DEVELOPMENT OF SECURED REMOTE DISPLAY USING DIGITAL CONTROLLER

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR AWARD OF THE DEGREE OF

MASTER OF ENGINEERING

IN

CONTROL & INSTRUMENTATION

SUBMITTED BY:

NITIN GOEL

UNIVERSITY ROLL NO. 8663

COLLEGE ROLL NO. 03/C&I/04

UNDER THE GUIDANCE OF

Dr. VISHAL VERMA

(Asstt. Professor)

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI COLLEGE OF ENGINEERING

DELHI-110042

JUNE-2006

CERTIFICATE

This is to certify that dissertation entitled “Development of Secured Remote Display Using Digital Controller” has been submitted by Nitin Goel as Partial fulfillment for the degree of Master of Engineering in Control & Instrumentation in the Department of Electrical Engineering, Delhi College of Engineering, University of Delhi. This is a record of his own work carried out by him under my supervision and guidance. The matter embodied in this major thesis report has not been submitted for the award of any other degree or diploma course.

Dr. Vishal Verma
Assistant Professor
Department of Electrical Engineering
Delhi College of Engineering
Delhi-110042

ACKNOWLEDGEMENT

I would like to take this opportunity to express my thanks to Prof. Pramod Kr. Gupta Head of the Department and Mr.Vishal Verma my supervisor for their guidance and encouragement throughout the realization of this project. I am indebted to them for the provoking discussions we had during the course of my project. I am also thankful to them for their constructive criticism and appraisal of the work.

I offer my special thanks to my friends Aditya, Deepak and Rohan, for their co-operation throughout the project work by coming to project lab and providing moral support. Finally, I like to acknowledge the help rendered by Mr. Karan Singh, Sr. Mechanic control system lab for the help rendered to this thesis work.

I was greatly encouraged by my guide who gave both moral and technical support. I am really grateful to him for all his guidance and support.

At last but not the least I would like to express my sincere thanks to all those who contributed directly and indirectly to completion of my project.

Thanks

30th June 2006

Nitin Goel

Roll No. 8663

ABSTRACT

This project deals with the development of a system which can display data or information at remote locations with security, and flexibility of sequencing and timing control over data to be displayed. There are few techniques available in the market for such display of system but those methods are non-flexible and non-programmable as they are not able to select data or information to be displayed and their time duration of display at display unit through programmable logic.

To establish the proposed data transfer module the digital data from source (telephone lines etc.) is acquired and collected in different memory locations through microcontroller interface through control of modern signals through suitable programming it is been made available for display at remote locations. A two way communication has been provided making it suitable for variety of applications requiring two way data flow.

The control and interface built around 8 bit microcontroller which stores the data at desired memory locations and with proper programming, sequence the display that data to display unit with control over its sequencing and timing.

This thesis present design and development of new secured programmable remote display systems based on digital controller. The entire work presented in the thesis is broadly discussed in two parts, hardware interface circuits and software development.

The sections on hardware interface system discuss the design procedure of the developed circuits, interface standards, logic and design compatibility.

The software section which deals with software design discuss various modules developed through assembled language. The programs are firstly assembled with on chip simulator and then transferred to kit to operate on hardware interface where various modules of system are finally interfaced to form a complete system.

The prototype developed has secured procurement of data at remote location so that the system can only be accessed by approved user only. The secured data acquired from remote location also embodies the signature for desired instant of time and duration at display unit.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC - Analog to Digital Converter

ALE - Address Latch Enable

CS - Chip Select

EIA - Electronic Industries Association

IDE - Integrated Development Environment

LCD - Liquid Crystal Display

RAM - Random Access Memory

REN - Receive Enable

RI - Receive Interrupt

ROM- Read Only Memory

RS- Register Select

SBUF - Serial Buffer

SCON - Serial Control

SFR - Special Function Register

TI - Transmit Interrupt

UART - Universal Asynchronous Receiver/ Transmitter

INTRODUCTION

1.1 GENERAL

We know that in present era mobile computing and wireless technology have become essential productivity tools and they are bringing convenience and drastic changes to our life. There are many situations which require collection of digital data from one remote location for display at site some examples are display of information over specified interval of time under difficult environmental conditions at remote locations etc.

This work presented in this thesis aims to develop a multipurpose system to gather either analog or digital data from one specified location with transfer protected by password, store this data to external memory connected with microcontroller, controls the display sequence of digital data and their duration of display on LCD or projectors with interface circuit connected to display system.

Application to such system may include the information regarding departure and arrival of various trains displayed at screens for passengers at different locations which may be away from actual platform.

The other application may include information or instruction made available to workers at workstation of remotely located industrial unit from its registered office in city. System should be compact and small with cost considerations taken into effect at the same time.

The system has flexibility of operations that is it is able to select the locations at which data collected from remote locations can be stored and this data stored at different locations can be displayed at display unit with proper sequencing control over stored data as well as the duration of time for which this data can be displayed.

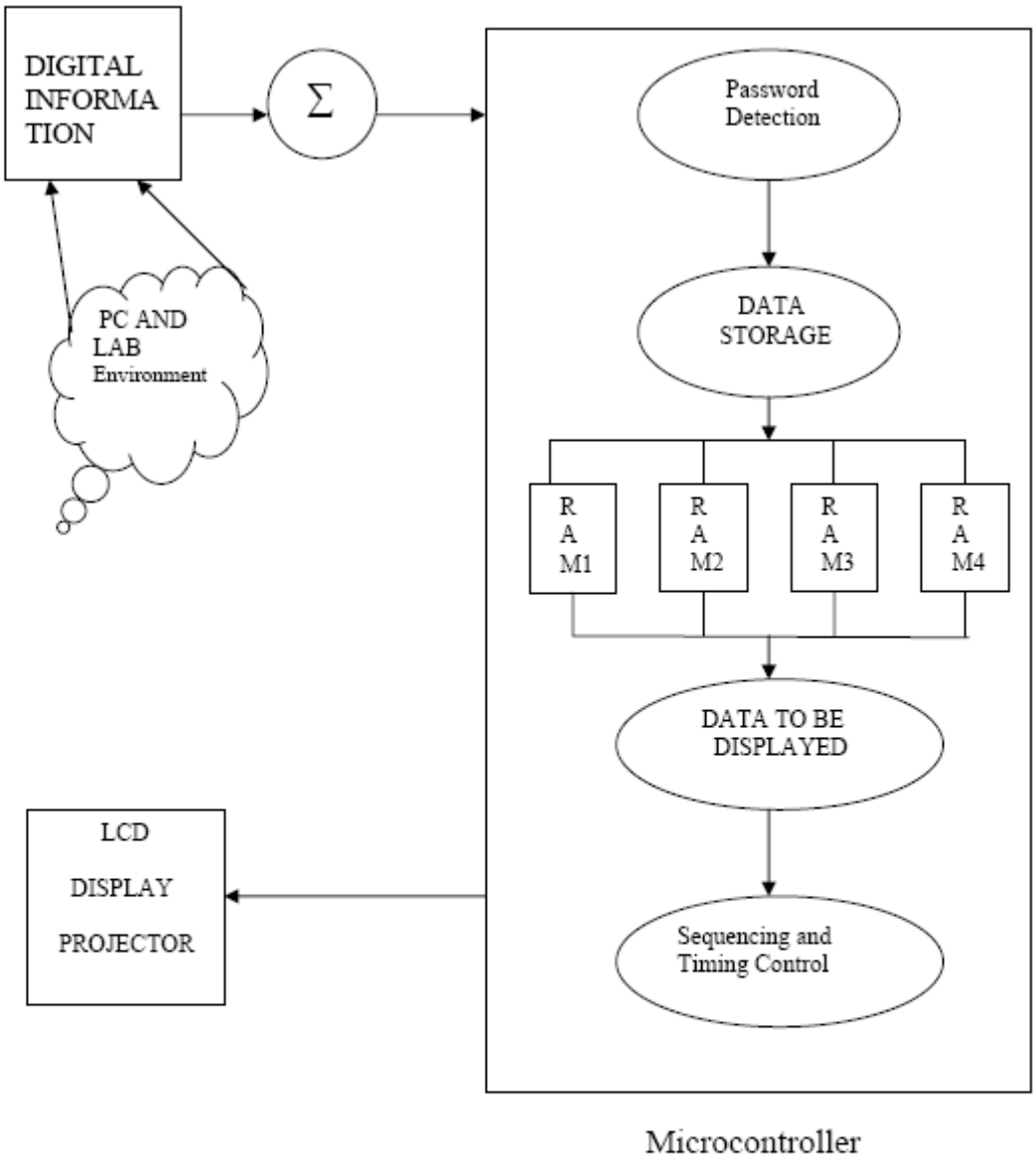


FIG 1.1 Block Diagram for General Overview

1.2 PROBLEM FORMULATION

The project is aimed to develop a flexible/programmable secured prototype that can control and communicate from a distance with control over sequence timing, duration of data to be displayed.

The proposed project is an attempt to provide solution to remotely located difficult to react display system flexibility and programmability as per the external conditions.

To achieve the proposed information system following subsystems are required:

- Data Collection
- Data compression and transmission
- Data Storage
- Data Presentation

The subsystems are integrated together using hardware and software configurations with proper testing and verification to provide a solution which can perform desired task.

