SECURITY ACCESS AND CONTROL USING RSA ENCRYPTION ON THE 8051 MICRO CONTROLLER

PROJECT WORK DONE AT

INFOGNANA SYSTEMS

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS

OF BHARATHIAR UNIVERSITY,COIMBATORE.

SUBMITTED BY

RAJESH RAMACHANDRAN

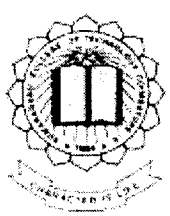
Reg. No:9938M0626

GUIDED BY

Mrs Latha,M.E.

Dept. of Computer Science and Engineering

Kumaraguru College of Technology, Cbe



Dept. of Computer Science and Engineering

Kumaraguru College of Technology,Cbe

Coimbatore-641 006

May 2002

Department of Computer Science and Engineering

Kumaraguru College of Technology

(Affiliated to Bharathiar University)

Coimbatore – 641 006.

CERTIFICATE

This is to certify that the project work entitled

SECURITY ACCESS AND CONTROL USING RSA ENCRYPTION ON THE 8051 MICRO CONTROLLER

Done by

RAJESH RAMACHANDRAN
Reg. No: 9938M0626

Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of Bharathiar University.

Professor and Head

Internal Guide

Submitted for the university examination held on 9/5/2002

Internal Examiner

External Examiner

INFOGNANA

SRJ INFO JNANA SYSTEM (Pr) Ltd.,
COIMBATORE STOCK EXCHANGE BUILDING
683 - 686, TRICHY ROAD, 3rd FLOOR,
COIMBATORE - 641 005
Ph :320067, 320085
e-mail : infognana@yahoo.com

April 25, 2002

To Whomsoever It May Concern

This is to certify that **Mr.Rajesh Ramachandran** of Kumaraguru College of Technology,Coimbatore has done a project with us on "**SECURITY ACCESS AND CONTROL USING RSA ENCRYPTION ON THE 8051 MICRO CONTROLLER (Code name:Info Crypt)**" for **TEXAS INSTRUMENTS (USA)** from December' 2001 to April' 2002.

He has successfully completed his **part of the project**. He was punctual and dedicated towards his project.

For **SRJ INFOGNANA SYSTEM (Pr) LTD**

MANAGING DIRECTOR.

DECLARATION

I here by declare that the project entitled **SECURITY ACCESS AND CONTROL USING RSA ENCRYPTION ON THE 8051 MICRO CONTROLLER** submitted to Bharathiar University as the project work of Master of Computer Applications Degree, is a record of original work done by me under the supervision and guidance of **Mr Babu Unnikrishnan**, Chief of Developments,INFOGNANA SYSTEMS,Coimbatore and **Mrs. Latha M.E.** Lecturer, Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore and this project work has not found the basis for the award of any Degree/Diploma/Associateship/Fellowship or similar title to any candidate of any University.

Place: Coimbatore.

Date: 30/04/2002

Signature of the Student

Countersigned by:

Acknowledgement

I express my profound respect and sincere gratitude to **Dr. K.K. Padmanaban Ph.D,** *Principal, Kumaraguru College of Technology, Coimbatore,* for his kind co-operation in allowing me to take up this project work.

I record my sincere thanks to **Dr. S. Thangasamy Ph.D,** *Head of the Department, Computer Science and Engineering, Kumaraguru College of Technology,* for allowing me to take up the project at *INFOGNANA SYSTEMS private Limited, Coimbatore*

I wish to extend my gratitude to **Mr. Babu UnniKrishnan,** *Chief of Developments INFOGNANA SYSTEMS private Limited, Coimbatore,* for allowing me to carry out the project at their organization and guiding me in completing the project successfully.

I am greatly privileged to express my deep gratitude to my guide **Mrs. Latha,M.E. ,***Dept of Computer Science and Engineering, Kumaraguru College of Technology*, for her valuable advice and encouragement.

I also owe my sincere thanks to **Mrs. Geetha Vellingiri,** *Course coordinator, Master of Computer Applications, Kumaraguru College Of Technology, Coimbatore* for her guidance and immense support throughout my project work.

I express my sincere and heart felt thanks to **Mr. James G Urakkadan BE and Miss Priya. MCA**, of *INFOGNANA SYSTEMS private Limited, Coimbatore,* for their support in finishing the project. express my sincere and heart felt thanks to Jose De Jesus Angel, Angel(Buenos Aires, Argentina) who has guided me through out this project with whom I was in constant touch via email. I also take immense pleasure in thanking everyone at INFOGNANA who were directly or indirectly involved in the success of this project.

Rajesh Ramachandran

SYNOPSIS

The project entitled **SECURITY ACCESS AND CONTROL USING RSA ENCRYPTION ON THE 8051 MICRO CONTROLLER** which has been done partly for **INFOGNANA SYSTEMS** for their clientele **TEXAS INSTRUMENTS** (USA) is an embedded systems project by nature .The product in its hardware form encompassing software does encryption of textual data .Its ideal in situations where by there is a security risk if conventional encrypting software is installed on the system. The product has been built in Borlands C++ on system and Keil C from Keil Software on micro controller.

Although not really embedded in nature into any particular device, the product can be built as part of any other system. The **software** for the project is divided in to **two phases**

i)Encryption of data and its interaction with the system

ii)Connection of the Embedded device to a network(such as the internet)and its subsequent transmission using a protocol ideal for embedded systems.

This project is all about the Phase 1 i.e. Encryption of data and its interaction with the system.

CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The requirements for a hardware solutions has increased duly in recent times owing to decrease in hardware costs and the advantages associated with hardware such as secrecy, compatibility, portability, ability to withstand external disturbances etc. This project aims to provide the software for the device under construction as specified by our clients Texas Instruments(USA).The software is under the testing phase at the moment and further improvements to it are being made.

Objectives

Consider a situation whereby computers connected to a public network such as a Telephone line need to send confidential data .This can be done so using common encrypting software across all systems on the network. Modern Software Engineering has crossed all bounds that nowadays there exist Re-engineering tools which can be used to break into software thereby revealing details of the inner working of the software and other associated details.
Therefore the existence of even the most powerful encrypting software incorporating all the most sophisticated tampering mechanisms do not go fool proof. If broken into the consequences could prove fatal for any organization both Civilian as well as Military when espionage occurs

Solution:

Since it can be seen that anything incorporated within the computer system does not remain safe, the solution is to have it external to the system. Thus the purpose of this project.

A Brief Introduction to Micro controllers with relevance to Embedded Systems

As Microprocessors have become smaller and cheaper, more and more products have Microprocessors "embedded" in them to make them smart. Such products as VCR's, Digital watches, elevators, automobile engines, thermostats, industrial control equipment and medical instruments are driven by these microprocessors **and their software**. The term embedded system means any computer system hidden in any of these products.

The size of embedded software written for micro controllers is small in size depending upon size of available memory(usually not extending more than a few hundred kilobytes of ROM) unlike software written for desktop computers. Therefore we cannot write any desktop applications such as word processing or image viewers for embedded systems. Embedded software for embedded systems are written to perform functions that are typical of micro controllers and in most cases these functions are limited in nature unlike desktop software. E.g. limited code to run a stepper motor, code to receive a character from another device and to send it to another, code to perform small computations like addition, subtraction, small encryption algorithms etc.

What is the difference between micro controllers and microprocessors?

Some people use the term micro controller for the very small end of the range of available microprocessors .Although there is no general accepted definition for micro controller, most people use it to mean a small, slow microprocessor with some RAM, and ROM, other than what is built in and a collection of pins that can be set high or low or sensed by the software directly.

Since the principles of programming a micro controller are the same as those for programming a microprocessor the **term microprocessor will mean both.**

What does it take to understand embedded systems programming?

Anyone can understand the structure of a micro controller ,its associated components ,how they are interrelated to each other.

Does any engineering background required for embedded systems programming?

Physical quantities such as voltage levels, speed of transfer, clock generation etc. are represented as **constants** in programming. Their conversion in to corresponding **analog** or **digital** signals are taken care of by the structure of the entire hardware .Therefore an embedded systems programmer need not bother about the technical details.

Lots of manufacturers supply Micro controllers, E.g.: Intel, Motorola, Hitachi, Arm, Philips, Atmel etc .

Their basic structure remains the same except for a few enhancements such as more memory or extra timer or extra port etc. They are accompanied by their instructions manual explaining how to program them. Understanding of the basic instruction set is more or less enough to understand micro controller programming.

Engineering study more or less explains how these signals of different types represented as constants are converted to hardware signals. They are of no relevance to Embedded systems programmers as mentioned above. In other words the micro controller is nothing but an **empty shell** requiring programmers to have a basic knowledge of its **design** and **instruction set** to be programmed as per requirements. Therefore no engineering knowledge is required to do embedded systems programming.

About the Computer System

The IBM- PC is a versatile one in its design in the sense that we can connect external devices to it and interface it. This provision is made through the Serial, Parallel and USB Ports.

The Device

Code named **Info Crypt**, This device which is a Micro controller(explained in the System Study Chapter will be connected to the Serial Port and will incorporate the Encryption Algorithm. The Algorithm considered is **RSA** .Programming for this will cover the following Areas:

-User Interface Design

-Serial Port Programming(On System)

-Serial Port Programming(On Device)

-RSA Algorithm

The RSA Algorithm

An innovation by Ron Rivest, Adi Shamir & Leonard Adleman,this encryption algorithm is based upon a public key pair and a private key pair.

Mechanics of RSA encryption:

The "key" of an RSA cipher is three numbers: The first is Xpub, the public exponent, the second is Xpriv, the private exponent, and the third is Mod, the modulus The message M (which must be shorter than Mod) is interpreted as a number.

It is broken into chunks as necessary to meet the length requirement. This number is raised to the power of Xpub, modulo Mod, to give C, the cipher text. C may in turn be raised to the power of Xpriv, modulo Mod, to give M. The public key is the pair (Xpub, Mod). The private key is the pair (Xpriv, Mod). If Bob and Alice want private communications, they each generate a pair of keys, then may exchange public keys using a non-secure channel, such as email or telephone.

Thereafter, when Bob desires to send Alice a message, he can encrypt it using Alice's public key and send it via a non-secure channel. When Alice receives it, she uses her private key to decrypt it and read it. The nice part is that Sam, even though he has been able to read every message sent over the non-secure channel, cannot decrypt that message; he would need Alice's private key, which she has never had to send.

Example

```
P  = 61    <= first prime number (destroy this after computing E and D)
Q  = 53    <= second prime number (destroy this after computing E and D)
PQ = 3233  <= modulus (give this to others)
E  = 17    <= public exponent (give this to others)
D  = 2753  <= private exponent (keep this secret!)

Your public key is (E,PQ).
Your private key is D.

The encryption function is: encrypt(T) = (T^E) mod PQ
                                       = (T^17) mod 3233

The decryption function is: decrypt(C) = (C^D) mod PQ
                                       = (C^2753) mod 3233
```

To encrypt the plain text value 123, do this:

encrypt(123) = (123^17) mod 3233

= 337587917446653715596592958817679803 mod 3233

= 855

To decrypt the cipher text value 855, do this:

decrypt(855) = (855^2753) mod 3233

=

504328889584160687344228991273944666314538783600355093155549675645010556286120825599787442454281100543834986542893363849302464514415078517209179665478263530709963803538732650089668607477182974582295034295040790358184594095637793858659893688380836028401325097686207669773967533250542826093475735137988063256482639334453092594385562429233017519771900169249169128091505960191787601713497254392792156967017899021343071464689712796102771813783945869677289869342365240311693217089269617643726521315665833158712459759803042503144006837883246101784830717585474547252069688925995892544366701432205469543174002285500923863694244485597333306305160738530286321930291350374547194675777671357954965202919790505781532871558392070303159585937493663283548602090830635507044556588963193180119341220178269233441013301164806963340240750469525886698765866900622402410208846650753026395387052663193358473410948761562271260373275973603752373883641480889484380961577570453800810794698006673487779588375828998513279307035335512750904399481789790548993381217329458535447413268056981087263348285463816885048824346588978393334662544540066196452187666947955280230884124659482392751057704911332902568430650522925614273038983208900705151105525061899417123177795157979429711795475296301837843862913977877661298207389072796767202350113992715819642730764074189891904868607481245493157953743771244160143876506914586819640227602776686953090395131496831909732450545234594477256587887692693353918692354818518542420923064996406822184490119135710885424428521120773712238311054554312653073940759278908226060431711333957522660344516452597631618427745904320191345289329932161307440532227470572894812143586831978415597276496357090901215131304157569209798518321041155969357848833665315951327344675243940875769777890849012691532284208094963079297247130442219424390659030814289393O

29158483087368745078977086921845296741146321155667865528338164806795
45594189100695091965899085456798072392370846302553545686919235546299
57157358790622745861957217211107882865756385970941907763205097832395
71346411902500470208485604082175094910771655311765297473803176765820
58767314028891032883431850884472116442719390374041315564986995913736
51621084511374022433518599576657753969362812542539006855262454561419
25880943740212888666974410972184534221817198089911953707545542033911
96453936646179296816534265223463993674233097018353390462367769367038
05342644821735823842192515904381485247388968642443703186654199615377
91396964900303958760654915244945043600135939277133952101251928572092
59788751160195962961569027116431894637342650023631004555718003693586
05526491000090724518378668956441716490727835628100970854524135469660
84481161338780654854515176167308605108065782936524108723263667228054
00387941086434822675009077826512101372819583165313969830908873174174
74535988684298559807185192215970046508106068445595364808922494405427
66329674592308898484868435865479850511542844016462352696931799377844
30217857019197098751629654665130278009966580052178208139317232379013
23249468260920081998103768484716787498919369499791482471634506093712
56541225019537951668976018550875993133677977939527822273233375295802
63122665358948205566515289466369032083287680432390611549350954590934
06676402258670848337605369986794102620470905715674470565311124286290
73548884929899835609996360921411284977458614696040287029670701478179
49024828290748416008368045866685507604619225209434980471574526881813
18508591501948527635965034581536416565493160130613304074344579651083
80304062240278898042825189094716292266898016684480963645198090510905
79651307570379245958074479752371266761011473878742144149154813591743
92799496956415653866883891715446305611805369728343470219206348999531
91764016110392490439179803398975491765395923608511807653184706473318
01578207412764787592739087492955716853665185912666373831235945891267
87095838000224515094244575648744840868775308453955217306366938917023
94037184780362774643171470855830491959895146776294392143100245613061
11429937000557751339717282549110056008940898419671319709118165542908
76109008324997831338240786961578492341986299168008677495934077593066
02207814943807854996798945399364063685722697422361858411425048372451
24465580270859179795591086523099756519838277952945756996574245578688
38354442368572236813990212613637440821314784832035636156113462870198
51423901842909741638620232051039712184983355286308685184282634615027
44187358639504042281512399505995983653792227285847422071677836679451

343638070865797742198535953931662799887897216959634553463364979492211
130176613162074772661131070123214037138822702217232330854726795330151
079980622538354589480248200431447261915961905260340690619309392907241
102849487001671729695177034679099794409750637649296356755580071162181
277276031829217903502904860909762662853966270243925368902563371014711
683274045045830602286763142158159900791642627700054612322919219299711
699076901690259464681041412142044724026616582756805241668614733933221
659591270064563044741608529167218700704514464979322666873214634674901
411858867608368403061906957869900965213906752050197440767765104388511
519416193184799191349243881528220384647292694460849152999588185988551
195149066307311777238132267516945882593638786107243025659809149010321
783848214011365567849341024315124828645291703141004001201636482998531
251663490560537945850894244038552524554777922401046148907527451634251
139921637383568141490479320374263373019878254056996191635201938969821
544786313097737491544784276345325939987417001381631981166453772089441
002854850002696859826445621837941167021518477219093392321850877757901
959332676311413129619398495926138987901669710881027663862316769405721
959325380786434441005121380250817976227237972103521967732684419464861
164029610598990277105325704570163326134310764177000432371524746263931
990118997278453629493036369149008810605312316300090101508393318801161
682151638931046666595137827498923745560511004016477716822716267270781
370122424655126487845492350418521674263831897333324346744490397800171
846897264054621480241241258338435017048853206014756878623180940900121
632419690922520226798801134080730122162644041338873926005230960723861
158554965158001034746119792130767224543803671883253708606713311325811
992279755227718486484753261243028041779430909389923709380536520464621
551472678849615277327411926570911661358008414542148768731039444105421
796393085308968803656085047721445921725001265007170689694281546275631
704588389042191773981906487319080148287390581594622278672774186101111
027632479729041222119941173882045263357017590906786281592815199822141
576527968538925172187200900703891385628400073322585075904853480465641
543498370732876259358914278543182665872946080723896522915990217388871
957736477387265746104008225511241827200961681888284938946788104688471
312655417262097890567845810965179753008730631546490302112133528180841
761229904095764278573163641248809309497707395675884229631711584645691
842024551090298823985179536841258914463527918973076838340736961314091
745229856386682726910433575176771288945278813686239650666540898943941
951619120021607778988768647364818378253248466991683072812203107919351

$$
\begin{aligned}
&64666840159148582699993374427677252275403853322196852298590851548110\\
&40229657916338257385513314823459591633281445819843614596306024993617\\
&53097925561238039014690665163673718859582772525683119989984646027216\\
&46279764077057074816406450769779869955106180046471937808223250148934\\
&07851137833251073753823403466269553292608813843895784099804170410417\\
&77608463062862610614059615207066695243018438575031762939543026312673\\
&77406936404705896083462601885911184367532529845888040849710922999195\\
&65539701911191919188327308603766775339607722455632113506572191067587\\
&51186812786344197572392195263333856538388240057190102564949233944519\\
&65959203992392217400247234147190970964562108299547746193228981181286\\
&05556588093851898811812905614274085809168765711911224763288658712755\\
&38928438126611991937924624112632990739867854558756652453056197509891\\
&14578114735771283607554001774268660965093305172102723066635739462334\\
&13638045914237759965220309418558880039496755829711258361621890140359\\
&54234930424749053693992776114261796407100127643280428706083531594582\\
&30594632682786127020335698034614324569702148437 5 \bmod 3233\\
&= 123
\end{aligned}
$$

1.2 ORGANIZATION PROFILE

Infognana Systems, a Coimbatore based company with a liaison office in Dallas ,Texas ,USA is headed by a group of US based NRI' s working for Texas Instruments undertaking embedded projects across a wide range of development platforms. The company is about 2 years old and the projects undertaken are the low end types. The reason being that Embedded systems programming is not very familiar to many in the IT Industry. By undertaking small projects programmers are able to understand embedded systems more and its underlying concepts.

