



Research Article / Araştırma Makalesi

AN APPLICATION OF A REMOTE MACHINE AUTOMATION VIA AN ANDROID SMART DEVICE

İlhan TARIMER*, Ertan DAĞISTANLI

Muğla Sıtkı Koçman University, Faculty of Technology, Department of Information Systems Engineering, Kötekli Kampüsü, MUĞLA

Received/Geliş: 04.12.2012 Revised/Düzeltilme: 28.01.2013 Accepted/Kabul: 15.02.2013

ABSTRACT

In this study, a remote machine automation of a heating pump and an electro–pneumatic driller deployed at the heat center of the university campus has been realized via an Android smart device which has Wi-Fi or 3G access without wire. In order to do this aim, an electronic card has been designed; regarding to the card, C #software adapted to the card has been developed. The application card is connected to a host computer over USB protocol by a software; a smart device which has Android operating system has been communicated over USB protocol with the host computer which has the software. It is expected that the remote automation made in this work would decrease commitment of time, and service costs.

Keywords: Android, USB port, PIC, C#, remote control.

BİR ANDROID CİHAZ İLE UZAK NOKTA DENETİMLİ MAKİNE OTOMASYONU UYGULAMASI

ÖZET

Bu çalışmada, bir ısı merkezindeki bazı makinelerin, Wi-Fi veya 3G erişimine sahip Android bir cihaz ile kablosuz olarak uzak noktadan denetimli otomasyonu gerçekleştirilmiştir. Bu amaçla elektronik bir kart tasarlanmış ve bununla uyumlu bir yazılım hazırlanmıştır. Uygulama kartı, USB protokolü üzerinden sunucu bir bilgisayara bir yazılım vasıtasıyla bağlanmış; bir Android işletim sistemli cihaz, USB portu üzerinden yazılımın bulunduğu sunucuyla haberleştirilmiştir. Geliştirilen sistem sayesinde uzak noktadaki makineler anlık, gerçek zamanlı denetimli ve olarak çalıştırılabilmektedir. Gerçekleştirilen bu uzak otomasyon sisteminin zaman mekân bağılılığını ve servis maliyetlerini azaltması beklenmektedir

Anahtar Sözcükler: Android, USB portu, PIC, C#, uzak nokta denetimi.

1. INTRODUCTION

Smart systems is being integrated to human life together with developing technology, so, time and energy saving are possibly being made. Controlling smart systems has been eased and deleted obligation of users ought to be at working environment, since that internet diffused to every field. Whenever they wished both control and feedback parts have been able to occur.

*Corresponding Author/Sorumlu Yazar: e-mail/e-ileti: itarimer@mu.edu.tr, tel: (252) 211 17 22

Normally, there are some remote control systems made by mobile phones; they are generally paid by SMS and a certain amount of fee must be paid for sending each of SMSs. In order to do this, it is necessary to use specially produced tablet PCs and nevertheless the communication is rather slow, since this system uses serial and parallel interfaces [10] comparing to using of USB.

As known, Google has developed Android operating system and codes of this operating system have been presented as a library [1, 8]. In the study, it has been wanted from internet applications by Android operating system and Android based device. Darcy and Conder have stated about Android and wireless communication protocols, developed a client – host application [2]. Android operating system and basic programming fundamentals have introduced in a book [3]. Rakesh has made a circuit by using USB interface for controlling a DC motor with PIC 18F4550 [4]. Kuna and et.all have presented a context-aware system that uses machine-to-machine communication for retrieving sensor data collected by an Android operating system smartphone. Chena and et.all introduced principles of an embedded OSGi framework in a Google Android platform. They proposed vehicular Android/OSGi platform provides various unique features [6]. Kundu and Paul presented benefits of using Android in low power embedded devices and compared Android Java performances with sun embedded JVM [9].

It has been wanted that a smart device based on Android Operating System would be used for an industrial application in the study. In order to do this, an electronic card has been designed, a software has been written for a PIC micro controller, and a host application has been developed. Thereby, a remote machine automation and a system control have been done via an Android device using Android operating system.

