DEVELOPMENT OF AN ELECTRONIC MESSAGE DISPLAY

BY NANDOR JUAN SIMON

Thesis submitted in partial fulfilment of the requirements for the Masters Diploma in Technology to the Department of Electrical Engineering (light current) at the Cape Technikon.

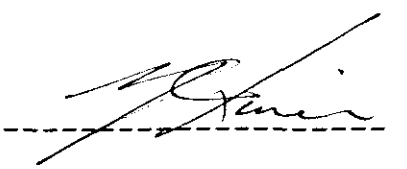
Research and Development Centre
TELKOM

CAPE TOWN
SOUTH AFRICA
1993

DECLARATION

I declare that the contents of this thesis represents my own work and the opinions herein are my own. It has not been submitted before for any examination at this or any other institute.

N. J. SIMON



(Signature)

ABSTRACT

In the last decade the advertising industry has developed into an advanced science which increasingly relies on the use of electronic utilities and modern technology. The advent of the microcontroller has made it possible to incorporate electronic intelligence into advertising utilities. This thesis describes the design, development and functioning of a stand alone programmable electronic message display as required by the Electrical Engineering Department of the Cape Technikon.

OPSOMMING

Die advertensie bedryf het in die laaste dekade ontwikkel tot n gevorderde wetenskap wat al hoe meer staatmaak op elektroniese hulpmiddels en moderne tegnologie. Die uitvinding van die mikroverwerker het dit moontlik gemaak om n mate van elektroniese intellegensie te inkorporeer in adverterings hulpmiddels. Hierdie verhandeling beskryf die ontwerp, ontwikkeling en funksionering van 'n onafhanklike programmeerbare elektroniese advertensie vertoonbord soos benodig deur die Elektriese Ingenieurs Departement van die Kaapse Technikon.

ACKNOWLEDGEMENTS

I would sincerely like to thank all the staff and colleagues at the Research and Development Centre of Telkom for all the help received, advice and ideas given during this project.

I would like to thank Mr. Kleinhans who initiated the idea for the development of this project, his colleagues and the Cape Technikon for the financial support given during the development and design of this project.

I especially would like to thank my dearest wife, Anél, for her moral support and assistance.

CONTENTS.

1. INTRODUCTION.

We are in the midst of an electronic revolution where computers and high-tech electronic equipment is the order of the day. Daily the human races try to invent ways and means to simplify their daily activities, enhance their way of living and try to create more sufficient methods of doing business. Small business advertising is one of the industries in South Africa that has not yet been influenced by electronics (not taking television and radio into account). Small businesses have to rely on newspaper adds, pamphlets and brochures to advertise their products. Programmable electronic message displays is the future in advertising. These displays are an inexpensive, attention grabbing way to advertise or sell a product or service and will work day and night without advertising costs. These display's have various applications e.g.

- Electronic notice boards at company entrances to inform staff of meetings, etc.
- Announcing a sale or special offer in shop windows.
- Announcing business hours at entrances.
- In vehicles to display slogans, advertisements or political messages.
- At cinemas, theatres, etc. to show current and forth coming attractions.
- In pubs to inform about special occasions or bands performing.
- In bus terminals, railway stations or airports to

announce and update bus, train or plane arrivals.

Based upon the above applications, the system was designed and constructed using a 8051 series microcontroller, memory and peripheral components. Software was written in low level microcontroller language.

Most of the relevant information for the system design was obtained from technical literature (data books, technical manuals, etc.) and at the Research and Development Centre (TELKOM S.A.) in Cape Town.

2. OBJECTIVES.

Electronic message displays currently available in South Africa are imported from abroad and is not manufactured in this country.

The major objective of this project is to design and develop such a message display to accommodate the growing need in the advertisement industry for such equipment.

Proposed specifications:

- The screen will consist of 20 character dual line display.
- Highly visible message using a LED matrix.
- Extremely easy to program.
- Using the ASCII character set in two font types.
- Attractive high quality Perspex housing.
- Messages stored in non-volatile memory (no battery backup necessary)
- Stores up to 10 programs.

- Low power consumption.
- Stand alone unit.
- Power down facilities.

The objective of this thesis report is to provide a detailed outline and analysis of the proposed system.

3. HARDWARE IMPLEMENTATION.

On having decided to use a microcontroller it was decided to use a CMOS type microcontroller because of its versatility, simplicity and low power consumption. The INTEL 80C51/FA-1 (ROMLESS version) was chosen for this purpose. The decision to use the ROMLESS version is justified by the fact that greater flexibility is given to the system, for expansion or modification, by simply changing the program in the external EPROM.

3.1 CENTRAL PROCESSING UNIT (CPU).

The features of the 80C51/FA-1 core are:

* 8-bit CPU Optimised for control applications
* Extensive Boolean Processing (single bit logic) Capabilities
* 64 Kilobytes (64K) Program Memory Address Space
* 64K Data Memory Address Space
* 256 Bytes of On-Chip Data Read Only Memory (RAM)
* 32 Bi-directional and Individually Addressable Input/Output (I/O) Lines (Four I/O ports)

* Three 16-bit Timer/Counters
* Programmable Counter Arrays
* 7 Interrupt Sources
* Programmable Serial Channel
* TTL and CMOS Compatible Logic Levels
* MCS-51 Compatible Instruction Set
* Power Saving Idle and Power Down Modes
* On-Circuit-Emulation Mode

For control situations the 80C51/FA-1 is ideal as it supports one-bit or Boolean processing operations. This incorporates special features such as the direct manipulation and testing of individual bits and the use of single-bit variables for performing logical and arithmetic functions.

3.1.1 ADDRESS/DATA BUS.

The CPU of the 80C51/FA-1 has a 16-bit internal address bus, which consists of an 8-bit multiplexed address/data bus (AD0-AD7) and represents the lower part of the 16-bit address bus and address lines A8 - A15 which represent the higher part of the 16-bit address bus.

3.1.2 INPUT/OUTPUT PORTS.

The 80C51/FA-1 has 4 ports and for this application the ports will be used as follows:

1) Port 0 (pins P0.0 - P0.7) is an 8-bit open drain bi-directional I/O port. It is used as the low-order

address and data bus during access to the EPROM, RAM, Keyboard Encoder and the Programmable Peripheral Interface (PPI). (See Appendix C-2 and C-3)

2) Port 2 (pins P2.0 - P2.7) is an 8-bit bi-directional I/O port with internal pull-up resistors. This port emits the high-order address byte during access to the EPROM, RAM, Keyboard Encoder and Programmable Peripheral Interface. (See Appendix C-2 and C-3)

3) Port 1 (pins P1.0 - P1.7) is an 8-bit bi-directional I/O port with internal pull-up resistors. This port is bit addressable and used to scan, reset and clock the display.

The port pins are used as follow:

P1.0 SERDATA (Serial Data to the Display)

P1.1 RRES (Reset Shift Registers on the Display)

P1.2 REGCLK (Clock Data into Shift Registers)

P1.3 - P1.7 Not used

(See Appendix C-2 and C-3)

4) Port 3 (pins P3.0 - P3.7) is an 8-bit bi-directional I/O port with internal pull-up resistors. The READ $\overline{\text{(RD)}}$ and WRITE $\overline{\text{(WR)}}$ lines as well as the external Keyboard interupt (KEYINT) and Power down interupt (PWRINT) are connected to this port.

P3.2 $\overline{\text{INT0}}$ (Interrupt from the Keypad Encoder)

P3.3 $\overline{\text{INT1}}$ (Interrupt from the Power down switch)

P3.6 $\overline{WR}$ (Write line to Peripheral Components)

P3.7 $\overline{RD}$ (Read line to Peripheral Components)

P3.1 - P3.2 & P3.4 - P3.5 Not used

(P3.2; P3.3; P3.6; P3.7 Appendix C-2 and C-3)

3.1.3 RESET OF MICROCONTROLLER.

The reset input (RST), will reset the microcontroller if the RES1 line is held high for at least two machine cycles while the oscillator is running. The RST pin is connected through a 10µF electrolytic capacitor to +5V (VCC) and through a 10KΩ resistor to ground (GND), to enable the microcontroller to reset automatically when power is applied to the system. (U1 pin 9 - Appendix C-2)

3.1.4 OSCILLATOR OF CONTROLLER.

The CPU makes use of its internal on-chip oscillator with a 12 MHz crystal connected between the XTAL1 and XTAL2 pins of the controller. The two legs of the crystal is connected through two 30pF capacitors to GND. (U1 pins 18 & 19 - Appendix C-2)

3.1.5 THE OCTAL ADDRESS LATCH.

The address latch enable (ALE) output is connected to the control (C) input of the 74HC573 octal D-type latch. During external accesses to I.C.'s the ALE line pulse high to latch the low byte (AD0-AD7) of the address into the octal latch. Addresses A8 -A15 are connected directly to the appropriate pins of the various I.C.'s that require addressing. (U1 pin 30 - Appendix C-2)

3.1.6 PROGRAM MEMORY. (EPROM - 27C128A)

In this application the enableaddress ($\overline{EA}$) pin of the controller is connected to GND to enable the controller to access external program memory. (U1 pin 31 - Appendix C-2) or the program memory the ultra violet erasable 27C128A EPROM was used (U5 - Appendix C-2). The memory consists of an array of 16384 x 8-bit words. The chip employs advanced CHMOS circuitry for systems requiring low power, high performance speeds and immunity to noise. The 27C128A has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip enable ($\overline{CE}$) is the power control and is connected to GND to permanently select the EPROM (U5 PIN 20 - Appendix C-2). The PGM and VPP pins are connected to VCC and is only used when programming the chip. Code memory is normally non-volatile and only READ functions are available. To access external memory, the 8 lower bits of the address are multiplexed out on the port 0 pins, and must be latched by the 74HC573 (U3) 8-bit latch, activated by the ALE control pin. At the same time, the 8 higher address bits are multiplexed out on port 2. The read function for code memory is controlled by the $\overline{PSEN}$ (program segment enable) pin from the controller, and the program data is read into the controller through Port 0.

3.1.7 DATA MEMORY. (RAM - 62C256)

The 62C256 static RAM was chosen for external data memory. The memory array consists of 32768 x 8-bit words

which make up the 32K which can be addressed. Only 16K of data memory is used due to the 14 address lines available. The 15th address pin (U6 pin 1 Appendix C-2) of the RAM is connected to ground. The RAM has three control pins, chip select ($\overline{CE}$), output enable ($\overline{OE}$) and write enable ($\overline{WE}$). These pins have to be logically active in order to write data to the device or to obtain data from the device. The $\overline{OE}$ and $\overline{WR}$ are directly connected to the $\overline{RD}$ and $\overline{WR}$ outputs of the controller respectively. The $\overline{CE}$ pin is the power control pin used for device operation.

3.1.8 THE SMARTSOCKET (DS1213C).

The DS1213C smart socket is a 28-pin DIP socket with a built-in controller circuit and an embedded lithium energy source. It accepts either an 8K X 8 or a 32K X 8 CMOS static RAM. When the socket is mated with a CMOS RAM, it provides a complete solution to problems associated with memory volatility. The smart socket monitors incoming Vcc for an out-of-tolerance condition. When such a condition occurs, an internal lithium source is automatically switched on and write protection is unconditionally enabled to prevent garbled data to be written to the RAM. The smart socket uses only pins 28 and 20 for RAM control. All other pins are passed straight through to the socket receptacle.

Advantage: Using the smart socket saves printed circuit board space since the combination of smart socket and

memory uses no more area than the memory alone. It is also more reliable than a external lithium back up power source and its life expectancy is approximately ten years.

(See Appendix C-2 for data sheets.)

Operation:

The DS1213C smart socket performs five circuit functions required to battery back up a CMOS memory.

First, a switch is provided to direct power from the battery or Vcc supply, depending on which is greater. This switch has a voltage drop of less than 0.2 volts.

The second function is power-fail detection. Power-fail detection occurs between 4.75 and 4.5 volts. The DS1213C constantly monitors the Vcc supply. When Vcc falls beyond 4.75 volts, a precision comparator detects the condition and inhibits the RAM chip enable.

The third function accomplishes write protection by holding the chip enable signal to the memory to within 0.2 volts of Vcc or battery supply. If the chip enable signal is active at the time power fail detection occurs, write protection is delayed until after the memory cycle is completed to avoid corruption of data. During nominal power supply conditions the memory chip enable signal will be passed through to the socket receptacle with a maximum propagation delay of 20ns.

The fourth function the DS1213C performs is to check battery status to warn of potential data loss. Each time that Vcc power is restored to the SmartSocket the battery voltage is checked with a precision comparator. If the

battery supply is less than 2.0 volts, the second memory cycle is inhibited.

The fifth function the SmartSocket provides is battery redundancy. In many applications it is desirable to use two batteries to ensure reliability. The DS1213C provides a internal isolation switch which provides for the connection of two batteries. During battery back up the battery with the highest voltage is selected for use. If one battery fails the other immediately takes over.

3.1.9 THE 3-TO-8 LINE ADDRESS DECODER (74HC138).

The processor is memory-mapped and selection of the various peripheral devices e.g. RAM, Programmable peripheral interface (PPI) and Programmable Keyboard Interface (PKI) is done by the 74HC138 3-to-8 line decoder (U2), under processor control via address lines A14 and A15. The 74HC138 has three binary selected inputs (A, B & C). These A, B & C inputs determine which one of the eight normally high outputs will go low. The device is enabled, by connecting the G2A and G2B pins to GND and the G1 pin to VCC. (U2 Appendix C-2)

The outputs are allocated as follow:

CS0 - RAM (Random Access Memory)

CS1 - PPI (Programmable Peripheral Interface)

CS2 - PKI (Programmable Keyboard Interface)

3.1.10 THE PROGRAMMABLE PERIPHERAL INTERFACE (PPI).

This application uses only ports A and B. When power is applied to the system the 82C55A will automatically reset

through the reset circuit which consists of a 4.7µF cap.(C14) connected to VCC and a 10KΩ resistor (R2) connected to GND.

Address lines A0 and A1 are connected to the A0 and A1 inputs of the 82C55A and is used to address the chip. The D0 - D7 data lines are directly connected to the multiplexed address/data port (port 0). The PORT A lines (PA0 - PA7) and PORT B lines (PB0 - PB7) of the 82C55A are connected to the top eight and bottom eight transistor switches of the display respectively. (U7 Appendix C-2)

3.1.11 THE PROGRAMMABLE KEYBOARD INTERFACE (PKI).

The Intel 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel microprocessors. The keyboard portion provides a scanned interface to a 64-contact key matrix. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8x8-bit FIFO (First in first out 8-byte buffer). A key entry will set the interrupt output line (IRQ) of the 8279 to the micro controller via a 74HC04 inverter (U10 Appendix C-2).

The 8279 is provided with a CLK input. The one output of the system oscillator (OSC2) is connected to the CLK input of the 8279. This enables the 8279 to scan the keyboard by using its internal timing circuitry and therefore relieving the CPU from scanning the keyboard.

The 8279 shares the same reset circuitry as the 82C55A (PPI).

The I/O control section uses the $\overline{CS}$, A0, $\overline{RD}$ and $\overline{WR}$ lines to control the data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$ (Enabled from the 3-to-8 line decoder output CS2). The character of information, given or desired by the CPU, is identified by the A0 input. A logic one means that the information is a command or status. A logic zero means the information is data. The $\overline{RD}$ and $\overline{WR}$ lines determine the direction of data flow to and from the 8279.

Output keyboard scan lines SL0 - SL2 are connected to the A, B and C inputs of a 3-to-8 line decoder (74HC138) respectively. Internally in the 8279 a binary upcounter is continuously emitting a binary code to these lines, which in turn will activate (active low) an output of the decoder thus scanning the eight columns of the keyboard. The eight input lines RL0 - RL7 are internally monitored to detect a key depression.