The Organization has a workforce of 7 budding programmers of which 6 are undertaking projects as well are in training. Their prime focus are on the markets of USA and Israel. Their current projects include Anti Lock Braking System design, Cruise Control Design, Encryption, Industrial Printing automation design(undertaken for a local company here in Coimbatore) and small scale Device Driver Development.

The Organization at the moment concentrates on software only and will be employing experienced people in hardware design in the near future for which interviews are being conducted.

2. SYSTEM STUDY & ANALYSIS

2.1 Existing System available & its limitations

As indicated above, The existing system which in the general sense has a lot of limitations requires a hardware implementation running its corresponding software which will ensure complete secrecy of data transmitted over any public network.

What if the device gets into the wrong hands?

There are no known methods at present in which the code stored in a devices ROM is obtainable. Of course this fact can not be overlooked. But the degree of security is far more higher than conventional encrypting software.

2.2 Proposed System

A Hardware product which is to be connected to the serial port, incorporating RSA Encryption which in turn will be routed back into the system for further processing or normal transmission to destination machine through the network.

2.3 Requirements of the new System

a) Software

-Easy to use interface is the primary concern. The user should not fear that he \ she is using a complicated device and so must be very careful in its usage. nor should any complications be felt while using the device lest they should switch to conventional methods of Encryption.

-The System should not interfere in the normal working of other processes nor cause any Operating System violations.

-The System should operate on devices with a varying degree of Baud Rates (The rate at which Data is transmitted in a given interval of time).

-Provision for connection should by default at COM Port 1 and extended to cover at least 3 more ports i. e COM Port 2,COM Port 3 & COM Port 4.

-Since this is a Technical Software an extensive Help system is required to guide new users in its usage.

- After usage any intermediate files created by the software should be removed from the system to avoid access by unauthorized users.

b) Generation of Error Reports

includes the following two categories of Errors encountered in Serial Transmission:

a)Initialization Error

occurs as a result of improper settings in the transmission parameters such as

-Port Selection
-Framing
-No of Stop Bits
-Baud Rate Selection

b)Transmission Errors

occurs as a result of data Transmission viz.

-Framing Errors
-Parity Errors
-Overrun Errors
-Timeout

c) Log Creation

A Text File is created recording the following details

-Most recent date on which Info Crypt was used
-Time
-Duration

2.4 User Characteristics

Ideal for Military and Industrial usage .This product can be used in situations where by its catastrophic to use conventional encrypting software as mentioned in the System Study & limitations page.

The Users interface acts silently in the background and a button pops up when text files with the following extensions is opened i. e

- MS Word Files
- Acrobat PDF Files
- Word Pad Files, Note Pad Files

Upon clicking the button The program to interface with the device is invoked. The user has to set the parameters i. e

- COM Port Selection
- Baud Rate Selection

At the same time the files contents are loaded in the Transmitting Window of interface software. When the send button is selected the data is transmitted to the device and the encrypted resulted can be viewed in the Receivers Window.

All Efforts have been made to make the software as user friendly as possible in the best possible way.

3. PROGRAMMING ENVIRONMENT

3.1 Hardware Configuration (Minimum Requirements)

On System:

-Pentium 233 MHz

-32 MB RAM

-1 RS 232 Configured Serial Port

-1 MB of Hard Disk for Program Storage

External:

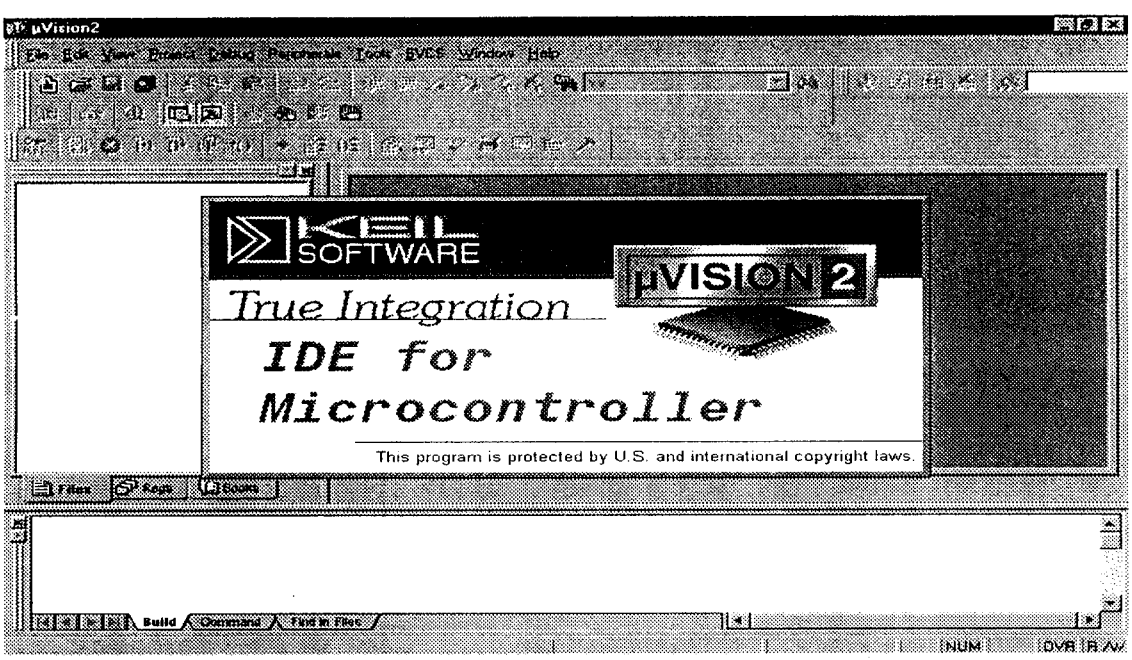
-MCS-51 NMOS single-chip 8-bt micro controller with

32 I/O lines, 2 Timers/Counters, 5 Interrupts/2 priority levels

4 KB ROM, 128 Bytes on-chip RAM.

Micro controller Programming

mu-vision 2 from Keil Software: This is a cross compiler that supports assembly language programming as well as C programming for the 8051 micro controller from a wide range of manufacturers. The programmer has a choice to code in a normal High Level language or in Assembly. After compilation a Hex file can be generated since only Hex codes can be understood by the micro controller. This Hex file can be downloaded in to the simulation kit for execution. The IDE is as shown below.



RS 232 Configured Cables

D Type(9 Pin Connectors)

Length : approximately not more than meters in length

3.2 Software Requirements

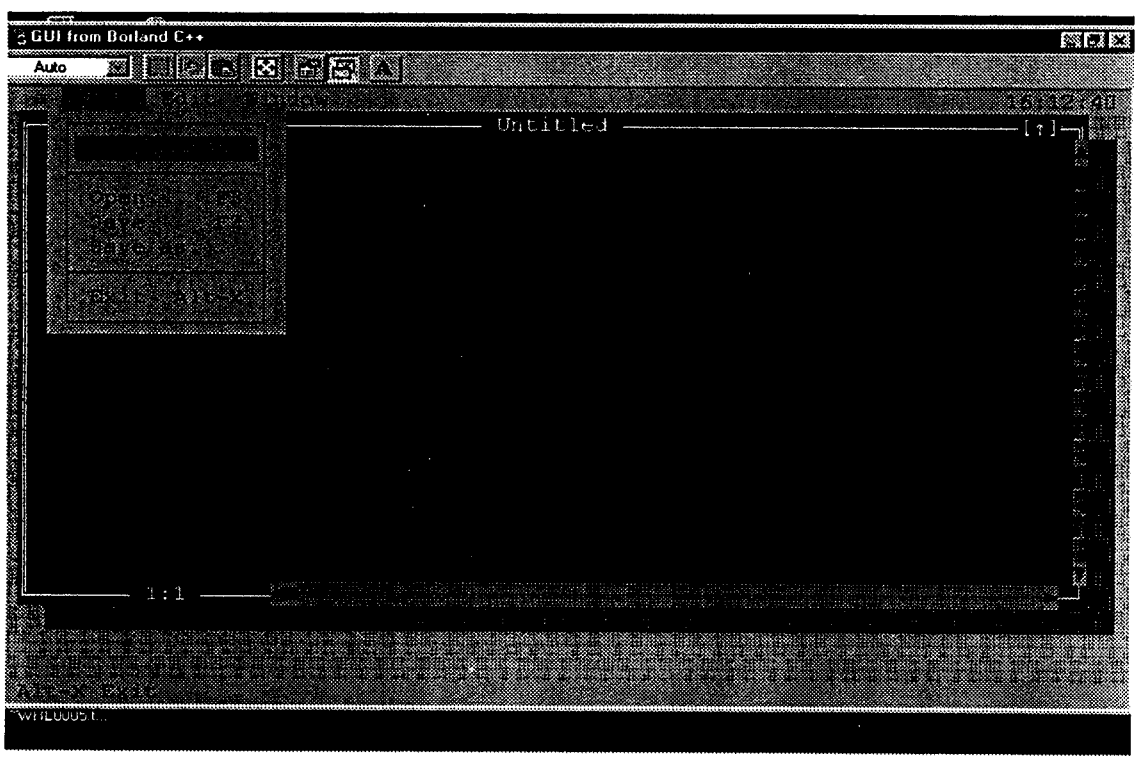
On System Programming

Borland C++ ver 3.1 with Application Framework Creation for DOS
with extended support libraries for Serial Port Programming
from Jean J Labrosse with his book

Brief description about software tools used

Borland C++

The C++ Compiler from Borland provides efficient construction of programs following The OOP's paradigm. Unlike other compilers from other firms, Borlands compiler supports n creation of user interface components such as Menu Bars, Menu Items, Input Box, Buttons etc.A sample application is as shown below.



Why choose Borland C++ than other compilers like Visual C++, Visual Basic etc.

True, that it will be much easier to design user interfaces as well as code for Serial port programming in these platforms however since they are dependant on the Windows Platform as these tools have been designed to be they cannot be ported to non-Microsoft Platforms. A working version in Java will be implemented when this phase is successful to make it platform independent.

How were the User Interface Components built in C++

Borland C++ extension to Turbo Vision(an application framework for creating DOS character mode applications. Turbo Vision is in part a user interface toolkit. It has components that let you construct dialog boxes, overlay windows, build menu based applications etc. In addition Turbo Vision specifies how these components will work together to form a full application as shown in the sample application above.

The advantages of DOS based Turbo Vision compared either Windows development include

- Applications can run on a wider range of machines because character mode user interfaces require less power.
- Applications require less development effort ,which makes it easier to recoup your costs even when a niche market is targeted.
- Turbo Vision is an ideal upgrade path for improving the interface of existing DOS character-mode applications.

Turbo Vision is a C++ class library and it fully uses C++ capabilities. Turbo Vision is an **event driven system** which means it responds to all mouse events.

About extended support libraries from Jean J Laborsse

The complexities involved in Serial port programming in C++ are high since no standard commands are available to implement them with one single command unlike other languages like Visual Basic and Visual C++. There are lots of support libraries available from Corporations, University Archives and individuals but none proved to be more useful than the one supplied by **Jean J Labrosse** along with his book **Embedded Systems Building Blocks, Ready to use modules in C**.

Reasons:

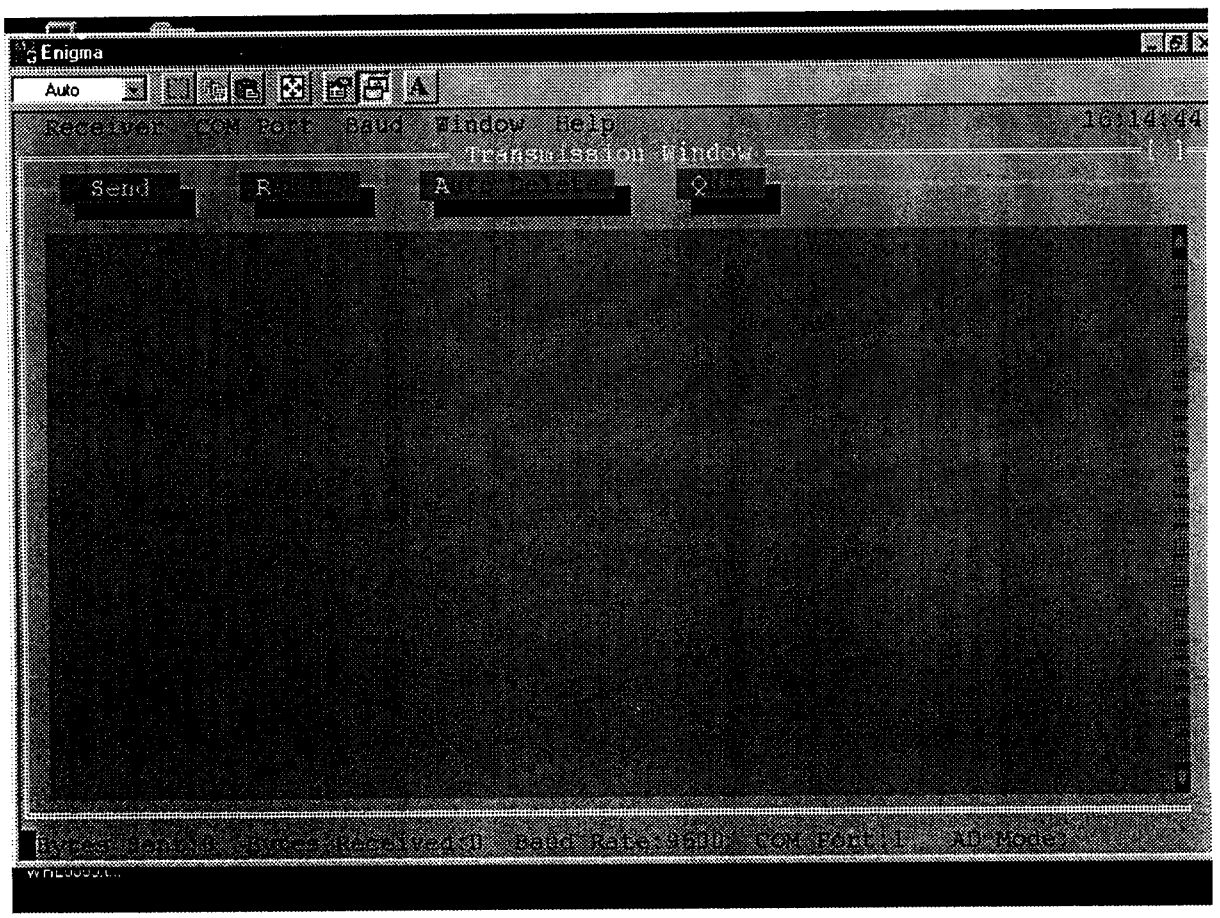
Support for other interfaces like

- RS 232C
- RS 485
- Synchronous Serial Data Transfer
- Parallel Port Programming

