure and manage this type of networks.

The best way to train students and recycle professionals is to have a big infrastructure with many devices that can be accessible from anywhere at any time to let them practice. The courses about networks and new technologies based on the realization of practices on real devices, gives as a result, well-trained professionals capable of facing any challenge [1] [2].

The use of many network devices, allows students to do useful practices, getting reproduce any scenario that a student in his professional stage in a company could find [3].

More and more, it is tending to use collaborative teaching methods to improve learning, training and acquisition of skills required by any higher education [4]. However, in on-line courses where students can be in different cities, it is very difficult to carry out laboratory collaborative practices. But if instructors were in possession of a device management platform, where they can control the access to devices and even communicate in real time with their students, these problems would be solved.

The use of laboratory practices based on real devices improve the learning outcomes of training and experience for students [5], minimizing the difference between the lab practices and scenarios or problems that can happen in a company.

In this paper, we have developed a console server to access and manage network devices remotely. This is a software application for control of a teaching laboratory for courses about network management. The application is executed on a computer with a network operating system (NOS), which will be responsible for managing the access to several network devices. Network devices such as routers and switches have integrated a console port for their configuration. Our application is able to handle simultaneously up to 12 console ports, for the configuration of 12 devices. The advantage of our system compared to other existing systems is that it is a low-cost solution. This application has been developed initially for the management of network devices. However, its application scope is very wide and it may extend its use in Internet of things (IoT).

The rest of the paper is structured as follow. Section 2 shows some of the main proposals relating to the implementation of solutions for remote access to laboratories. Section 3 explains the composition of our console server and the hardware used. To manage our application and the access for all users, we need a NOS. Section 4 defines the NOS used and its settings. Section 5 explains the operation of the developed application and the operation of the protocol. It also shows how to connect to the console server. Finally, conclusions and future work are show in section 6.

### II. RELATED WORK

It is easy to find several proposals regarding the possibility of implementing virtual labs. An example is the work presented by J. Lloret et al. [6], where they describe the experience carried out in the subject of computer networks at the Polytechnic University of Valencia. Authors implemented a laboratory that allows remote access to real devices for the students of the engineering of telecommunications.

We find a system similar to the one presented in this paper. The NDG NetLab developed by Cisco [7]. It is a computer whose physical format is a rack unit connected to other auxiliary network equipment, through which it controls the groups of network equipment such as routers and switches, enabling users to access to the system and the topologies laboratories. The architecture of this system is a

server that controls the network laboratory equipment by certain actuators that turn on, shut down and restart the devices by orders of NetLab. With NetLab, a user may reserve time to use laboratory equipment and can perform exercises remotely through the Internet as if he was directly connected to them. However, the use of this system requires a license very expensive.

As Lloret et al. commented in their work [8], the combination of the new tools provided by e-Learning platforms and the new technologies, with a minimum combination of the traditional education instruments, shows satisfactory results, as much for the student as for the company.

In this paper, we show the development of a system that makes a real laboratory accessible from any place, at any time with any device (if the device has an application that allows Telnet connections).

### III. SYSTEM DESCRIPTION

In a teaching academy for networking courses, we can find a lot of network devices like routers and switches. These devices usually have a console port to be configured by the users. However, until now it is necessary to stay physically in the same place to connect the cable and configure it.

In this section we will see how we solved this problem and the equipment and hardware used for this.

Today it is relatively simple that a building used to teaching has a broadband connection to the Internet. We have used this fact to deploy our proposal. As we can see in Fig. 1, we have a preassembled network topology, which may be formed by several devices. All of them will be connected directly to the console server, which will be connected to Internet. The implementation of the console server will not preclude that instructors and local students could use the equipment. However, the possibility of getting them from the Internet through the server opens new possibilities for teaching. In this case, a local instructor can be teaching a class of students who are at home or in different place. It also raises the possibility that an instructor can teach a class on real devices without being physically at the academy.