A shift (SHIFT) and control (CNT) input are also available to enable the keyboard to be expanded to 192 different key values. (U8 Appendix C-2)

3.2 THE DISPLAY.

The Display consist of a 120 x 16 Light Emitting Diode (LED) array. The array can be devised into two lines, each line able to display twenty alpha-numeric characters

making up a total of forty characters of information to be displayed on the two lines.

3.2.1 SERIAL IN PARALLEL OUT SHIFT REGISTERS. (74HC164)

In this application the 74HC164 (U1 - 15 Appendix C-4) was chosen due to its high operating frequency of 50 MHz. The display houses fifteen registers which are cascaded in a serial configuration. This shift register has two serial inputs (A and B) which are connected, and eight parallel outputs. Serial data is applied to the first shift register (U1 pin 1 and 2 Appendix C-4) by the SERDATA pin of the Microcontroller (U1 pin 1 Appendix C-2). The eighth parallel output of each shift register is connected to the serial data input of the next shift register in the cascade, to enable the data to be shifted from the first output of the 1st register (U1 pin 3 Appendix C-4), to the eighth output of the 15th register (U15 pin 13 Appendix C-4). The CLK inputs (U1 pin 8 Appendix C-4) of the registers are commonend and the shift registers are clocked simultaneously. This enables the data to shift to the next output, or register. The clock is supplied by the REGCLK output of the microcontroller (U1 pin 3 Appendix C-2) and is a positive going transition. The CLEAR inputs of the registers are commonend and the registers are simultaneously cleared by a "low" level at the CLEAR input, received from the RRES output of the Microcontroller (U1 pin 2 Appendix C-2). The outputs of the shift registers are used to activate one of the 120 columns of the display.

3.2.2 LED DRIVER CIRCUIT (ULN2804A).

The ULN2804A (U16 - U30 Appendix C-4) each contain eight Darlington transistors with common emitters and integral suppression diodes for inductive loads. The ULN2804A driver was chosen because of its unique drive capabilities. The driver has a current rating of 500mA continues and is more than sufficient to drive a column of 16 LED's. Each column of 16 LED's draws a maximum of 320mA. The 16 TIP31 NPN-transistors are used to interface the data from the programmable peripheral interface (8255) to the LED array and switches the +12V through to the anodes of the LED's. The cathodes of the LED's in each column are connected to a output of a driver. The anodes of the LED's in each row of the display are commoned and connected to a emitter of one of the transistor switches. Resistors R1 - R16 are used as voltage dividers in conjunction with the LED's, to protect the LED's from breakdown in the event when the scanning of the display is disabled by a microcontroller failure.

3.2.3 THE LIGHT EMITTING DIODES (LED's).

In this application LEDs were chosen because of their higher illumination capabilities, their longer life span and their lower cost, in relation to the 7 X 5 matrix LED blocks. The display consists of 120 columns and 16 rows which adds up to a total of 1920, 5mm, red, diffused LED's. Diffused LEDs were used to increase the viewing angle of the display. The display has a 120 degree

viewing angle and can comfortably be seen in day time. Each LED has a luminous intensity of 500 millicandellas (mcd).

3.3 THE POWER SUPPLY UNIT.

The electronic message display is powered from a mains powered power supply located inside the housing. The live and neutral 220 VAC rails pass through a 220 VAC switch with a build-in light, to enable the user to disconnect the display from the AC source, and a 5A fuse to protect the power supply from overloading. A 220V (5A) mains transformer (T1 Appendix C-5) with a power rating of 60VA and a 18 VAC output is used as the power source.

The processor, peripheral devices and EPROM are sensitive to voltage variations and therefore a stable constant voltage supply is needed.

The 18 VAC output of the transformer is rectified by a full wave bridge rectifier (BR1 Appendix C-5). Two smoothing capacitors C1 (100pF) and C2 (2200μF) are connected across the output of the bridge. The output of the bridge, approximately 25.5V, is connected via the cap's to two LM338KA voltage regulators REG1 and REG2.

The output of REG1 supplies +5V through a smoothing capacitor (C3), diode (D1) and 2A fuse (F2), to ensure a stable DC voltage supply for the processor board and the shift registers on the display board.

The output of REG2 supplies +12V through a smoothing capacitor (C4), diode (D2) and 2A fuse (F3) to the display drivers on the display board.

The output voltages of both REG1 and REG2 can be adjusted by RV1 and RV2 respectively to ensure the correct outputs needed. (See Appendix C-5)

3.4 THE KEYBOARD.

The 66-key keyboard is used to operate, program and enter data onto the display. Positive action dome type keys were used to limit key bounce to a minimum. Key debouncing is done by software. The keys are configured in a 11 X 6 matrix and is housed in a easy-to-handle, slim line, perspex box. The keyboard is connected to the EMD via a ribbon cable. The cable terminates on a plug which is plugged into the display box, and enables the user to detach the keyboard when not needed.

3.5 THE DISPLAY HOUSING.

The equipment is housed in a black perspex box which provides protection against sunlight, dust and water. perspex were chosen because of its durability.

The display screen is covered by a red transparent Perspex sheet to enhance the intensity of the display and reflect sunlight.

4. SOFTWARE DEVELOPMENT.

The execution program for the electronic message display was written in 8051 microcontroller assembler language using the ASM51 compiler. Different software test routines were written to initialise each peripheral and test the operation of the chips within the microcontroller board environment. Once the operation of the peripherals were tested, the different test routines were combined and an execution program was written. A modular approach was used when the execution program was written to simplify the writing thereof, debugging and to ease the understanding of the programming sequence.

4.1 PROGRAM DEVELOPMENT TOOLS.

The Unidux DICE-8051 in-circuit emulator was used for the software development of this project. The DICE can be used successfully with just a CRT (Cathode Ray Tube) terminal for standalone in-circuit emulation or with most host computers to provide a full development system. An in-circuit emulator is recognised as the best debugging and integration tool, providing a valuable shortcut and enhanced productivity for design engineers.

The emulator offers the following features:

* Offers a complete debugging facility.
* Real-time trace: The DICE-8051 collect information into memory during the real-time trace and the operator can set a variety of sampling start/stop conditions.

* Standalone workstation: By simply connecting to a CRT terminal, you can perform in-circuit emulation and your host computer can then be free for other purposes.
* Memory mapping: The DICE-8051 is supplied with 64K bytes program memory and 64K bytes of data memory.
* A variety of breakpoints: A powerful breakpoint monitor system is provided with DICE-8051 to enable the design engineer to control and analyse the target system.
* Dual CPU's (Central Processing Unit): The DICE-8051 employs two CPU's, one as a monitor and a second to replace the target system CPU.
* Built-in PROM programmer: The DICE-8051 incorporates a PROM programmer for the 8751 processor.
* Easily connected to your host computer: The DICE-8051 communicates via a RS232 serial interface to the host computer. The DICE-8051 can operate in transparent mode to enable easy connection and downloading of code and symbol tables from your development system to the DICE.
* XON/XOFF Protocol: The DICE uses this protocol and enables file transfers between the host computer and the DICE irrespective of the host computers type and operating system.
* Execution time: The DICE comes standard with a timer so that the development engineer can measure the exact time of his program.

By removing the microcontroller from the target system and replacing it with the emulator's probe, the emulator is now in the circuit and behave as if it was the target microcontroller. Thus, the emulator can access all signals and data to which the target's microcontroller accesses. This allows the user to control, test and check almost all possible functions in his target system.

4.2 MEMORY MAP.

The 80C51FA-1 microcontroller's internal 16-bit address bus is capable of addressing 64KBytes of program memory and data memory respectively. The address bus is common to all components on the microcontroller board connected to the bus. When an address is generated onto the bus by the microcontroller, all components will see the address but only one will be selected. Therefore in order for the microcontroller to address the correct component a memory map is needed to help with the decoding.

The memory map used for this project can be seen in figure 1. on page 20.

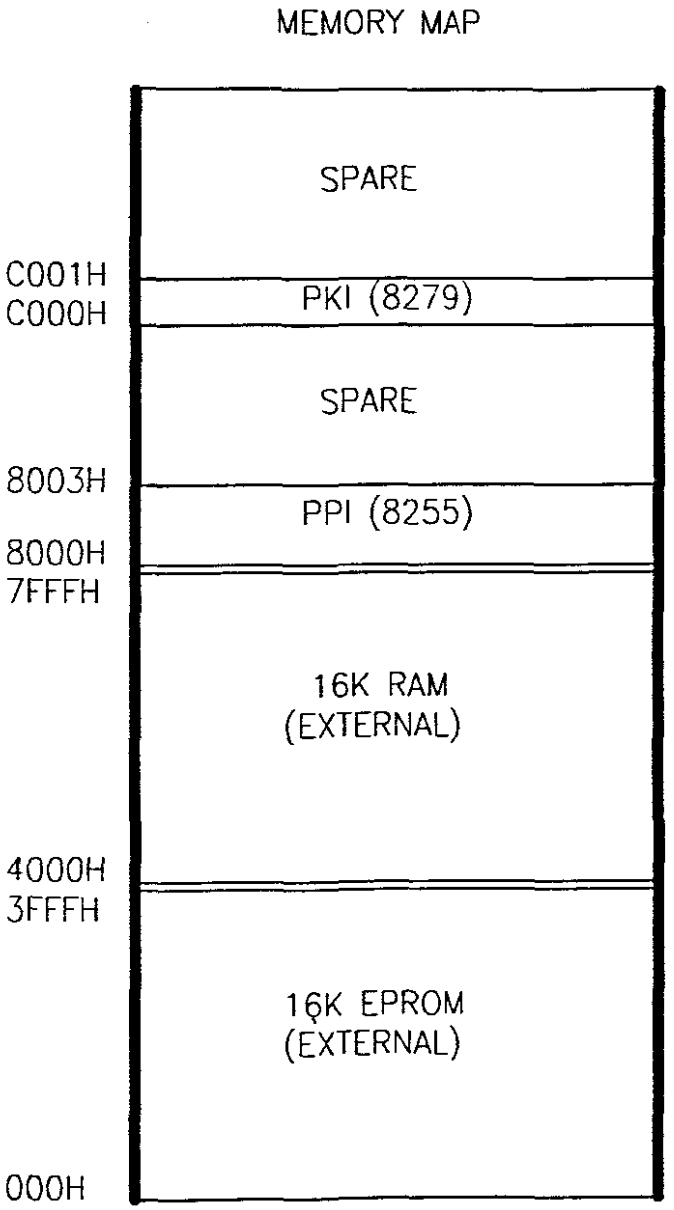


Figure 1.

4.3 ADDRESSING AND CONTROL WORDS.

4.3.1 THE EPROM.

As discussed previously the EPROM is selected by the $\overline{\text{PSEN}}$ line and no control words are necessary. The address space for the EPROM is located at the bottom of the memory map and ranges from address 0000H - 3FFFH which represents 16KBytes of program memory. Any address

emitted by the microcontroller onto the address bus, which falls into the above-mentioned range, will select the EPROM.

4.3.2 THE RAM.

The RAM is selected by $\overline{CS0}$ and does not require a control word. The address space for the RAM is located above that of the EPROM and ranges from address 4000H - 7FFFH which represents 16KBytes of data memory. Any address emitted by the microcontroller onto the address bus, which falls into the above mentioned range, will select the RAM for a read or write function.

4.3.3 THE PARALLEL PERIPHERAL INTERFACE.

Access to the Interface from the system bus is done through two address lines A0, A1 and a Chip-Select input CS1, decoded to allocate 4 sequential address locations to the interface. The data read and write addresses (A0, A1) for Ports A, B and C(not used) are 8000H, 8001H and 8010H respectively, while address 8011H is used to write and read a control word to select the modes and the input/output choice of the ports (see memory map). There are three basic modes of operation that can be selected by the system software:

Mode 0	-	Basic input/output
Mode 1	-	Strobed input/output
Mode 2	-	Bi-directional bus

In this application Ports A and B are both programmed for Mode 0. This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to and read from the specified port. In this application they are both programmed as outputs. The initialising programming routines is included in the boot and set-up routines of the system.

Figure 2. illustrates the control word for set-up and normal operation.

The Control Word: (To Address A1, A0 = 11)

1 Word	Port A & C Mode 0/1/2		Port A I/$\overline{O}$	Port C I/$\overline{O}$	Port B & C Mode 0/1	Port B I/$\overline{O}$	Port C I/$\overline{O}$
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

8 0 H

Figure 2.

4.3.4 THE PROGRAMMABLE KEYBOARD INTERFACE.

Access to the PKI from the system bus is done through two lines, address line A0 and a chip-select input $\overline{CS2}$. To select the PKI, two addresses C000H and C001H is used (see memory map). Address C000H is used to read and write data to the PKI. The commands in figure 3. and figure 4. program the PKI's operating modes. The commands are sent on the data bus with $\overline{CS2}$ low and A0 high and are loaded

to the PKI on the rising edge of $\overline{WR}$.

The PKI is programmed for "encoded scan keyboard" with N-key rollover, with the controller emitting the address C001H, onto the address bus, followed by the data of 02H.

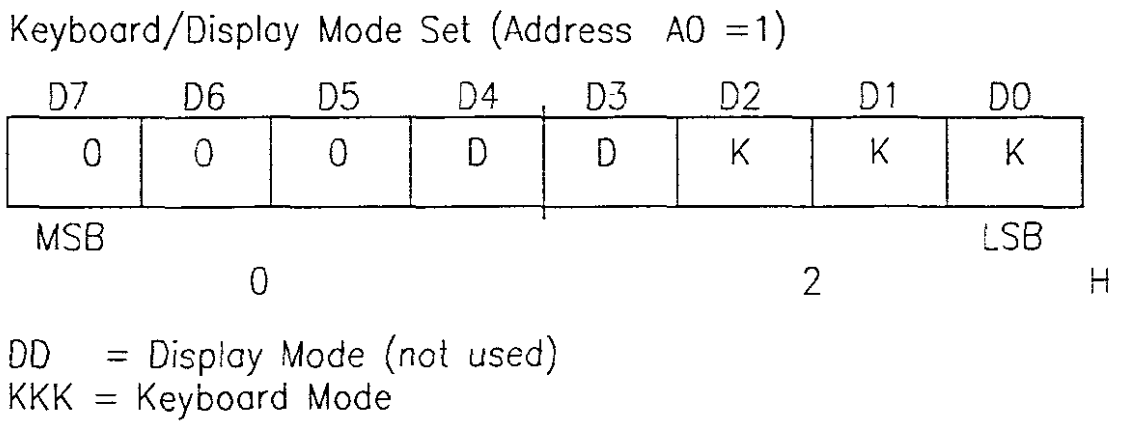


Figure 3.

Scanned Keyboard Mode with N-Key Rollover:

With N-key rollover each key depression is treated independently from all the others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO (First In First Out register). Any number of keys can be depressed and another can be recognised and entered into the FIFO. If a simultaneous depression occurs, the keys are recognised and entered according to the order the keyboard scan found them.

The PKI is also programmed for "Program Clock" which programs the appropriate keyboard scan and debounce times.

Program Clock:

All timing and multiplexing signals for the PKI are generated by an internal prescaler. The prescaler divides the external clock (pin 3) by a programmable integer. As illustrated in figure 4, bits PPPPP determine the value of this integer which can range from 2 to 31. For this application a integer of 20 was chosen. A integer/divisor of 20 and a clock of 12MHz will yield a frequency of 600kHz and will give a scan time of 1.67µsec and a debounce time of 3.3µsec.

Keyboard Program Clock Set (Address A0 =1)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	P	P	P	P	P

MSB LSB

3 4 H

PPPPP = Programmable integer

Figure 4.