1.3 SCOPE OF THE WORK

In light of aforesaid shortcomings there is a need to develop such display system where digital data or information can be programmed remotely with security at various levels secured procurement are endorsed so that only the desired user can work on the system and other user cannot access the system.

The system has flexibility of operations that is it is able to select the locations at which data collected from remote locations can be stored and this data stored at different locations can be displayed at display unit with proper sequencing control over stored data as well as the duration of time for which this data can be displayed.

The proposed system has to be compact and modular in structure so that system may be generalized, portable and at same time cost considerations are taken into count. Thus the problems which are being faced with earlier developed systems of these kinds can be easily overcome with advancement in technology.

1.4 PLAN OF IMPLEMENTATION

The prototype should be able to collect the data properly and at the display end it should be properly presentable or displayed. The digital or analog data from remote location is collected with the help of modem and analog to digital converters through serial communication interface to externally collected storage locations with 8051 microcontroller and with assembly programs the microcontroller is able to control the sequencing of data and duration of display at display unit.

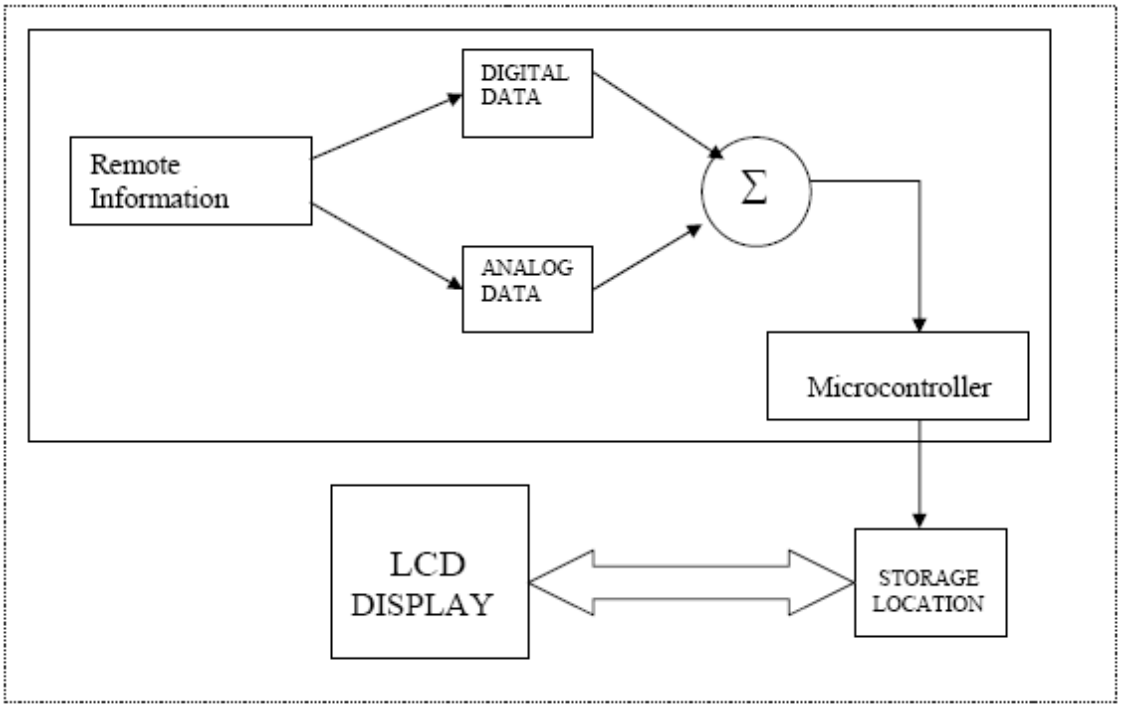


FIG 1.2 Block Diagram for Plan of Implementation

The work discussed in this thesis has been divided into different phases of development in first phase study of different hardware components and their interfacing so as to derive proper performance with software program developed in second phase assembled and executed on microcontroller to perform desired task of system. Finally integrated system with hardwire circuits to assemble the developed programs on microcontroller interface with hardware circuits to perform the sequencing and timing control of data under secured environment.

At input end following activities has to be performed

- Data Collection from remote locations
- Data Conversion and Compression (if required)
- Feeding this digital data through microcontroller for storage
- Command for storage at specific locations
- Check Password for security reasons and allow the user to develop data

At the output end following activities has to be performed:

- Data Display at Display unit
- Proper sequencing of program
- Data De-Compression
- Proper timing control

These two group of activities are integrated and tested and verified through proper channel so that the results are obtained with little troubleshooting and making small changes as required in desired programs.

1.5 ORGANISATION OF THESIS

The work of thesis has been divided into Five Chapters to clearly demonstrate the design and development of whole prototype where First Chapter briefly describes the general overview of project problem formulation its scope and its plan of implementation.

In Second Chapter the configuration of the system its development the operation principle and hardware components description used in this project is presented in great detail with subsequently sections describing data conversion, display system interface with description of memory modules and other necessary details regarding development of project.

In Third Chapter describes the modules for assembly language programs description of data communication software used sequence of execution of programs and developed assembly programs for integrated system development.

In Fourth Chapter the prototype development of project its verification how the hardware and software portion are tested and combined for successful running of the project and results obtained in the experiment from the set up are summarized with discussions.

In Fifth Chapter the overall conclusion of the project work and how further work can be carried on for further advancement of this project has been described.

In the end references and appendices related to work are given which are helpful in getting interacted with more details of system.

SYSTEM CONFIGURATION AND DEVELOPMENT

2.1 GENERAL

The development of the prototype requires certain hardware components of specifications as per the system configuration these hardware components are connected in proper manner to develop the system. The various aspects of the prototype configuration and its development has been described in subsequent sections in a way to provide insight of design, component selection etc. required by the system for its development and their compatibility issues to provide desired results.

2.2 PRODUCT OVERVIEW

There are varieties of products which are used for transmission of information or data. Some products deals with display and some of them are integrated systems. The problems associated with these products not only include their fixed structures and mode of operation, rigidity, duration of timing control and sequencing of data etc. Different frames of data are stored in the EPROM and data is only sequenced to produce animated or multiple displays. The process is generally repeated for dedicated displays or messages. Such display system possess rigidity of operation and control.

The prototype developed in this work able to overcome the aforesaid shortcomings posed by due to systems in present use. The proposed product will be in great use and shall provide flexibility for wide area of applications with remote access.

Application includes information or instruction regarding arrival and departure of international and domestic flights timing at airport authorities thus preventing inconvenience to the passengers.

The overall product view shows that with developing the prototype a wide range of problems related to secured display can be overcome easily.

2.3 BASIC PRINCIPLE OF OPERATION

The basic principle of operation of the proposed prototype is discussed in this section. The digital data is collected, extracted with the help of MODEMs, placed in desired memory locations with tags and upon its priority, timing, duration is decomposed and sequenced by digital controller for its display at display units which can be LCD panel or projector or VGA card.

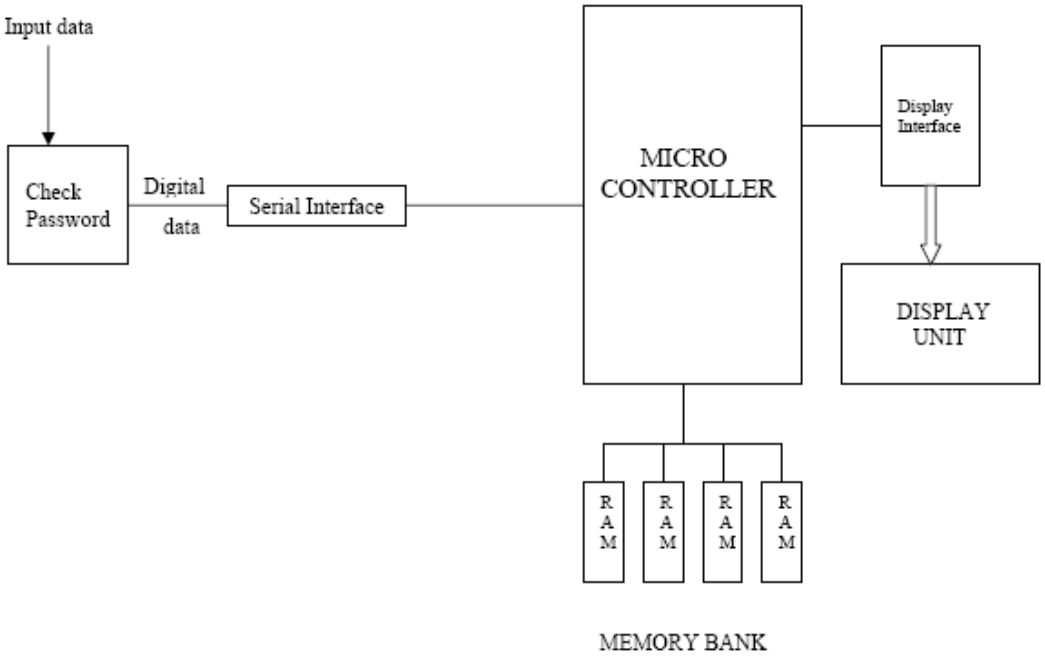


Fig 2.1 Basic Principle of Operation of Proposed system

The input information or data is received as first byte consists of password. The communication is only established when password matches with the stored password. The next set of data after matching of password consists of header information which provide programming information of data to be received.