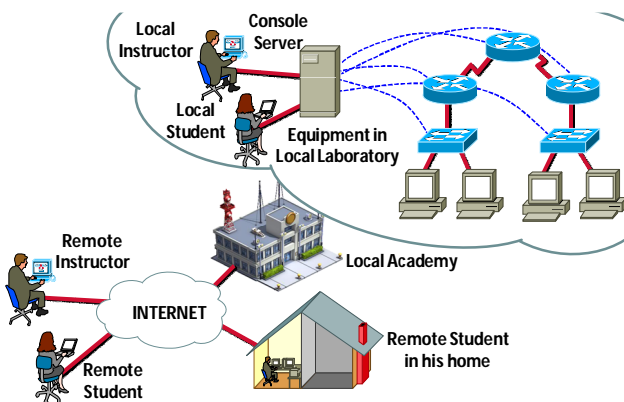


Figure 1. System Architecture.

The computer chosen is a computer-server, Fujitsu TS-C870I. This computer is formed by two processors Intel Pentium III of 550 MHz, with 512 MB of memory RAM. Because the processing capacity of the equipment that should be performed is not too high, these characteristics are sufficient. It is only need a 10 GB hard drive to store and run the OS. Moreover, this computer contains only two integrated serial ports that are insufficient for our proposal. Therefore, we need 2 expansion cards, which allow us to multiply the number of serial ports. To do this, we have used:

- Sunix's 4079T, serial and parallel combo expansion card [9]: It is a serial and parallel combo expansion card [9]. It offers two ports of DB9 Male connection plus one internal parallel port. The serial ports have a data transfer rate up to 921.6 kb/s and the parallel ports offer speed up to 2.7 Mb/s. This board is supported for Windows CE, Windows 98SE / ME / NT/ 2000 / XP / 2003 / Vista, Linux, DOS (See Fig. 2).
- Quatech ESC-100D ISA Eight Port RS-232 Serial Board (DB-25) [10]: This board utilizes a single PCI slot to provide eight independent asynchronous serial ports that share a single interrupt. Serial port connections are made via standard DB-25 male connectors. This board is supported for Windows 95/98/Me/NT/2000/XP/VISTA, OS/2, DOS. It allows a data transfer rate up to 921.6 kb/s, full hardware and software flow control and full modem control (See Fig. 3).

With this hardware, we have 12 serial ports (COM ports), which can be used to configure 12 devices simultaneously. This has been our basic configuration, but it can be easily extended by adding more serial expansion cards (even using USB ports).

Table 1 shows the port distribution in function of available hardware. It is also shown the number of TCP port that is assigned to each serial port.

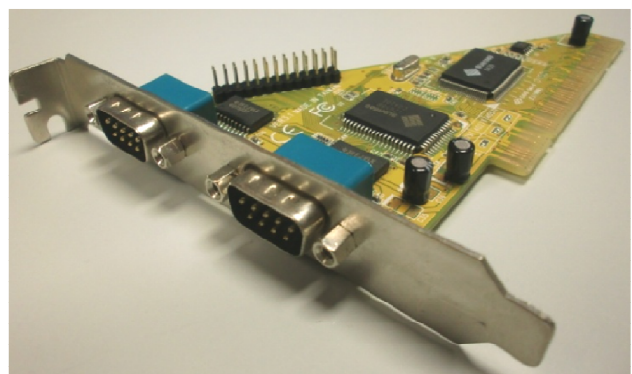


Figure 2. Sunix's 4079T, serial and parallel combo expansion card.

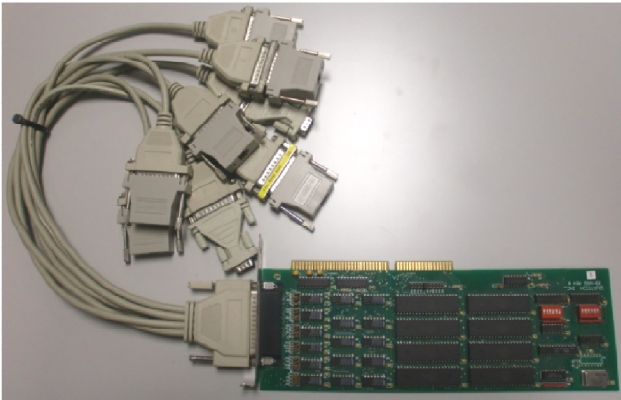


Figure 3. Quatech ESC-100D ISA board.