4.3.5 THE DISPLAY.

Access to the display from the system bus is done through the programmable peripheral interface (PPI), which acts as an addressable latch, and control lines RRES, SERDATA and REGCLK (see Display hardware description for functions). To display the operational information

(Static Characters) on the screen the following sequence is pursued (See flowchart A ; Appendix A-10 and figure 5. on page 27 for illustrations).

- Depending on which message to be displayed, the address of the start of the array would be saved at a data location.
- The different software counters is initialised:

  Display column counter - CNTR2 = 7FH (120 columns)

  Display character counter - CNTR1 = 13H (20 char./display line)

  Array counter - CNTR3 = 05H (5 data positions/char.)
- The control lines are initialised:

  Serial data line - SERDATA = 'HIGH'

  Register reset line - RRES = 'HIGH'

  Register clock - REGCLK = 'HIGH'

  Serial data is applied to the first register (reg. no. 1), all registers (reg. no. 1 - 15) are enabled and the serial data is clocked through to the first output of register no. 1. The SERDATA line is deactivated and no further serial data is available on the shift registers.
- The data to be displayed on line 1 column 1 of the display is fetched from the array.
- The data is processed in the STATICTABLE, written to port A of the PPI and displayed.
- The data to be displayed on line 2 column 1 of the display is fetched from the array.
- The data is processed in the STATICTABLE, written to port B of the PPI and displayed.

- The register clock is deactivated and a delay routine is activated to enable the LED's to light up sufficiently.
- The data is cleared from the display.
- The column counter, CNTR2 is decremented and a question is asked if the contents of CNTR2 has reached 'ZERO'(end of display), if not the sequence is repeated. The second time round the second clock pulse will shift the data previously at output 1 of register no. 1 to output 2 of register no. 1. The data in the array is processed with that of the STATICTABLE and displayed. This sequence continues from output 1 of register no. 1 to output 8 of register no. 15 until the above mentioned question is true.
- All counters and registers are cleared, and the software checks if an interrupt has occurred, which vectors the software to the keyboard routine. If no keyboard interrupt has occurred the sequence is repeated and the message is displayed permanently.

The display is scanned from right to left, from column 1 to column 120. The scanning is so fast that it appears to the observer as if all the columns of the display is activated simultaneously. For different display modes the data is processed differently but uses the same display sequence.

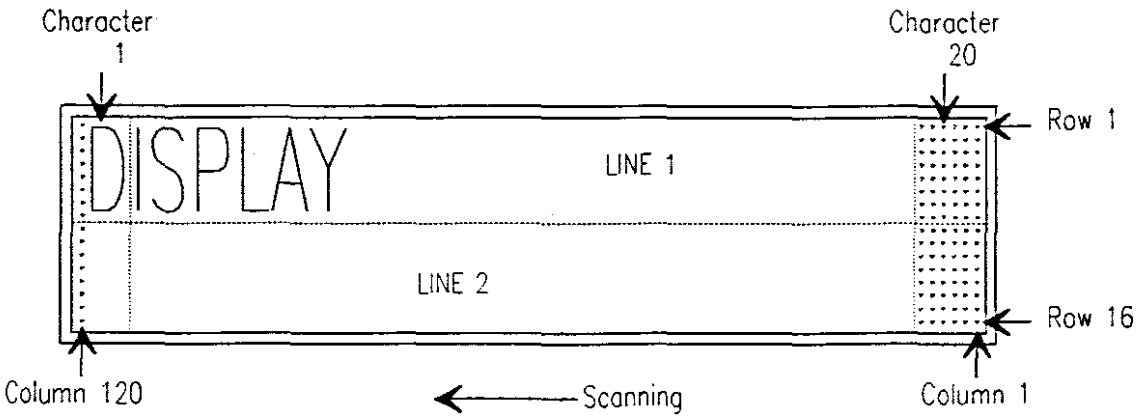


Figure 5.

4.3.6 POWER DOWN MODE

A black push-button is provided on the side of the display housing. When the button is depressed, a 'low' is applied to a interrupt ($\overline{\text{INT0}}$) pin of the controller. The microcontroller vectors to the interrupt routine and an instruction that sets the PCON.1 bit in the PCON reg. causes that to be the last instruction executed, before going into power down mode. In the power down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and special function registers. The controller's $\overline{\text{PSEN}}$ and ALE lines are held low, which disables any program execution. The only exit from power down mode for the 80C51 is a hardware reset. The reset redefines the SFR's, but does not change the data stored in the on-chip RAM. The reset that terminates power down also frees the oscillator and the controller starts executing code from EPROM. This function is used to reduce power consumption during times when the display is not used.

5.0 OPERATING INSTRUCTIONS.

5.1 Front Page of Manual.

TECHDISPLAY

EASY KEY WITH AUTOMODE

PROGRAMMING MANUAL

5.2 TABLE OF CONTENTS PAGE

5.3 SECTION 1.

Start-up and Easy Key Mode.

The Easy Key operation makes entering messages as simple as writing your own name. Just plug your TECHDISPLAY into any standard electrical outlet, insert the keyboard and switch the unit on. Then:

- The screen will display:

  ****** WELCOME TO ******
  ****** TECHSIGNS *******

  and after a delay, will disappear.

- The screen will now display:

  RECALL SAVED PROGRAM
  RUN/YES PROGRAM/NO

  1. Depress the 'PROG' key.

- The screen will display the following:

  EASY KEY MODE >ENTER
  SPECIAL KEYS > SPECIAL

  2. Depress the 'ENTER' key for Easy Key Mode.

- If a previous program has been entered the following message will appear, otherwise continue to step no. 4:

  *********** WARNING *************
  *****EXISTING PROGRAM******

- This screen disappears after a delay, and the screen will display:

OVERWRITE PROGRAM Y/YES V/VIEW N/NO

3. Depress the **'YES'** key to overwrite and continue the sequence, or the **'VIEW'** key to run the program or **'NO'** to restart the sequence.

- The next message will appear on the screen:

SELECT LETTER SIZE BIG OR SMALL

4. Depress the **'BIG'** key to select big letters or depress the **'SMALL'** key to select small letters.

Enter your messages (up to 200 characters for small letters and 100 characters for a big letter selection). If more than the allowable characters is entered the display will prompt you. When a mistake is made, use the **'BACKSPACE'** key to correct the error.

5. Depress the **'ENTER'** key after your program is entered.

That's all there is to it! Your messages will now be displayed from right to left on your screen.

NOTE: If at any stage a wrong option has been entered, depress the **'PREV SCRN'** key and the screen will return to the previous menu.

5.4 SECTION 2.

Auto Mode Programming:

A "mode" is the manner in which a message is displayed on the display. If you want the unit to automatically select the display and clear modes for you, you should choose the Auto Mode Function. The following sequence is used.

- The screen will display:

  ****** WELCOME TO ******
  ****** TECHSIGNS *******

  and after a delay, will disappear.

- The screen will now display:

  RECALL SAVED PROGRAM
  RUN/YES PROGRAM/NO

  1. Depress the 'PROG' key to enter a program.

- The screen will display the following:

  EASY KEY MODE >ENTER
  SPECIAL KEYS > SPECIAL

  2. Depress the 'SPECIAL' key.

- The screen will prompt the following message:

  AUTO MODE > ENTER
  SPECIAL MODE > SPECIAL

  3. Depress the 'ENTER' key to select Auto Mode.

- The next screen will appear:

SELECT PROGRAM NUMBER 1 - 9

4. Select a program number 1 to 9

- If a previous program was entered at this program number location, the following message will appear, otherwise continue to step no. 4:

********* WARNING ********** ***EXISTING PROGRAM****

- This screen disappears after a delay, and the screen will display:

OVERWRITE PROGRAM Y/YES V/VIEW N/NO

5. Depress the 'Y' key to overwrite the program and continue the sequence, or the 'V' key to run the program or 'N' to restart the sequence.

- If YES has been entered, the display prompts the following:

CHOOSE DISPLAY MODE DUAL OR SINGLE

6. Depress the 'DUAL' key for two line display and the 'SINGLE' key for one line display. The next screen is only used for single line displaying. If the operator depressed the 'DUAL' key skip the next sequence.

- The next message will appear on the screen:

SELECT LETTER SIZE BIG OR SMALL

7. Depress the **'BIG'** key to select big letters or depress the **'SMALL'** key to select small letters.

- The next screen is displayed momentarily:

ENTER FROM KEYBOARD THE NEXT SCREEN

8. Enter your data onto the screen. When a mistake is made, use the **'BACKSPACE'** key to correct the error. When data is entered, depress the **'ENTER'** key.

- The following screen will display:

SELECT NEXT SCREEN ENTER/Y PROGRAM/SAVE

9. If the **'ENTER'** key is depressed the operator is able to enter another screen, (number of screens limited to the amount of screens allocated to each program).If more than the allowable screens is requested the display will prompt the operator. When the **'PROGRAM'** key is pressed the program will be saved and the next screen will prompt the operator.

- The screen will now display:

  RECALL SAVED PROGRAM
  RUN/YES PROGRAM/NO

  The sequence can now be repeated with another program, choose a different sequence or to run the program previously entered.

NOTE: If at any stage a wrong option has been entered, depress the **'PREV SCRN'** key and the screen will return to the previous menu.

To run a program refer to section 4.

5.5 SECTION 3.

Advanced Special Key Programming:

If the operator wishes to enter specific modes for messages, use the following procedures:

- The screen will display:

**** WELCOME TO ****
**** TECHSIGNS *****

and after a delay, will disappear.

- The screen will now display:

RECALL SAVED PROGRAM
RUN/YES PROGRAM/NO

1. Depress the 'PROG' key to enter a program.

- The screen will display the following:

EASY KEY MODE >ENTER
SPECIAL KEYS > SPECIAL

2. Depress the 'SPECIAL' key.

- The screen will prompt the following message:

AUTO MODE > ENTER
SPECIAL MODE > SPECIAL

3. Depress the 'SPECIAL' key to select Auto Mode.

- The next screen will appear:

SELECT PROGRAM
NUMBER 1 - 9

4. Select a program number 1 to 9

- If a previous program was entered at this program number location, the following message will appear, otherwise continue to step no. 4:

********* WARNING **********
EXISTING PROGRAM*

- This screen disappears after a delay, and the screen will display:

OVERWRITE PROGRAM
Y/YES V/VIEW N/NO

5. Depress the 'Y' key to overwrite the program and continue the sequence, or the 'V' key to run the program or 'N' to restart the sequence.

- If YES has been entered, the display prompts the following:

CHOOSE DISPLAY MODE
DUAL OR SINGLE

6. Depress the 'DUAL' key for two line display and the 'SINGLE' key for one line display. The next screen is only used for single line displaying. If the operator depressed the 'DUAL' key skip the next sequence.

- The next message will appear on the screen:

SELECT LETTER SIZE
BIG OR SMALL

7. Depress the 'BIG' key to select big letters or depress the 'SMALL' key to select small letters.

- The next screen is displayed momentarily:

ENTER FROM KEYBOARD THE NEXT SCREEN

8. Enter your data onto the screen. When a mistake is made, use the 'BACKSPACE' key to correct the error. When data is entered, depress the 'ENTER' key.

- The next screen will appear:

SELECT SCREEN MODE S/T/C/L/R/I/O/B

9. If the operator depresses any of the above mentioned characters, the corresponding display mode will be saved. Refer to section 6. for descriptions of the display modes and what they do.

- The following screen will display:

SELECT CLEAR MODE N/T/B/C/S/SPEC. FUNC.

10. If the operator depresses any of the above mentioned characters, the corresponding clear screen mode will be saved. Refer to section 6.

for descriptions of the clear screen modes and what they do.

- If the **'SPEC. FUNC.'** key was depressed the following screen will appear. If another key was depressed this sequence should be ignored.

SELECT GRAFICS NO. 1 - 3

11. Select a graphics clear mode. Refer to section 6. for descriptions of the graphics clear screen modes and what they do.

- The following screen will display:

SELECT NEXT SCREEN ENTER/Y PROGRAM/SAVE

12. If the **'ENTER'** key is depressed the operator is able to enter another screen, (number of screens limited to the amount of screens allocated to each program).If more than the allowable screens is requested the display will prompt the operator. When the **'PROGRAM'** key is pressed the program will be saved and the next screen will prompt the operator.

- The screen will now display:

RECALL SAVED PROGRAM RUN/YES PROGRAM/NO

The sequence can now be repeated with another program, choose a different sequence or run the program previously entered.

NOTE: If at any stage a wrong option has been entered, depress the **'PREV SCRN'** key and the screen will return to the previous menu.

To run a program refer to section 4.

5.6 SECTION 4.

Program Execution.

If a program has been entered, and the operator wants to execute the program, the following sequence has to be pursuit.

- The screen will now display:

RECALL SAVED PROGRAM
RUN/YES PROGRAM/NO

  The operator must depress the **'RUN'** key

- The screen will now display:

ENTER PROGRAM
NUMBER 1 - 10

  The operator must select the program he previously entered. At the depression of the program number the program starts to execute. If another program is to be executed, the operator must depress the **'ESC'** key to stop the execution and repeat the above sequence.

NOTE:

If no program is present at the program number selected, the display will return to the previous screen.

5.7 SECTION 5.

5.7.1 LED Test:

If the operator suspects faulty LED's the following sequence has to be followed.

- The screen will display:

  ****** WELCOME TO ******
  ****** TECHSIGNS *******

  and after a delay, will disappear.

- The screen will now display:

  RECALL SAVED PROGRAM
  RUN/YES PROGRAM/NO

  1. Depress the **'PREV SCRN'** key to enable LED-test.

5.7.2 Troubleshooting:

If difficulties are encountered at any stage during the programming or execution of a program, the operator has to depress the **'RED RESET PUSH-BUTTON'** on the side of the display housing. If further difficulties are encountered please contact your agent.

5.7.3 Power Down:

The TECHDISPLAY provides a unique power down facility for applications where power consumption is critical.
Power Down is done by depressing the **'BLACK PUSH-BUTTON'** on the side of the display housing.

5.8 SECTION 6.

5.8.1 Display Modes.

- This screen prompts the operator to enter a screen mode:

SELECT SCREEN MODE S/T/C/L/R/I/O/B

If the operator depresses any of the above mentioned characters, the corresponding display mode will be saved. Refer to table below for definitions of screen modes.

S = Static Characters

T = Characters enter the screen from TOP and BOTTOM.

C = Characters enter the screen from the CENTRE towards the top and bottom of the screen.

L = Characters enter the screen from the RIGHT and move to the LEFT across the screen.

R = Characters enter the screen from the LEFT and move to the RIGHT across the screen.

I = Characters enter the screen from the INSIDE towards the OUTSIDES of the screen.

O = Characters enter the screen from the OUTSIDES towards the INSIDE of the screen.

B = Characters segments enter the screen from the right and build the characters to be displayed towards the right.

5.8.2 Clear Mode.

- The screen prompts the operator to enter a clear mode:

SELECT CLEAR MODE N/T/B/C/S/SPEC. FUNC.

If the operator depresses any of the above mentioned characters, the corresponding clear screen mode will be saved. Refer to table below for definitions of clear modes.

N = The screen is cleared by displaying the next screen.

T = The screen is cleared from the TOP of the display.

B = The screen is cleared from the BOTTOM of the display.

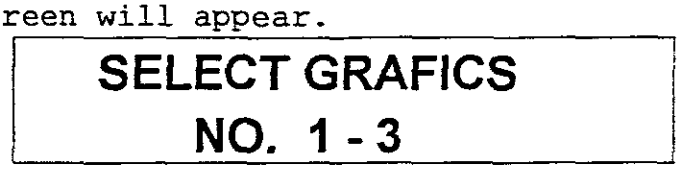
C = The screen is cleared from the CENTRE towards the top and bottom of the display.