2. ELECTRONIC HARDWARE OF THE WORK

A PIC microprocessor is an interfaced circuit that steers and controls all micro processing elements. Even though there are many USB interfaced micro controllers commercially, PIC 18F4550 has been selected in the study, since that micro controllers have additive interfaced USBs have more costs, and their complex inner structures. The USB interface module is embedded in such micro controller. Besides this, it allows to USB access, and it has got five ports like Ports A, B, etc.

C programming language is frequently being started to use for micro controller applications [7]. As known, Assembly language is such program that provides a programmer to dominate absolutely to a micro controller. It is more suitable selecting C programming language for a micro controller application; since code pursuing is time consuming in long programs and debugging in programs are rather difficult.

USB is known as universal serial communication technology, and it is used in many electronic informatics devices. A USB protocol provides maximum 5 volts, 500–900 mili amperes, and it carries data with pins of Data+ and Data-. It is necessary to know vendor ID and product ID of a USB for controlling it. The numbers of vendor and product ID define that they state the producer's ID and the product's ID. Whenever they want to develop a product, a vendor ID ought to be purchased. It has been communicated that the vendor and product IDs are being matched together with each program which controls the device in the developed USB based device. In case of not being matched of the IDs, it cannot be accessed to the device to be controlled and there doesn't seem a control possibility. That is why it is obliged that the IDs within the hardware and software ought to be same with each.

In order to read the data sent to USB interface, *usb_get_packet* function is used in CCS C compiler [11]. Hence, the data came to the USB device with vendor ID and product ID in the written C project can be read. This function's prototype is as:*int16 usb_get_packet(int8 endpoint, int8 *ptr, int16 max);*

The function of *usb_get_packet* writes the information taken from USB buffer to the PIC's local memory until to end point (EP) value. Once *usb_get_packet* function is called, if no packet in, an invalid result is obtained. For preventing this, USB_KBHIT methods ought to call before *usb_get_packet* function.

In order to send data to USB interface, *usb_put_packet* function is used in CCS C compiler. Hence, the data can be sent to USB device stated in C project with vendor and product IDs. This function's prototype is as:

*int1 usb_put_packet(int endpoint, int *ptr, int16 len, USB_DTS_BIT tgl);*

The *usb_put_packet* function sends the packages (packets) stated with *ptr* until to end point of USB.

3. THE DEVELOPED CONTROL CARD AND SOFTWARE FOR THE WORK

A heating circulation pump, a fuel filler motor, hot water circulation pump, and boiler burner in a boiler room and an electro pneumatic drilling machine will be operated by an Android smart device from far distance as automatically in the application. Therefore, an automation will be made by USB based PIC micro controllers in the work. The developed automation is called as Elektroid. The Elektroid software for the host has been developed within .NET technology platform. All output processes of the control card are being run according to the data comes from the USB interface. Output status info are being sent to computer via USB interface. The electronic schematic of this card is given in fig. 1.

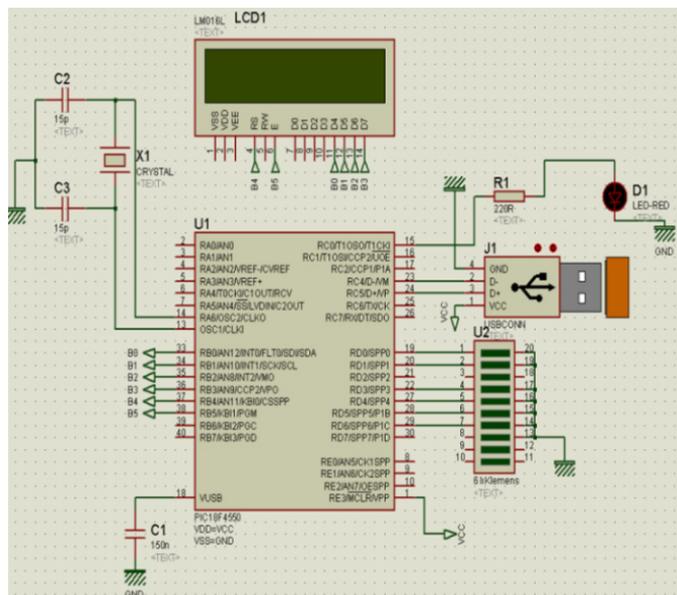


Figure 1. Electronic circuit diagram of the control card

The developed host application 'Elektroid' consists of electronic control card and the section which provides communication with Android device. Some routines written as program blocks within the second part of the application are shown in below lines. In the fig. 2, there are the units which are PIC 18F4550, LCD, USB female socket, crystal and LED. The D+ and D- points of USB interface are connected to PIC. The data come are transmitted to the solid state relays from D port of the PIC.