Fig 2.1 shows the block diagram of basic signal acquisition and output principles. Based on the information in the header received by the microcontroller; it selects the input digital data. The digital data is collected in digital form through serial interfacing and is stored in appropriate place in memory device, the starting address of which is stored as a tag. The frames of data are stored in the form of digital data after proper compression.

The tags of the memory provide information about the starting address of display data stored at a particular memory. Based on priority, duration of display, time of display received through header received by the microcontroller the sequencing and timing control is provided.

Depending on the determined sequence the data is assembled in the buffer memory after decompression (if required). This display buffer provides continuous feed to display interface to provide requisite display.

Upon reaching the end of data the display process is repeated in accordance with duration of time assigned to a particular message or display. The microcontroller on expiry of time loads next data on display buffer so that feed to the display interface may be regular. The display buffer operates as circular buffer.

Thus the basic operation of storage of digital data and desired control to display data is achieved in this prototype whose hardware and software configurations are being described in subsequent section of this chapter.

2.4 DEVELOPMENT OF HARDWARE

This section deals with the basic hardware components description and their interface connections required to develop the prototype. The whole section is divided into three subsections receiving data or information, storing digital data in memory banks, displaying the data at display unit. Thus hardware components required are:

a) Modem Based Interference

b) Memory Bank and Decoders

c) Display Interference

2.4.1 Modem Based Interference

The first step is the detection of password by the controller; on its successful match, the next digital data is sent to the controller for information about their programmability and control of digital data for display. The communication is established through modem. The data is assembled through modem interface and UART (Universal Asynchronous Transmitter and Receiver). Data conversion and compression (if required) is performed in this subsection Figure 2.2 shows the block diagram of modem based interface for establishing communication from remote location, conversion of serial data to parallel, serial interface with microcontroller etc.

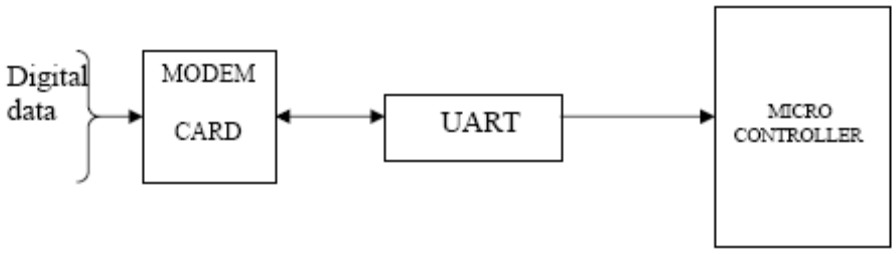
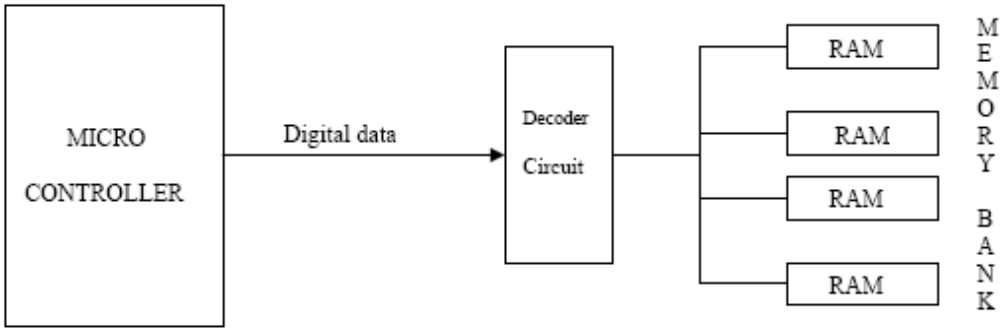


FIG 2.2 Block Diagram for Modem Interface

The Figure 2.2 above shows that digital data received through the modem interface and UART. Thereafter, data is assembled in parallel form and then compress data transferred to microcontroller.

2.4.2 Memory Bank and Decoder

The storage interface deals with storage of data collected from modem interface through microcontroller. The data is collected in memory devices with starting address is stored as a tag. The data is to be stored at specific memory location of the memory bank, the memory bank section is determined by decoder circuit.



The Figure 2.3 shows, data handling after reception from modem interface, and its storage into the memory bank with the help of decoding circuit. The data is collected in specified location and is controlled for its sequencing and duration of display at display unit. The detailed description of memory bank is given in subsequent sections.

FIG 2.3 Block Diagram for Memory Decoder

Digital data

2.4.3 DISPLAY INTERFACE

This section deals with the display interface, which enables the display of data stored in memory banks with the help of the controller. Display interface utilizes a memory buffer which acts as a cyclic buffer for temporary storage of the digital data for subsequent display on display unit continuously until next data is available in the buffer for display. The next data in loaded after the timer interrupt.

The Fig 2.4 shows the connections of integrated system with display interface connected to the controller. Data collection, storage, and display are performed in a sequential way to have control over duration and sequence control through controller.

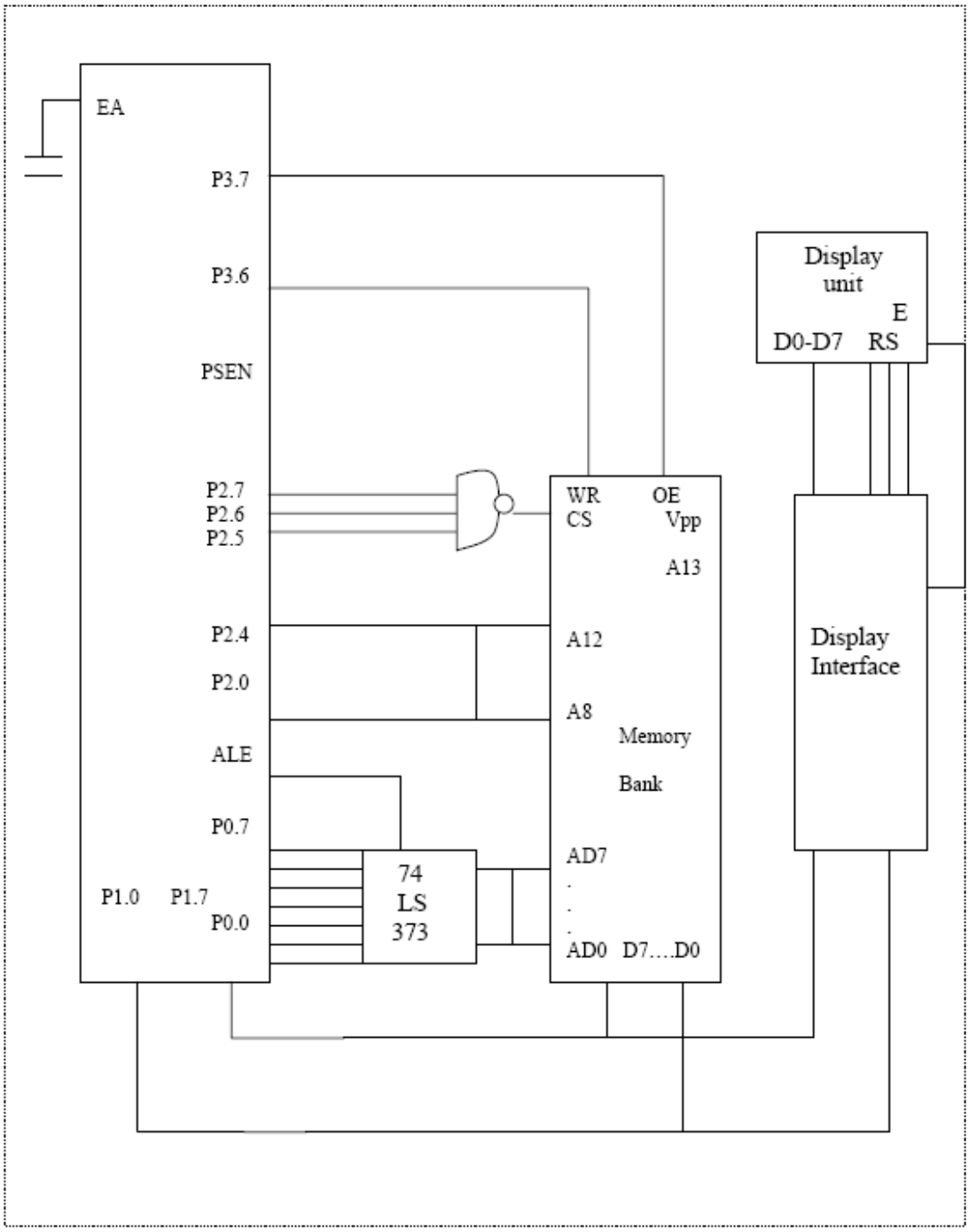


FIG 2.4 CONNECTION DIAGRAM OF COMPLETE INTEGRATED INTERFACING

2.5 MICROCONTROLLER AND INTERFACE HARDWARE

2.5.1 8051 Microcontroller

The 8051 is an 8-bit microprocessor originally designed in the 1980's by Intel that has gained great popularity since its introduction. Its standard form includes several standard on-chip peripherals, including timers, counters, and UART's, plus 4kbytes of on-chip program memory and 128 bytes (note: bytes, not Kbytes) of data memory, making single-chip implementations possible. Its hundreds of derivatives, manufactured by several different companies (like Philips) include even more on-chip peripherals, such as analog-digital converters, pulse-width modulators, I2C bus interfaces, etc. Costing only a few dollars per IC, the 8051 is estimated to be used in a large percentage all embedded system products.