S = This screen is cleared from the sides towards the inside of the display.

Spec. Func. = The screen is cleared by graphics.

5.8.3 Graphics Mode.

- If the **'SPEC.. FUNC.'** key was depressed the following screen will appear.



The operator selects a number 1-3 to select a graphical clear mode picture.

1 = The PUCMAN moves from right to left across the screen.

2 = The DUCK moves from right to left across the screen.

3 = The CATERPILLAR moves from right to left across the screen.

6. PROBLEMS ENCOUNTERED.

The major problem encountered during the design of this project, concerned the weak illumination capabilities of the display. This was caused by the fast scanning of the display, which caused the LED's not to switch on to their maximum capability.
This problem was overcome by using a different method of driving the LED array, brighter LED's and a software delay in the display routine.

The second problem encountered was the difficulty in writing the assembler program for this project. The program for this thesis resulted in a extremely long program, which gave enormous problems when debugging. This problem prevailed due to the time that would have been lost if a high-level language had to be learned and the completed amount of software already written.

During the feasibility study and system analysis stages of this project difficulty was found in acquiring general information regarding electronic message display's. This was due to the lack of literature available on message display's and the reluctance of the few private companies to assist in acquiring such information, due to their fears of future competition and a reduced market share.

7. CONCLUSION.

The electronic message display that was designed, developed and built for this project, surpasses the original requirements as laid down by the Electrical Engineering Department of the Cape Technikon. This is largely due to the amount of research that went into the designing and development of the electronic message display.

The time and effort spent on designing and developing this project has lead to an excellently functioning display unit. The need for such a display at the Cape Technikon does not end there, as this display is a useful tool for use in the advertising industry and small business community.

The experience and knowledge gained will contribute to the success of designing future display's and other projects.

8. FUTURE SUGGESTIONS.

Software development is an on-going process that is susceptible to future enhancements. A possible enhancement, is to write the software of this project in a high-level language, to increase flexibility of the project and to simplify the debugging of the software.

Because of the problems encountered with the brightness of the display, the suggestion is to change the scanning of the display. Using separate programmable LED drivers which need no scanning, to each access a 8 x 8 matrix of LED's, and therefore will increase the brightness of the display. It would then be possible to program each LED separately, enabling the programmer to graphically enhance the display modes, and for the operator to design his own graphics on the display screen.

A possibility of marketing a future upgraded version of this project must strongly be considered, due to the advancement of electronic communications in the advertising industry.

9. BIBLIOGRAPHY.

Abel, Peter. 1987. **IBM PC Assembler Language and Programming.** Prentice-Hall International Editions.

Dallas Semiconductors Corporation. 1992-1993. **Product Data Book.** Dallas, Texas.

Intel Corporation. 1990. **Embedded Applications.** Intel Literature Sales, Santa Clara.

Intel Corporation. 1991. **Peripheral Components.** Intel Literature Sales, Santa Clara.

Intel Corporation. 1992. **Embedded Microcontrollers and Processors Vol. 1.** Intel Literature Sales, Santa Clara.

Motorola Semiconductors. 1981. *Power.* Motorola Inc.

National Semiconductor Corporation. 1982. **Transistor Data book.** Santa Clara, California.

National Semiconductor Corporation. 1984. **Logic Data book Vol. 1.** Santa Clara, California.

National Semiconductor Corporation. 1988. **CMOS Logic Data book.** Santa Clara, California.

National Semiconductor Corporation. 1988. **Linear Data book**. Santa Clara, California.

National Semiconductor Corporation. 1988. **Memory Components Handbook**. Santa Clara, California.

Spectratech. March - June 1992. **RS Catalogue**. Elvey International Group.

Texas Instruments. 1989. **ALS/AS Logic Data Book**. Texas Instruments.

Unidux. **DICE-8051 User's Manual**. Unidux Inc.

10. LIST OF ABBREVIATIONS:

µF	Microfarrad
µsec.	Microseconds
A0-A15	Address lines
AD0- AD7	Address/Data lines
add.	Address
Amp	Ampere
ASCII	American Standard Character Information Interchange
CLK	Clock
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DC	Direct Current
DIP	Dual Inline Package
e.g.	example
EMD	Electronic Message Display
EPROM	Electrical Programmable Read Only Memory
FIFO	First in First out
GND	Gound
H	Hexidesimal
I/O	Input/Output
I.C.	Integrated Circuit
KBytes	Kilo Bytes
KΩ	Kilo Ohms
LED	Light Emitting Diode
mA	milli Ampere
mcd	millicandellas
MHz	Megga Hertz

mm	millimeter
no.	number
ns	nanoseconds
OE	Output Enable
pF	pico Farrad
PKI	Programmable Keyboard Interupt
PPI	Programmable Peripheral Interface
PWRDWN	Power Down
RAM	Random Access Memory
RD	Read
reg.	registor
REGCLK	Register Clock
RRES	Register Reset
RST	Reset
SERDATA	Serial Data
TTL	Transistor Transistor Logic
V	Volts
VA	Volts/Ampere
VAC	Voltage Alternating Current
VCC	Supply Voltage
VPP	Programming Voltage
WR	Write

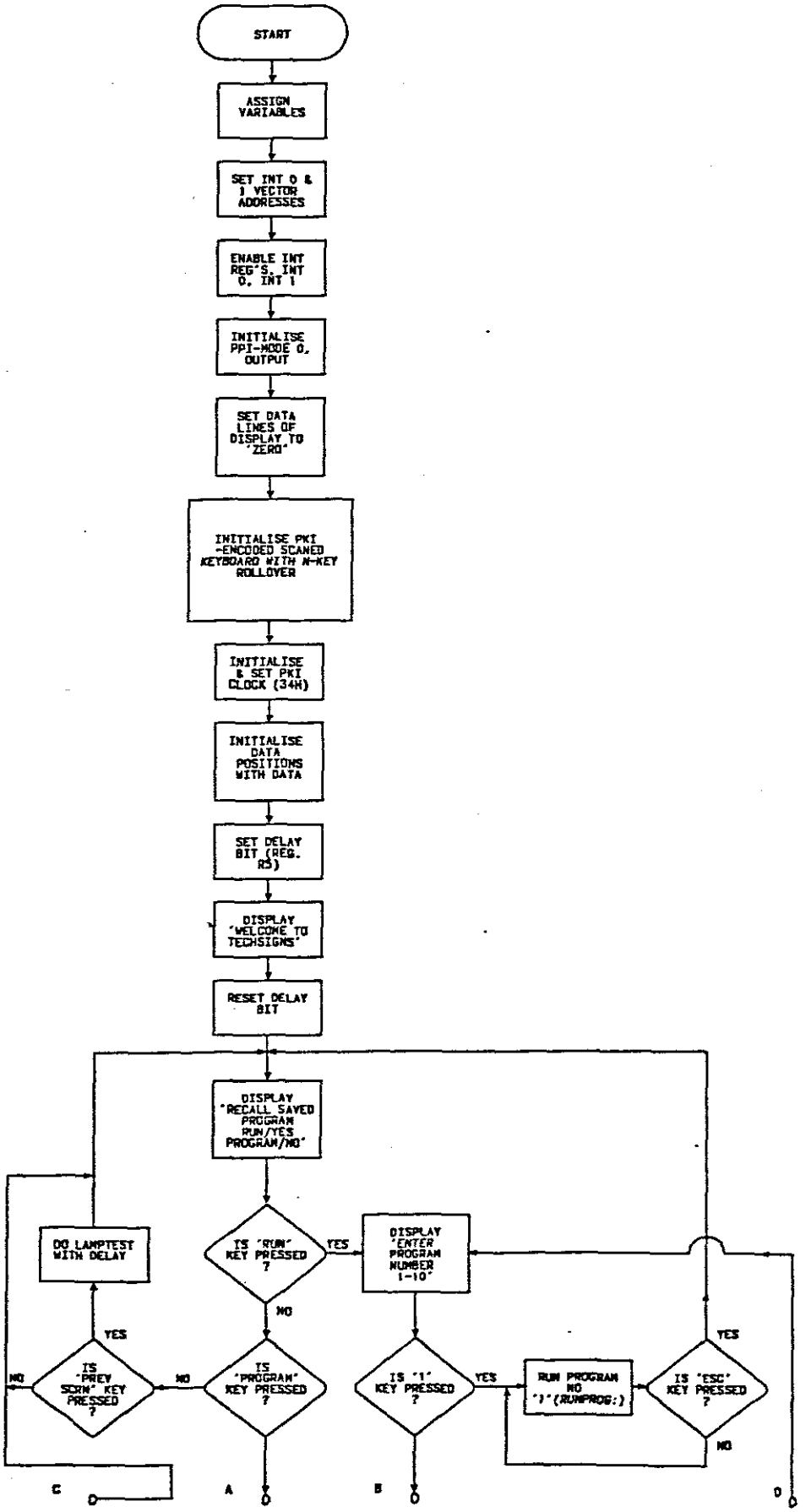
Appendix

APPENDIX A.

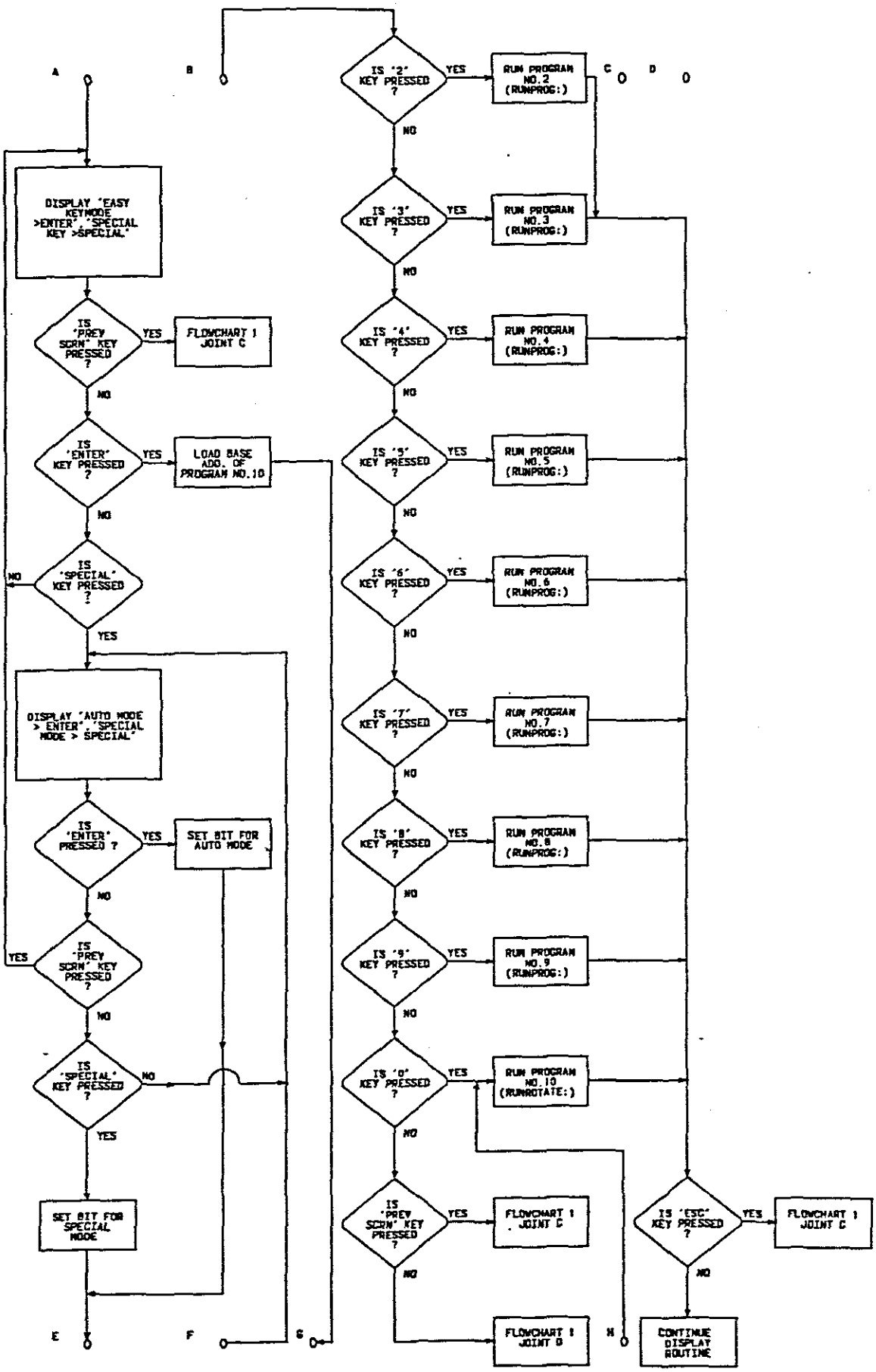
FLOWCHARTS.

INDEX.

