World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 82-85, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2235

Remote PC Administration Using F_BUS Protocol

K.P. Thooyamani, R. Udayakumar and V. Khanaa

School of Computing Science, Bharath University, Chennai-73, India

Abstract: As we know many software professionals are day-to-day undergoing many problems like Optical migraine, Myopia, other optical disorders, Lumbago and spondylitis and so to avoid and overcome all such problems and relieve the professionals from such dangerous ills, a new protocol called NOKIA F-BUS implemented in mobile phones is introduced. This paper entitled "REMOTE PC ADMINSTRATION USING F-BUS PROTOCOL" is a real time application which helps to remotely access the system administration using NOKIA F-BUS protocol. This paper mainly focuses on three major modules – Client, Server and Mobile component. For Client we install a particular component in the entire Client machine. Server which is a non-dedicated Server we can run it from all machines and the mobile component will interact with the Server application and passes the commands to the Server and the Server works depend on the commands passed by the mobile.

Key words: RS-232 · ETSI · F-BUS · DTR · RTS

INTRODUCTION

As we already known that the user must have to manually control the personal computer by physically present in front of the machine. The **Turn on** and **Turn down** operations are performed manually. Hardware status like Hard Disk, Process name, Processor speed, Ram, Mouse, keyboard setting and other things are displayed by physically controlling the machine [1, 2]. We can't operate the system remotely by residing at far away distance and also it is also time consuming to operate the system manually by physically present in front of the machine [3]. In order to overcome this situation we will go for remotely accessing the system administration using NOKIA F-BUS Protocol [4]. The F_BUS protocol standard is European Telecommunications Standard Institute (ETSI). Consider the Fig. 1 the mobile will get into the server via the data cable i.e RS-232 connected. Then corresponding actives performed with respect to the commands/requests in the SMS received. The server will respond to the mobile using the mobile connected to the server. The server will transfer the command/request to the corresponding PC [5, 6]. Every communication among the server and the mobile will be done using F – Bus Protocol. After extracting the information from the Mobile,

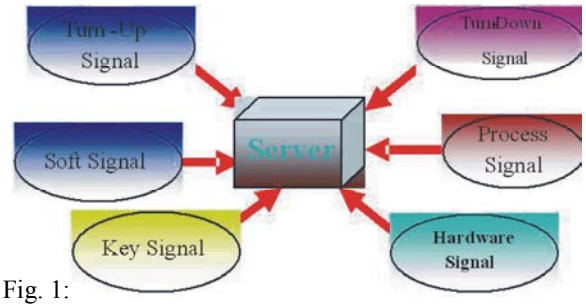

Fig. 1:

the server communications with the corresponding Clients. The Server controls the client through TCP/IP. After the Client passed information to server threw TCP/IP [7].

Implementation Details: The implementation details consist of three modules. They are Server [Application], client [Component] and a Mobile Component Interaction [8]. For the client we install a particular component in the entire client machine. Server which is a non-dedicated server we can run it from all the machines and the mobile component will interact with the server application and Passes Commands to the server and the server works depends on the command passed by the mobile.

Corresponding Author: Dr. R. Udayakumar, School of Computing Science, Bharath University, Chennai-73, India.

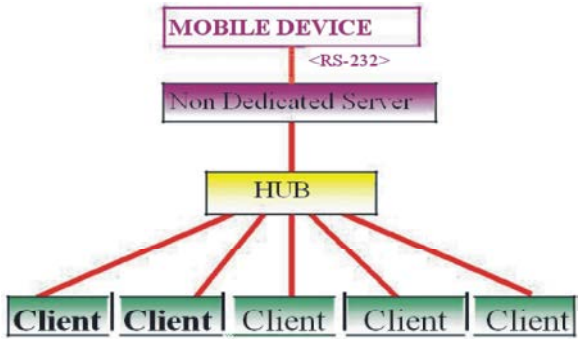

Fig. 2:

Server Application: The End user of this system is going to use this server application. This application is a Non-Dedicated Server [9]. We can view the currently running machine on the server. If the End user selects a particular machine he can view details like Software, Task Manager and Process Viewer. End user can install or deinstall software from the server if the software supports the silent mode.