The 8051 memory architecture includes 128 bytes of data memory that are accessible directly by its instructions. A 32-byte segment of this 128 byte memory block is bit addressable by a subset of the 8051 instructions, namely the bit-instructions. External memory of up to 64 Kbytes is accessible by a special "MOVX" instruction. Up to 4 Kbytes of program instructions can be stored in the internal memory of the 8051, or the 8051 can be configured to use up to 64 Kbytes of external program memory. The majority of the 8051's instructions are executed within 12 clock cycles.

We developed assembly language of the 8051 and a C based 8051 instruction-set simulator in some research directions. One of those directions is a tuning environment, wants to modify the 8051 architecture to be more power efficient for a particular program. You see, a particular 8051 will probably execute the same program over and over for its lifetime, so it would be nice to orient the 8051 towards that program.

The Major Features of 8051 are:

* 8-Bit CPU optimized for control applications.

* Extensive Boolean Processing Capabilities

* 64K Program Memory Address Space

* 64K Data Memory Address Space

* 4K Byte of On Chip Program Memory

* 128 Bytes of on Chip Data RAM

* 32 Bidirectional and individually addressable I/O Lines

* Two 16 bit Timer or Counters

* Full Duplex UART (Universal Asynchronous Receiver/ Transmitter)

* Interrupts with Priority Level

* On chip Oscillator and Clock Circuitry

A Microcontroller has a CPU in addition to fixed amount of RAM, ROM I/O Ports and Timer all on a single chip i.e processor and other functional are embedded together on one chip. The fixed amount of on chip ROM RAM and no. of I/O ports in microcontroller makes them ideal for many applications in which cost and space are critical. In General purpose Microprocessor RAM, ROM and I/O ports are connected externally to make them functional although the addition of these makes the system bulkier and much more expensive they have the advantage of versatility that designer can decide amount of external functionality at its hand. In many applications, the space it takes the power it consumes the price per unit are much more critical than the computing power in these applications microcontroller can be used widely.

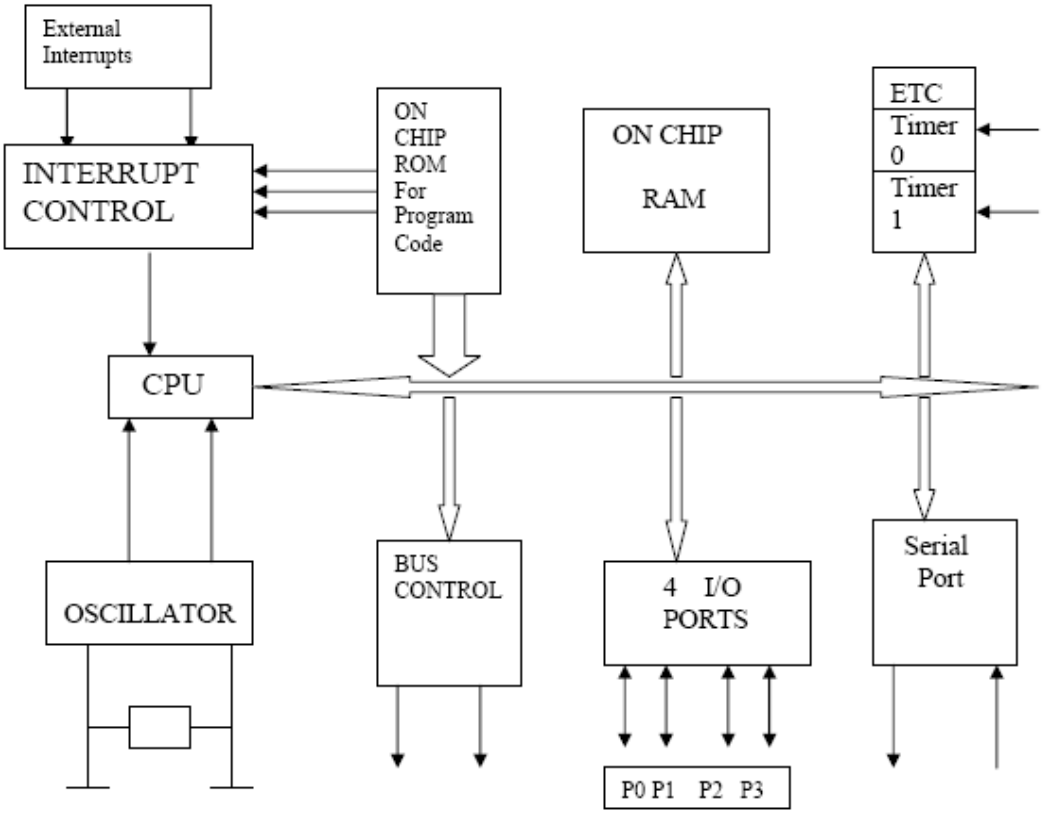


FIG 2.5 INTERNAL ARCHITECTURE OF 8051 IS:

2.5.2 Setting the serial port

The first thing we must do when using 8051's integrated serial port is to configure it. This lets us tell the 8051 how many data bits we want the baud rate we will be using and how baud rates will be determined.

The serial interface is full duplex and data is transmitted and received by SBUF register On writing to SBUF register it loads the transmit register and on reading the SBUF a separate received buffer is accessed the serial transmission is controlled by Special Function Register SCON, definition of what each bit of SFR represents:

Bit	Name	Bit Address	Function
7	SM0	9FH	Serial port mode bit 0
6	SM1	9EH	Serial port mode bit 1
5	SM2	9DH	Multiprocessor communication enabled
4	REN	9CH	Receiver enable
3	TB8	9BH	Transmit bit 8
2	RB8	9AH	Receive bit 8
1	TI	99H	Transmit Flag
0	RI	98H	Receive Flag

The function of SM0 and SM1 by an additional table:

SM0 SM1 Serial mode Function Baud Rate

0	0	0	8-bit Shift reg	Oscillator/12
0	1	1	8-bit UART	Set By Timer 1*
1	0	2	9-bit UART	Oscillator/32 *
1	1	3	9-bit UART	Set by Timer 1*

* The baud rate indicated in this table is doubled if PCON.7 (SMOD) is set the SCON, SFR allows us to configure the serial port.

The first four bits are configuration bits SM0 and SM1 let us set the serial mode to a value between 0 and 3 inclusive the four modes are defined in the chart immediately above selecting the serial mode selects the mode of operation and also determines how

the baud rate will be calculated. In modes 0 and 2 the baud rate is fixed based on oscillator frequency. In modes 1 and 3 the baud rate is variable based on how often Timer 1 overflows.

The next bit SM2 is a flag for Multiprocessor communication generally whenever a byte has been received the 8051 will set RI Flag. This lets the program know that a byte has been received and that it needs to be processed when SM2 is set the RI flag will only be triggered if the 9^{th} bit received was "1" if SM2 is set and a yte is received whose 9^{th} bit is cleared the RI flag will never be set This can be useful in certain advanced serial applications one will always almost want to clear this bit so that the flag is set on reception of any character.

The next bit REN "Receive Enable" if you want to receive data via serial port set this bit. The last four bits are operational bits they are used when actually sending receiving data they are not used to configure the serial port.

The TB8 is used in modes 2 and 3 in modes 2 and 3 a total of nine data bits are transmitted The first 8 data bits are the Bits of main value and 9^{th} bit is taken from TB8 if it is set and value is written to serial port the data bits will be written to serial line followed by set 9^{th} bit if TB8 is clear the 9^{th} bit will be clear.

The RB8 also operates in the same modes 2 and 3 which function essentially the same as TB8 but on the reception side.

TI means Transmit Interrupt when a program writes a value to serial port a certain amount of time will pass before the individual bits of byte are clocked out and serial port.

RI means Receive Interrupt it function similarly to TI bit but indicates that a a byte has been received thus whenever 8051 has received a complete byte it will trigger RI bit to let the program know that it needs to read the value quickly before another byte is read.

2.5.3 Setting the serial port baud rate

Once the serial port mode has been configured the program must configure the serial port's baud rate this only applies to serial port modes 1 and 3 the baud rate is determined based on oscillator's frequency when in mode 0 and 2. In mode 0 the baud rate is always oscillator frequency divided by 12 i.e. our crystal is 11.059Mhz mode 0 baud rate

921.583 baud in mode 2 the baud rate is always oscillator frequency divided by 64 so baud rate will be 172.797 baud.

In mode 1 and 3 the baud rate is determined by how frequently timer 1 overflows the more frequently timer 1 overflows the higher the baud rate. There are many ways one can cause timer 1 to overflow at a rate that determines a baud rate but the most common method is tyo put timer 1 in 8 bit auto reload mode and set a reload value TH1 that causes timer 1 to over flow at a frequency appropriate to generate a baud rate.

To determine that must be placed in TH1 to generate agiven baud rate we must use the following equation:

TH1= 256-((Crystal/ 384)/ Baud)

To obtain 19200 baud on a 11.059 Mhz crystal we have to set TH 1 to 254.5.