4. SYSTEM DESIGN AND DEVELOPMENT

4.1Input Design

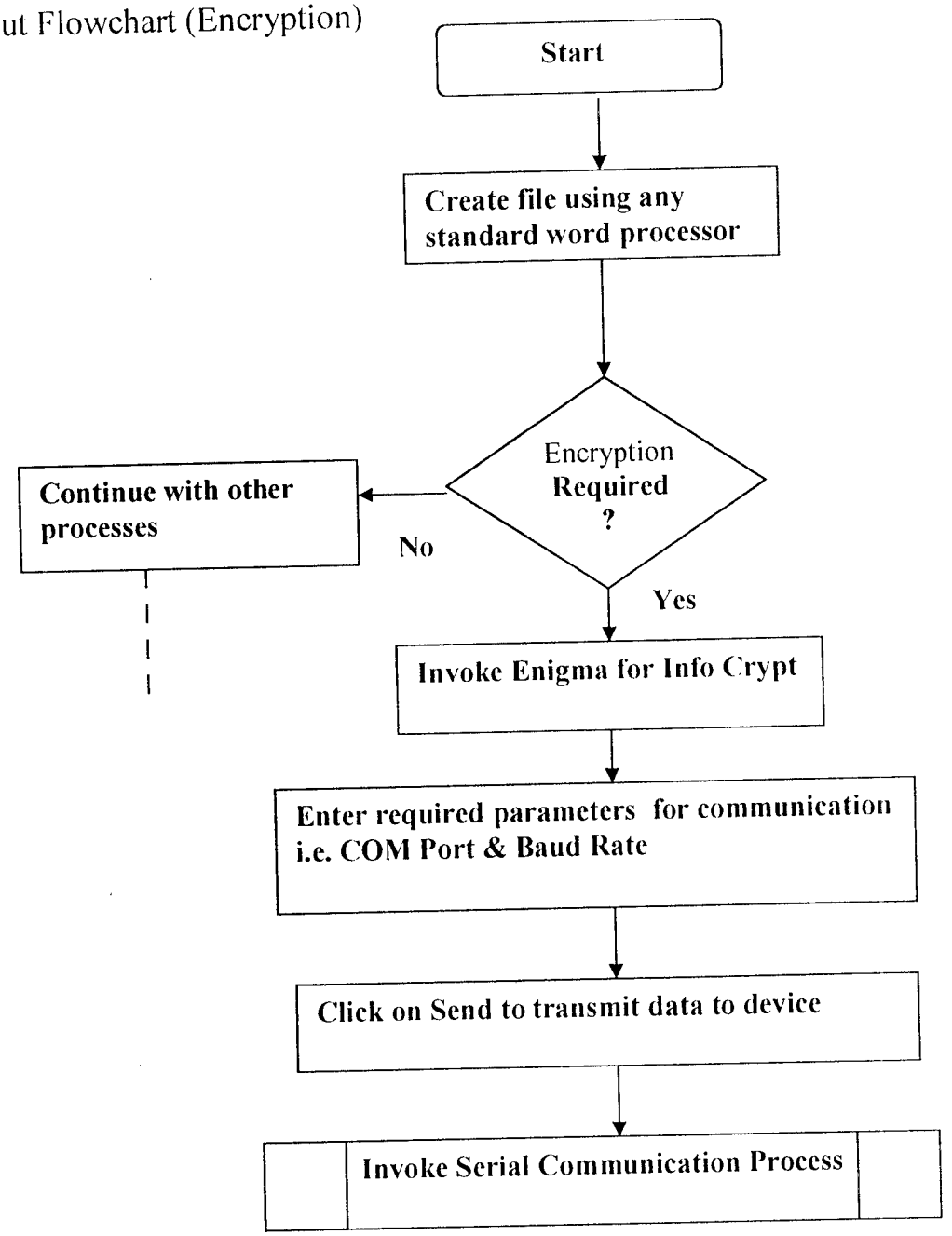
Input to Enigma is through the transmit window ,which will in turn be used as input to Info Crypt after setting all required parameters as shown below.



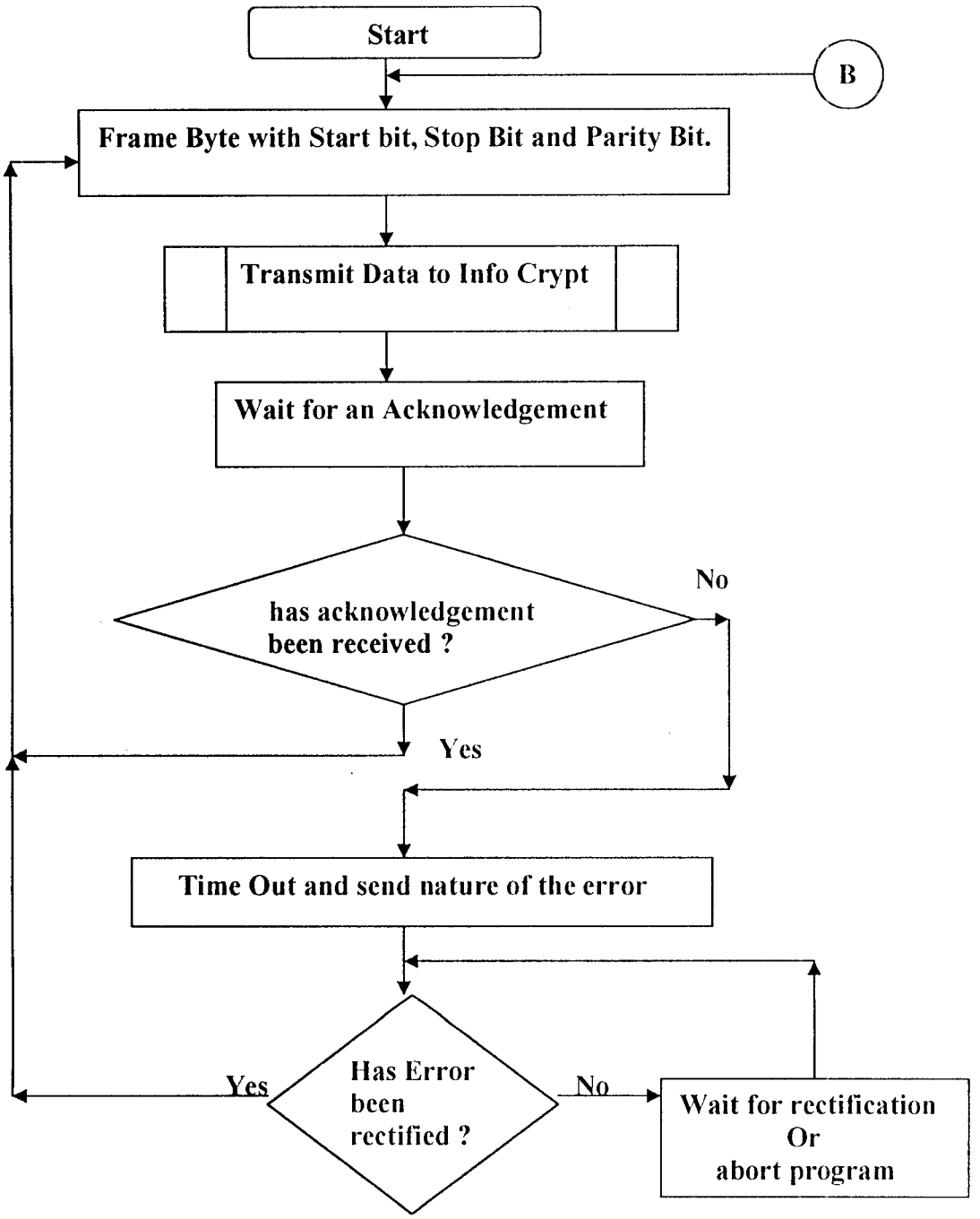
4.2 Process Design

In terms of flowcharts as shown below.

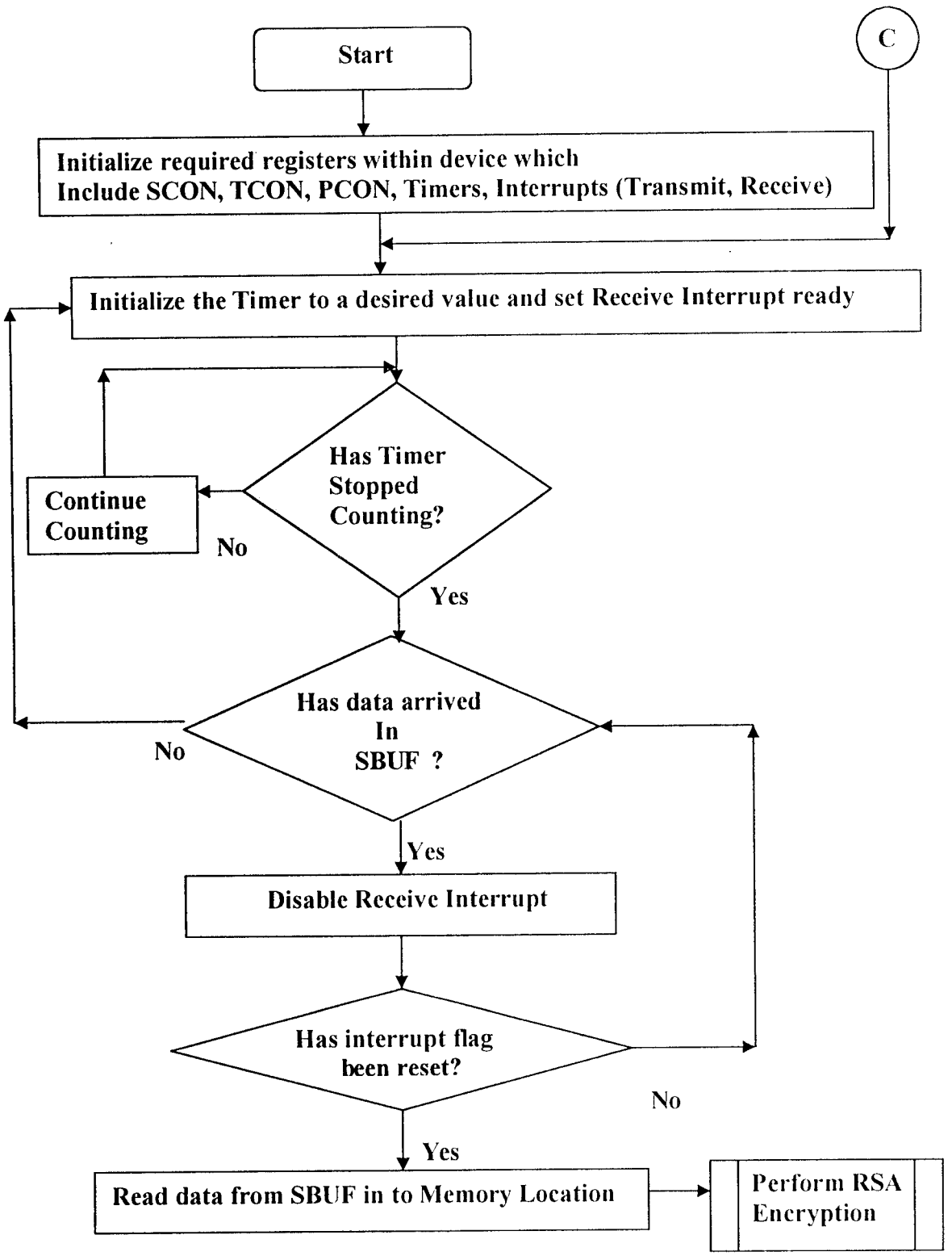
Input Flowchart (Encryption)



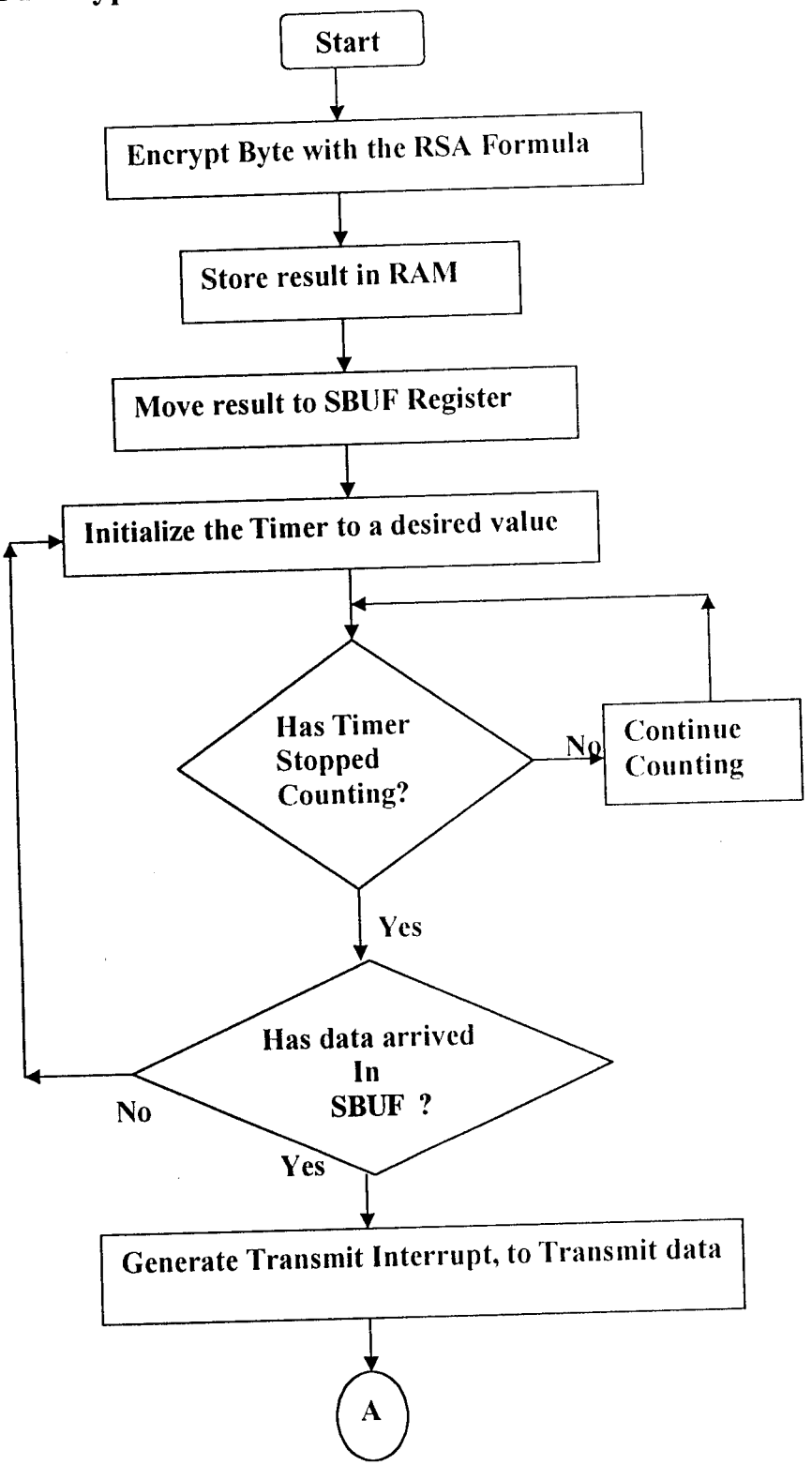
Serial Communications Sub-System Flowchart

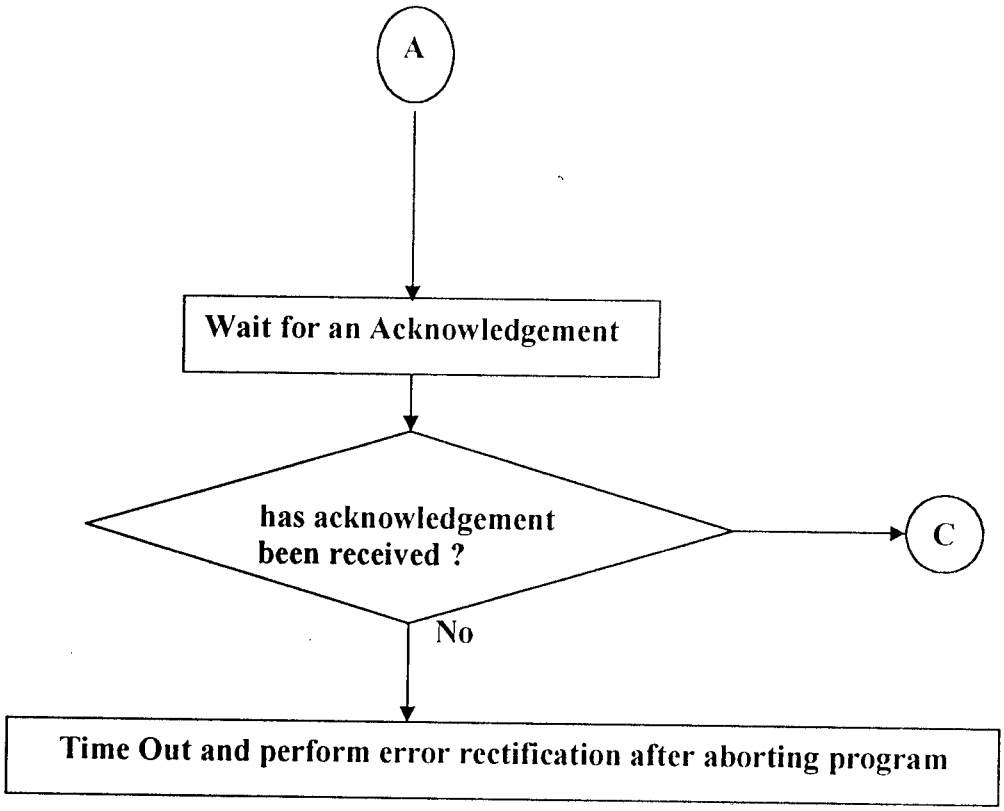


Serial Communication(Info Crypt)