TABLE I. SERIAL PORT AND TCP ASSIGNMENT

Console Connection	Compute Fujitsu TS-C870I		
	Device	Port connection	TCP Port Assignment
COM 01	Motherboard Serial Port	1	1101
COM 02		2	1102
COM 03	Sunix's 4079T, expansion card	1	1103
COM 04		2	1104
COM 05	Quatech ESC-100D PCI board	1	1105
COM 06		2	1106
COM 07		3	1107
COM 08		4	1108
COM 09		5	1109
COM 10		6	1110
COM 11		7	1111
COM 12		8	1112

#### IV. SERVER CONFIGURATION

So far, we have a computer made up of 2 expansion cards, which allow you to configure up to 12 network devices. But the access management to each port must be done through an application that must run under a NOS. The choice of a NOS or another, depends mainly on the infrastructure that we need. In this section we will see the server configuration that has been selected as well as the NOS that has been used.

- **Level of network security.** This decision is based on the types of security that are considered more appropriate. Networks server-based allow more options regarding the security that workgroup could offer. On the other hand, when security is not a factor to be considered, it would be more appropriate, the utilization of a workgroup network.
- **Number of network users.** When the number of users is small, sometimes it is more practical and easy to manage a working group that a client/server

network, where the maintenance, updating and management of resources will be small.

- **Number of computers in the network.** As in the previous case and for the same reasons, if we have a several computers on the network, it may be better to work in a client/server network.
- **Interoperability of the network.** After identifying the needs of security, users and computers on the network, the next step is to determine the types of interoperability required in the network to get it to behave as a unit.

Unlike the NetWare operating system, an OS not widespread, Windows combines a desktop computer OS and a network OS in a single OS. Microsoft provides client OS versions and server OS versions. On the other hand, the GNU/Linux OS's, and in particular the Ubuntu distributions, also offer different versions for client and server as Ubuntu Server.

GNU/Linux is a general-purpose operating system, multiuser and multitasking. The most popular OS based on GNU/Linux Debian, Ubuntu and Solaris. Typically, a GNU/Linux system is made up of a central computer and multiple terminals for users. This OS not only includes the networking features, designed specifically for large networks, but also some applications for personal computers. GNU/Linux OS's work well on an autonomous system, and as a result of its ability of multitask it also works well in a networked environment.

For our system, we chose the version of Windows 2003 Server from Microsoft. On the computer we installed the 2 expansion cards that we have presented in the previous section. We have also modified the allocation of IRQs for different resources that the OS performs by default, since some of the IRQs were shared by the hardware of the motherboard and the new hardware installed.

Finally, the setting of the serial ports in terms of transmission speed is 9600 bauds, 8 data bits, 1 stop bit and no hardware flow control.

#### V. PROGRAM CODE

In order to establish a secure communication between remote users and the equipment installed in the academy or learning center, we need an application that manages the access to the console server and devices connected to it. In this section, we will see the operation of the application developed.

The object **TRS232TCPXX** (where XX indicates the number of COM port), executes and loads the initial configuration of serial ports that will be used. These settings include the name of connection, the data transfer rate of the serial port in bauds, the port name that will be used and the TCP port number that is assigned to that COM port. This configuration will run every time that the service starts, and it is preconfigured for the 12 COM ports which can be used (See Fig. 4).

**Program Code for TRS232TCPXX object**

```

__fastcall TRS232TCP10::TRS232TCP10(TComponent* Owner):
TService(Owner) {

    const char * IpCommandLine = GetCommandLine();
    CommandLine=AnsiString(IpCommandLine);
    CommandLine.Delete(1,1);
    Path=ExtractFilePath(CommandLine);
    TIniFile *Config = new TIniFile(Path+"Serial2TCP.ini");
    ComPort1->Open();
    ServerSocket->Active=true;
    ServerSocket->Open();

}

```

Figure 4. Program Code for TRS232TCPXX object.