2.5.4 Writing to serial port

Once the serial port has been properly configured the serial port is ready to receive and send data. If you thought that configuring the serial port was simple, using the serial port will be a breeze.

To write a byte to serial port one must simply write the value to the SBUF (99h) SFR for eg. If you wanted to send the letter “A” to the serial port it could be accomplished as:

```
MOV SBUF, #’A’
```

Upon execution of above instruction the 8051 will begin transmitting the character via the serial port as transmission is not instantaneous it takes a measurable amount of time to transmit we need to be sure that a character is completely transmitted before transmitting next character.

CLR TI MOV SBUF,#’A’

19

JNB TI, Target

The above three instruction will transmit a character and wait for TI bit to be set before continuing. The last instruction says “Jump if the TI bit is not set to Target” thus 8051 will pause on JNB instruction until the TI bit is set by 8051 upon successful transmission of the character.

Reading the serial port

To read a byte from serial port one must simply read the value stored in the SBUF (99h) SFR for eg. If you wanted to read the letter “A” from the serial port it could be accomplished as:

MOV A,SBUF

Upon execution of above instruction the 8051 will begin receiving the character via the serial port as reception is not instantaneous it takes a measurable amount of time to receive

JNB RI,Target MOV A,SBUF

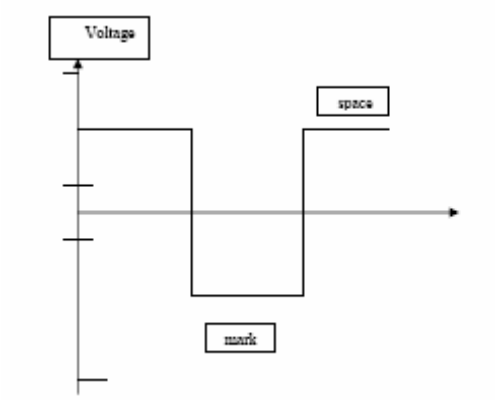
The above instructions will receive a character and wait for RI bit to be set before continuing.

2.6 RS-232 SERIAL COMMUNICATION:

The most common serial interface in use around the world today is an Electronic Industries Association (EIA) standard called RS232-C this standard developed jointly by Bell Labs and the EIA in 1969 is used to connect computers to other computers as well as other laboratory instruments, printers and a host of other peripherals. The RS232-C standard establishes the signal level, timing, pin designation used for interconnection.

The original RS232-C specification set data transfer rate at 19.2 Kbps over distance upto 16m.The standard also specifies the voltages level for binary'0' binary'1' data a signal level b/w 3V and 25V is considered as binary '0' and a signal level b/w -3V and -25V is considered as binary '1' also referred as " Space" and "Mark" respectively. For control or handshake signals a control line is said to be "ON" if signal level is b/w 3V and 25V and "OFF" if it is b/w -3V and -25V. Since the power supplies can only deliver voltages ranging between -12V and +12V these value marks the outer voltage limits for the RS 232-C interfaces in PC's.

This information is summarised as:



RS 232-C SIGNAL STANDARDS

Signal level	Binary data value	Control signal state
3V to 25V	0 (Space)	ON
-25V to -3V	1 (Mark)	OFF
-3V to 3V	Not Defined	Not Defined

The RS232-C interface contains a UART which takes a 5V pulses of parallel data from the transmitting devices bus and converts them to a serial stream of + 12Volt pulses. This stream of data is converted back into parallel data by the RS232-C interface in the receiving device. Every PC has two serial ports and they are referred to as COM1 and COM2.

Serial port comes in two sizes. There are the D type 25 pin connector and the D type 9 pin connector both of which are male on the back of PC thus you will require a female connector on your device pin connection for the 9pin and 25 pin D type connector.

FIG 2.6 VOLTAGE LEVELS IN RS 232

SERIAL PINOUT (D25 and D9 Connectors)D Type 9 Pin and D Type 25 Pin Connectors

D Type-25 pin no.	D Type-9 pin no.	Abbreviation	Full Name
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request to Send
Pin 5	Pin 8	CTS	Clear to Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

2.7 DISPLAY SYSTEM INTERFACE

The prototype development last stage is to display the digital data on the display unit and this final stage is accomplished with display interface connected to the system we can use LCD or Projectors or VGA card as the display unit depending upon the availability and requirement of system.

An LCD is an “intelligent” display in that it contains integrated electronics. The microprocessor need only provide several initialisation bytes to set the display for an operating mode to fit the project. The display has one register into which commands are sent and one register into which data to be displayed are sent. The two register are differentiated by a single control line the interface to LCD consist of eight data lines D0-D7 and control lines Enable (E) Register Select (RS) and Read/Write (R/W). The data

lines are used to transfer both commands and data to be displayed. The LCD will treat the incoming bytes as commands when RS is low and as data to be displayed when RS is high.

The enable pulse on the LCD is a positive going pulse that is synchronous with and same length as active low write or read depending on whether the LCD is being written to or read. It is convenient to display information in system with little or no memory or to be able to read to read the LCD status register for handshaking purposes.

If the display is to be read the R/W line would be connected to the system WR so that the R/W would be high when a read occurs or low for a write LCD's usually require some initialisation commands these are described in the data sheets for particular display used.

LCD Command Codes

Code(Hex)	Command to LCD Instruction
1	Clear display Screen
2	Return Home
4	Decrement Cursor
6	Increment Cursor
5	Shift Display right
7	Shift Display left
8	Display off, Cursor off
A	Display off, Cursor on
C	Display on, Cursor off
E	Display on, Cursor blinking
F	Display on cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to left
1C	Shift the entire display to right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	Two lines and 5*7 matrix

2.8 MEMORY MODULE

The prototype to be developed requires memory source where digital data collected from remote locations through the microcontroller and code program can be stored the data stored in these memory chips or sources is available for display at display unit.

The memory space is the most important part in the development of prototype as memory is required for storage of both the code program and digital data which is to be displayed. The code program can be stored in internal memory of the controller or externally connected ROM depending upon the configuration and memory space required for the task the internal memory organisation of 8051 is shown below:

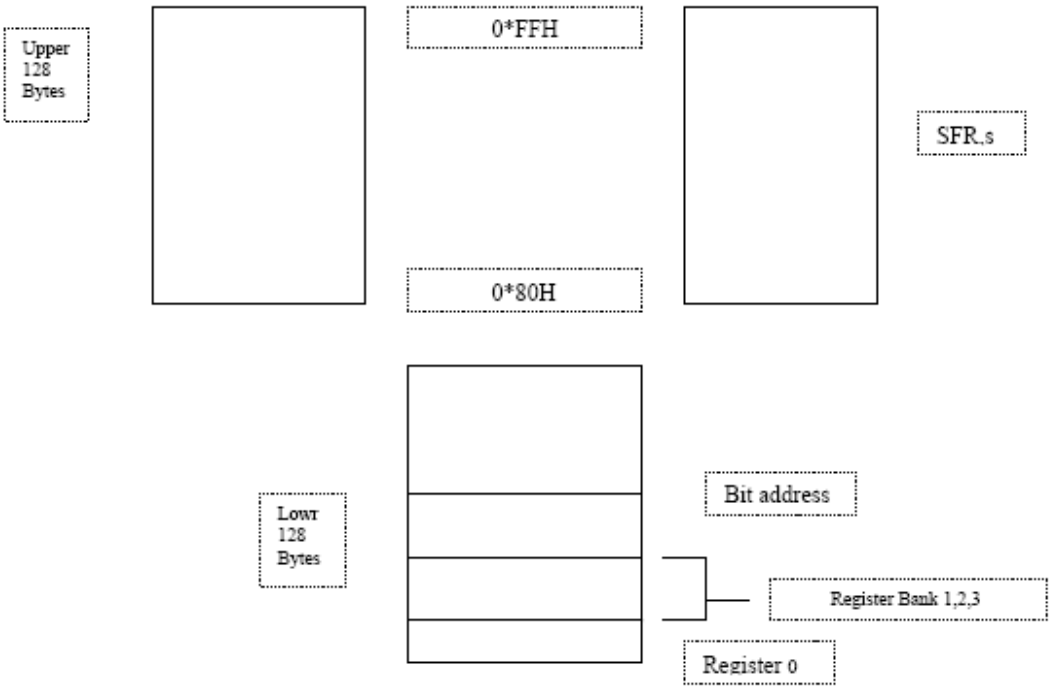


FIG 2.7 Internal Memory Organizations

The digital data or information is stored in external RAM which is connected externally to the controller through latches which provides proper sequencing for accessing of both of address and data on the bus cycles through ALE pulse generated by the latch. The data stored at different memory locations on external RAM is available for microcontroller to move to display unit where the programming logic of controller controls which location data is to be transferred on the buses for display to provide the sequencing control by the prototype.

The external RAM connected to the system can be Static or Dynamic RAM depending on the cost and configuration required by the system the list of various RAM's are attached in the appendix section.

The memory module thus forms the most required hardware part for development of system without which the development of desired prototype will not be completed.

SYSTEM SOFTWARE DEVELOPMENT

3.1 GENERAL

This section deals with building of assembly language programs for performing the desired operation through the controller. The sequence in which execution of programs on the controller should be done and finally transferring this programs to controller.