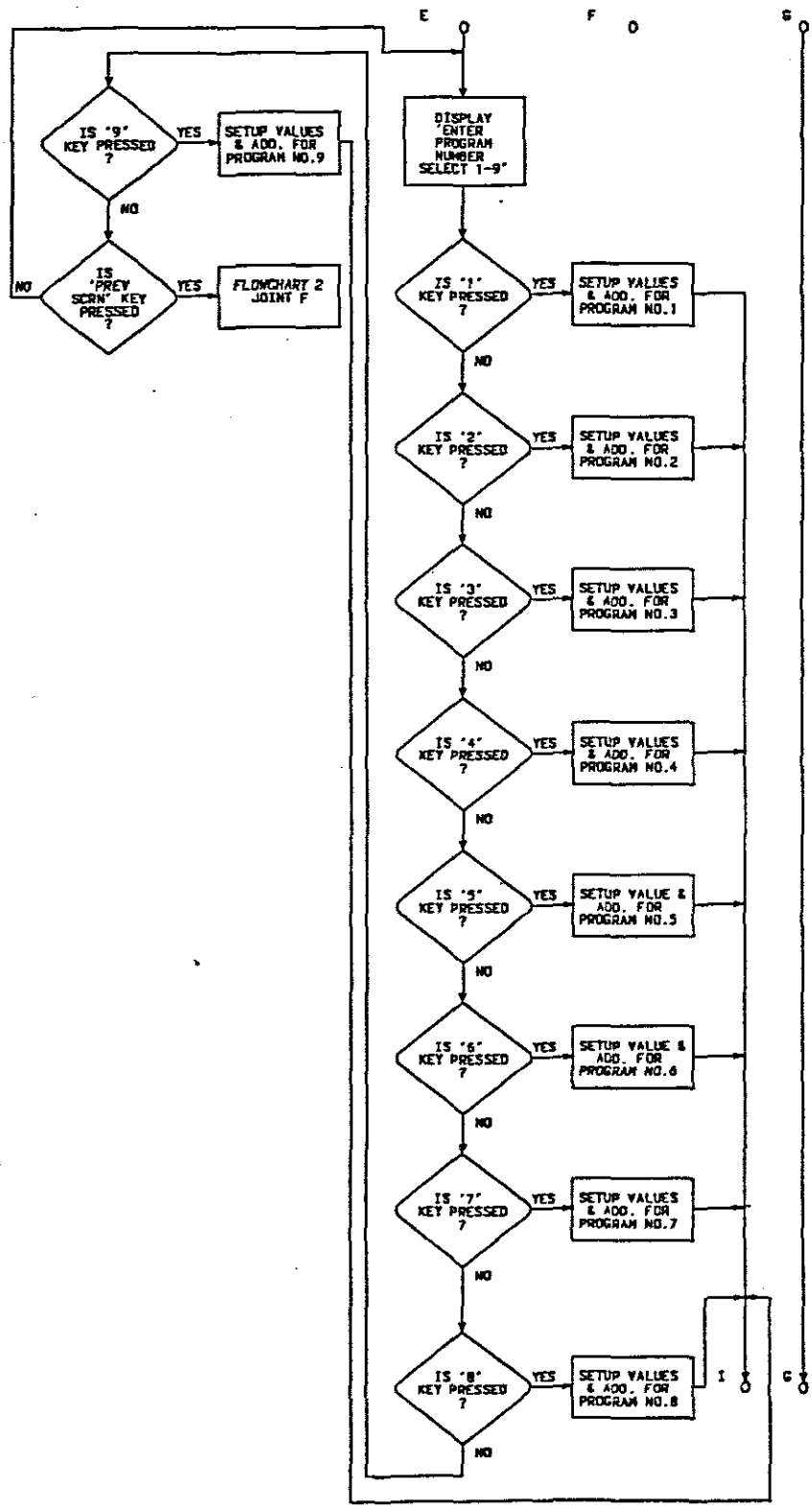
MAIN PROGRAM
FLOWCHART 1



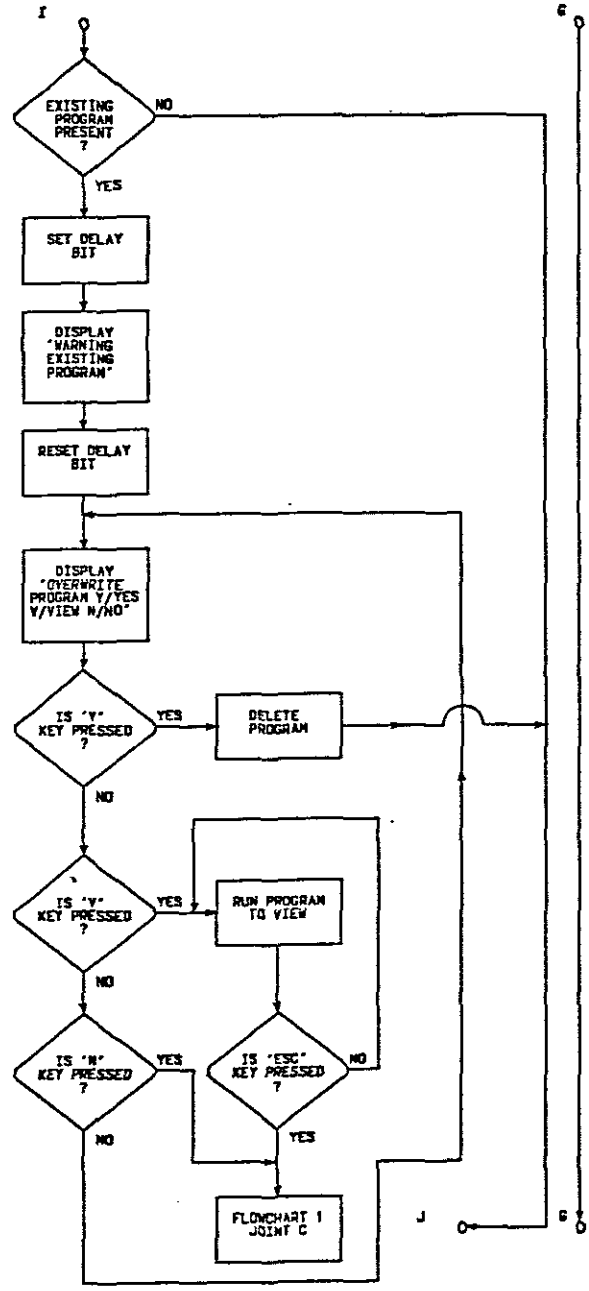
MAIN PROGRAM
FLOWCHART 2

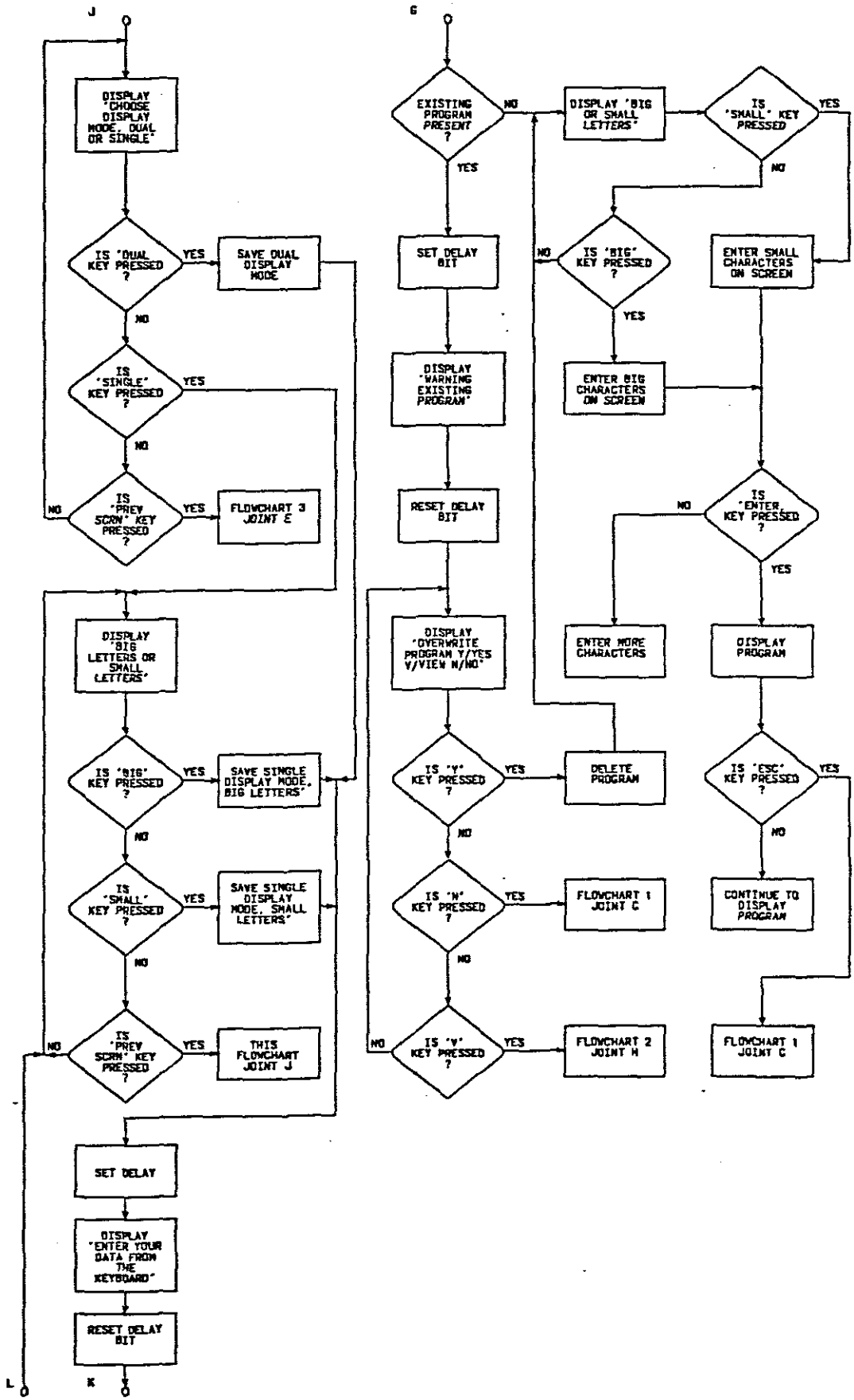


MAIN PROGRAM
FLOWCHART 3

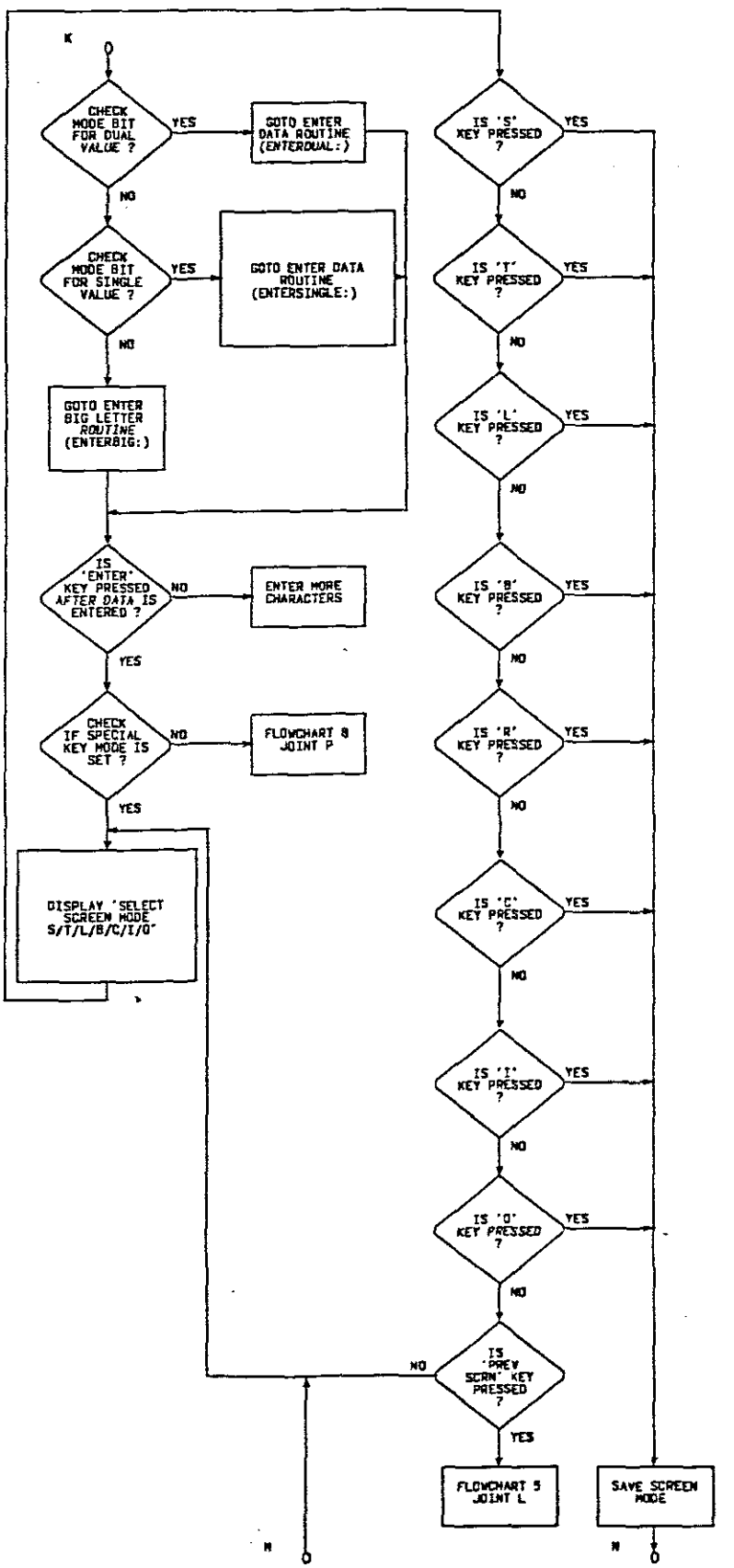


MAIN PROGRAM
FLOWCHART 4

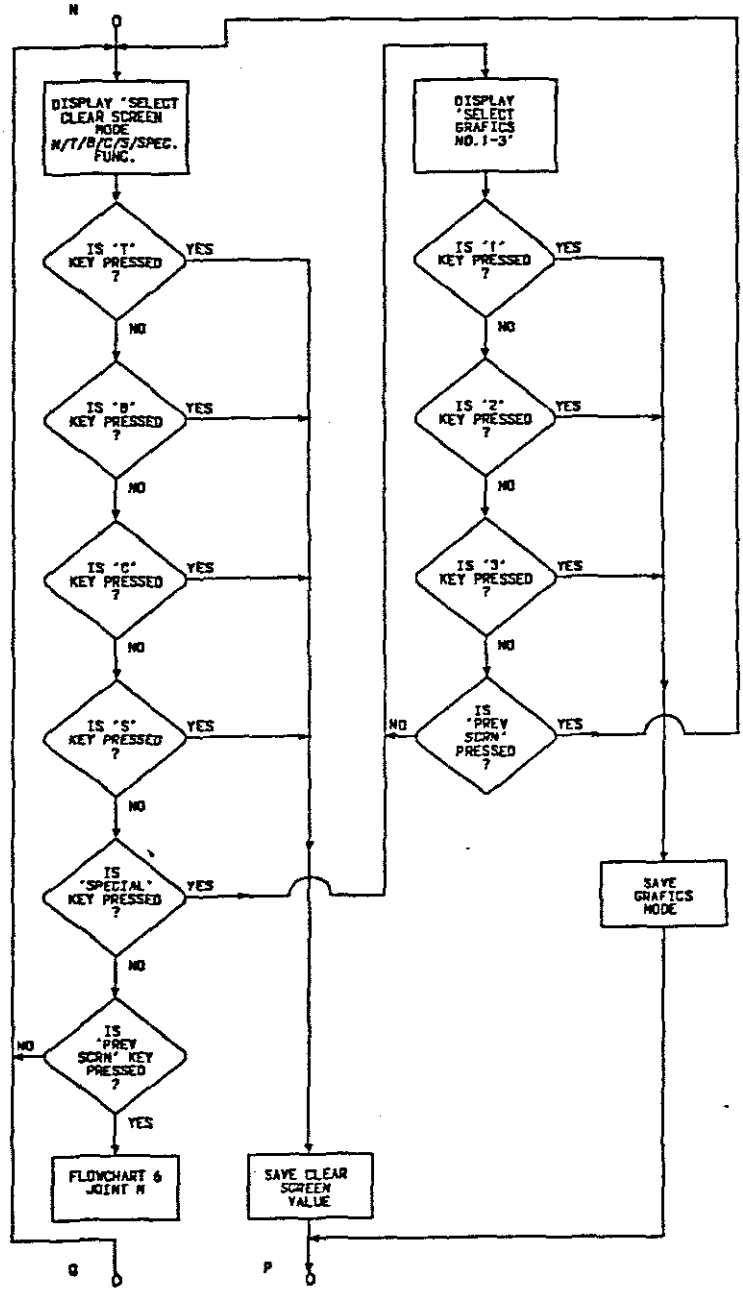



J
DISPLAY 'CHOOSE DISPLAY MODE, DUAL OR SINGLE'
IS 'DUAL' KEY PRESSED ?
YES
SAVE DUAL DISPLAY MODE
NO
IS 'SINGLE' KEY PRESSED ?
YES
NO
IS 'PREV SCRN' KEY PRESSED ?
NO
YES
FLOWCHART 3 JOINT E
DISPLAY 'BIG LETTERS OR SMALL LETTERS'
IS 'BIG' KEY PRESSED ?
YES
SAVE SINGLE DISPLAY MODE, BIG LETTERS'
NO
IS 'SMALL' KEY PRESSED ?
YES
SAVE SINGLE DISPLAY MODE, SMALL LETTERS'
NO
IS 'PREV SCRN' KEY PRESSED ?
NO
YES
THIS FLOWCHART JOINT J
SET DELAY
DISPLAY 'ENTER YOUR DATA FROM THE KEYBOARD'
RESET DELAY BIT
L
K
G
EXISTING PROGRAM PRESENT ?
NO
YES
SET DELAY BIT
DISPLAY 'WARNING EXISTING PROGRAM'
RESET DELAY BIT
DISPLAY 'OVERWRITE PROGRAM Y/YES V/VIEW N/NO'
IS 'Y' KEY PRESSED ?
YES
DELETE PROGRAM
NO
IS 'N' KEY PRESSED ?
YES
FLOWCHART 1 JOINT C
NO
IS 'V' KEY PRESSED ?
NO
YES
FLOWCHART 2 JOINT H
DISPLAY 'BIG OR SMALL LETTERS'
IS 'SMALL' KEY PRESSED
YES
NO
IS 'BIG' KEY PRESSED ?
NO
YES
ENTER BIG CHARACTERS ON SCREEN
ENTER SMALL CHARACTERS ON SCREEN
IS 'ENTER' KEY PRESSED ?
NO
YES
ENTER MORE CHARACTERS
DISPLAY PROGRAM
IS 'ESC' KEY PRESSED ?
YES
NO
CONTINUE TO DISPLAY PROGRAM
FLOWCHART 1 JOINT C