When the service has been initialized, the computer remains in a listening state until it receives a TCP connection request. If a connection request is generated, the object **ServerSocketClientConnect** is executed. When it is running, this object analyzes the port, looking for open connections. If there is any open connection, it is closed, to attend the next connection request (See Fig. 5).

**Program Code for TRS232TCPXX object**

```

1 void __fastcall TRS232TCP10::ServerSocketClientRead(TObject
  *Sender, TCustomWinSocket *Socket){
2 Initialize Variables;
3 Generate file Log.txt;
4 Generate file Users.txt;
5 USERNAME:
6 Collect New character of username;
7 if(character Ascii es "13"){
8 Save username in variable;
9 Goto PASSWORD;
10 } else{
11 Goto USERNAME;
12 }
13 PASSWORD:
14 Collect New character of password;
15 if(character Ascii es "13"){
16 Save password in variable;
17 Open file Log.txt;
18 Open file Users.txt;
19 Search string of characters in file Access.txt;
20 if(cadena caracteres=true){
21 Message: access granted;
22 Save in file Log.txt;
23 }else{
24 Message: access deneed;
25 Save in Log.txt;
26 Close connection;
27 }
28 } else{
29 Goto PASSWORD;
30 }

```

Figure 5. Program Code for TRS232TCPXX object.

If no further connections, the system will request a user name and password. At this time, the object **ServerSocketClientRead** is initialized. This object is

responsible for analyzing all characters entered for the username and password, until it finds an "Enter" (See Fig. 6).

When the username and password is entered, the program will compare these names with the names defined in the file users.txt. If the username and password are correct, the application displays "access granted". In other case it will show "access denied". A register called log.txt, stores all system activity, gathering the data about users that are accessed, date and time, as well as the IP address.

If the application allows the access to the server, the port will open, allowing traffic between TCP port and COM port selected. At this time, the user is connected to the console of the device, allowing to work without problems. However, if access has not been enabled, the port will remain closed.

**Program Code for ServerSocketClientRead object**

```

void __fastcall TRS232TCP10::ServerSocketClientConnect(TObject
*Sender, TCustomWinSocket *Socket) {

    if (ServerSocket->Socket->ActiveConnections>1) {
        Socket->Close();
        return;
    }
    Socket->SendText("\r\nUser:");
    LName=true;
    LPass=false;
    Login=false;
    Password="";
    UserName="";
    ComPort1->Open();
}

```

Figure 6. Program Code for ServerSocketClientRead object.

For each port, there is available an application or service. The services are left enabled to start automatically when the Console server is started. Fig. 7, shows the diagram of our program.

Fig. 8 shows the operation of the protocol and message exchange that are registered when a remote user wants to access to any device connected to the console server.

In it, we can see that the user sends a service request to the server. The service request is a TCP connection to the server. The server sends the response to this request, to establish this connection.

When the connection is established, the server sends the request for username to remote user. In this case, the remote user will respond with a new message, leading the user name information. The server sends the request for password that the user must respond. Finally, the server sends the message of action to take. If the comparison of the characters gathered coincides with one of the entries stored in the file of access, the connection is allowed. Otherwise, access is denied and the application closes the connection.