3.2 MODULES OF ASSEMBLY LANGUAGE

The whole development work of software programs has been divided into small modules where each module has its own specific task to perform. There are modules for development of each task the whole development is divided in sequential manner from decoding and matching the password detection to the final display of data at display unit.

A) Module for password detection

The aim of this module is to have a first level of secured control so that system cannot be accessed by other user. The task is implemented with the help of password detection and matching. If it does not match the stored password it will ignore the request, and if it is correct the microcontroller allows stream of data to be stored onto memory devices in the memory bank.

B) Storing digital data onto memory units

The module is used to store digital data or information from remote input point to desired memory location through serial interfacing which is done with proper checking of transmit and receive flag status of controller. The data collected is assembled in serial to parallel conversion sections, in external memory devices is stored.

C) Display data to LCD Checking Busy Flag

This module transfers the data stored in the memory devices to display unit using buffer for its onward transmission to display units for display. Based on the received

information the timer is programmed to device the duration of display, which is controlled by interrupt. The sequencing control of this data can be achieved with small sub modules developed with the help of memory tags and clock information.

3.3 SEQUENCE OF EXECUTION OF PROGRAMS

A backbone sequencing software program is developed which sequences various developed modules to integrate into system software for program operation of the proposed prototype.

The sequence in which the programs are executed begins with a password detection to provide first level of security purpose for remote access so that only valid user may access the system.

Then it allows the microcontroller to store digital data or information in the external memory banks connected to the system at desired locations according to priority. If the user wishes to flash the message only once then data can be transferred to display unit directly at highest priority with display reset.

The data stored in external memory devices for display is controlled for sequencing and timing and duration by digital controller. The data is substituted in display buffer or circular buffer of the display interface.

The problem of limited parallel ports of microcontroller can be circumvented by expanding the I/O ports of the microcontroller. The modules of assembly language program are transferred to the controller through data communication software.

This sequence plays a vital part in development as without proper sequencing of modules of programs in microcontroller, requisite operation cannot be obtained from the system.

3.4 ASSEMBLY PROGRAMS OF INTEGRATED SYSTEM

There are large numbers of assembly language programs which are being developed for performing the desired task of storing and display of data through digital controller.

```
/*This program performs password detection before data is stored in external or internal RAM locations connected to Microcontroller*/

cseg at 0 ; absolute segment starting at (origin) 0

org 0000h ; location from where program execution begins

mov 90h,#04h

mov dptr,#003fh ; Password character location

back: mov a,90h

mov r5,a

clr a

movc a,@a+dptr ; move the character to accumulator

jz 2100h ; location where start of programming can be done

cjne a,05h,target ; comparison of characters

inc dptr

sjmp back ; jump back till all characters matched

target: mov r4,#1 ; no. of times password check takes place

djnz r4,target
```

```
ajmp powerdown ; Jump to power down
powerdown: mov 87h,#2h ; value for switching off microcontroller
end
/*Program to store password in external or internal RAM location*/
cseg at 0 ; absolute segment starting at (origin)
org 0000h ; location from where program execution begins
mov 89h,#20h ; set Timer Mode
mov 8dh,#0fah ;Set Baud Rate
mov 98h,#50h ; Set serial control transfer register
setb 8eh
mov a,#'r' ;mov character to acc
acall recv ;jump to completely receive
mov a,#'e'
acall recv
mov a,#'a'
acall recv
mov a,#'d'
acall recv
mov a,#'y'
```

```
acall recv
recv: mov 99h,a ;serial buffer for transmitting data
here: jnb 99h,here ;stay here till TI flag is not set(i.e. 1)
mov dptr,#0041h ;location where data to be stored
movx @dptr,a ;store data in RAM location
inc dptr
clr 99h ; Clear TI flag
ret
end ; End of program
```

/*Program that will digitize an input voltage and storing value in external RAM location 5000h to 53e7h with specific time delay*/

```
cseg at 0 ; absolute segment starting at (origin) 0
org 0000h ; location from where program execution begins
adc: mov dptr,#5000h ;load data pointer with external RAM address
clr 0b2h ;generate Cs to ADC
next: clr 0b3h ;generate WR pulse
setb 0b3h
clr 0b4h ;generate RD pulse
mov a,90h ;get data
```

```
setb 0b4h ;end of RD pulse
movx@dptr,a ; store in external RAM
inc dptr ;point to next location
mov a,83h
        cjne a,#53h,wait
mov a,82h
cjne a,e8h,wait
sjmp done ;finished if both test pass
wait: mov r1,#74h ;delay for specific time
here: djnz r1,here
sjmp next
done: sjmp done ;simulate rest of program
end ;end of program
/* Program to run in burst mode collect and display to output*/
cseg at 0 ; absolute segment starting at (origin) 0
org 0000h ; location from where program execution begins
start: mov 89h,#20h ;Set Timer1 mode1
mov 8dh,#0fdh ; Set or load timer
mov 98h,#50h ;8 Bit UART Receive enabled
```

```
mov 88h,#0 ;nothing running
setb 8eh ;Start timer 1
input: call Serin ;get character
cjne a,#3ah,input ;wait for 3ah
call ghd ;get a byte
mov 0f0h,a ;number of bytes
call ghd
mov 83h,a ;msb of address
call ghd
mov 82h,a ;lsb of address
call ghd
jz getlin
call ghd
ljmp 2000h ; jump to run program
get lin: call ghd ;get data byte
movx @dptr,a ;store in memory
inc dptr ;next address
djnz b,getlin ;do until no data
call ghd
```

```
jmp inp
ghd: call getnyb ;get hex digit serially
swap a ;exchange nibbles
mov r7,a
call getnyb
orl a,r7 ;combine nibbles
ret ;return
getnyb: call Serin
jnb 0e6h
clr 0e5h
add a,#9 ;adjust for ASCII
getret: anl a,#0fh ;mask upper nibble
ret
Serin: jnb 98h,Serin ;ready?
clr 98h ;clear flag
mov a,99h ;get byte
ret ;return with character
end ;end of program
```

```
/*Program to send data to LCD for display from RAM while checking busy flag*/
cseg at 0 ; absolute segment starting at (origin) 0
org 0000h ; location from where program execution begins
mov a,#38h ;LCD 2 lines 5*7 matrix
acall command ;write command to lcd
mov a,#0eh ;LCD command for cursor on
acall command ;write command to lcd
mov a,#01h ;Clear LCD
        acall command ;write command to lcd
mov a,#06h ;Shift cursor right command
acall command ;write command to lcd
mov dptr,#5100h ;address data is stored
movx a,@dptr ;move to acc
acall lcd ;display data
till: sjmp till ;stay here
command: acall ready ;is LCD ready
mov 90h,a ;issue command code
clr 0b2h ;RS=0 for command
clr 0b3h ;R/W=0 to write to LCD
```

```
setb 0b4h ;E=1 for H to L pulse
clr 0b4h ;E=0 latch in
ret
lcd: acall ready ;is LCD ready
mov 90h,a ;issue command code
setb 0b2h ;RS=1 for command
clr 0b3h ;R/W=0 to write to LCD
setb 0b4h ;E=1 for H to L pulse
clr 0b4h ;E=0 latch in
ret ;return back
ready: setb 97h ;make P1.7 input port
clr 0b2h ;RS=0 access command register
setb 0b3h ;R/w=1 read command register
back: clr 0b4h ;E=1 for H to L pulse
setb 0b4h ;E=0 H to L pulse
jb 97h,back ;stay until busy flag=0
ret
end ;end of program
```

```
/* Program to display data to LCD checking busy flag using 8255 chip Connected to D0-D7 pins of LCD and to microcontroller*/
cseg at 0 ; absolute segment starting at (origin) 0
org 0000h ; location from where program execution begins
mov a,#80h ; all 8255 ports as output
mov r0,#23h ;load control register address
movx @r0,a ;issue control word
mov a,#38h ;LCD 2 lines 5*7 matrix
acall command ;write command to lcd
mov a,#0eh ;LCD command for cursor on
acall command ;write command to LCD
mov a,#01h ;Clear LCD
acall command
mov a,#06h ;Shift cursor right command
acall command
mov a,#'D' ;Display data
acall display ;Send data to LCD display
mov a,#'C' ;Display data
acall display ;Send data to LCD display
```

```
mov a,#'E'
acall display
mov a,#'D'
        acall display
mov a,#'e'
acall display
        mov a,#'l'
acall display
        mov a,#'h'
acall display
        mov a,#'i'
acall display
command: mov r2,a ;save acc. value
mov a,#90h ;PA=IN to read LCD status
mov r0,#23h ;Load control reg address
movx @r0,a ;configure PA=IN and PB=OUT
mov a,#00000110b ;RS=0,R/W=1,E=1 Read command
mov r0,#21h ;load port B address
movx @r0,a ;RS=0 R/w=1 for RD and RS pin
```

```
mov r0,#20h ;load port A address
ready: movx a,@r0 ;read command reg
rlc a ;move D7 into carry flag
jc ready ;wait until LCD is ready
mov a,#80h ; all 8255 ports as output
mov r0,#23h ;load control register address
movx @r0,a
        mov a,r2 ;get back value to LCD
mov r0,#20h
movx @r0,a
mov r0,#21h
        mov a,#00000100b ;RS=0,R/W=0,E=1 for H to L
movx @r0,a
nop
nop
mov a,#00000000b ;RS=0,R/W=0,E=0 for H to L
movx @r0,a ; latch in LCD data pin info
ret ;return back
display: mov r2,a
```

```
mov a,#90h
mov r0,#23h
movx @r0,a
mov a,#00000110b
mov r0,#21h
movx @r0,a
mov r0,#20h
ready1:movx a,@r0
rlc a ;move D7 into carry flag
jc ready1 ;wait until LCD is ready
mov a,#80h
mov r0,#23h
movx @r0,a
        mov a,r2
mov r0,#20h
movx @r0,a
mov r0,#21h
mov a,#00000101b ;RS=1,R/W=0,E=1 for H to L
movx @r0,a ;activate RS R/W E pins
```

```
nop ;make E pin wide enough
nop
mov a,#00000001b ;RS=1,R/W=0,E=0 for H to L
movx @r0,a ;latch in LCD data pin info
ret ;return back
    end ;end of program
```

DELAY IN OUTPUT OF DESIRED TIME:

We can introduce delay of desired amount by using following instructions:

1. Divide desired time delay to be introduced with 1.085 microseconds.
2. Perform 65536-n, where n is decimal value in step1.
3. Convert the result into Hex YYXX is the initial value to be loaded into Timer's Registers.
4. Set TL=XXh and TH=YYh in timer which is used.