The host software for Windows operating system has been written by using Visual Studio 2010 IDE and .NET framework 2.0 with C#™ programming language [7]. So, the developed software can be run in a computer which has Windows operating system and framework 2.0. There is a powerful Socket class of .NET which prepared for powerful internet/intranet and at same time umng.dll which is used for USB control has been designed for .NET languages as well. A licensed language called as ElectronicDeviceConnection.dll has been used for controlling USB interface. By help of this dll the data bytes are being sent and read to the device which has a constant vendor and product ID. An object ought to derive from class of Electronic Device Connection after the dll file is added to the project as reference. This object of USB interface has got control aiming functions such as ConnectDevice, GetOutPutStates, SendValueToPIC(byte[]my_buffer) and DisconnectDevice .

Some routines of the developed host application ‘Elektroid’ are given in Table 1.

Table 1. Interaction of Android client and C# server programs

Android Client Codes	Function	C# Server Codes	Function
<pre>InetAddress serverAddr = InetAddress.getByname(ServerIpAddress); socket = new Socket(serverAddr,Integer.parseInt(Port));// Server.Activity.SERVERPORT); okuyucu = new BufferedInputStream(socket.getInputStream()); yazici = new BufferedOutputStream(socket.getOutputStream()); cThread = new Thread(new ClientThread()); cThread.start();</pre>	<p>Connection is being expected from configured IP addresses</p>	<pre>ServerIP = IPAddress.Parse(ipAddress); Listen = new TcpListener(ServerIP, Convert.ToInt64(portAddress)); Listen.Start(); t = new Thread(new ThreadStart(StartRead)); t.Start(); Log(DateTime.Now.ToString() + " Listening was started...");</pre>	<p>Data transmitting is being begun to client</p>
<pre>comingData = new byte[15]; okuyucu.read(comingData); byte[] sifreyolla = ("pass"+ Password).getBytes(); printer.write(sendpassword); printer.flush(); comingData = new byte[15]; okuyucu.read(gelenVeri); String approval = parse(comingData); if(onay.equals("7975")) { picOutputs= "CheckData".getBytes(); yazici.write(picOutputs); printer.flush(); picOutputs = new byte[21]; okuyucu.read(picOutputs); }</pre>	<p>It is requested that the systems are connected to the server after password verification done with the server.</p>	<pre>Soket = Listen.AcceptSocket () ; Log ("The Connection come was Accepted"); Network = new NetworkStream (Soket); Read = new StreamReader (Network); byte [] very, buffer; int socketStatus; System.Text.ASCIIEncoding encoding = new System.Text. ASCIIEncoding Data = encoding.GetBytes ("48"); SendData (veri); buffer = new byte [8192] ; while (true) { socketStatus = Soket.Receive(buffer); Array.Resize(ref buffer, socketStatus); If (socketStatus !=0) { string data = Parse(buffer); if (data.Contains ("pass")) prgStatus = ProgramStatus.PassReceived; else if (data == "DataCheck") prgStatus=ProgramStatus.OutputRequestReceive d else if (data.Contains ("Q") prgStatus = ProgramStatus.ControlDataReceived; Return Delegate(prgStatus.data); }</pre>	<p>It is accepted that the connection request would come. The program status (prgStatus) is being determined according to the incoming data, and this status is being transmitted as information to the main program. Hence, the data came from Android Client can be determined.</p>
<pre>String result= parse(getOutPut); if(result.equals("OK")) { Machine cannot be opened = false; Toast.makeText(context, "Operation completed", 3000).show(); } else if(result.equals("NO")) { Machine cannot be opened = true; Toast.makeText(context, "Output cannot be changed !!!", 3000).show(); btnActive.setChecked(btnActive.isChecked()); Machine cannot be opened = false; }</pre>	<p>These codes precede the results which comes from server relating with the changed output operation.</p>	<pre>public bool SendData(byte [] data) { try { Soket.Send(data); return true; } catch (Exception) { Return false; } }</pre>	<p>The data stated is being sent to the server.</p>

The host application called as Elektroid consists of two parts: Electronic control card and section of providing communication between Android device and machine systems. Some routines of the developed software have been given below table (Table 1).