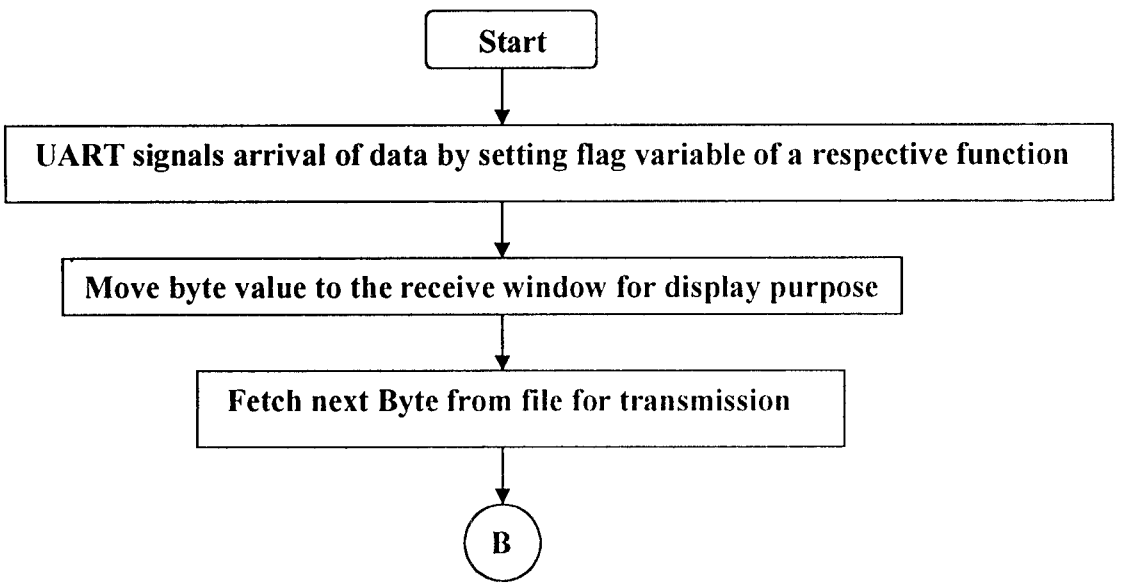


RSA Encryption




A
Wait for an Acknowledgement
has acknowledgement been received ?
C
No
Time Out and perform error rectification after aborting program

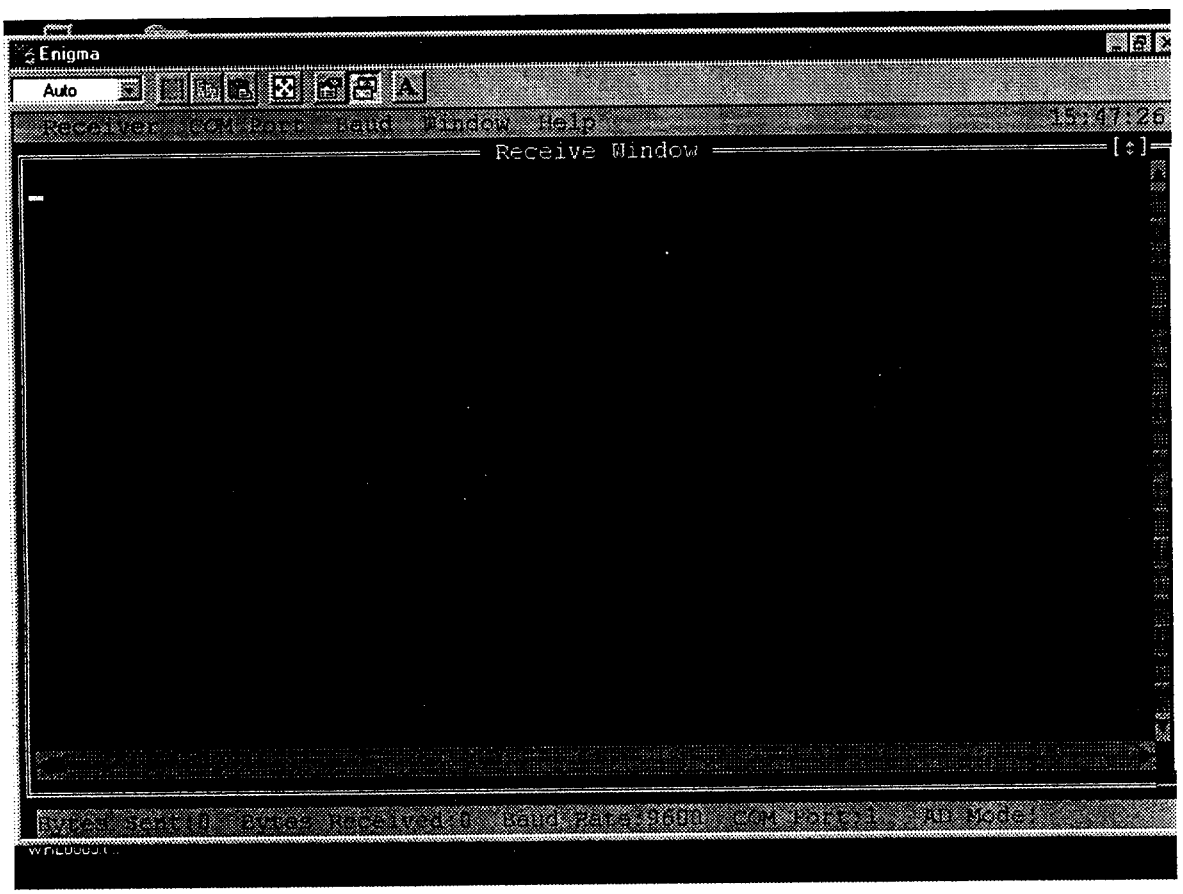
Upon Reception of Data in System



4.3 Output Design

The output to enigma is Cipher Text which will appear in the receivers window

as shown below



Analysis

The Cipher text produced should provide a sufficient challenge to anyone who would like to break the code.

5.SYSTEM TESTING AND IMPLEMENTATION

5.1 System Testing

Main Test Criteria:

Successful Encryption of data to be viewed in the Receivers window.

5.1.1 Stages in Testing(Unit Testing)

a)Successful loading of file into to Enigma, its output to be viewed in the transmit window.

Result:	Problems	Remedy	Conclusion
Successful	Appearance of yet unknown symbols towards end of each line.	Not decide as yet	No mistakes in data been received

b)Transmission of individual byte to device

Result	Problems	Conclusion
Successful	None so far	Correct Error reports are been received as expected of Serial Transmission in case of errors encountered else transmission successful.

c)Receival of byte in Info Crypt

Result	Problems	Conclusion
Successful	None so far.	Byte received correctly and transferred to correct memory location.

d)Encryption of byte

Result	Problems	Conclusion
Successful	None so far	Correct Encryption of byte .

e)Transmission of encrypted byte to system

Result	Problems	Conclusion
Successful	None so far	Correct receival of byte into the system

5.1.2 Other Test Criteria(for small individual modules)

Secondary Operations

which includes the following

i)Reset

This option resets Enigma to the default values

ii)Quit

User Interfaces

which includes

i)Zooming of Windows

ii)Tiling of Windows

iii)Next

iv)Previous

v)Toggle

vi)Appearance of Error Messages, Help Window & About Boxes.

General Conclusion

All working correctly as per requirements.

Unknown Factor

Clogging of the system with huge volumes of data as file sizes get larger. This is yet to be uncovered through further testing.

5.1.3 Integration Testing

Results

The various unit-tested modules were integrated and tested in accordance to their relation with other modules and they were found to be working in perfect harmony

5.1.4 Validation Testing

Results

a) Operating Systems

Enigma worked well with the following Operating Systems

-Windows 95

-Windows 97

-Windows 98

-Windows ME

-Windows NT/2000

b) Serial Ports

Enigma was also tried on 4 serial Ports and found to be working correctly

c) RS 232 Connectors

Type D.

Tested to a distance of **3 meters** and found to be responding correctly.

Condition!

Only a Cable correctly configured for Info Crypt can be used for communication purpose else no results will be displayed.

d) On Existing Encryption Software

Enigma worked successfully and other party encryption software too worked successfully **without any interference** from Enigma.

e) On other software.....

like games, image processors, audio and media players, voice recognition systems etc. **No interference** was to be found

5.2 System Implementation

To successfully implement the system developed, planning is necessary. Proper planning is done to take care of the following:

- The implication of the system as a real time system.
- Resources available for the system using the product.
- Changeover:

Changeover is the process of shifting from the existing system to the new system The changeover from the existing system to the new system takes place after system testing and found to meet all requirements needed by the company.

Implementation procedures of Info Crypt

- Install Enigma through the set up program. Existence of Enigma on system should not pose a security threat as no coding algorithms are present.
- Connect RS 232 Cables to the Systems Serial port and in turn connect Info Crypt to the system.
- Use as Plug and Play with concerned applications.

5.3 System Refinements & Feedback

Moments of meeting held on 10th April 2002

Topic : General Discussion about Info Crypt

Meeting called by: Rajesh Ramachandran

Objectives: To discuss the progress made so far on Enigma & Info Crypt and to bring in new ideas to the product

Members: 6

Main Suggestion to existing product: To include a **File Save Option** to save the contents in the receivers window in to a file.

Arguments: Saving of an encrypted file in the system is hazardous and will violate the principle of what Info Crypt is designed for. However so as not to be too restrictive this option maybe provided for with the user keeping in mind of the dangers of keeping the file on system.

Other Suggestions: To provide for encryption for sound, pictures and moving files on the same product.

Building of the product

Upon successful testing of the product which was concluded on the 24th of April, the product will be built in to a more compact device incorporating its own power supply, casing ,wiring etc. This is the responsible of the more Technical competent Hardware design team of qualified engineers.

Expected date of submission to Client

June 20th.

CONCLUSION

Conclusion

The first phase of the project is fully functional as per requirements although the fact that clogging of the system cannot be overlooked by huge volumes of data due to the slow nature of serial transmission. However the product is still under testing on simulation kits and no obscurities have been uncovered so far.

There are so many products in the form of chips which incorporate Encryption algorithms available in the market. While undertaking this project we learnt how it could be done and the significant advantages of doing so. The following objectives were obtained.

- Encryption on an external system providing good security
- Even if system gets in to unauthorized hands, there is no way of breaking the software containing the code using hacking tools since its not resident on the system
- Even if Info Crypt gets in to unauthorized hands there are no known methods of reading ROM Resident code.

Since the first phase of the project is more or less successful the second phase will commence soon i.e Connection of the Embedded device to a network(such as the internet)and its subsequent transmission using a protocol ideal for embedded systems.

SCOPE FOR FUTURE DEVELOPMENT

Scope for future Development

Connection of Info Crypt is possible to only one system which means at any given time only one file can be encrypted at a time. For parallel processing of several files many devices may have to be attached to many serial ports connected to the system which will prove to be quite inconvenient .A more broader **RS 485 standard** can be used which follows polling i.e. each device connected to a common serial port will be repeatedly polled for data transmission at a fixed time intervals.

Scope also exists for encryption of non textual files like pictures ,media etc though not possible on the same product with the same encryption algorithm owing to memory space restrictions. In other words Info Crypt can take several forms to encrypt non textual matter incorporating more suitable algorithms. This process of doing so is underway by examining conventional software which do so.

Internet Resources

i) www.microlink.co.uk/rs232.html

ii)www.howstuffworks.com/serial-port1.htm

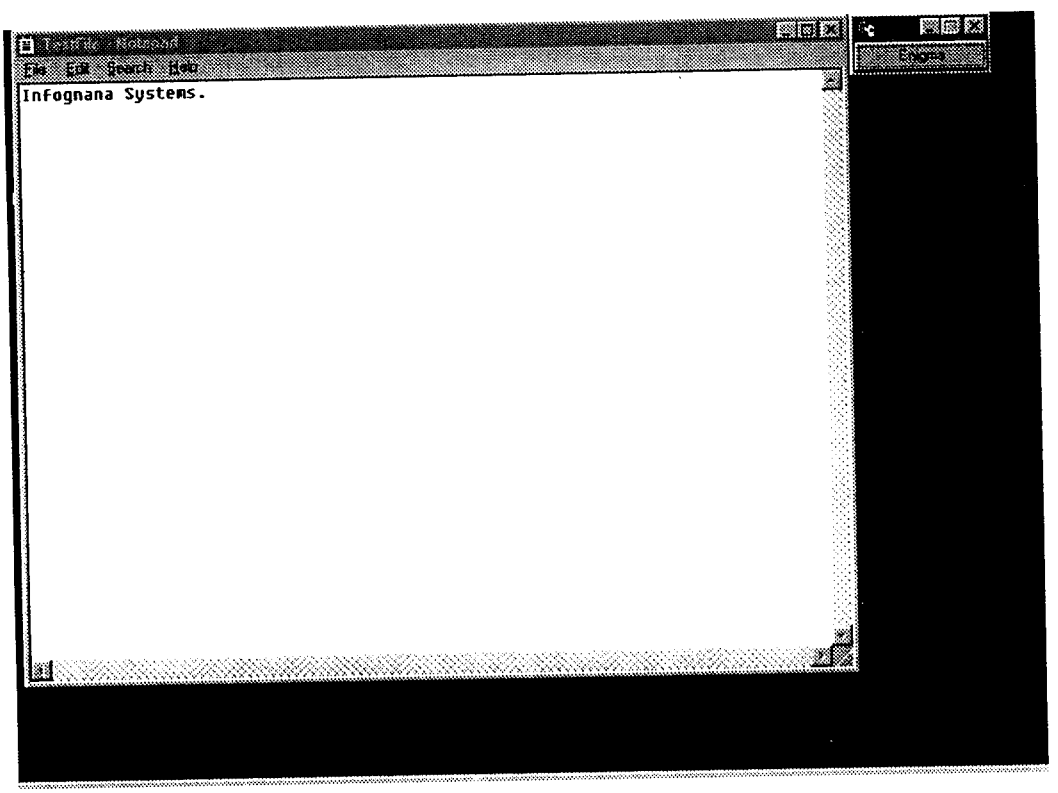
iii)www.taltech.com/TALtech_web/resources/intro-sc.html

iv)www.primenumbers.com

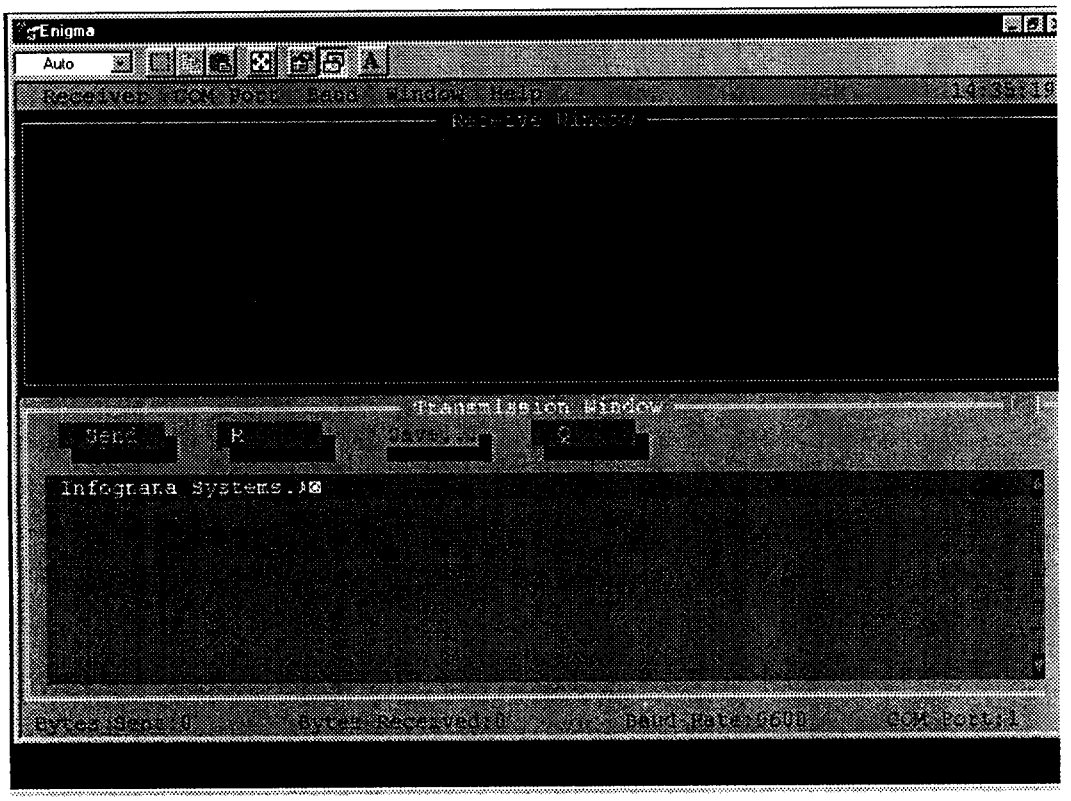
APPENDIX

The following sample outputs labeled from (ii) onwards are the subject of this report.