Use following instructions:

MOV TMOD,#01h or 10h

AGAIN: MOV TL1,#XXh

MOV TH1,#Yyh

SETB TR1

BACK: JNB TF1,BACK

CLR TR1

CLR TF1

SJMP AGAIN

BAUD RATE can be simply doubled of the value which is being used by these three instructions: MOV A,PCON ; Move contents of PCON to Acc

SETB ACC.7 ; make D7=1

MOV PCON,A ; SMOD=1 double Baud Rate

Thus with help of following instructions a simple means of doubling the Baud Rate.

3.5 DATA COMMUNICATION SOFTWARE The overall communication software required between PC and microcontroller is either we are using Crosstalk or DATA COM Transfer package provided along with the kit which enables the serial communication between the two for data transfer basically the developed assembly language programs on which microcontroller work is being performed is saved in its internal memory and send to PC via its serial port. The data available for display is at parallel ports of controller and can be transferred to display unit.

Data transmission and Reception Algorithm

In Transmission the steps are:

- Initialise the Timer and Port
- Check for end of conversion
- Inform the receiver of the beginning and end of transmission and parity check
- Convert parallel word into stream of Bits
- Transmit one Bit at a time first low going start bit 8 data bits and last one high stop bit

In Reception the steps are:

- Initialise the Timer and Port
- Wait for Start bit and recognize the beginning of Transmission
- Receive Serial bits one at a time and convert into parallel byte
- Check for errors and recognize end of transmission

PROTOTYPE DEVELOPMENT, TESTING AND VERIFICATION

4.1 PROTOTYPE DEVELOPMENT

The development of prototype is accomplished through a number of steps involving analysis and design. These steps are:

1) Design and development of Hardware and Software

2) Test and Verification of the Hardware

3) Test and verification of the Software

4) Testing of the integrated system

DESIGN AND DEVELOPMENT OF HARDWARE AND SOFTWARE

The product development has certain Hardware and Software requirements. These requirements are being introduced and described in previous chapters. While developing a prototype the basic step is to plan according to these requirements the layout is done, components are selected, and compatibility is seen before performing operation on these module components. This hardware are integrated in useful manner so that set up can be made for product development and software are run on the microcontroller to get the desired results out of the set up.

TEST AND VERIFICATION OF THE HARDWARE

The Hardware module to be integrated is described in previous chapter. The serial interface is connected between microcontroller and input digital data to feed display through display through display units through parallel transmission. The microcontroller is used for the purpose of executing the program and performs desired operation. Individual test are performed separately to ascertain there appropriate operation. Their compatibility is tested through artificially generated signals.

TEST AND VERIFICATION OF THE SOFTWARE

The integrated software is developed as various parts with assembly language programs. The developed programs are checked with the help of On chip Simulator before they are integrated.

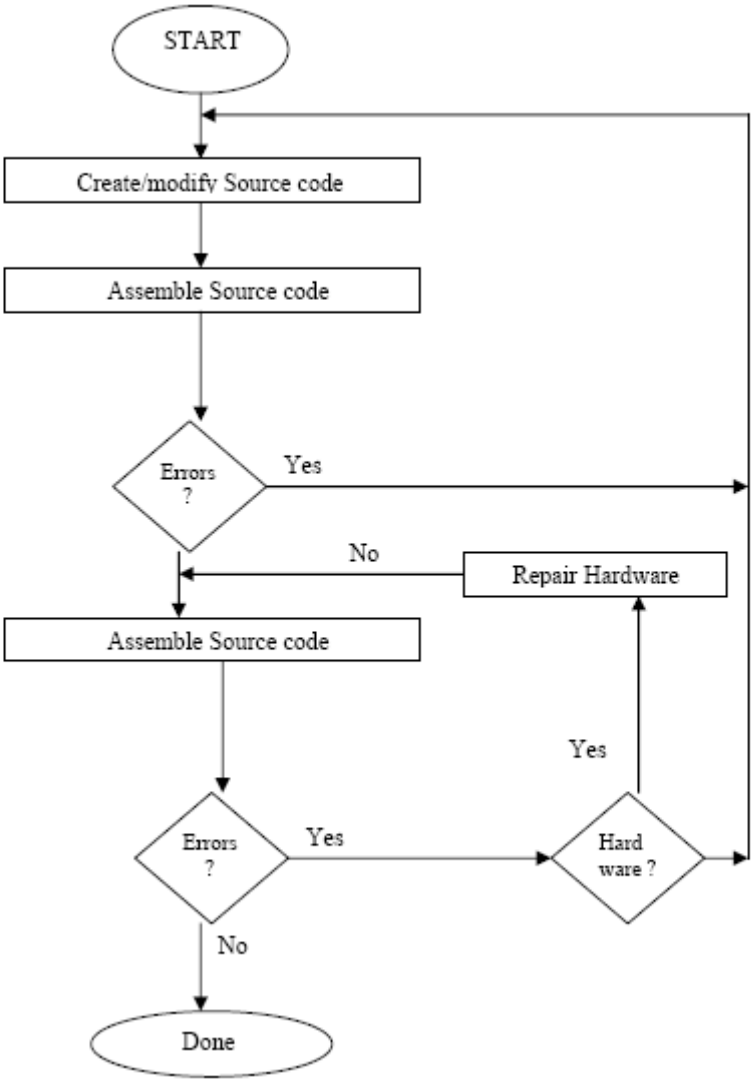


FIG 4.1 FLOW CHART FOR DEVELOPMENT PROCESS

Reads51 (Rigel's Embedded Applications Development System, V4.xx) is an Integrated Development Environment (IDE) that currently supports Rigel's 8051 family embedded

controller boards. The IDE supports several tool chains, most notably the previous READS V3.x absolute assembler and rChipSim51 V3.x simulator. This makes Reads51 an upgrade IDE for those who would like to use the previous software tools. Compatibility of software is tested through artificially generated signals.

TESTING OF THE INTEGRATED SYSTEM

The programs developed are run on On Chip Simulator after they are being successfully assembled they are being run on the hardware set up of the project and the system is tested by running it over several time periods and review the data for accuracy. The integrated software is successfully run on the system to verify its desired operation.

4.2 RESULTS AND DISCUSSIONS

After integration of hardware and software as a complete integrated system the results are taken from developed prototype. Where hardware and software has been tested for conditions on considered under simulated cases, before storing code and data program on hardware it is tested and verified with help of on chip simulator. The results are shown in figure 4.2 – 4.3 to display the stored data with hardware used while developing the prototype in the laboratory.

The desired prototype has been developed and satisfactory results are obtained with troubleshooting of the hardware. This work has been implemented with number of specific applications depending upon the requirements.