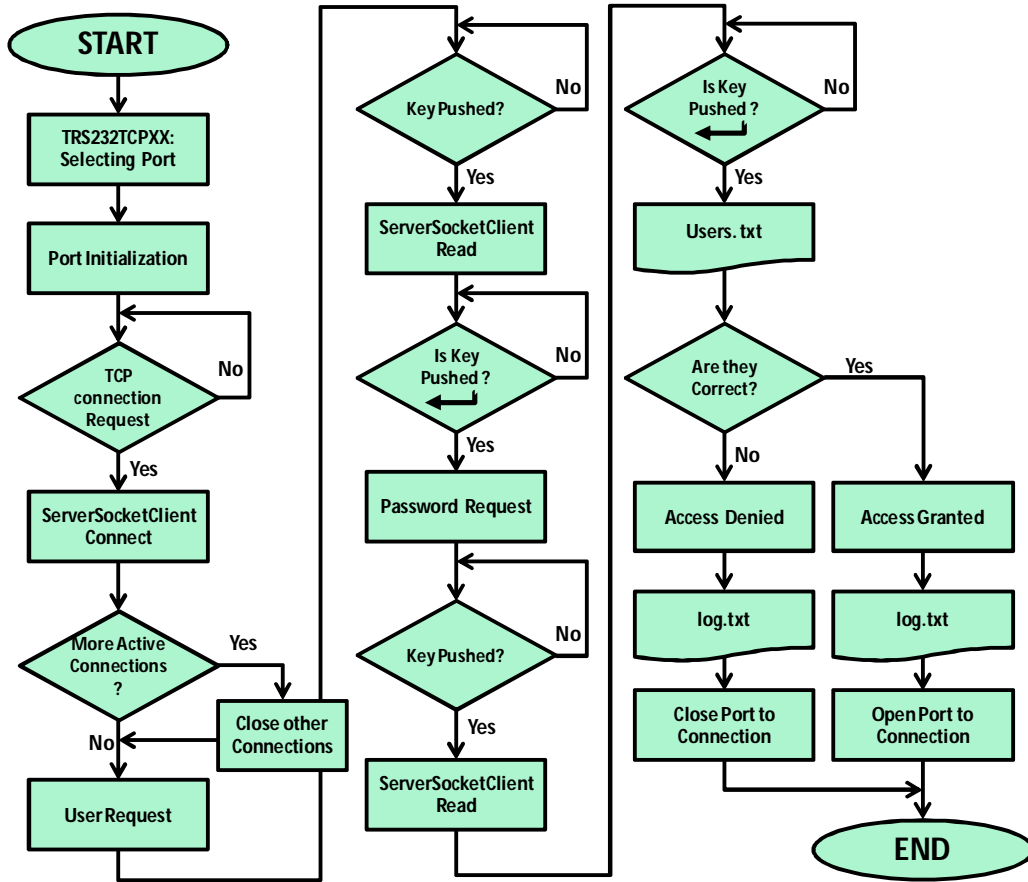


Figure 7. Flow diagram of program

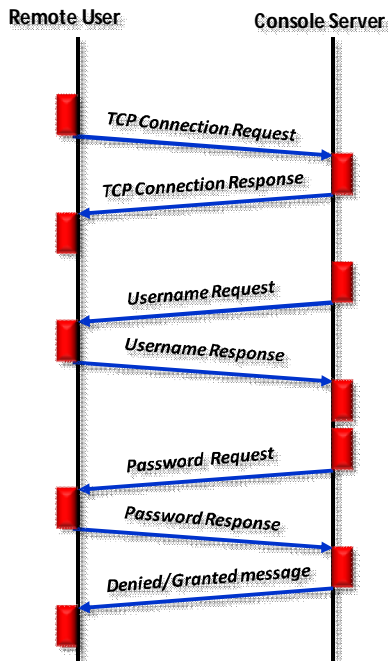


Figure 8. Protocol Operation.

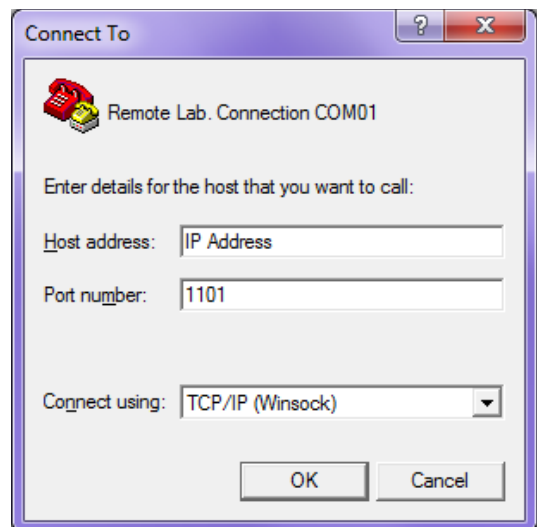


Figure 9. Connection configuration.