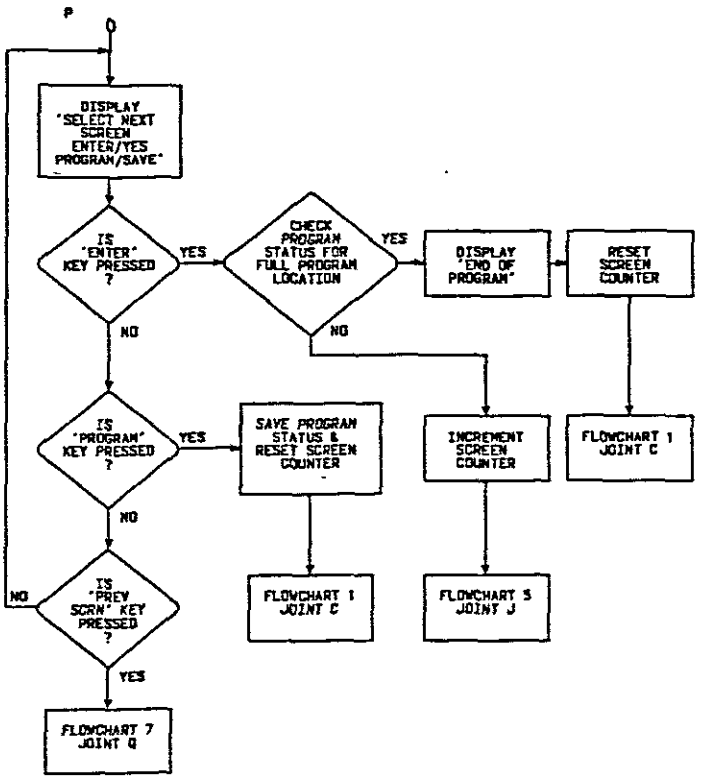
MAIN PROGRAM
FLOWCHART 6



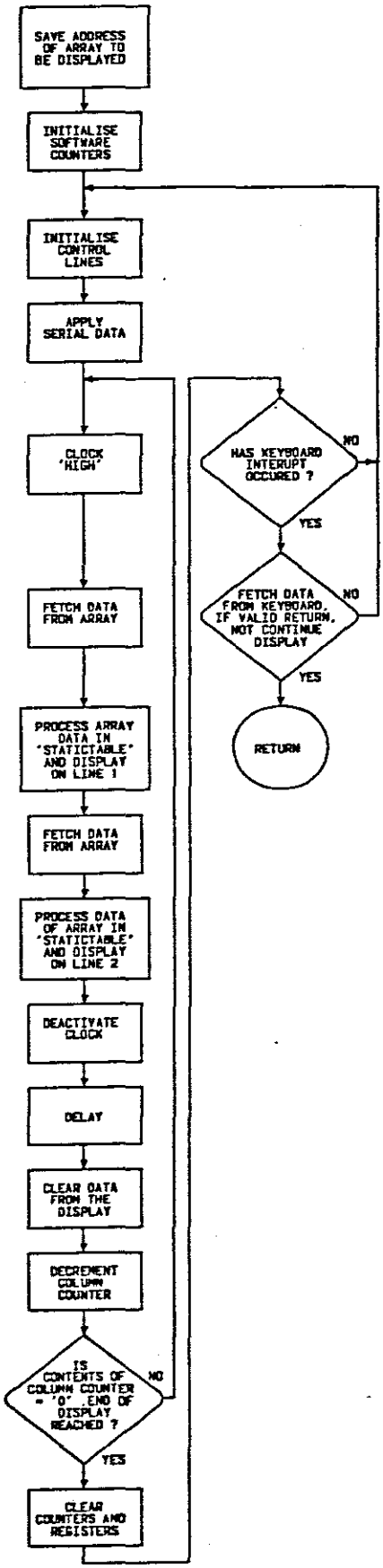
MAIN PROGRAM
FLOWCHART 7



MAIN PROGRAM
FLOWCHART 8



FLOWCHART A
DISPLAY OF OPERATIONAL INFORMATION
STATIC1:



APPENDIX B.

PROGRAM LISTING.

INDEX.

