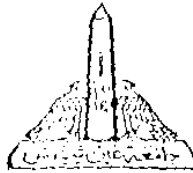


021302
A
G



Ain Shams University
Faculty of Engineering



COMMUNICATION SOFTWARE FOR TERMINAL
EMULATOR; DESIGN AND IMPLEMENTATION

A THESIS

BY

GAMAL AHMED HUSSEIN *Ahmed*

57091

B.Sc. Computer & Systems Engineering

Submitted in partial fulfillment of the requirements for
the degree M.Sc. in Computer & Systems Engineering

SUPERVISORS

Prof. Dr. OSMAN A. BADR

Dr. AHMED Z. BADR

Ain Shams University

Ain Shams University

1996





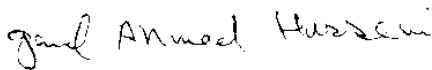
Statement

This dissertation is submitted in partial fulfillment for the degree of Master of Science in Computer & Systems Engineering, to Ain Shams University.

The work included in this was carried out by the author at the laboratories of the department of Computer & Systems Engineering, Ain Shams University.

No part of this thesis has been submitted for a degree or qualification at any other University.

Signature:



Name: Gamal Ahmed Hussein

ACKNOWLEDGMENTS

I wish to express my gratitude to Prof. Osman A. Badr, Computers and Systems Engineering Dept. Faculty of Engineering, Ain Shams University, who made many useful suggestions and for his continuous guidance and help throughout the work.

I also wish to thank Dr. Ahmed Z. Badr, Computers and Systems Engineering Dept., Faculty of Engineering, Ain Shams University, who guided the manuscript efficiently through the various stages of editing , for the helpful advises he gave me and for encouragement in presenting this work and his hand-by-hand support .

I also thank everyone who criticizes the project in order to get a good thesis image .

Examiners committee

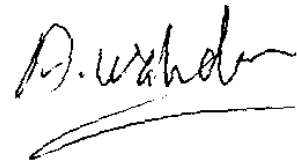
Name, Title and Affiliation

Signature

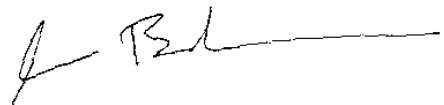
1- Prof. Dr. M. Z. Abdel- Mageed
Systems & Computer Eng. Dep.
Faculty of Engineering,
Al-Azhar University



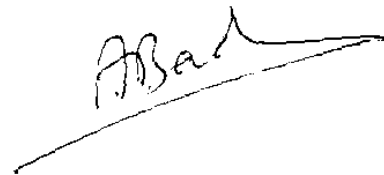
2- Prof. Dr. A. M. Wahdan
Computer & Systems Eng. Dep.
Faculty of Engineering,
Ain Shams University



3- Prof. Dr. Osman A. Badr
Computer & Systems Eng. Dep.
Faculty of Engineering,
Ain Shams University



4- Dr. Ahmed Z. Badr
Computer & Systems Eng. Dep.
Faculty of Engineering,
Ain Shams University



Date: 2 / 1 / 1996

TO WHOM TRYING TO MAKE SOMETHING USEFUL

ABSTRACT

of M.Sc thesis by

Hussein, Gamal Ahmed. COMMUNICATION SOFTWARE FOR TERMINAL EMULATOR;
DESIGN AND IMPLEMENTATION. Computer and Systems Eng. Dep., Faculty
of Engineering, Ain Shams University

This thesis presents a proposed terminal emulator model (**GenTerm**). It consists of a stand alone real time application and a set of utilities implementing a practical way to transfer files between the Personal Computers and Hosts. GenTerm allows the user to make use of both, the personal computers and the hosts. A set of terminals is emulated making a less expensive hardware piece more powerful than using different expensive terminals.

Developing a terminal emulator is somehow similar to writing an Operating System. Also the terminal emulator, as a real-time program, must service two pieces of gears, the keyboard and the communication, both of them are operating in real time and neither of them is synchronized to the running task.