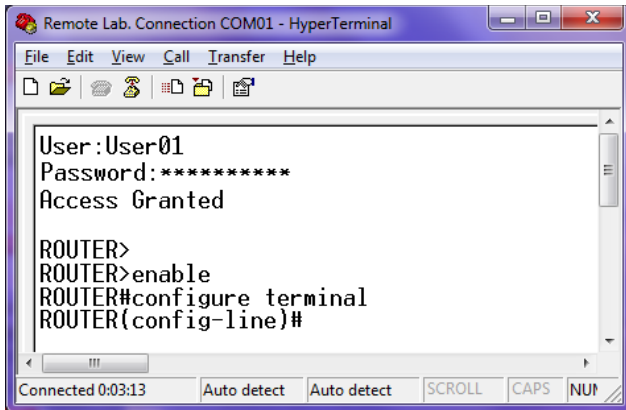


Figure 10. Hyperterminal connection with a router.

Finally, we activate all the services to run when the OS starts. We tried to connect to port COM01. To do this, it is used the Hyperterminal and the application is configured to connect to the COM01 port. To do this, we will define the IP address, the TCP port, which we have defined for COM01 and specify that the connection is done using the TCP protocol. Fig. 9 shows an image of the program

Fig. 10 shows a picture of access to a router. In it, we can see how the application requests the user name and password. As the access is allowed, we can directly access the router configuration.

## VI. CONCLUSION

In this paper, we presented the implementation of a console server for remote management of network devices that can be found in real teaching laboratories. This low cost solution has been designed with the objective of enabling students to do practices on real devices. In addition, the remote access allows students work more time with real devices.

This system could be used also to control applications and devices over the Internet and favor the development and evolution of the field of IoT.

## REFERENCES

- [1] J. Lloret, J. R. Diaz, J. M. Jiménez, and M. Esteve. "Aprendizaje Colaborativo en Profesionales de Nuevas Tecnologías". International Institute of Informatics and Systemics (IIS). Revista Iberoamericana de Sistemas, Cibernética e Informática. Vol. 2, Iss. 2, July 2005.
- [2] J. Lloret, J. R. Diaz, and J. M. Jiménez, "A Collaborative Learning and Evaluation Method In Telematics", In proceedings of Sefi Annual Conference 2004, Valencia (Spain), September 8-10, 2004.
- [3] S. Sendra, A. Canovas., M. Garcia, and J. Lloret, "Cooperative assessment in the hands on skills of computer networks subjects", in proceedings of IEEE Education Engineering Conference (EDUCON 2010), Madrid (Spain), April 14-16, 2010.
- [4] M. García, H. Coll, M. Edo, and J. Lloret, "Mixing collaborative learning techniques for practice evaluation in networking," in proceedings of EAEEIE Annual Conference, Valencia (Spain), June 22-24, 2009.
- [5] J. Lloret, J. M. Jiménez, F. Boronat, J. Tomás, and J. R. Díaz. "Utilización de diversas metodologías didácticas para desarrollar las habilidades de los estudiantes de Ingeniería Técnica de Telecomunicaciones", in proceedings of Congreso Internacional de Docencia Universitaria e Innovación (CIDUI 2006). Barcelona (Spain), July 5-7, 2006.
- [6] J. Lloret, J. M. Jimenez, J. R. Diaz, and G. Lloret, "A remote Network laboratory to improve University classes", in proceedings of the 5th WSEAS/IASME International Conference on ENGINEERING EDUCATION (EE'08), Heraklion (Greece), July 22-24, 2008
- [7] Netlab Home page. Available at: <http://cisco.cit.ie/Netlab.html>. [Last access: May 15, 2012]
- [8] J. Lloret, J. R. Diaz, and J. M. Jiménez, "Creation and Development of an E-Learning Formative Plan", In proceedings of Sefi Annual Conference 2004, Valencia (Spain), September 8-10, 2004.
- [9] Quatech ESC-100D board features. Available at: <http://www.dpishop.com/quatech-esc100d-pci-eight-port-rs232-serial-board-db25-p-204.html> [Last access: May 15, 2012]
- [10] Sunix's 4079T board features. Available at: <http://www.sunix.com.tw/product/4079t.html> [Last access: May 15, 2012]