Output (i) is the result of Windows Programming done in Visual C++.

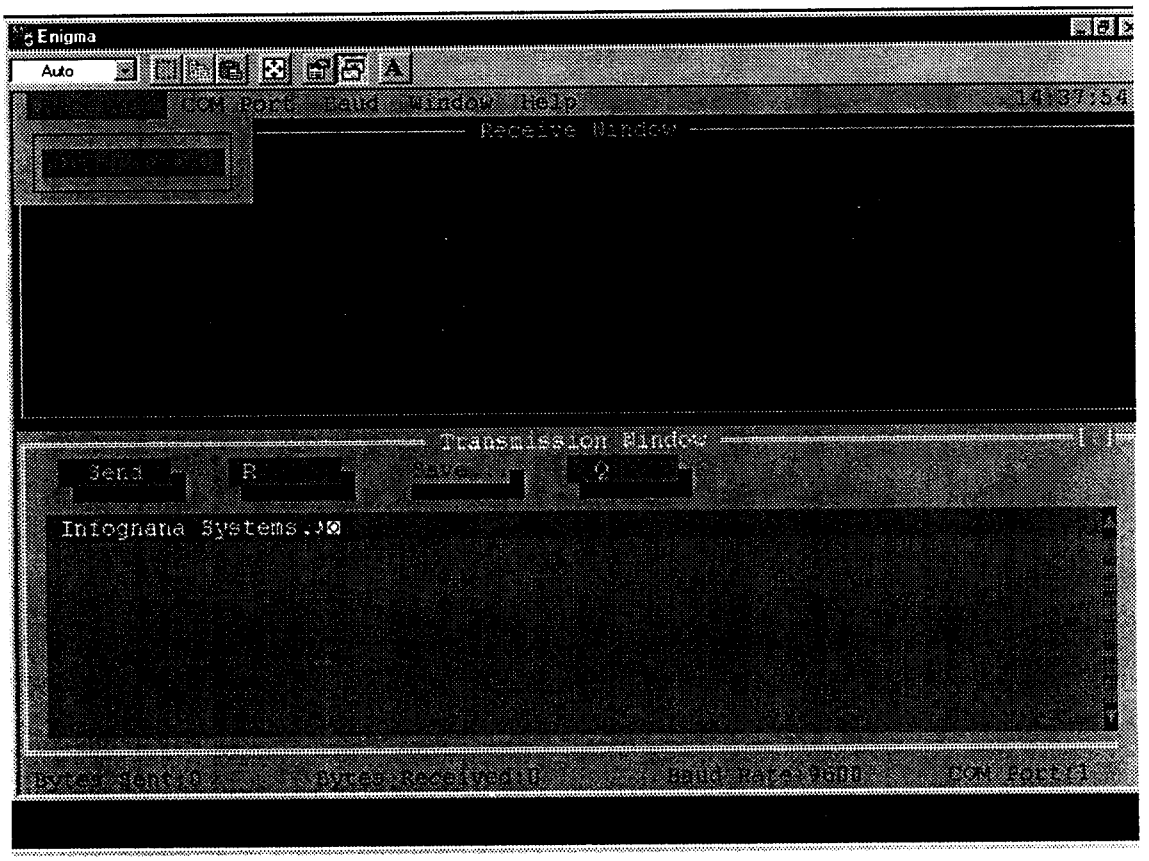


(ii) This is the default screen that pops up when the enigma button is clicked. The screen is split in to two windows a receive window and Transmission window as shown above.



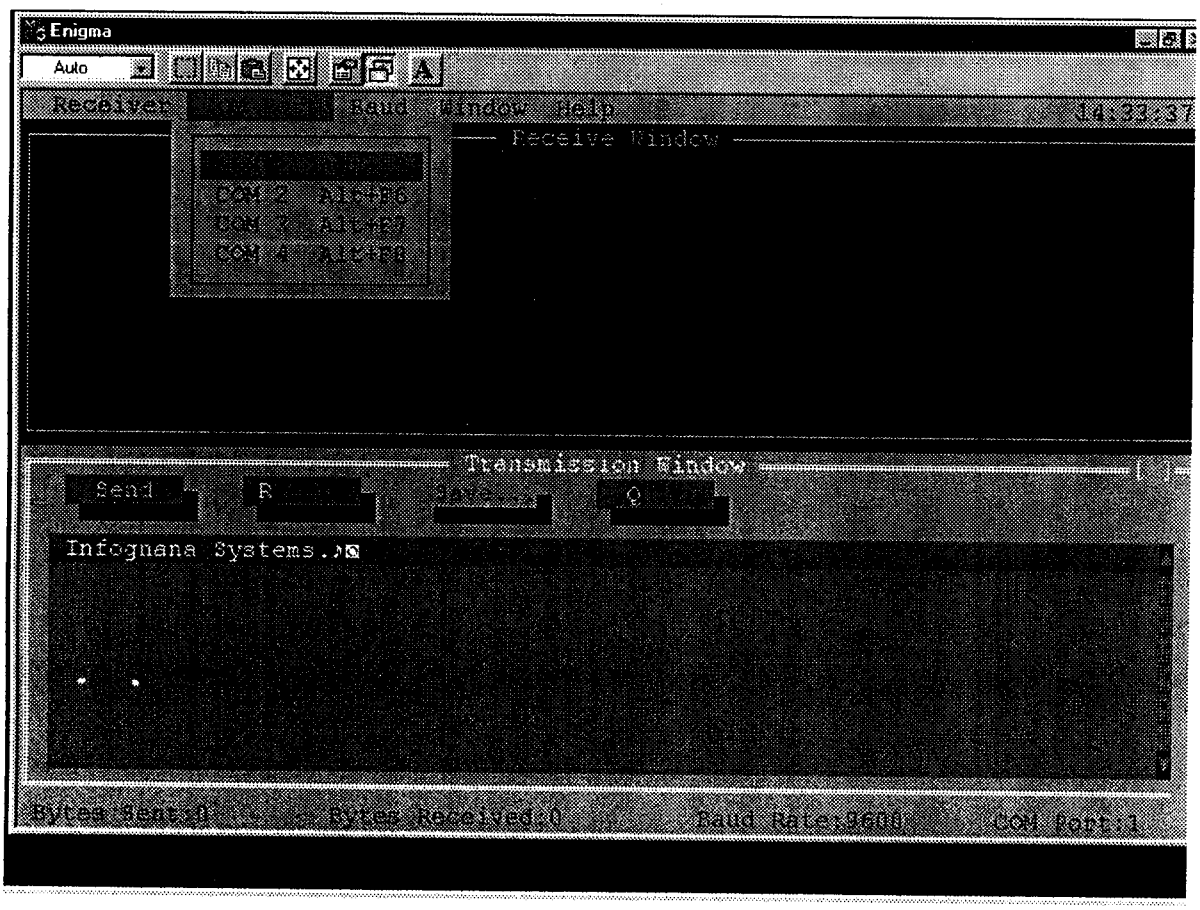
Other components include a Menu Bar, Commands on the Transmission Window and a Status bar. By default the Save button on the Transmission window is dimmed. Note that data found in the word processors window is loaded into the Transmission Window.

(iii)



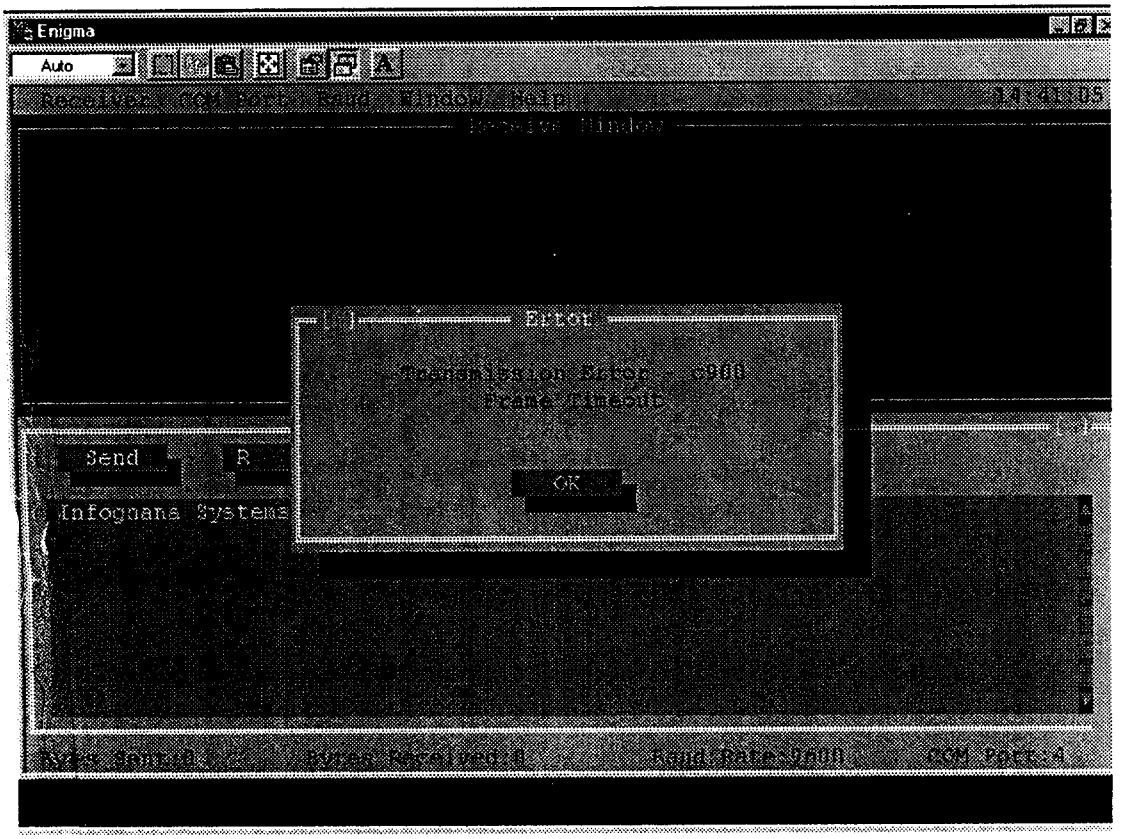
Bringing down Receiver on the Menu Bar and clicking on Clear will clear the contents of the Receive Window from previous sessions of Enigma though this is not necessary since it is done so when ever Enigma is executed.

(iv)



Select the COM Port to which Enigma is connected to as shown above.

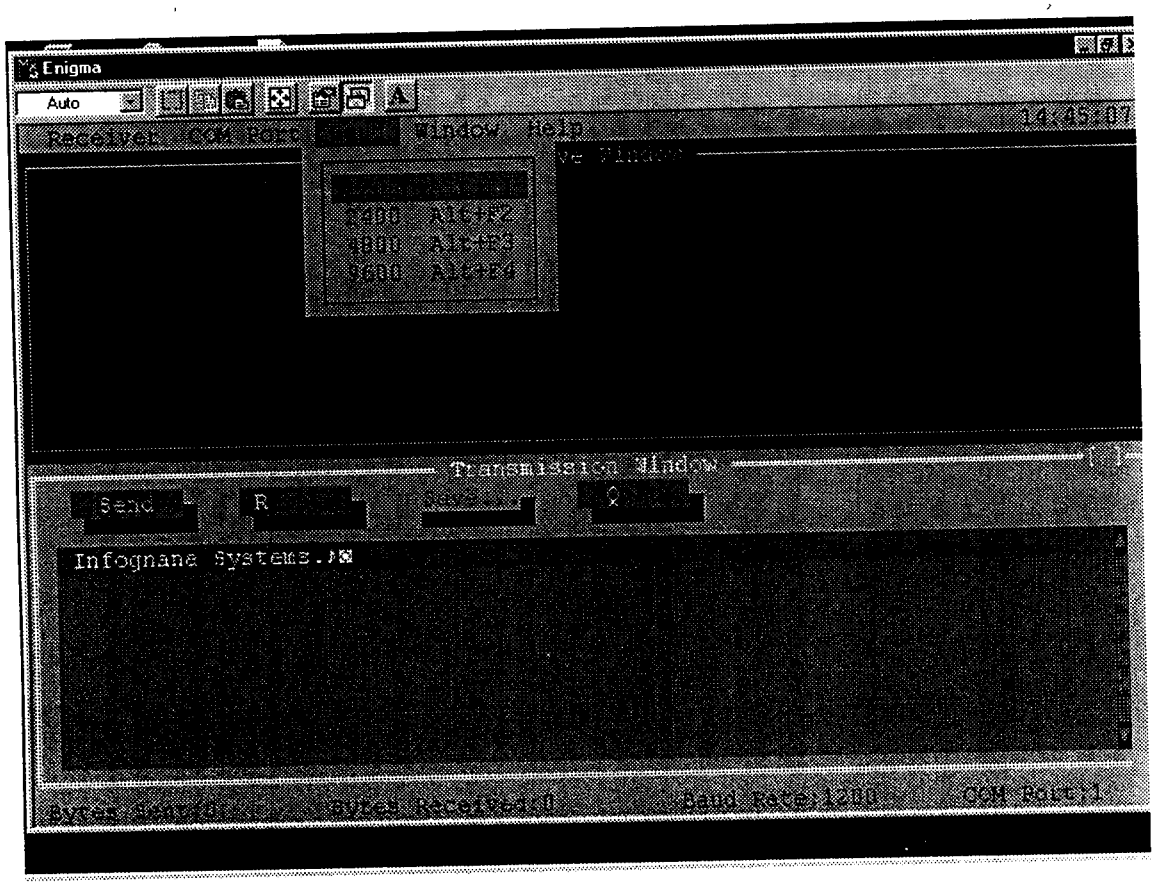
(v)



Assuming the Port Selected is not available,a Transmission Error will be displayed.

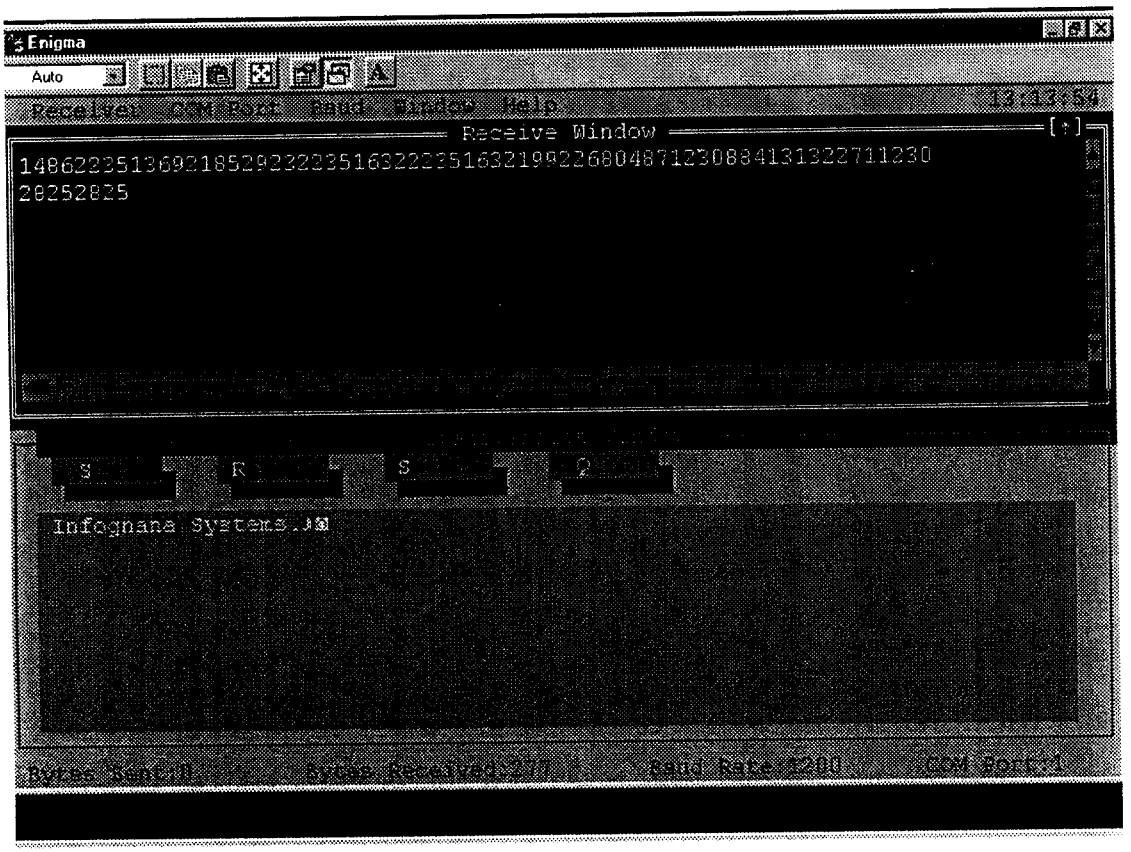
Selectan available port to which Info Crypt is connected to.