Client Component: The client component which we installed in the client machines has the six signals

Turn up Signal: When we turned on the client machine, the client component sends a startup signal to the server. Sever accept the signal and it displays the client machine name in the server application. We show the client machine name in the server application in an icon format.

Turn Down Signal: When we turned off the client machine, these time client component sends a signal to the server. Server accepts the signal and it remove the client machine name from the application. We have to remove the client machine name which is shown in an icon format [10].

Soft Signal: This is the mainstay of this project we have the things like handling software, process and hardware. When the client needs a software details it sends a software list to the server. In this case we can install and deinstall software. There were two things in the installation and deinstall- interactive mode and silent mode.

Process Signal: This will send the details i.e., currently running process to the server. If the server wants to kill or terminate a process it will send that details to the client. So the client starts the work for terminating the process. It will refresh the status of process and this will be reflected in the server process screen.

Key Signal: Key Operation like log off, shutdown and restart can be made from the server. We can also shut down entire client machine available on the network [11].

Hardware Signal: In this Client machine sends the hardware status to the server. Hardware status like Hard Disk, Process name, Processor speed, Ram, Mouse, keyboard setting and other things. This also includes the free space available on the system [12].

Mobile Communication with Server: The Mobile will communicate with the Server with the help of F-Bus Protocol. Here a Detail description about the F-Bus Protocol and their functionalities. *About F-Bus Protocol*.

Most Nokia phones have F-Bus and M-Bus connections that can be used to connect a phone to a PC or a micro controller. The connection can be used for controlling just about all functions of the phone, as well as uploading new firmware etc. This bus will allow us to send and receive SMS messages. The very popular Nokia 3310/3315 has the F/M Bus connection under the battery holder [13]. This is a bit of a pain to get to and requires a special cable to make the connection. The left picture above shows the 4 gold pads used for the F and M Bus.

The Difference of M-Bus Protocol and F –
Bus Protocol: M-Bus is a one pin bi-directional bus for both transmitting and receiving data from the phone. It is slow (9600bps) and only half-duplex. Only two pins on the phone are used. One ground and one data. M-Bus runs at 9600bps, 8 data bits, odd parity and one stop bit. The data terminal ready (DTR) pin must e with the request to send (RTS).

F-Bus is the later high-speed full-duplex bus. It uses one pin for transmitting data and one pin for receiving data plus the ground pin. Very much like a standard serial port. It is fast 115,200bps, 8 data bits, no parity, one stop bit. For F-Bus the data terminal ready (DTR) pin must be set and the request to send (RTS) pin cleared.

F-Bus Frame Header

Byte	**0:**	F-Bus Frame ID. We are on Cable (0x1E) See Part 1.
Byte	**1:**	Destination address.
Byte	**2:**	Source address.
Byte	**3:**	Message Type or 'command'. 0x02 (SMS Handling).

Byte 4 & 5: Message length. In our case it is 0x0059 bytes long or 89 bytes in decimal

(SMS)Short Message Service Frame Header

Byte 6 to 8: Start of the SMS Frame Header. 0x00, 0x01, 0x00 **Byte 9 to 11:** 0x01, 0x02, 0x00 = Send SMS message

(SMSC) Short Message Service Centre (12 Bytes)

Byte 12: SMS Centre number length. 0x07 is 7 bytes long. This includes SMSC Number Type and SMS Centre Phone Number

Byte 13: SMSC Number type
Example

0x81-unknown
0x91-international
0xa1-national

Byte 14 to 23: (Octet format) SMS Center Phone Number In this case +61 411990010

(TPDU) Transfer Protocol Data Unit

Byte 24: Message Type

Byte 25: Message Reference if SMS Deliver & Validity Indicator used (Not used).