The screen view of Elektroid host application is given in fig. 2. The screen view after setting with the ctrl card and taking output situation is given as shown in fig. 3.

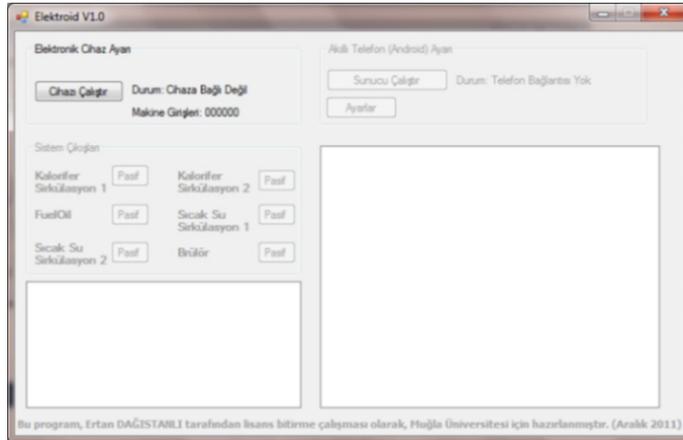


Figure 2. Screenshot of Elektroid host screen

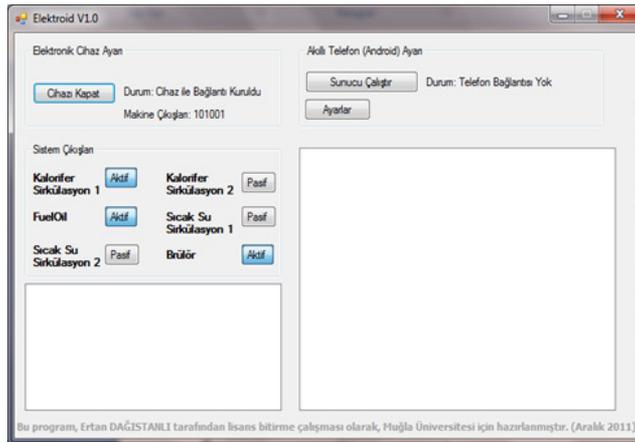


Figure 3. Screenshot of communicating with the card

As shown in fig. 3, the left side of the screen communicates with the control card, and the right side of the screen provides communicating with the smart phone/device using Android. The right side of this interface keeps the adjusting and Logs regards to communicate. Once the electronics control card is plugged to the USB interfacing socket of the computer, it brings the computer to become ready to serve without turning off the computer by means of USB's plug-in features. In case of the connection is done without any fault, the control LED located at the card starts flashing. When this sign is seen, it is inferred that both the program written and the card mounted are ready to communicate with each others. To do this, the button seen at Elektroid Host interface screen-shot is clicked on. If the procedure goes on correctly (without any problem), an information is seen related with communicating between the device and the software in section of the label called as 'Durum' at the interface (screen view). So, it would be seen with the LEDs' flashing as signaling 1 (working machine(s) connected to the card) or 0 (not working machine(s) connected to the card). Therefore, as referencing situations at outputs of the machine(s), the buttons at outputs which control machine(s) come to active or passive forms.

It is understood that the three machines have been now activated as seen from the screen in fig. 4. Once the application software provides the connection with the machines, an activation approval is taken from the control card. It is momentarily informed to the program at the first opening whether the machines are activated or not.

Communication and Controlling with an Android Device

It has been considered that a TCP/IP based client – server application in which Android operating system was uploaded based upon communication protocol with a smart telephone. The Android based application is determined as ‘client’. On the other hand, the application written within C# program has a mission as ‘server’. The Android client software is being communicated with the host software over internet. Operating flowchart of this communication is given in fig. 5.