(vi)



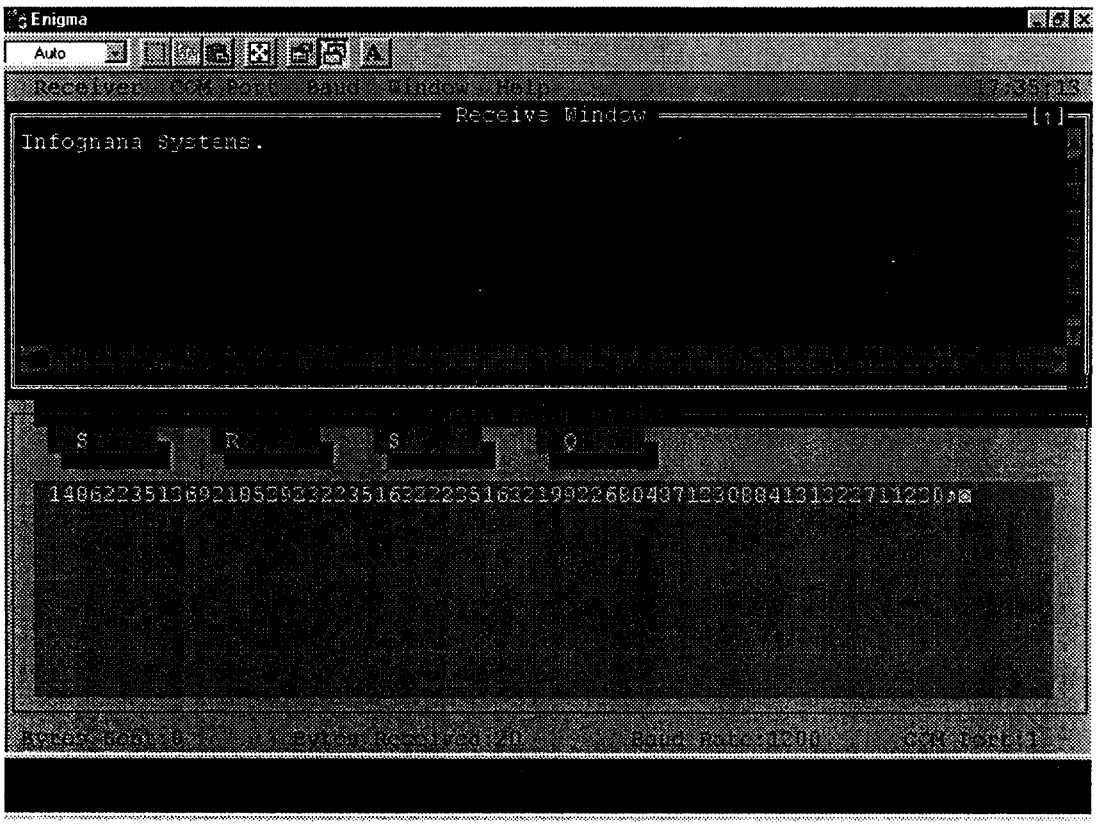
Select Baud Rate from Baud Menu

(vii)



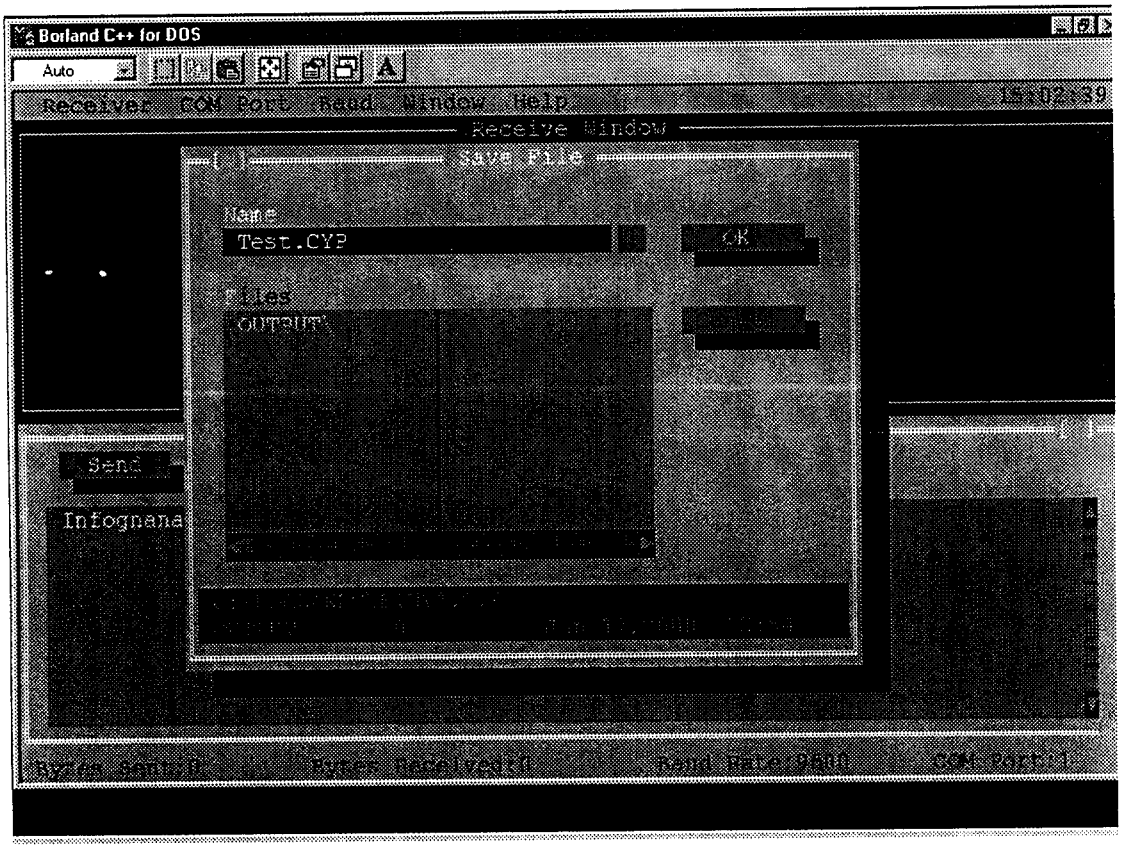
As seen above data in the Transmitt window has been encrypted by Info Crypt and its result viewed in the Receive Window.Save has also been highlited.The contents can be saved into a file.

(viii)



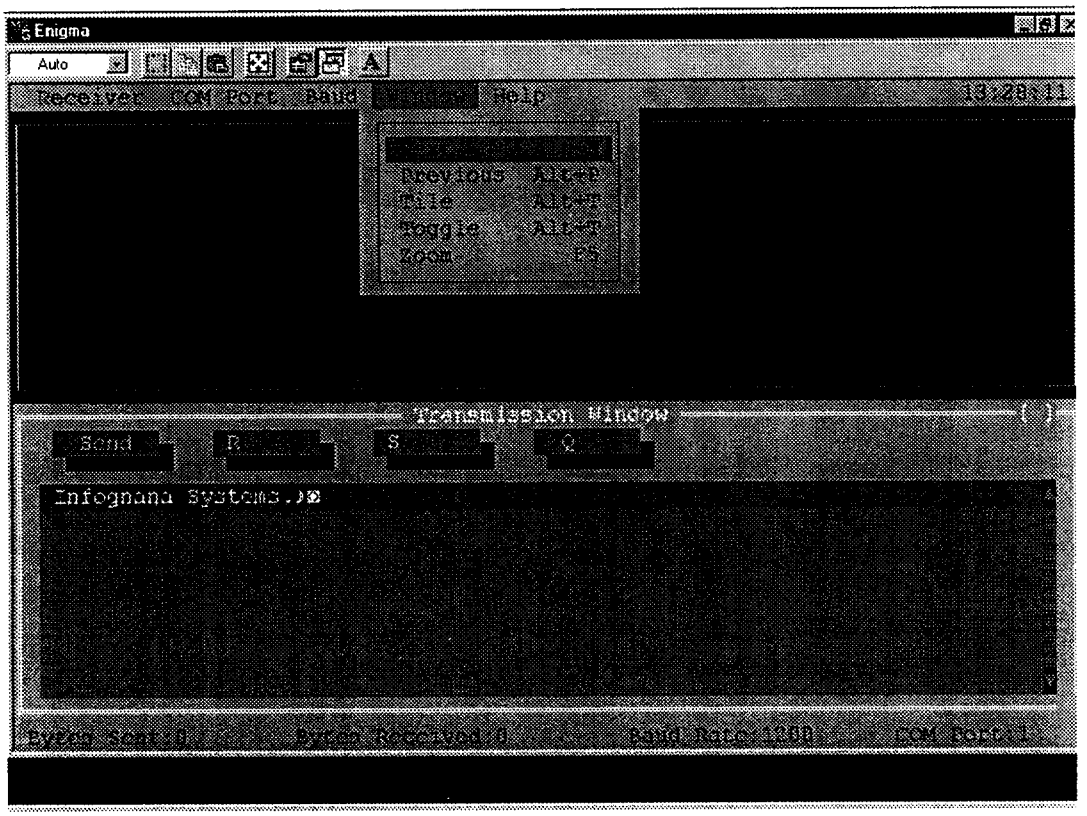
The same holds true for decryption as shown above.

(ix)



When contents are to be saved the Save Dialog Box appears with the default extension .**CYP** for encrypted data and **.DYP** for decrypted data.The receivers window is automatically cleared of contents and the Status Bar values restored to default values.

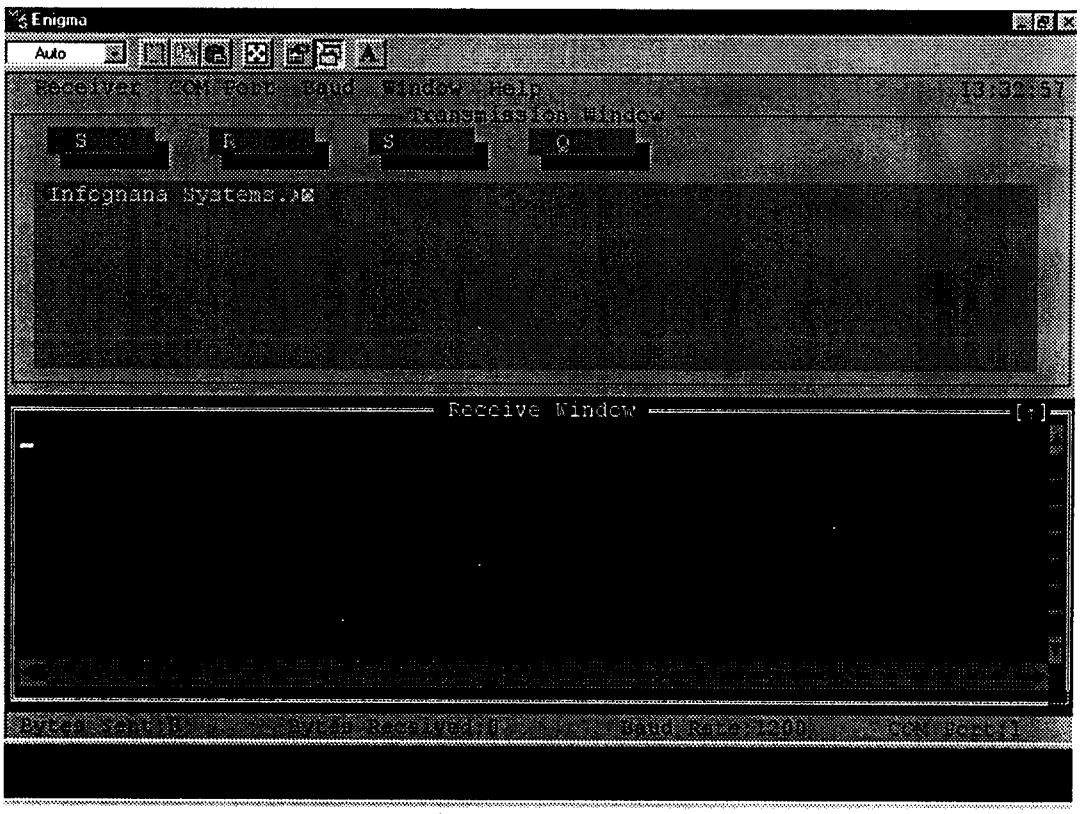
(x)



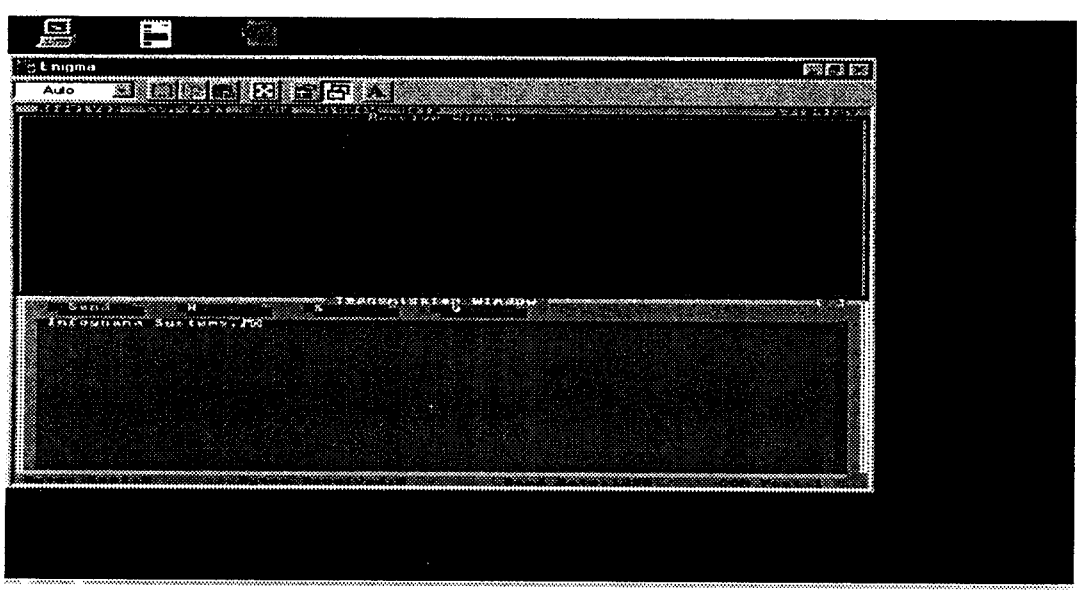
The Window Menu provides a set of window settings commands.