PROGRAM LISTING.

```
;TITLE                          ELECTRONIC MESSAGE DISPLAY
;SUBTITLE                       ASSEMBLER PROGRAM
;                               WRITTEN BY N.J. SIMON
;*****************************************************************
                        DATASEGMENT
;*****************************************************************
EPROM_H         DATA    30H
EPROM_L         DATA    31H
RAM_H           DATA    32H
RAM_L           DATA    33H
CNTR1           DATA    34H
CNTR2           DATA    35H
RAMVAL          DATA    36H
CURS_H          DATA    37H
CURS_L          DATA    38H
CNTRL           DATA    39H
OFFSET          DATA    3AH
CNTRC           DATA    3BH
DPHSAVE         DATA    3CH
NR1_H           DATA    3DH
NR2_H           DATA    3EH
NR3_H           DATA    3FH
NR4_H           DATA    40H
NR5_H           DATA    41H
NR6_H           DATA    42H
NR7_H           DATA    43H
NR8_H           DATA    44H
NR9_H           DATA    45H
NR10_H          DATA    46H
NR_H            DATA    47H
NR_L            DATA    48H
PROGNUM         DATA    49H
MODEDISP        DATA    4AH
MODESCRN        DATA    4BH
SCRNCNTR        DATA    4CH
CNTR3           DATA    4DH
LOOP1           DATA    4EH
LOOP2           DATA    4FH
LOOP3           DATA    50H
LOOP4           DATA    51H
EVENT           DATA    52H
EVENT2          DATA    53H
WORKVAL         DATA    54H
RVAL            DATA    55H
VALUE           DATA    56H
PROG_H          DATA    57H
PROG_L          DATA    58H
CLRSCRN         DATA    59H
CNTR4           DATA    60H
DISPLAY         DATA    61H
PROG1           DATA    62H
PROG2           DATA    63H
PROG3           DATA    64H
PROG4           DATA    65H
PROG5           DATA    66H
PROG6           DATA    67H
PROG7           DATA    68H
PROG8           DATA    69H
PROG9           DATA    70H
SCRNCOUNT       DATA    71H
PIC1_H          DATA    72H
```

```
PIC1_L          DATA    73H
PIC2_H          DATA    74H
PIC2_L          DATA    75H
PIC3_H          DATA    76H
PIC3_L          DATA    77H
PIC             DATA    78H
AUTO            DATA    79H
CLEAR           DATA    7AH
PCON            DATA    87H

;*************************************************************
                        CODESEGMENT
;*************************************************************
            ORG     00H
            LJMP    INIT

            ORG     03H
            LJMP    KEYREAD

            ORG     13H
            LJMP    PWRDWN

;*************************************************************;
MAIN PROGRAM
;*************************************************************
;-------------------------------------------------------------
;         INITIATE DSEG. ,PPI,PKI,DISPLAY REGISTERS
;-------------------------------------------------------------
            ORG     30H
INIT:
            MOV     PCON,#00H
            MOV     IE,#00H
            SETB    EA              ;ENABLE INTREG.
            SETB    EX0             ;ENABLE INT0
            SETB    EX1             ;ENABLE INT1
            MOV     TCON,#00H
            SETB    IT0             ;ENABLE INTERUPT '0'
            SETB    IT1             ;ENABLE INTERUPT '1'
            MOV     P1,#00H         ;RESET DISPLAY

            MOV     DPTR,#8003H     ;PHERIPHERAL INTERFACE ADD.
            MOV     A,#80H
            MOVX    @DPTR,A         ;INIT PPI TO MODE 0,OUTPUT
            MOV     DPTR,#8001H
            MOV     A,#00H          ;SET DATA LINES
            MOVX    @DPTR,A         ;       OF DISPLAY TO
            DEC     DPL             ;               ALL ZERO'S
            MOVX    @DPTR,A

            MOV     DPTR,#0C001H    ;KEYBOARD CONTROLLER ADD.
            MOV     A,#02H
            MOVX    @DPTR,A         ;SET MODE OF 8279
            MOV     A,#34H
            MOVX    @DPTR,A         ;PROGRAM CLOCK OF 8279

            MOV     CNTR1,#77H
            MOV     CNTR2,#4FH
            MOV     NR1_H,#40H
            MOV     NR2_H,#44H
            MOV     NR3_H,#48H
            MOV     NR4_H,#4CH
            MOV     NR5_H,#52H
```

```
            MOV     NR6_H,#58H
            MOV     NR7_H,#5EH
            MOV     NR8_H,#66H
            MOV     NR9_H,#6EH
            MOV     NR10_H,#76H
            MOV     PROG1,#04H
            MOV     PROG2,#04H
            MOV     PROG3,#04H
            MOV     PROG4,#06H
            MOV     PROG5,#06H
            MOV     PROG6,#06H
            MOV     PROG7,#08H
            MOV     PROG8,#08H
            MOV     PROG9,#08H
            MOV     CNTRL,#00H
            MOV     CNTRC,#00H
            MOV     PROGNUM,#00H
            MOV     MODEDISP,#00H
            MOV     MODESCRN,#00H
            MOV     SCRNCNTR,#00H
            MOV     AUTO,#00H
            MOV     CLEAR,#00H

;-------------------------------------------------------------
;MAIN PROGRAM:    DISPLAY AND KEYBOARD FUNCTIONS
;-------------------------------------------------------------
CONT5:
            MOV     R5,#01H          ;SET DELAY
            MOV     DPTR,#SCNT1      ;'**** WELCOME TO ****'
            LCALL   STATIC1          ;'**** TECHSIGNS *****'
            MOV     R5,#00H          ;RESET DELAY
CONT7:
            MOV     DPTR,#SCNT2      ;'RECALL SAVED PROGRAM'
            LCALL   STATIC1          ;'  RUN/YES PROGRAM/NO'
            CJNE    R7,#0F9H,CONT11 ;CHECK FOR 'RUN'KEY F3H
CONT9:
            MOV     DPTR,#SCNT3      ;'ENTER PROGRAM NUMBER'
            LCALL   STATIC1          ;'SELECT 1-10
            CJNE    R7,#0C4H,RUN2
            MOV     RAM_H,NR1_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
CONT11:
            LJMP    CONT6
RUN2:
            CJNE    R7,#0CCH,RUN3
            MOV     RAM_H,NR2_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN3:
            CJNE    R7,#0D4H,RUN4
            MOV     RAM_H,NR3_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN4:
            CJNE    R7,#0C5H,RUN5
            MOV     RAM_H,NR4_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
```

```
RUN5:
            CJNE    R7,#0CDH,RUN6
            MOV     RAM_H,NR5_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN6:
            CJNE    R7,#0D5H,RUN7
            MOV     RAM_H,NR6_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN7:
            CJNE    R7,#0C6H,RUN8
            MOV     RAM_H,NR7_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN8:
            CJNE    R7,#0CEH,RUN9
            MOV     RAM_H,NR8_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN9:
            CJNE    R7,#0D6H,RUN10
            MOV     RAM_H,NR9_H
            MOV     RAM_L,#00H
            LCALL   RUNPROG
            LJMP    CONT7
RUN10:
            CJNE    R7,#0CFH,RUN11
RUN12:
            MOV     RAM_H,NR10_H
            MOV     RAM_L,#00H
            MOV     DPTR,#7FF0H
            MOVX    A,@DPTR
            CJNE    A,#0FDH,CONT67
            LCALL   RUN_ROTATE
            LJMP    CONT7
CONT67:
            LCALL   RUN_ROTATEBIG
            LJMP    CONT7
RUN11:
            CJNE    R7,#0FBH,CONT13 ;CHECK FOR 'PREV SCRN' KEY
            LJMP    CONT7
CONT13:
            LJMP    CONT9
CONT21:
            CJNE    R7,#0FBH,CONT74 ;IS 'PREV. SCRN 'KEY
                                     PRESSED
            MOV     R5,#01H         ;SET DELAY BIT
            MOV     DPTR,#SCNT17    ;LAMPTEST
            LCALL   STATIC1
            MOV     R5,#00H
CONT74:
            LJMP    CONT7
CONT6:
            CJNE    R7,#0F1H,CONT21 ;CHECK FOR 'PROGRAM' KEY
                                     ;0F1H
CONT47:
            MOV     DPTR,#SCNT12    ;'EASY KEY MODE >ENTER'
            LCALL   STATIC1         ;'SPECIAL KEY >SPECIAL'
```

```
        CJNE    R7,#0F8H,CONT46 ;? 'ENTER' KEY PRESSED
        LJMP    NO10
CONT46:
        CJNE    R7,#0FBH,CONT48 ;? PREV. SCRN KEY PRESSED
        LJMP    CONT7
CONT48:
        CJNE    R7,#0F0H,CONT47 ;? 'SPECIAL FUNCTION'KEY
                                   PRESSED
CONT53:
        MOV     DPTR,#SCNT13    ;' AUTO MODE > ENTER  '
        LCALL   STATIC1         ;'SPECIAL MODE>SPECIAL'
        CJNE    R7,#0F8H,CONT52 ;? 'ENTER' PRESSED
        MOV     DISPLAY,#01H    ;SET DATA POSITION FOR
                                 ;AUTO MODE
        LJMP    NO1
CONT52:
        CJNE    R7,#0F0H,CONT55 ;? 'SPECIAL FUNCTION' KEY
                                 ;PRESSED
        MOV     DISPLAY,#0FFH   ;SET DATA FOR SPECIAL MODE
        LJMP    NO1
CONT55:
        CJNE    R7,#0FBH,CONT53 ;? 'PREV. SCRN KEY PRESSED
        LJMP    CONT47
NO1:
        MOV     VALUE,#0FFH     ;SET FOR PREV SCRN
        MOV     DPTR,#SCNT16     ;'ENTER PROGRAM NUMBER'
        LCALL   STATIC1         ;'SELECT 1-9'
        CJNE    R7,#0C4H,NO2    ;NO1
        MOV     PROGNUM,#01H
        MOV     SCRNCOUNT,PROG1
        MOV     RAM_H,NR1_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO2:
        CJNE    R7,#0CCH,NO3    ;NO2
        MOV     PROGNUM,#02H
        MOV     SCRNCOUNT,PROG2
        MOV     RAM_H,NR2_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO3:
        CJNE    R7,#0D4H,NO4    ;NO3
        MOV     PROGNUM,#03H
        MOV     SCRNCOUNT,PROG3
        MOV     RAM_H,NR3_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO4:
        CJNE    R7,#0C5H,NO5    ;NO4
        MOV     PROGNUM,#04H
        MOV     SCRNCOUNT,PROG4
        MOV     RAM_H,NR4_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO5:
        CJNE    R7,#0CDH,NO6
        MOV     PROGNUM,#05H
        MOV     SCRNCOUNT,PROG5
        MOV     RAM_H,NR5_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO6:
        CJNE    R7,#0D5H,NO7
```

```
        MOV     PROGNUM,#06H
        MOV     SCRNCOUNT,PROG6
        MOV     RAM_H,NR6_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO7:
        CJNE    R7,#0C6H,NO8
        MOV     PROGNUM,#07H
        MOV     SCRNCOUNT,PROG7
        MOV     RAM_H,NR7_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO8:
        CJNE    R7,#0CEH,NO9
        MOV     PROGNUM,#08H
        MOV     SCRNCOUNT,PROG8
        MOV     RAM_H,NR8_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO9:
        CJNE    R7,#0D6H,NO11
        MOV     PROGNUM,#09H
        MOV     SCRNCOUNT,PROG9
        MOV     RAM_H,NR9_H
        MOV     RAM_L,#00H
        LJMP    CONT14
NO10:
        MOV     PROGNUM,#0AH
        MOV     RAM_H,NR10_H
        MOV     RAM_L,#00H
        LJMP    CONT50
NO11:
        CJNE    R7,#0FBH,CONT30 ;? PREV SCRN KEY PRESSED
        LJMP    CONT53
CONT30:
        LJMP    NO1
CONT14:
        MOV     NR_H,RAM_H
        MOV     DPH,RAM_H
        MOV     DPL,#0F0H
        MOVX    A,@DPTR
        MOV     SCRNCNTR,#00H
        JZ      CONT35          ;? EXISTING PROGRAM
        MOV     DPTR,#SCNT9
        MOV     R5,#01H         ;SET DELAY
        LCALL   STATIC1         ;'WARNING EXISTING
                                 PROGRAM'
        MOV     R5,#00H         ;RESET DELAY
CONT38:
        MOV     DPTR,#SCNT10
        LCALL   STATIC1         ; 'OVERWRITE PROGRAM'
        CJNE    R7,#0E4H,CONT36 ;? 'Y' KEY PRESSED
        LCALL   CLEARPROG
        LJMP    CONT35
CONT36:
        CJNE    R7,#0DBH,CONT37 ;? 'V' KEY PRESSED
        LCALL   RUNPROG
        LJMP    CONT7
CONT37:
        CJNE    R7,#0CAH,CONT38 ;? 'N' KEY PRESSED
        LJMP    CONT7
CONT35:
        MOV     DPTR,#SCNT5     ;'CHOOSE DISPLAY MODE'
```

```
        LCALL   STATIC1          ;'DUAL OR SINGLE'
        CJNE    R7,#0FFH,CONT8   ;CHECK FOR 'DUAL' KEY
        MOV     MODEDISP,R7      ;SAVE DUAL DISPLAY MODE
        LJMP    CONT10
CONT8:
        CJNE    R7,#0FEH,CONT31 ;CHECK FOR 'SINGLE' KEY
        LJMP    CONT26
CONT31:
        CJNE    R7,#0FBH,CONT35 ;CHECK FOR 'PREV SCRN' KEY
        MOV     A,VALUE
        CJNE    A,#0FFH,CONT57  ;? BACK TO NEXT SCRN
        LJMP    NO1
CONT57:
        DEC             SCRNCNTR
        LCALL   PROGSTATUS
        LJMP    CONT45
CONT26:
        MOV     DPTR,#SCNT8      ;BIG LETTERS OR SMALL
                                  ;LETTERS
        LCALL   STATIC1
        CJNE    R7,#0FCH,CONT25 ;? IS 'BIG' KEY PRESSED
        MOV     MODEDISP,R7      ;SAVE SINGLE DISPLAY MODE
        LJMP    CONT10           ;       BIG LETTERS
CONT25:
        CJNE    R7,#0FDH,CONT32 ;? IS 'SMALL' KEY PRESSED
        MOV     MODEDISP,R7      ;SAVE SINGLE DISPLAY MODE
        LJMP    CONT10           ;       SMALL LETTERS
CONT32:
        CJNE    R7,#0FBH,CONT26 ;CHECK FOR 'PREV SCRN' KEY
        LJMP    CONT35
CONT10:
        MOV     DPTR,#SCNT4
        MOV     R5,#01H          ;SET DELAY BIT
        LCALL   STATIC1          ;'ENTER YOUR DATA FROM THE
                                       ;KEYPAD'
        MOV     R5,#00H          ;RESET DELAY BIT
        MOV     A,SCRNCNTR
        JZ      CONT12
        INC     RAM_H
        MOV     RAM_L,#00H
CONT12:
        MOV     A,MODEDISP
        CJNE    A,#0FFH,CONT22  ;? IS IT DUAL OR SINGLE MODE
        LCALL   ENTERDUAL        ;ENTER DATA ROUTINE
        LJMP    CONT17
CONT22:
        CJNE    A,#0FDH,CONT27  ;? SINGLE SMALL LETTERS
        LCALL   ENTERSINGLE
        LJMP    CONT17
CONT27:
        LCALL   ENTERBIG
CONT17:
        MOV     A,DISPLAY
        CJNE    A,#0FFH,CONT56  ;? SPECIAL KEY MODE
        MOV     DPTR,#SCNT6
        LCALL   STATIC1          ;'SELECT SCREEN MODE
                                  ;S/T/L/B/C/I/O
        CJNE    R7,#0C3H,CONT18 ;? IS 'S' KEY PRESSED
        MOV     MODESCRN,R7      ;SAVE SCREEN MODE
        LJMP    CONT15
CONT56:
        LJMP    CONT45
```

```
CONT18:
        CJNE    R7,#0CBH,CONT19 ;? IS 'T' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT19:
        CJNE    R7,#0E9H,CONT20 ;? IS 'L' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT20:
        CJNE    R7,#0C8H,CONT23 ;? IS 'B' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT23:
        CJNE    R7,#0EAH,CONT24 ;? IS 'R' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT24:
        CJNE    R7,#0D0H,CONT28 ;? IS 'C' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT28:
        CJNE    R7,#0D1H,CONT29 ;? IS 'I' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT29:
        CJNE    R7,#0D2H,CONT49 ;? IS 'O' KEY PRESSED
        MOV     MODESCRN,R7     ;SAVE SCREEN MODE
        LJMP    CONT15
CONT49:
        CJNE    R7,#0FBH,CONT17 ;? 'PREV SCRN' PRESSED
        LJMP    CONT35
CONT15:
        MOV     DPTR,#SCNT11
        LCALL   STATIC1         ;'SELECT CLEAR SCREEN'
        CJNE    R7,#0CAH,CONT39 ;? 'N' KEY PRESSED
        MOV     CLRSCRN,#00H
        LJMP    CONT45
CONT39:
        CJNE    R7,#0CBH,CONT40 ;? 'T' KEY PRESSED
        MOV     CLRSCRN,R7
        LJMP    CONT45
CONT40:
        CJNE    R7,#0C8H,CONT41 ;? 'B' KEY PRESSED
        MOV     CLRSCRN,R7
        LJMP    CONT45
CONT41:
        CJNE    R7,#0D0H,CONT42 ;? 'C' KEY PRESSED
        MOV     CLRSCRN,R7
        LJMP    CONT45
CONT42:
        CJNE    R7,#0C3H,CONT43 ;? 'S' KEY PRESSED
        MOV     CLRSCRN,R7
        LJMP    CONT45
CONT43:
        CJNE    R7,#0F0H,CONT44 ;? 'SPECIAL' KEY PRESSED
        MOV     CLRSCRN,R7
CONT60:
        MOV     DPTR,#SCNT15
        LCALL   STATIC1         ;'SELECT GRAFICS NO 1-3
        CJNE    R7,#0C4H,CONT61 ;?'1' KEY PRESSED
        MOV     PIC,R7
        LJMP    CONT45
```

```
CONT61:
        CJNE    R7,#0CCH,CONT62 ;?'2' KEY PRESSED
        MOV     PIC,R7
        LJMP    CONT45
CONT62:
        CJNE    R7,#0D4H,CONT63 ;?'3' KEY PRESSED
        MOV     PIC,R7
        LJMP    CONT45
CONT63:
        CJNE    R7,#0FBH,CONT60 ;?'PREV SCRN' KEY PRESSED
        LJMP    CONT15
CONT44:
        CJNE    R7,#0FBH,CONT70 ;? 'PREV SCRN' KEY PRESSED
        LJMP    CONT17
CONT70:
        LJMP    CONT15
CONT45:
        MOV     DPTR,#SCNT7
        LCALL   STATIC1         ;'SELECT NEXT SCREEN
                                 ;ENTER/Y PROGRAM/SAVE'
        CJNE    R7,#0F8H,CONT16 ;?IS 'ENTER' KEY PRESSED
        LCALL   PROGSTATUS
        DJNZ    SCRNCOUNT,CONT58;? LAST SCREEN OF PROGRAM
        MOV     DPTR,#SCNT14
        MOV     R5,#01H         ;SET DELAY BIT
        LCALL   STATIC1         ;'END OF PROGRAM'
        MOV     R5,#00H
        LJMP    CONT59
CONT58:
        INC     SCRNCNTR
        MOV     VALUE,#00H
        LJMP    CONT35
CONT16:
        CJNE    R7,#0F1H,CONT34 ;?IS 'PROGRAM' KEY PRESSED
        LCALL   PROGSTATUS
CONT59:
        MOV     SCRNCNTR,#00H   ;RESET SCREEN CNTR
        LJMP    CONT7
CONT34:
        CJNE    R7,#0FBH,CONT45 ;CHECK FOR 'PREV SCRN' KEY
        LJMP    CONT15
CONT50:
        MOV     DPTR,#7FF0H
        MOVX    A,@DPTR
        JZ      CONT64
        MOV     R5,#01H
        MOV     DPTR,#SCNT9     ;'WARNING EXISTING PROGRAM'
        LCALL   STATIC1         ;DISPLAY MESSAGE
        MOV     R5,#00H
CONT71:
        MOV     DPTR,#SCNT10    ;OVERWRITE PROGRAM Y/N/V
        LCALL   STATIC1
        CJNE    R7,#0E4H,CONT69 ;? IS 'Y' KEY PRESSED
        LCALL   ROTOVERWRITE    ;OVERWRITE PROGRAM NR10
        LJMP    CONT64
CONT69:
        CJNE    R7,#0CAH,CONT72 ;? IS 'N' KEY PRESSED
        LJMP    CONT7
CONT72:
        CJNE    R7,#0DBH,CONT71 ;? IS'V' KEY PRESSED
        LJMP    RUN12
CONT64:
        MOV     DPTR,#SCNT8     ;? BIG OR SMALL LETTERS
```

```
            LCALL   STATIC1
            CJNE    R7,#0FDH,CONT65 ;? IS SMALL LETTERS PRESSED
            MOV     DPTR,#7FF0H
            MOV     A,R7
            MOVX    @DPTR,A
            LCALL   ENTERROTATE     ;ENTER SMALL ROUTINE
            MOV     RAM_H,NR10_H
            MOV     RAM_L,#00H
CONT51:
            LCALL   RUN_ROTATE      ;DISPLAY SMALL ROTATE
                                          ;ROUTINE
            LJMP    CONT7
CONT65:
            CJNE    R7,#0FCH,CONT66 ;? IS BIG LETTERS PRESSED
            MOV     DPTR,#7FF0H     ;SAVE SMALL OR BIG LETTERS
                                          ;VALUE
            MOV     A,R7
            MOVX    @DPTR,A
            LCALL   ENTERROTATEBIG  ;ENTER BIG ROUTINE
            MOV     RAM_H,NR10_H
            MOV     RAM_L,#00H
            LCALL   RUN_ROTATEBIG   ;DISPLAY BIG ROTATE ROUTINE
            LJMP    CONT7
CONT66:
            LJMP    CONT64