Byte 26: Protocol ID. Refer to GSM 3.40 - 9.2.3.9 TP-Protocol-Identifier (TP-PID) **Byte 27:** Data Coding Scheme.

Byte 28: message size is 0x33 in hex or 51 bytes long in decimal. this is the size of the Unpacked message

Destination's Phone Number (12 Bytes)

Byte 29: Destination's number length. **Byte 30:** Number type

Byte 31 to 40: (Octet format) Destination's Phone Number

Validity Period (VP)

Byte 41: Validity-Period Code. Time period during which the originator considers the short message to be valid. **Byte 42 to 47:** Service Center Time Stamp?? For SMS-Deliver

SMS Message (SMS-SUBMIT)

Byte 48 to 92: This is the SMS message packed into 7 bit characters. SMS Point-to-Point Character Packing **Byte 93:** Always 0x00

The F-Bus usual ending

Byte 94: Packet Sequence Number **Byte 95:** Padding Byte - String is old and requires to be even! **Byte 96 & 97:** Odd & even checksum bytes.

If the phone receives a valid frame it should reply with something like this below, to say it got the message. Reply frame sent from my Nokia 3310 (showed as a Hex dump)

Byte: 00 01 02 03 04 05 06 07 08 09 **Data:** 1E 0C 00 7F 00 02 02 03 1C 72

This is just like the above Acknowledge command frame. The destination and source addresses are swapped, as this is a frame from the phone to the PC. This message is two bytes long with the first byte representing the message type received (0x02) and the next byte, the sequence number (0x03). The last two bytes are the checksum and should be checked to make sure the data is correct.

Example
Sample frame sent to NOKIA 3310

Byte:	00	01	02	03	04	05	06	07
	08	09	10	11	12	13	14	15
Data:	1E	00	0C	D1	00	07	00	01
	00	03	00	01	60	00	72	D5

This sample frame is used to get the hardware and software version from a NOKIA phone. It is good starting point to test if our implementation of the protocol is working.

CONCLUSION

We consider that many problems, the major ills such as Optical migrane, Myopia, other Optical disorders, Lumbago and spondylitis have Ofcourse created lot of trouble and disturbance to many of the software professionals. we propose a hope that this new protocol NOKIA F_BUS implemented in mobiles would bring a new change in system of application where we can eradicate such major ills of the professionals in software technology. This can be a new module which can be implemented by easy process. we hope this application will bring enthusiasm to the software professionals.

REFERENCES

1. Pragnell, M., L. Spence and R. Moore, 2000. The market potential for Smart Homes, York Publishing Services Ltd.
2. Peersman, G. and S. Cvetkovie, 2000. The Global Sys mobile Communications, pp: 15-23.
3. Collesei, S., P. Di Tria and G. Morena, 1994. Short Message service based applications in the GSM network, Proc. 5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. pp: 939-943.
4. ETSI TS 100901. Digital cellular telecommunications system (phase 2+); Technical Short Message Service (SMS) Point to-point (PP), 3GPP TS 03.40 version 7.50 release 1998.
5. Peacock, W., Nokia F-Bus Protocol, http://www.embedtronics.com/nokia/fbus. html
6. Intel Corporation, MCS 51 Microcontroller Family User's Manual, 1994.
7. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Improving Web Information gathering for personalised ontology in user profiles, Middle-East Journal of Scientific Research, ISSN:1990-923315 (12): 1675-1679.
8. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Detection of Material hardness using tactile sensor, Middle-East Journal of Scientific Research, ISSN:1990-9233 15(12): 1713-1718.
9 Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Blue tooth broad casting server, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1707-1712.
10. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1684-1690.
11. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1691-1697.
12. Udayakumar, R., Kumaravel A. Rangarajan, 2013. Introducing an Efficient Programming Paradigm for Object-oriented Distributed Systems, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(5S): 4596-4603.
13. Udayakumar, R., V. Khanaa and K.P. Kaliyamurthie, 2013. Performance Analysis of Resilient FTTH Architecture with Protection Mechanism, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4737-4741.