- **Next** and **Previous** will take you to the next or previous window depending upon where the cursors current position is

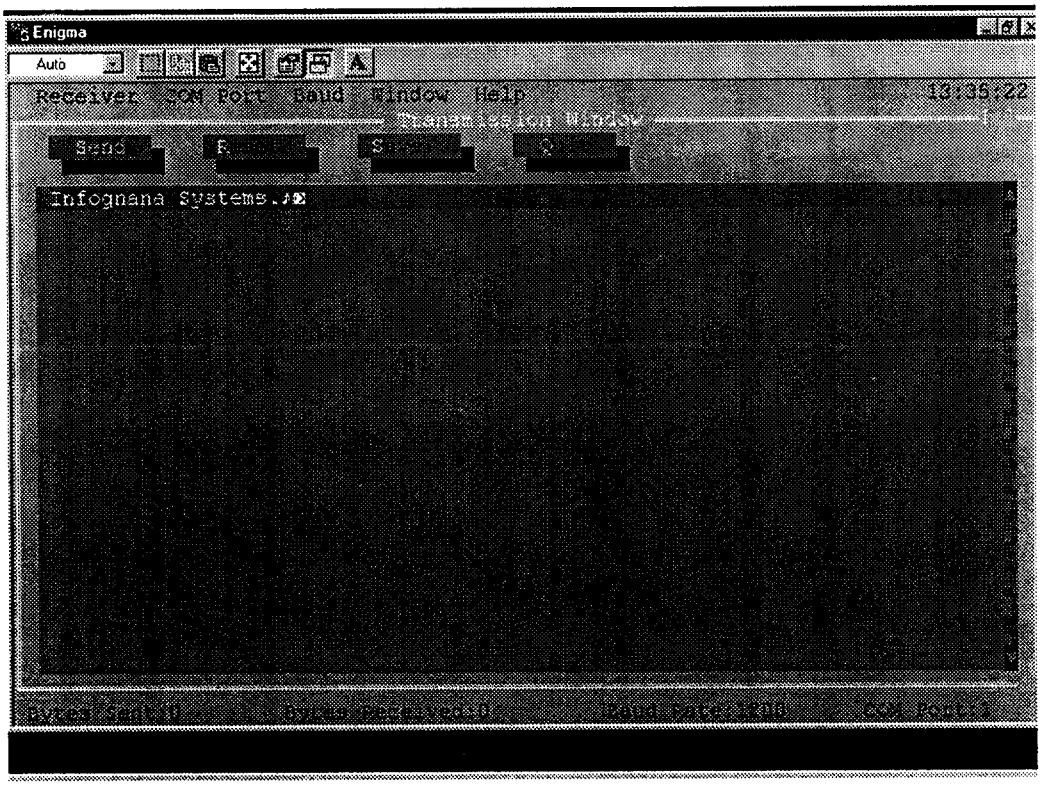
- **Tile** reverses the position of the windows as shown below

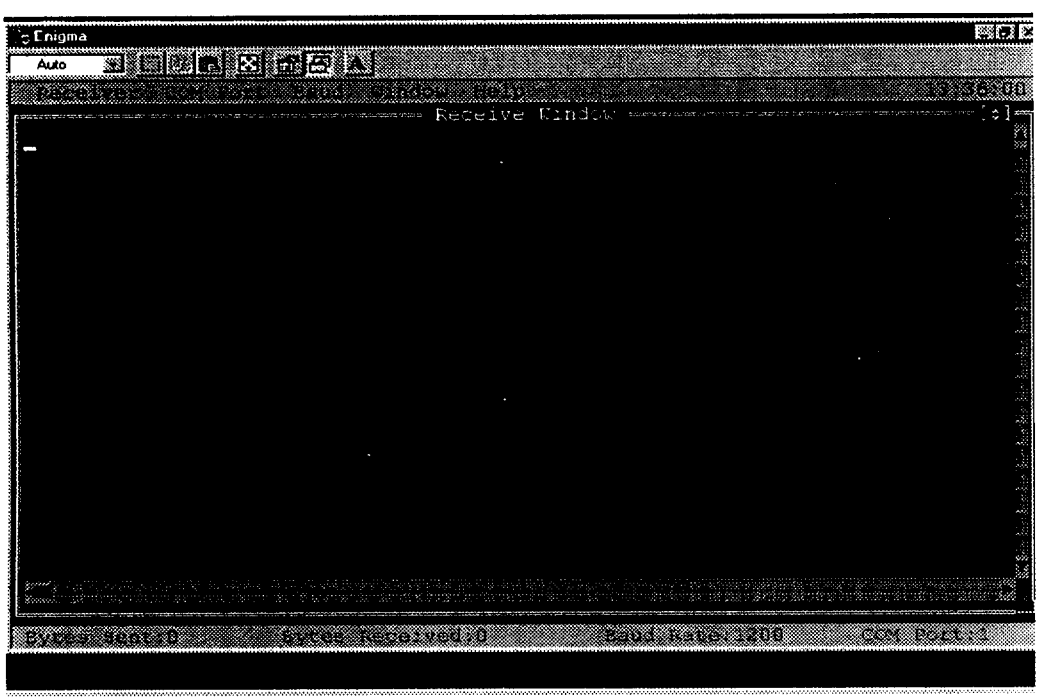


- Toggle will bring the appearance of the font to 8*8 as shown below

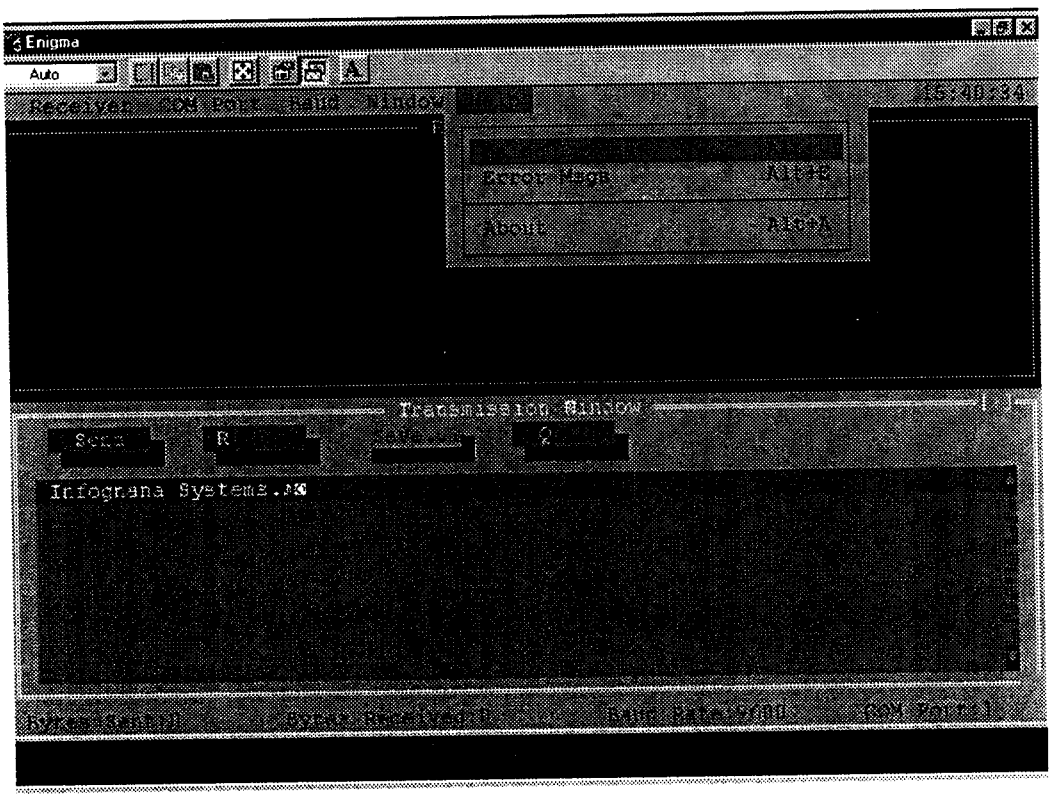


Zoom will zoom the window currently focused on to full screen as shown below

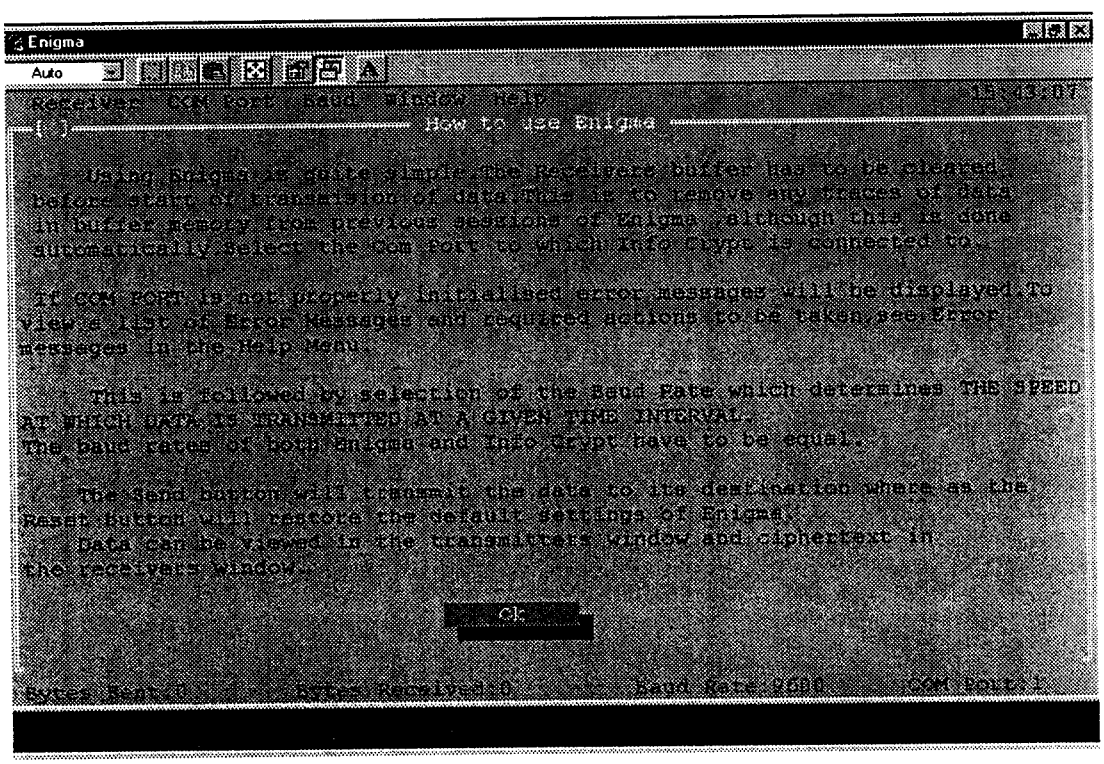




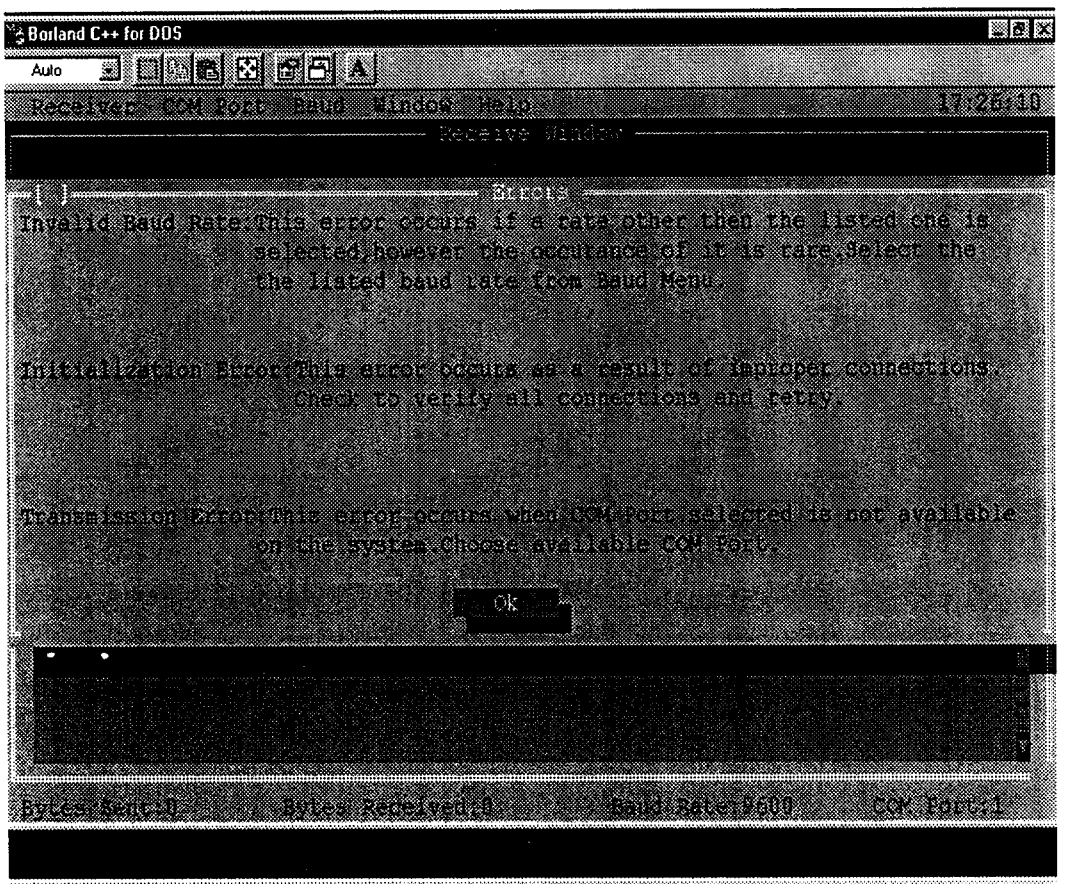
(xi) The Help Menu provides help related to two divisions



- **Using Enigma**:Relates how to use the product in a one page display as shown below

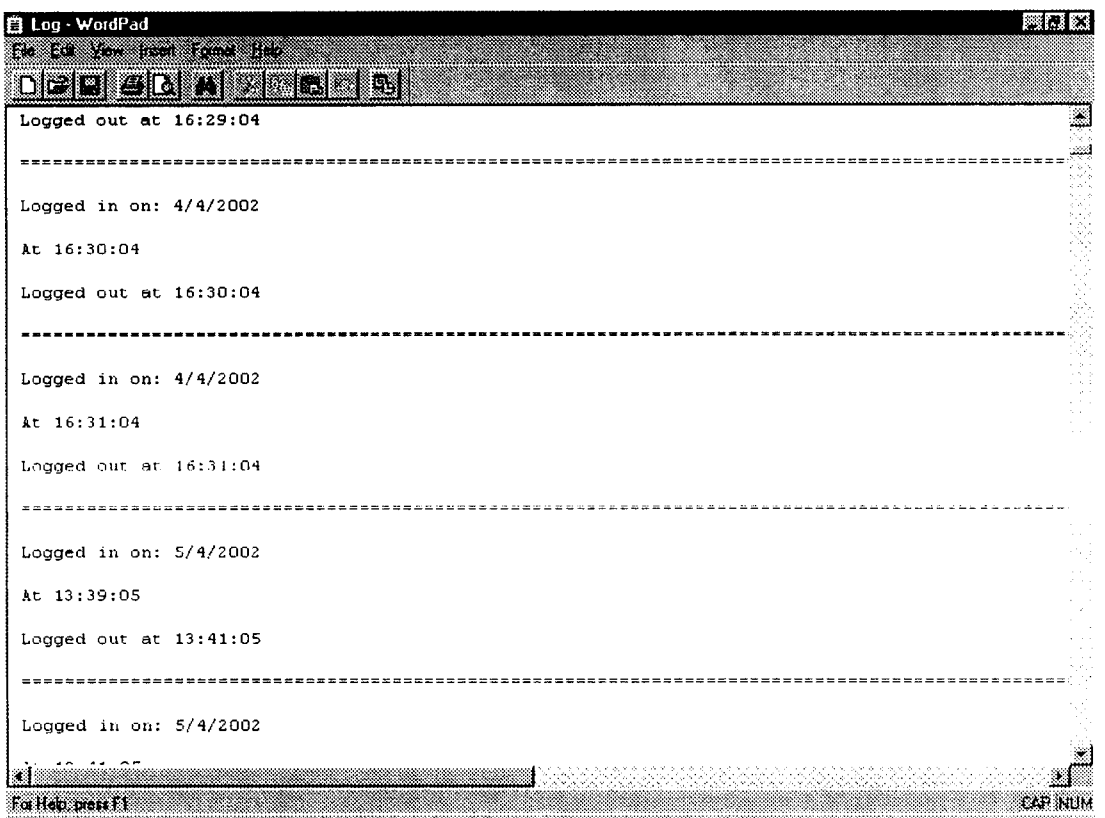


- **Error Messages**:relates to the possible errors likely in Serial Transmission and remedial solution to be followed.

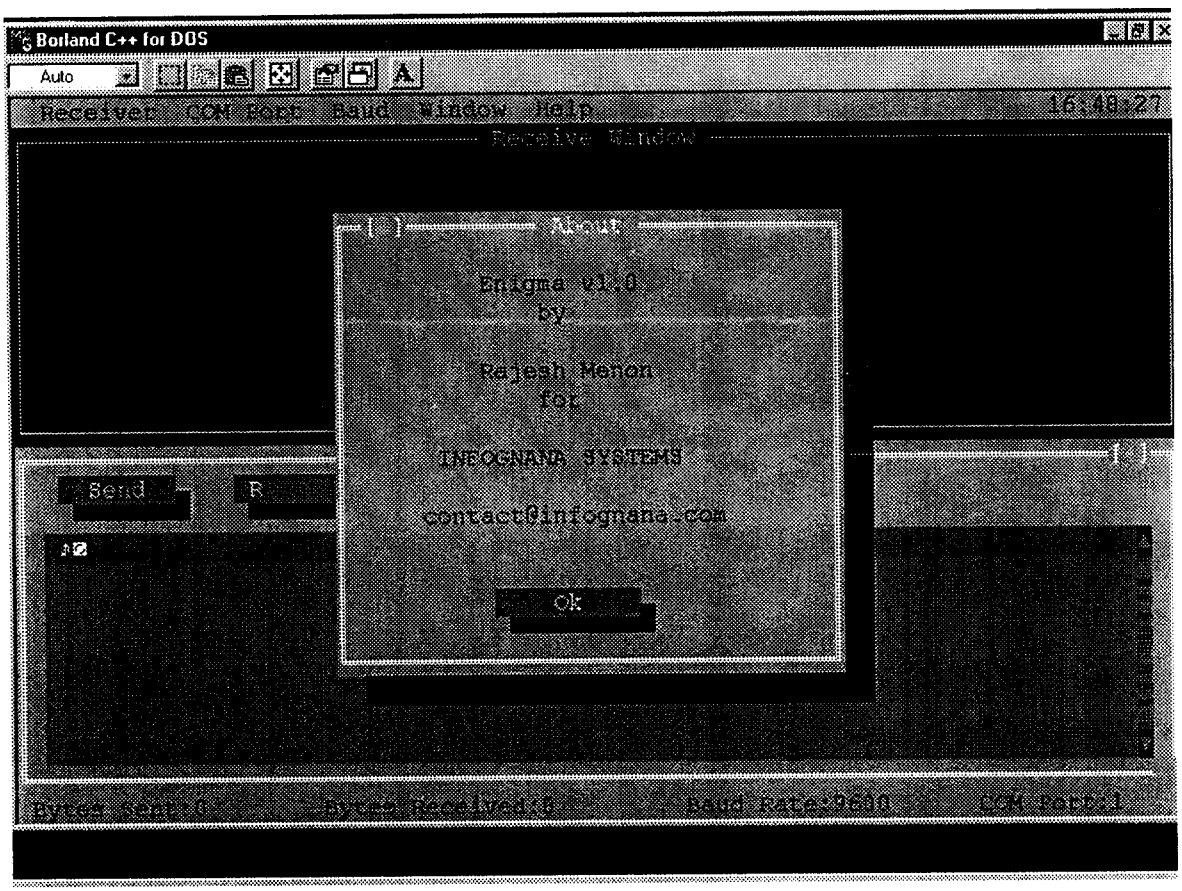


(xii)Enigma creates a log file in the background recording the following events

- Log in Date
- Log in time
- Logged out time.