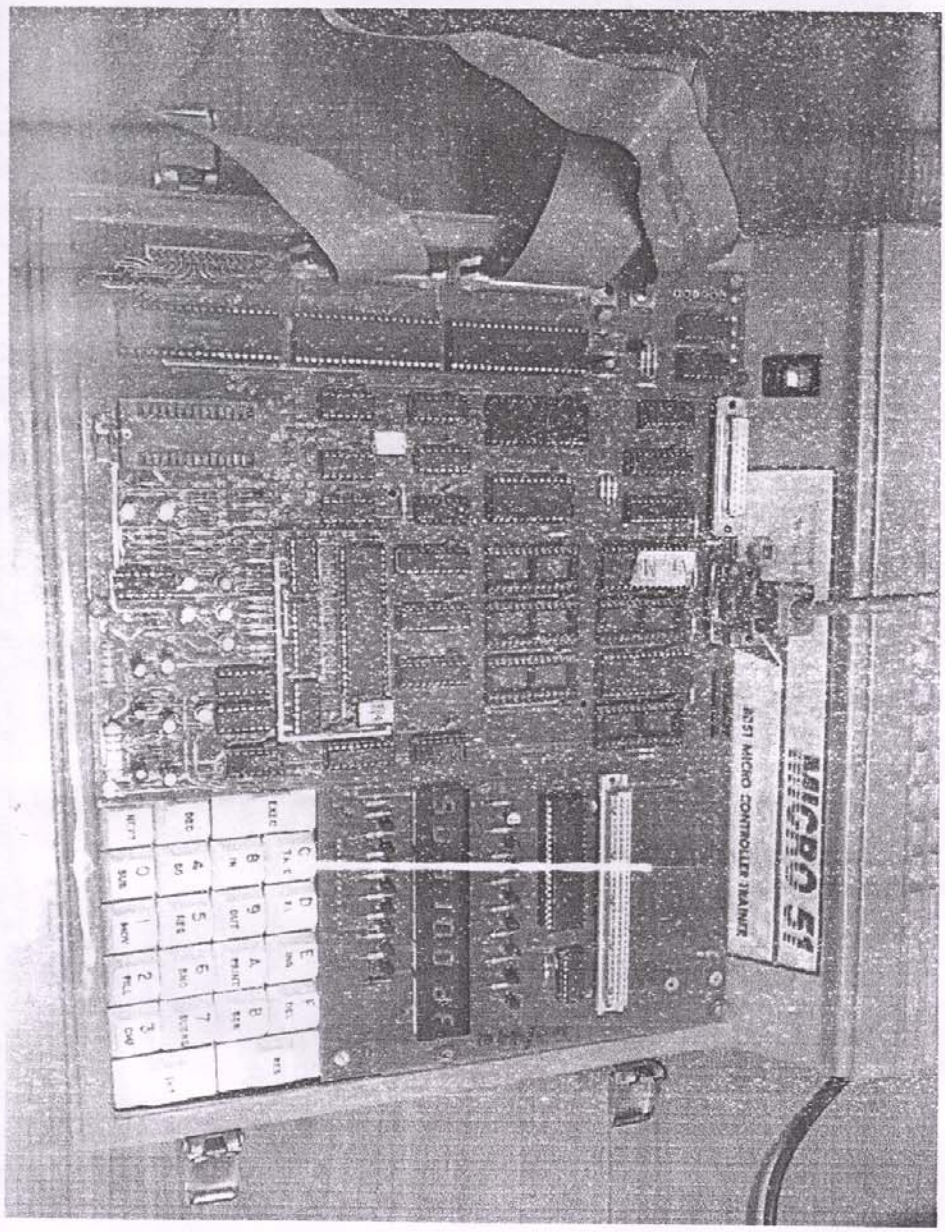
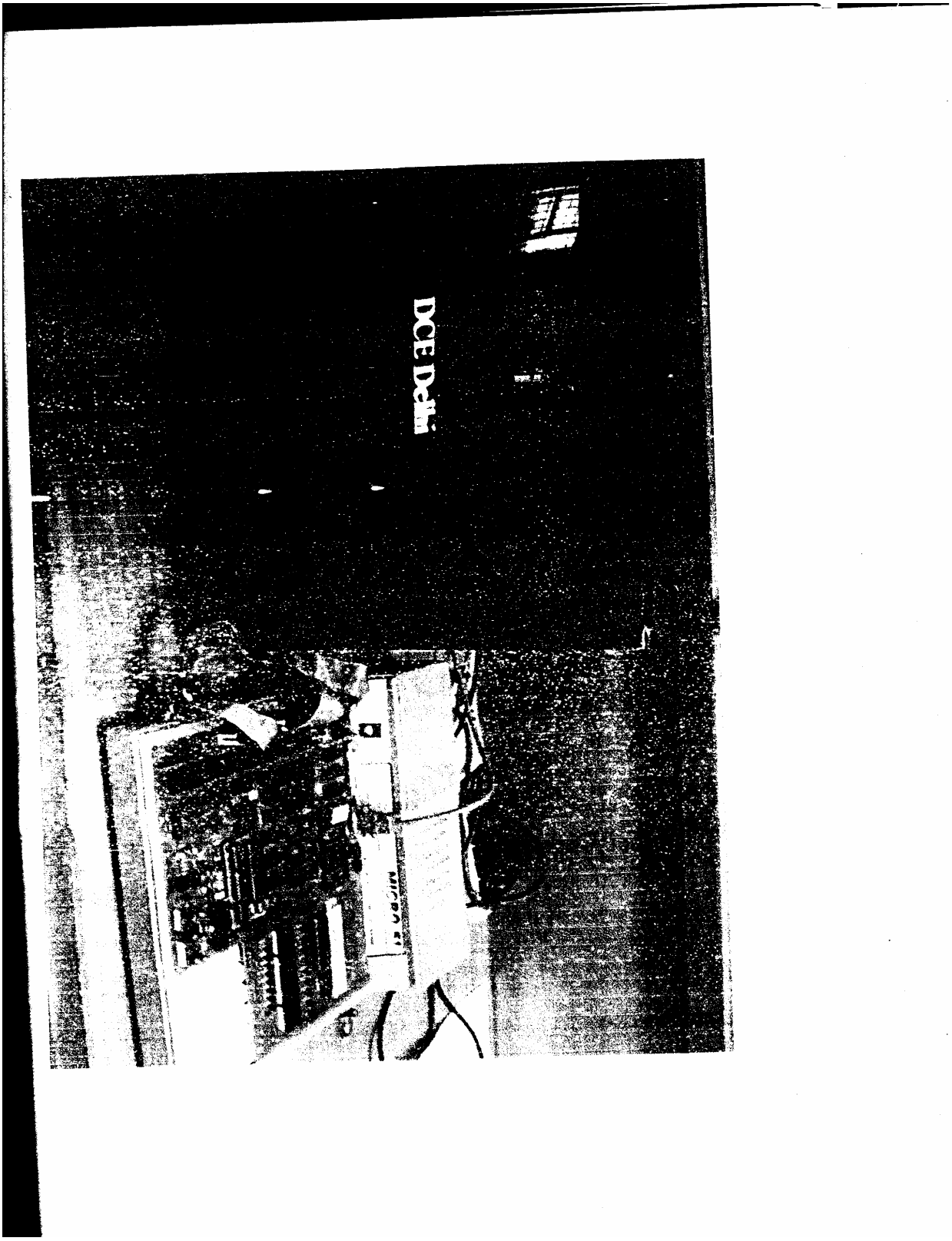


Fig 4.1 _ The microcontroller kit.



MAIN CONCLUSION AND FUTURE SCOPE

5.1 MAIN CONCLUSION

The Developed prototype in thesis work provides a secured, programmable and flexible way to control the display of data or Information from remote location with control of its sequence and time through digital controller (Microcontroller 8051). As of now there are compliant chipsets available to support this system. The developed prototype is feasible due to simplicity, portability and cost effectiveness.

The work in this thesis provides insight over an integrated system development involving both the Hardware and Software systems to perform simultaneously to attain the desired results. The Assembly programs developed are compatible to the hardware system developed, and the system is tasked through software with the help of On chip simulator.

The developed hardware modules are integrated with software to get the output in accordance with industrial specifications. The selected components are of commercial grade and are compatible with software speed so that the connections can be easily made to attain appropriate results. Although prototype has been developed with best available systems in the lab, there are limitations of its cost and performance while developing its prototype. But for a target board development the performance will improve and cost will come down drastically.

The developed prototype can be further upgraded to a system which will meet time to time requirements of an flexible system. We can use microcontroller of other make with additional features, higher memory, low cost. The space for data storage can be increased by using high capacity memory modules.

The software portion can be improved by using several fail safe microcontroller communication operation and modes so that information signal can be send with greater ease more fast and accurately. The work can be improved by utilizing programs or methods to work well in the noisy environment and interface with other devices in vicinity.

5.2 FUTURE SCOPE OF WORK

The developed prototype controls the display of data with its sequence and timing control, and can be used for numerous applications. This developed prototype can be used for display of advertisements and information at Malls, Stations, Theatres, Airport, industrial units and other places where different advertisement and information need to be displayed again and again with some specific time interval.

This work can be carried out further by detecting the type of input, be it is either analog or digital data with selector through different channels. The display unit can be safeguard from external interrupt sources, EMI radiations through proper shielding. The storage can be improved by using better storing methods and dual porting of memories. Thus the developed prototype can be in used for wide range of applications.

REFRENCES

[1] Muhammad Ali Mazidi**, "**The 8051 Microcontrollers and Embedded Systems", Pearson Education, Fifth Edition

[2] Kenneth J. Ayala, "The 8051 Microcontroller Architecture Programming and Applications", PRI Publication, Second Edition

[3] Richard Barnett, "The 8051 Family of Microcontrollers", Prentice Hall, Third Edition

[4] George Kennedy, "Electronic Communication System", Tata McGraw Hill, Third edition

[6] Douglas Hall, "Microprocessor and Interfacing", Tata McGraw Hill, Second edition

[7] B.P.Lathi, "Digital and Analog Communication", Winston Publication, Second Edition

[8] VI Microsystems, "User Manual and Technical Manual for MICRO-51", Chennai

[9] Yashwant Kanitkar, "Let Us C", BPB Publication, Sixth Edition

[10] Texas Instruments, "EIA-232 Interface Circuits" (Data Book)

[11] Han-Way-Huang, "Using the MCS-51 Microcontroller"

[12] www.google.com

[12] www.byte.com

[13] www.intel.com/m51

[14]www.pjrc.com/tech