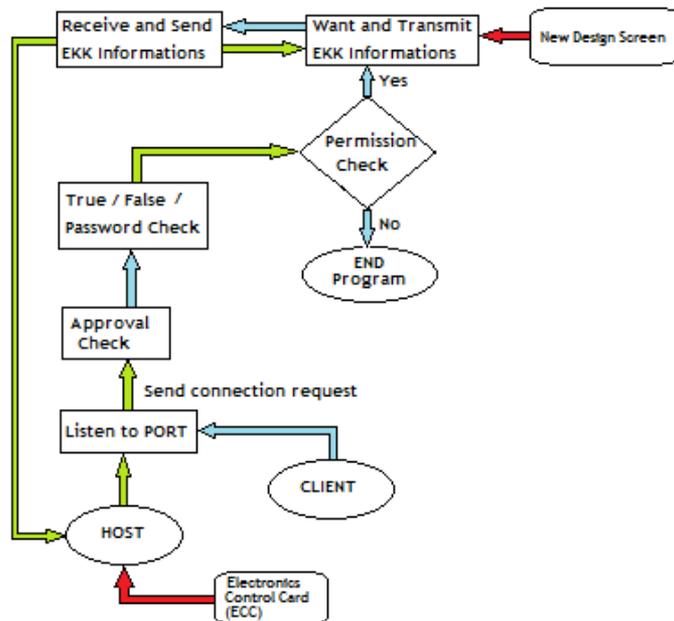


Figure 5. Flowchart of communication between client and host

In order to block any connection request comes from outside to the system, a password checking has been created during data communication as a security measure (Fig. 5). In case of the client stops with any reason, to receive back again the changes made at the electronic control card (ECC) by the host to the client, at the first connection the synchronization is provided as these output requests (informations) send to the client.

Common operations between Elektroid–Client–Android

As known, Android Operating System has been developed by Google for which needs low profile hardware such as smart phones, tablet computers, notebooks, and it has been written within C programming language platform together with the operating system, libraries, and other sub programs [1, 5]. The Android application software written by Java programming language by using Eclipse IDE to be connected to host both from internet and intranet. In the work, communication with the host has been realized over TCP/IP protocol within Android–client

operation by multi-threading technology. The screenshot of the developed application software 'ELECTROID 1.0' in the work is shown in fig. 6. Once the software wants to be run firstly, the screen will be seen as shown in fig. 6. This window occurs within 4 seconds after a voice warning.



Figure 6. Control screen of Elektroid 1.0 application

In order to provide data security between client and host, a password authentication is done before connecting to the host in this application as single side from settings menu. This password must be same as with the one determined at the server to be communicated, otherwise the Android application will automatically shut down itself. The IP address is the IPv4 address of the host to be connected itself. This address could belong to any computer either on the LAN or WAN. Moreover the port value must be same as with the software at the host, and it must be reached to the defined port. After settings completed, the device which operates the client software controls the system remotely over IP.

As shown from fig. 6, the button of "connect to the host" must be clicked on. Then the control screen will appear on the client device's screen. During the sequence, the Android application wants the output knowledge of the control card from the host, and its status 'active / passive' at the interface is being occurred according to this status.

After the sequence, the user will have the right of controlling the process anymore. Afterwards, the devices connected to the control card can be controlled; hence, it is possible to check for user both statuses of the devices continuously and instantaneously as well.

Using the developed software at machines' operation

In order to make the machines to run by the software, a PIC 18F4550 microprocessor embedded electronic control card is being interfaced to the computer. A host software by means of the USB interface is uploaded, and this software changes status of the relays as on/off according to the commands sent. All the SSR relays have been put in an electrical table as shown in fig. 7. It ought to be known that the card given in fig. 1 is supplied with an external voltage source for system stability and security.

The pump motors and an electro-pneumatic driller at a heating centre have been operated by the card as remotely on-line in the study (Fig. 7-8). The pump motors have been driven via the solid state relays since the motors take high nominal currents. The driver relays have been run remotely, and then the electronic control card has been connected successively to a notebook computer. All the system controls have been realized over a smart telephone.