(xiii)


Borland C++ for DOS
Auto
Receiver COM Port Baud Window Help
16:48:27
Receive Window
About
Enigma v1.0
by
Rajesh Menon
for
INFOGNANA SYSTEMS
contact@infognana.com
Ok
Send
Bytes Sent:0
Bytes Received:0
Baud Rate:9600
COM Port:1

CHAPTER 8

SCOPE FOR FUTURE ENHANCEMENTS

SCOPE FOR FUTURE ENHANCEMENTS

1) The knockout switch provides direct interconnection paths from the switch inputs to the outputs, allowing great simplification of the buffer design and thus achieves a more efficient switch.

2) The packet loss in any network is inevitable whether it is caused by transmission errors or buffer overflow. By allowing the packet switch itself to introduce a small amount of additional packet loss, the concentrator required at each switch output can be reduced from N*N to N*8 for arbitrarily large N. An N*8 concentrator is designed based on knockout matches commonly used in tournaments.

3) A new buffer sharing scheme joins the buffer associated with each of the concentrator output lines to form a simple first-in first-out buffer.

4) The lost packet rate of the knockout switch can be made as small as desired and the latency is the smallest achievable by any switch.

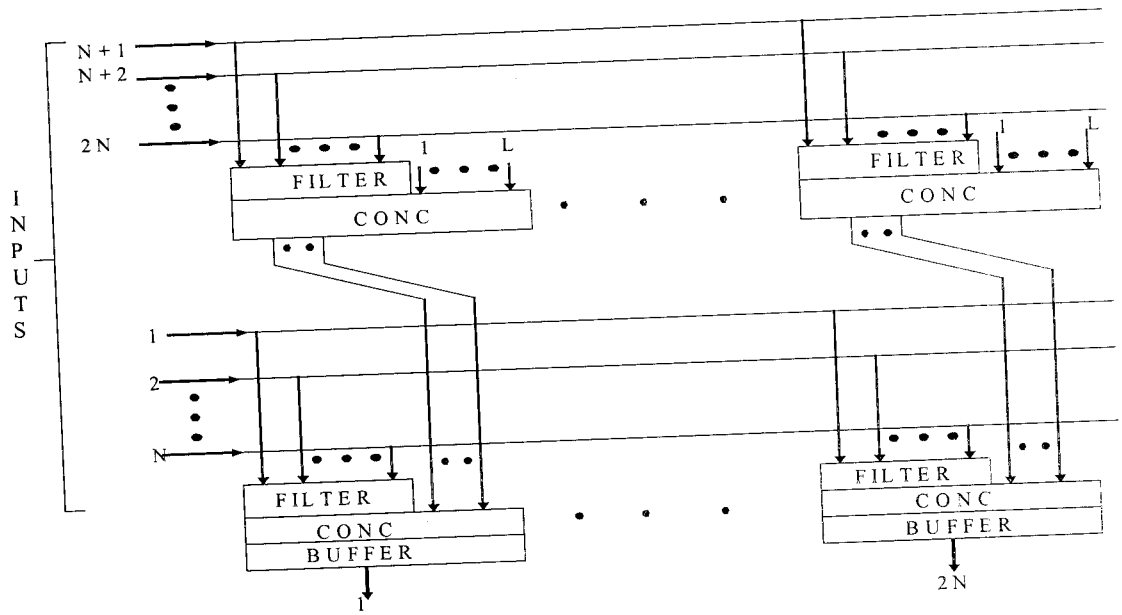
5) The interconnect allows:

- ✓ easy modular growth
- ✓ simple maintenance procedures
- ✓ design that can be made fault tolerant

8.1 MODULARITY

Often in the design of a switch ,the cost and complexity of the switch fabric play a secondary role to other factors such as the ability to grow the switch in a modular fashion, to maintain the switch without great difficulty and to provide the required degree of fault tolerance.

In addition to growing gracefully from 2*2 to N*N , the knockout switch can grow modularly from N*N to JN*JN,J=2,3.....One way to do this illustrated in the fig where we have provided each concentrator in the switch with L additional inputs for a total of N+L inputs and L outputs. The interface for each output in a JN*JN knockout switch consists of J separate N-bus interfaces daisy chained together. Specifically, each of the J interfaces for the one output contains a row of N packet filters and a (N+L)-to- L concentrator, with only the first interface containing the shared buffer structure with shifter and L FIFO buffers. These J individual components for each output are connected together by attaching the L outputs of the concentrator in the j^{th} interface to the L extra inputs on the concentrator in the $j\text{-}1^{st}$ interface. In effect, we have a convenient way of growing the knockout switch using a single (N+L)-to-L concentrator design and the same shared buffer (one for each output) independent of the switch size.



8.2 APPLICATIONS OF KNOCKOUT SWITCH:

Each of N=1000 inputs operating at 50Mbits/s, possible applications for a 50Gbits/s knockout packet switch include interconnects of multiprocessing systems, high speed local and metropolitan area networks, and local or toll switches for integrated traffic loads. With the advent of optical backplanes the integrated optoelectronic devices, the line speed and overall switch capacity could grow much larger.

CHAPTER 9

CONCLUSION

Chapter 9

CONCLUSION

The project "KNOCKOUT SWITCH - A Simple Modular Architecture for Packet Switching" has been successfully implemented by simulating the concentrator and buffer design for the switch by using the simulation package simjava.

A description of the various packages imported along with the interface index, class index, constructors and methods used in the simulation are stated in the appendix.

This project mainly demonstrates the routing of packets that arrive at the inputs of a knockout switch. By simulating the method in which the packets arrive at the output of the switch after traveling through the packet filters, concentrator and shared buffer we can conclude that the knockout switch is superior to conventional contention switches in design and functioning.

Through the project we are able to establish the following about the knockout switch architecture:

- It is designed such that it has low latency
- It is self routing and non blocking
- It has a simple interconnection topology that allows for easy modular growth
- Minimal disruption and easy repair of any fault.

CHAPTER 10

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] W.L.Hoberecht , "A layered network protocol for voice and data integration ,"IEEE J. Select. Areas Commn ., vol. SAC-1, pp. 1006-1013, Dec . 1983.

[2] "Design of an integrated services packet network," IEEE J. Select. Areas Commn ., vol. SAC-4, pp. 1373-1380, Nov. 1986.

[3] M.Decina and D.Vlack, Eds., Special Issue on Packet Switched Voice and Data Communication, IEEE J. Select. Areas Commn ., vol. SAC-1, Dec. 1983.

[4] J.S.Turner, "Design of a broadcast packet network," in Proc.IEEE INFOCOM '86, pp.667 to 675 , Apr.1986 .

[5] http://www.dcs.ed.ac.uk/home/hase/simjava

APPENDIX

Chapter 11

APPENDIX

package eduni.simjava

INTERFACE INDEX

- Sim_output

CLASS INDEX

- Sim_entity
- Sim_event
- Sim_port
- Sim_system

Class eduni.simjava.Sim_entity

java.lang.Object

|

+----java.lang.Thread

|

+----eduni.simjava.Sim_entity

CONSTRUCTORS

Sim_entity(String)

The standard constructor.

Sim_entity(String, String, int, int)

The constructor for use with the eduni.simanim animation package.

METHODS

add_param(Anim_param)

Add a parameter to this entity.

add_port(Sim_port)

Add a port to this entity.

body()

The method, which defines the behavior of the entity.

get_id()

Get the unique id number assigned to this entity

get_name()

Get the name of this entity

get_port(Sim_event)

Search through this entity's ports, for the one which sent this event.

get_port(String)

Search through this entity's ports, for one called name.

sim_hold(double)

Causes the entity to hold for delay units of simulation time.

sim_putback(Sim_event)

Put an event back on the defered queue.

sim_schedule(int, double, int)

Send an event to another entity, by id number and with no data.

sim_schedule(int, double, int, Object)

Send an event to another entity, by id number with data.

- sim_schedule(Sim_port, double, int)

Send an event to another entity, by a port reference with no data.

- sim_schedule(Sim_port, double, int, Object)

Send an event to another entity, by a port reference with data.

- sim_schedule(String, double, int)

Send an event to another entity, by a port name with no data.

- sim_schedule(String, double, int, Object)

Send an event to another entity, by a port name with data.

- sim_wait(Sim_event)

Hold until the entity recieves an event.

- sim_waiting()

Count how many events are waiting of this entity on the deferred queue .

Class eduni.simjava.Sim_event

java.lang.Object

|

+----eduni.simjava.Sim_event

public class Sim_event

extends Object

This class represents events which are passed between the entities in the simulation

CONSTRUCTORS

- Sim_event()

  Contructor, create a blank event.

METHODS

- event_time()

  Get the simulation time that this event was scheduled.

- from_port(Sim_port)

  Determine if the event was sent from a given port.

- get_data()

  Get the data passed in this event.

- get_dest()

  Get the unique id number of the entity, which received this event.

- get_src()

  Get the unique id number of the entity, which scheduled this event.

- get_tag()

  Get the user-defined tag in this event.

- scheduled_by()

  Get the unique id number of the entity, which scheduled this event.

- set_dest(int)

  Set the destination entity of this event.

- set_src(int)

Set the source entity of this event.

- type()

  Get the user-defined tag in this event

Class eduni.simjava.Sim_port

java.lang.Object

|

+----eduni.simjava.Sim_port

public class Sim_port

extends Object

CONSTRUCTORS

- Sim_port(String)

  Constructor, for stand-alone simulations.

- Sim_port(String, String, int, int)

  Constructor for use with the eduni.simanim package for animations.

METHODS

- get_dest()

  Get the unique id number of the destination entity of this port.

- get_dest_ename()

  Get the name of the destination entity of this port.

- get_pname()

  Get the name of this port.

- get_src()

Get the unique id number of the source entity of this port.

Class eduni.simjava.Sim_system

java.lang.Object

|

+----eduni.simjava.Sim_system

public class Sim_system

extends Object

This is the system class which manages the simulation. All of the members of this class are static, so there is no need to create an instance of this class.

VARIABLES

- SIM_ANY

  A standard predicate that matches any event.

- SIM_NONE

  A standard predicate that does not match any events.

CONSTRUCTORS

- Sim_system()

METHODS

- add(Sim_entity)

  Add a new entity to the simulation.

- clock()

  Get the current simulation time.

- current_ent()

  Find the currently running entity.

- get_entity(int)

  Find an entity by its id number.

- get_entity(String)

  Find an entity by its name.

- get_entity_id(String)

  Find out an entities unique id number from its name.

- get_num_entities()

  Get the current number of entities in the simulation

- initialise()

  Initialise the system, this function does the job of a constructor, and should be called at the start of any simulation program.

- initialise(Sim_anim, Thread)

  This version of initialise() is used by the standard animation package eduni.simanim

- initialise(Sim_output, Thread)

  This version of initialise() is for experienced users who want to use the trace output to draw a graph or an animation as part of the application.

- link_ports(String, String, String, String)

Link the ports the ports on two entities, so that events scheduled from one port are sent to the other.

- run()

  Start the simulation running.

- run_start()

  Start the simulation running.

- run_stop()

  Stop the simulation

- run_tick()

  Run one tick of the simulation.

- sim_clock()

  A different name for Sim_system.clock().

package eduni.simanim

CLASS INDEX

- ☐ Anim_applet
- ☐ Anim_entity
- ☐ Anim_param
- ☐ Anim_port
- ☐ Param_type
- ☐ Sim_anim

Class eduni.simanim.Anim_applet

java.lang.Object

```
  |
  +----java.awt.Component
        |
        +----java.awt.Container
              |
              +----java.awt.Panel
                    |
                    +----java.applet.Applet
                          |
                          +----eduni.simanim.Anim_applet
```

public abstract class Anim_applet

extends Applet

The superclass for all simulation animations. New animations should extend this and provide bodies for the methods anim_layout() and anim_init().

Anim_applet()

METHODS

actionPerformed(ActionEvent)

Updated action handler for 1.1 event model

- adjustmentValueChanged(AdjustmentEvent)

  Scroll bar event handler

- anim_completed()

  This method can be overidden in the subclass, and used to display results.

- anim_init()

  This method can be overidden in the subclass, and used to setup any GUI objects to be used for input to the animation.

- anim_layout()

  This method must be provided in the subclass, it should setup all the simulation entities, and link their ports.

- anim_relayout()

  Reinitialises all the animation entities, by calling anim_layout(), then redraws them.

- init()

  This method should not be overidden in the subclass, use anim_init() instead.

- run()

  This method should not be overidden in the subclass

- start()

  This method should not be overidden in the subclass

- stop()

  This method should not be overidden in the subclass

Class eduni.simanim.Anim_entity

java.lang.Object

|

+----eduni.simanim.Anim_entity

public class Anim_entity

extends Object

An animation entity, used to store display information about an entity for the animation.

CONSTRUCTORS

- Anim_entity(String, String)

  This constuctor should not be used directly, use the extended constructor in eduni.simjava.Sim_entity instead.

METHODS

- add_param(Anim_param)

  This method should not be used directly.

- add_port(Anim_port)

  This method should not be used directly.

- is_invisible()
- set_invisible(boolean)

•set_position(int, int)

This method should not be used directly.

Class eduni.simanim.Anim_param

java.lang.Object

|

+----eduni.simanim.Anim_param

public class Anim_param

extends Object

An animation parameter, parameters are used to display data relating to its parent entity during an animation.

VARIABLES

•enum_type

•HIDDEN

Display type: Don't display.

•NAME

Display type: Show parameter's name only.

•NAME_VALUE

Display type: Show both the parameter's name and its value.

•STATE

Display type: Show as the entity's bitmap.

- VALUE

  Display type: Show value only.

CONSTRUCTORS

- Anim_param(String, int, Param_type)

  Constructor, with co-ordinates defaulting to (0, 0).

- Anim_param(String, int, Param_type, int, int)

  Constructor, with Param_type object and (x, y) co-ordinate.

- Anim_param(String, int, String)

  Constructor, with co-ordinates defaulting to (0, 0).

- Anim_param(String, int, String, int, int)

  Constructor, with (x, y) co-ordinate.

METHODS

- get_intval()

  Get integer value of parameter.

- get_value()

  Get string value of parameter.

- set_value(int)

  Set value of parameter to integer.

- set_value(String)

  Set value of parameter to string.

Class eduni.simanim.Anim_port

java.lang.Object

|

+----eduni.simanim.Anim_port

public class Anim_port

extends Object

An animation port, used to store display information about the port for the animation

VARIABLES

- BOTTOM

  Attatch the port to the bottom side of the parent entity

- enum side

- LEFT

  Attatch the port to the left side of the parent entity

- RIGHT

  Attatch the port to the right side of the parent entity

- TOP

  Attatch the port to the top side of the parent entity

CONSTRUCTORS

- Anim_port(String, String)

  This constuctor should not be used directly, use the extended constructor in eduni.simjava.Sim_port instead.

METHODS

- draw_messages(Graphics)
- set_position(int, int)

This method should not be used directly.

Class eduni.simanim.Param_type

java.lang.Object

|

+----eduni.simanim.Param_type

public class Param_type

extends Object

An animation parameter type. Stores an array of strings representing the different states of a parameter, e.g. "Idle", "Busy" etc.

You need to specify a parameter type in order to get a parameter to display on a timing diagram.

Example code to add a parameter type from a user's entity constructor is below.

```
String[] wstate = {"idle","busy"};
add_param(new Anim_param( "State",
                  Anim_param.STATE,
                   new Param_type("wstate", wstate)
                   ));
```

+----eduni.simanim.Sim_anim

public class Sim_anim

extends Panel

Do not use this class directly, instead use Anim_applet.

CONSTRUCTORS

- Sim_anim(Anim_applet)

  Do not use this method directly

METHODS

- add_entity(Anim_entity)

  Do not use this method directly

- animate(Thread)

  Do not use this method directly

  Do not use this method directly

- draw_all_static()

  Do not use this method directly

- initialise()

  Do not use this method directly

- link_ports(String, String, String, String)

  Do not use this method directly

- paint(Graphics)

  Do not use this method directly

- println(String)

  Do not use this method directly

- update(Graphics)

  Do not use this method directly