The ISO Virtual Terminal Asynchronous model is followed as possible. GenTerm is optimized by using interrupt to catch characters from the keyboard and receive incoming characters from the host. The interrupt service routines is very short to maximize the baud rate to be used as possible. In the design of communication and keyboard buffers, we faced the producer-consumer problem which is solved using disabling interrupts and object oriented programming.

The implemented software includes a flexible terminal emulator, for DEC VT series, IBM 3101, video 900 series, and a special utility to add specifications for a new terminal. Also the software includes a practical way to upload and download files between PC's and VAX systems.

The GenTerm features was evaluated against a set of well known emulators features. Most of the GenTerm code is written in the "C++" programming language, with some assembly language routines for optimum performance.

The developed software **GenTerm** is designed and implemented on IBM PC and all its functions are tested.

**Key Words : Communication - Emulator - Asynchronous - Terminal-
Design**

LIST OF FIGURES

Figure	Description	Page
2.1	Network Topologies	7
2.2	Differences between LAN and WAN	8
2.3	XMODEM Data Flow	12
2.4	LAN Producers & Characteristics	17
2.5	Standards Organizations	20
2.6	The Open System Interconnection (OSI) reference model	23
2.7	The OSI model applied to PC LAN's	24
2.8	Classification Tree	25
2.9	Transmission of ASYN. Character	28
2.10	Transmission of ASYN. Message	30
2.11	Character Synchronization Block	31
2.12	Format of BISYNC	31
2.13	Byte Count Synchronization Frame	33
2.14	HDLC Protocol	33
2.15	HDLC Expanded 16-bit control field format	35
2.16	Master/Slave Polling Mechanism	36
2.17	ASYN. Block Size	36
2.18	Bisync Protocol	37
2.19	Bysync Block Size	37
2.20	SDLC Protocol	37
2.21	SDLC Block Size	38
2.22	Block Size Versus Protocol	40
2.23a	Mapping display objects to real display	40
2.23b	Some of the OSI virtual parameters	41
2.24	Virtual Terminal S-W Using Shared Data Structure a- A-Mode b- S-mode	43

2.25	Advanced Netware's relationship to the OSI Model	44
2.26a	Advanced Netware Network Interfaces	45
2.26b	IPX protocol Packet	47
2.27	NetWare workstation/server interaction	47
2.28	IEEE 802.3	49
2.29a	UART Registers	52
2.29b	RS-232-C Signals Relevant To PC	54
2.30	Asyn. Protocol Handshaking	55
2.31	Line-Control Register Definition	60
2.32	Bit Rate Assignment	61
2.33	Line Status Register definition	62
2.34	Interrupt Identification Register	63
2.35	Interrupt Enable Register	64
2.36	Modem Control Register	64
2.37	CCITT Frequency Assignments for Full_Duplex Modem	69
2.38	DTE/DCE Interface Using 25-Pin Connector	70
2.39	AT commands classification tree	70
3.1	Polled Terminal Emulator	78
3.2	Interrupt Based Terminal Emulator	79
3.3	Terminal Emulator Data Flow	79
3.4	Emulator Main Loop	81
3.5	Keyboard Service Routine	83
3.6	Communication Service Routine	83
3.7	NEXT Function	85
3.8	PUT Function	86
3.9	Empty Function	86
3.10	FIFO_buffer class	87
3.11	communication port structure	88
3.12	INITIALIZATION WORD	89
3.13	Flow Control	92
3.14	Keyboard Routine	93
3.15	CHARIN Routine	95

Figure	Description	Page
3.16	do_escape_sequence function	96
3.17	DO.ESCAPE.SEQUENCE FUNCTION	97
3.19	OUTPUT ESCAPE SEQUENCE FUNCTION	99
3.20a	CHAROUT Function	100
3.20b	Character Transmission	100
3.21a	Send String Function	101
3.21b	OUTESC_SEQ Function	101
3.22	Keyboard Interrupt Service Routine	102
3.23	Communication Interrupt Handler	105