Figure 7. SSR driver table and the controlled heating pump

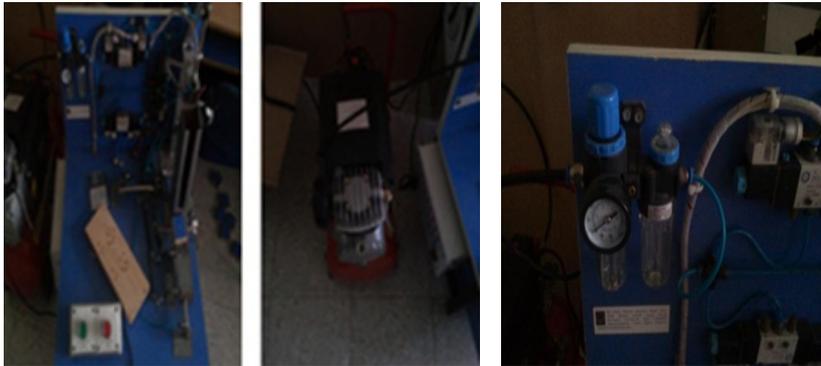


Figure 8. An electro–pneumatic driller

The pumps, the compressor, and pressure manometer have been controlled remotely by the device which runs the client applications. The electronic control card connected as parallel to the start and stop buttons has driven all the machines. As result, the need to put a technical worker for operating the machines at the heat center has been removed due to the developed study. Furthermore, some of the expenditures of the campus' heat center have been eliminated by the automation.

4. RESULTS AND DISCUSSIONS

In the study, we designed a card and written an adoptable software regarding to some industrial needs in order to facilitate work loads of a heat centre in Muğla University campus. By means of that remote control of the machines in a heat centre has kept time saving, and easiness of working conditions. Moreover system logs have also been saved for long durations, hence whenever an examination to be wanted with the logs, it would possibly become realistic.

The developed system in the study has lower production costs, if it is needed to compare the current products, and it needn't to use special devices like GSM modem, tablet PC, etc. All the machines have been driven by solid state relays operated by electronically. The communication between electronic control card and the relays have been provided with cables. In case of number of machines are changed, it is absolutely necessary to change all of the hardware

and software section of the system. The client application done in the study is worked with devices based on Android system.

An Android smart phone was used in the developed application, because of this, the system is based on internet and without charged, it's speedy, flexible and also user–friend. This system can be used both in several industrial works and in smart houses as well.

REFERENCES / KAYNAKLAR

- [1] Kiwi Inc. (2014) Android Developer Story. Available from <http://developer.android.com> [Accession date; 12.02.2014].
- [2] Conder S., Darcey L., “Android Wireless Application Development”, Upper Saddle River, NJ: Addison–Wesley, 2012.
- [3] Murphy M.L., “Beginning Android”, Apress and Springer Science+Business Media, New York, 2011.
- [4] Rakesh R. (2014) USB Interface Board Tutorial Using PIC18F4550. Available from <http://www.instructables.com/id/USB-DC-Motor-Controller-using-PIC18f4550-keyboa/> [Accession date; 12.02.2014].
- [5] Kuna M., Kolaric H., Bojic I., Kusek M., and Jezic G., “Android/OSGi-based Machine-to-Machine Context–Aware System”, ConTEL 2011, 11th International Conference on Telecommunications, Graz / Austria, ISBN: 978-953-184-152-8, p.p. 95–102, June 15-17, 2011.
- [6] Chena M-C., Chenb J-L., Changc T-W., “Android/OSGi-based vehicular network management system”, Elseiver Computer Communications, Vol. 34, Issue 2, 15 February 2011, p.p.169–183.
- [7] Hejlsberg A., Wiltamuth S., Golde P., “The C# Programming Language 2nd Ed.”, ISBN: 0–321–33443–4, Adobe Press, Indiana, USA, 2006,
- [8] Rogers R., Lombardo J., Mednieks Z., Meike B., “Android Application Development: Programming with the Google SDK 1st”, O'Reilly Media, Inc., ISBN:0596521472–9780596521479, p.334, 2009.
- [9] Kundu T.K., Paul K., “Android on Mobile Devices: An Energy Perspective”, 10th IEEE Int. Conference on Computer and Information Technology (CIT 2010), ISBN: 978–1–4244–7547–6, p.p. 2421–2426, Bradford, 2010.
- [10] Botterweck G., Hampe J.F., Stein S., Rosendahl A., “Mobile Home Automation: Merging Mobile Value Added Services and Home Automation Technologies”, Inf Syst E–Bus Manage, Springer–Verlag, 2008, p.p.275–299, 2009.
- [11] Çiçek S., “Programming PICs with CCS C”, Altaş Publishing and Electronics Trade Ltd. Company, p.600, Istanbul.