;-------------------------------------------------------------;
START OF SUBROUTINES
;-----------------------------------
;DISPLAY STATIC CHAR. ON THE SCREEN
;-----------------------------------
STATIC1:
            MOV     EPROM_H,DPH     ;SAVE BASE ADDRESS
            MOV     EPROM_L,DPL     ; OF DATA
            MOV     CNTR2,#7FH
            MOV     CNTR1,#13H
            MOV     CNTR3,#05H
CONT3:
            MOV     P1,#00H         ;RESET SHIFT REG.
            SETB    P1.0            ;SERIAL DATA = 'HIGH'
            SETB    P1.1            ;ENABLE REG'S
            SETB    P1.2            ;CLOCK 'HIGH'
            CPL     P1.0            ;SERIAL DATA = 'ZERO'
            LJMP    CONT1
CONT2:
            SETB    P1.2            ;CLOCK 'HIGH'
CONT1:
            MOV     DPH,EPROM_H
            MOV     DPL,EPROM_L
            MOV     A,CNTR1
            MOVC    A,@A+DPTR       ;FETCH DATA
            MOV     DPTR,#STATICTABLE
            CLR     C
            SUBB    A,#20H
            MOV     B,#06H
            CLR     PSW.2
            MUL     AB
            JNB     PSW.2,STAT1
            INC     DPH
            CLR     PSW.2
STAT1:
            CLR     C
            ADD     A,CNTR3
            JNC     STAT2
```

```
            INC     DPH
STAT2:
            MOVC    A,@A+DPTR
            MOV     DPTR,#8000H
            MOVX    @DPTR,A         ;DISPLAY UPPER CHAR.
            MOV     A,CNTR1
            ADD     A,#14H
            MOV     DPH,EPROM_H
            MOV     DPL,EPROM_L
            MOVC    A,@A+DPTR       ;FETCH DATA
            MOV     DPTR,#STATICTABLE
            CLR     C
            SUBB    A,#20H
            MOV     B,#06H
            CLR     PSW.2
            MUL     AB
            JNB     PSW.2,STAT3
            INC     DPH
            CLR     PSW.2
STAT3:
            CLR     C
            ADD     A,CNTR3
            JNC     STAT4
            INC     DPH
STAT4:
            MOVC    A,@A+DPTR
            RL      A
            MOV     DPTR,#8001H
            MOVX    @DPTR,A         ;DISPLAY LOWER CHAR.
            CPL     P1.2            ;CLOCK LOW
            LCALL   DELAY1          ;DELAY

            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DATA
            DEC     CNTR3
            MOV     A,CNTR3
            CJNE    A,#0FFH,CONT2
            MOV     CNTR3,#05H
            DEC     CNTR1
            MOV     A,CNTR1
            CJNE    A,#0FFH,CONT2   ;? END OF DISPLAY
            MOV     CNTR1,#13H      ;RESET CNTR1
            CJNE    R5,#01H,CONT4   ;CHECK FOR DELAYED SEQUENCE
            DJNZ    CNTR2,STAT5
            MOV     CNTR2,#6FH      ;DELAY VALUE
            MOV     CNTR1,#13H      ;RESET CNTR1
            RET
STAT5:
            LJMP    CONT3
CONT4:
            CJNE    R6,#01H,STAT5   ;CHECK IF INT0 IS SET
            MOV     R6,#00H         ;CLEAR INT0 REG.
            RET
;-------------------------------------
;DELAY1
;-------------------------------------
DELAY1:
            PUSH    ACC
            MOV     A,#25H
```

```
DEL1:
            DEC     A
            JNZ     DEL1
            POP     ACC
            RET
;-------------------------------------
KEYREAD:
            MOV     DPTR,#8000H
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A          ;CLEAR DATA TO DISPLAY
            MOV     DPTR,#0C001H     ;KEYBOARD ADD
            MOV     A,#40H
            MOVX    @DPTR,A          ;SETUP TO READ
            MOV     DPTR,#0C000H
            MOVX    A,@DPTR          ;READ CHAR.
            MOV     R7,A             ;SAVE CONTENTS OF ACC
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOVX    A,@DPTR
            MOV     R6,#01H
            MOV     A,#00H
            CLR     IT0
            RETI
;-------------------------------------------
;                POWER DOWN ROUTINE
;-------------------------------------------
PWRDWN:
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            MOV     PCON,#03H
            CLR     IT1
;-------------------------------------------
;        LAMPCHECK ROUTINE
;-------------------------------------------
LAMPCHECK:
LP3:
            MOV     CNTR1,#078H
            MOV     CNTR2,#0FFH
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     DPTR,#8001H
            MOV     A,#0FFH
LP2:
            SETB    P1.2
LP1:
            DEC     DPL
            MOVX    @DPTR,A
            LCALL   DELAY1
            INC     DPL
            MOVX    @DPTR,A
            LCALL   DELAY1
```

```
            CPL     P1.2
            DJNZ    CNTR1,LP2
            MOV     CNTR1,#78H
            DJNZ    CNTR2,LP3
            RET
;-------------------------------------------
;ENTER DATA ON THE SCREEN (STATIC 2 LINES MODE)
;-------------------------------------------
ENTERDUAL:
            MOV     CNTR1,#78H      ;SETUP CNTR1
            MOV     RAMVAL,#77H     ;SETUP RAM CNTR
            MOV     CNTR2,#04H
            MOV     LOOP1,#06H
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
DD9:
            MOV     DPH,NR_H
            MOV     DPL,NR_L        ;PROG1 DATA START
            MOV     R0,#40H
DD1:
            MOV     A,R0
            MOVX    @DPTR,A         ;WRITE CURSOR TO DISPLAY
            INC     DPL
            INC     CNTRL
            MOV     A,CNTRL
            CJNE    A,#05H,DD1      ;? END OF CURSOR

            MOV     CNTRL,#00H      ;RESET LETTER CNTR
            MOV     A,#00H
            MOVX    @DPTR,A         ;DISPLAY BLANK LINE
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
DD4:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H         ;RESET SHIFTREGS.
            SETB    P1.0            ;SERIAL DATA = '1'
            SETB    P1.1            ;ENABLE REGISTERS
            SETB    P1.2            ;CLOCK SERIAL DATA 'HIGH'
            CPL     P1.0            ;SERIAL DATA '0'
            LJMP    DD2
DD3:
            SETB    P1.2            ;CLOCK 'HIGH'
DD2:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR         ;FETCH DATA
            MOV     DPTR,#8000H
            MOVX    @DPTR,A         ;DISPLAY UPPER CHAR.
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPH,RAM_H
            MOV     DPL,A
            MOVX    A,@DPTR         ;FETCH DATA
            RL      A
            MOV     DPTR,#8001H
            MOVX    @DPTR,A         ;DISPLAY LOWER CHAR.
            CPL     P1.2            ;CLOCK 'LOW'
            DEC     RAM_L
            LCALL   DELAY1          ;DELAY
            MOV     A,#00H
            MOV     DPTR,#8000H
```

```
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A          ;CLEAR DATA FROM DISPLAY
        DJNZ    CNTR1,DD3        ;? END OF DISPLAY

        MOV     CNTR1,#78H
        CJNE    R6,#01H,DD4      ;? HAS INT. OCCURED
        CJNE    R7,#0EFH,DD12    ;? IS 'BACKSPACE' KEY
                                       ;PRESSED
        LCALL   BACKSPACE
        LJMP    DD1
DD12:
        CJNE    R7,#0F8H,DD14    ;? IS 'ENTER' KEY PRESSED
        MOV     DPH,NR_H
        MOV     DPL,NR_L
        MOV     A,#00H
DD13:
        MOVX    @DPTR,A
        INC     DPL
        DJNZ    LOOP1,DD13
        MOV     R6,#00H
        INC     RAM_L
        MOV     CNTRC,#00H       ;RESET CHAR. COUNT REG.
        RET
DD14:
        CJNE    R7,#0F0H,DD15    ;? IS SPEC. FUNC. KEY
                                       ;PRESSED
        LJMP    DD4
DD15:
        CJNE    R7,#0F1H,DD16    ;? IS PROG. KEY PRESSED
        LJMP    DD4
DD16:
        CLR     C
        MOV     A,R7
        SUBB    A,#0F8H
        JC      DD10             ;EXCLUDE FUNC.KEYS
        LJMP    DD4
DD10:
        MOV     A,CNTRC
        CJNE    A,#28H,DD11      ;? IS 40 CHAR. ENTERED
        LJMP    DD4
DD11:
        MOV     R6,#00H          ;RESET INT. REG.
DD5:
        MOV     A,R7             ;LOAD KEY DEPRESSED
        MOV     R7,#00H          ;CLEAR KEYSAVE REG.
        CLR     C                ;RESET CARRY FLAG
        SUBB    A,#0C0H          ;GET OFFSET OF TABLE
        MOV     B,#06H           ;TABLE MULTIPLICATION
                                       ;FACTOR
        MUL     AB
        MOV     OFFSET,A         ;SAVE OFFSET OFF CHAR.
                                       ;TABLE
        MOV     DPTR,#KEYTABLE   ;LOAD BASE OF THE TABLE
        MOV     A,DPH
        ADD     A,B              ;ADD HIGHER OFFSET
        MOV     DPHSAVE,A        ;HIGH BYTE OFFSET SAVE
DD7:
        MOV     DPTR,#KEYTABLE
        MOV     DPH,DPHSAVE
        MOV     A,OFFSET
        MOVC    A,@A+DPTR         ;FETCH DATA FROM KEYTABLE
        MOV     DPH,NR_H
```

```
            MOV     DPL,NR_L
            MOVX    @DPTR,A          ;WRITE TO RAM TO BE
                                          ;DISPLAYED
            INC     NR_L             ;SETUP NEXT RAM LOCATION
            INC     OFFSET           ;NEXT VALUE OF CHAR. IN TABLE
            MOV     A,OFFSET
            CJNE    A,#00H,DD6       ;? IS HIGHER BYTE NECCASARY

            INC     DPHSAVE
DD6:
            INC     CNTRL            ;CHAR. LINE CNTR. 1-6
            MOV     A,CNTRL
            CJNE    A,#06H,DD7       ;? END OF CHAR.

            MOV     RAM_H,NR_H
            MOV     RAM_L,NR_L
            MOV     OFFSET,#00H
            MOV     CNTRL,#00H       ;RESET VALUES
            INC     CNTRC            ;CHAR. CNTR 1-40
            MOV     A,CNTRC
            CJNE    A,#28H,DD8       ;? IS END OF DISPLAY
                                          ;REACHED
            LJMP    DD4
DD8:
            LJMP    DD9
;-------------------------------------------------
;       ENTER DATA ON SCREEN SINGLE LINE
;               MODE ONLY ONE SCREEN
;-------------------------------------------------
ENTERSINGLE:
            MOV     CNTR1,#78H       ;SETUP CNTR1
            MOV     CNTRC,#00H
            MOV     LOOP1,#06H
            MOV     RAMVAL,#77H      ;SETUP RAM CNTR
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
PP9:
            MOV     DPH,NR_H         ;PROG. DATA START
            MOV     DPL,NR_L
            MOV     R0,#40H
PP1:
            MOV     A,R0
            MOVX    @DPTR,A          ;WRITE CURSOR TO DISPLAY
            INC     DPL
            INC     CNTRL
            MOV     A,CNTRL
            CJNE    A,#05H,PP1       ;? END OF CURSOR

            MOV     CNTRL,#00H       ;RESET LETTER CNTR
            MOV     A,#00H
            MOVX    @DPTR,A          ;DISP. BLANK LINE
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A          ;CLEAR DATA TO DISPLAY
PP4:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            LJMP    PP2
```

```
PP3:
            SETB    P1.2
PP2:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R4,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,R4
            ANL     A,#0F0H
            SWAP    A
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,PP3       ;? END OF DISPLAY

            MOV     CNTR1,#78H
            CJNE    R6,#01H,PP4     ;? HAS INT. OCCURED
            CJNE    R7,#0EFH,PP12   ;? IS BACKSPACE KEY PRESSED
            LCALL   BACKSPACE
            LJMP    PP1
PP12:
            CJNE    R7,#0F8H,PP14   ;? IS ENTER KEY PRESSED
            MOV     DPH,NR_H
            MOV     DPL,NR_L
            MOV     A,#00H
PP13:
            MOVX    @DPTR,A
            INC     DPL
            DJNZ    LOOP1,PP13
            MOV     R6,#00H
            MOV     CNTRC,#00H
            RET
PP14:
            CJNE    R7,#0F0H,PP15   ;? IS SPEC. FUNC. KEY
                                          ;PRESSED
            LJMP    PP4
PP15:
            CJNE    R7,#0F1H,PP16   ;? IS PROG. KEY PRESSED
            LJMP    PP4
PP16:
            CLR     C
            MOV     A,R7
            SUBB    A,#0F8H
            JC      PP10            ;EXCLUDE    FUNC. KEYS
            LJMP    PP4
PP10:
            MOV     A,CNTRC
            CJNE    A,#14H,PP11     ;? IS 20 CHAR. ENTERED
            LJMP    PP4
PP11:
            MOV     R6,#00H
            MOV     A,R7
```

```
                MOV     R7,#00H
                CLR     C
                SUBB    A,#0C0H
                MOV     B,#06H
                MUL     AB
                MOV     OFFSET,A
                MOV     DPTR,#KEYTABLE
                MOV     A,DPH
                ADD     A,B
                MOV     DPHSAVE,A
PP7:
                MOV     DPTR,#KEYTABLE
                MOV     DPH,DPHSAVE
                MOV     A,OFFSET
                MOVC    A,@A+DPTR
                MOV     DPH,NR_H
                MOV     DPL,NR_L
                MOVX    @DPTR,A
                INC     NR_L
                INC     OFFSET
                MOV     A,OFFSET
                CJNE    A,#00H,PP6
                INC     DPHSAVE
PP6:
                INC     CNTRL
                MOV     A,CNTRL
                CJNE    A,#06H,PP7
                MOV     RAM_H,NR_H
                MOV     RAM_L,NR_L
                MOV     OFFSET,#00H
                MOV     CNTRL,#00H
                INC     CNTRC
                MOV     A,CNTRC
                CJNE    A,#14H,PP8
                LJMP    PP4
PP8:
                LJMP    PP9

;-------------------------------------------------
;       ENTER DATA ON SCREEN SINGLE LINE MODE
;       LARGE CHARACTERS
;-------------------------------------------------
ENTERBIG:
                MOV     CNTR1,#78H
                MOV     LOOP1,#0CH
                MOV     RAMVAL,#77H
                MOV     CNTR2,#04H
                MOV     NR_H,RAM_H
                MOV     NR_L,RAM_L
                MOV     CNTRL,#00H
VV9:
                MOV     A,NR_L
                ADD     A,#78H
                MOV     DPL,A
                MOV     DPH,NR_H
                MOV     R0,#40H
VV1:
                MOV     A,R0
                MOVX    @DPTR,A
                INC     DPL
                INC     CNTRL
                MOV     A,CNTRL
                CJNE    A,#0AH,VV1
```

```
        MOV     CNTRL,#00H
        MOV     A,#00H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
VV4:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
        LJMP    VV2
VV3:
        SETB    P1.2
VV2:
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,RAM_L
        ADD     A,#78H
        MOV     DPH,RAM_H
        MOV     DPL,A
        MOVX    A,@DPTR
        MOV     DPTR,#8001H
        MOVX    @DPTR,A
        CPL     P1.2
        DEC     RAM_L
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,VV3

        MOV     CNTR1,#78H
        CJNE    R6,#01H,VV4
        CJNE    R7,#0EFH,VV12
        MOV     CNTRL,#18H
        MOV     R3,#0CH
        LCALL   BIGBACKSPACE
        LJMP    VV1
VV12:
        CJNE    R7,#0F8H,VV14
        MOV     DPH,NR_H
        MOV     A,NR_L
        ADD     A,#78H
        MOV     DPL,A
        MOV     A,#00H
VV13:
        MOVX    @DPTR,A
        INC     DPL
        DJNZ    LOOP1,VV13
        MOV     R6,#00H
        MOV     CNTRC,#00H
        RET
```

```
VV14:
        CJNE    R7,#0F0H,VV15   ;? IS SPEC. FUNC. KEY
                                        ;PRESSED
        LJMP    VV4
VV15:
        CJNE    R7,#0F1H,VV16   ;? IS PROG. KEY PRESSED
        LJMP    VV4
VV16:
        CLR     C
        MOV     A,R7
        SUBB    A,#0F8H
        JC      VV10            ;EXCLUDE FUNC. KEYS
        LJMP    VV4
VV10:
        MOV     A,CNTRC
        CJNE    A,#0AH,VV11
        LJMP    VV4
VV11:
        MOV     R6,#00H
        LJMP    VV5
VV5:
        MOV     A,R7
        MOV     R7,#00H
        CLR     C
        SUBB    A,#0C0H
        MOV     B,#18H
        MUL     AB
        MOV     OFFSET,A
        MOV     DPTR,#BIGLETTERS
        MOV     A,DPH
        ADD     A,B
        MOV     DPHSAVE,A
VV7:
        MOV     DPTR,#BIGLETTERS
        MOV     DPH,DPHSAVE
        MOV     A,OFFSET
        MOVC    A,@A+DPTR
        MOV     R4,A
        INC     OFFSET
        MOV     A,OFFSET
        MOVC    A,@A+DPTR
        MOV     R1,A
        MOV     DPH,NR_H
        MOV     DPL,NR_L
        MOV     A,R4
        MOVX    @DPTR,A
        MOV     A,NR_L
        ADD     A,#78H
        MOV     DPL,A
        MOV     A,R1
        MOVX    @DPTR,A
        INC     OFFSET
        INC     NR_L
        MOV     A,OFFSET
        CJNE    A,#00H,VV6
        INC     DPHSAVE
VV6:
        INC     CNTRL
        MOV     A,CNTRL
        CJNE    A,#0CH,VV7

        MOV     RAM_H,NR_H
        MOV     RAM_L,NR_L
```

```
            MOV     OFFSET,#00H
            MOV     CNTRL,#00H
            INC     CNTRC
            MOV     A,CNTRC
            CJNE    A,#0AH,VV8
            LJMP    VV4
VV8:
            LJMP    VV9
            RET
;-----------------------------------------------
;       ENTER DATA ON SCREEN SINGLE LINE MODE
;       SMALL LETTERS MORE THAN ONE SCREEN
;-----------------------------------------------
ENTERROTATE:
            MOV     CNTR1,#78H
            MOV     CNTRC,#00H
            MOV     CNTRL,#00H
            MOV     RAMVAL,#77H
            MOV     DPTR,#7600H
            MOV     R1,#14H
            MOV     SCRNCNTR,#14H
LL18:
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#78H,LL18
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
LL9:
            MOV     DPH,NR_H
            MOV     DPL,NR_L        ;PROG. DATA START
            MOV     R0,#40H
LL1:
            MOV     A,R0
            MOVX    @DPTR,A         ;WRITE CURSOR TO DISPLAY
            INC     DPTR
            INC     CNTRL
            MOV     A,CNTRL
            CJNE    A,#05H,LL1      ;? END OF CURSOR

            MOV     CNTRL,#00H      ;RESET LETTER CNTR
            MOV     A,#00H
            MOVX    @DPTR,A         ;DISPLAY BLANK LINE

            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
LL4:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H         ;RESET SHIFT REG.
            SETB    P1.0            ;SERIAL DATA =ONE
            SETB    P1.1            ;ENABLE REG'S
            SETB    P1.2            ;CLOCK 'HIGH'
            CPL     P1.0            ;SERIAL DATA=ZERO
            LJMP    LL2
LL3:
            SETB    P1.2            ;CLOCK 'HIGH'
LL2:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR         ;FETCH DATA
```

```
        MOV     R4,A            ;SAVE DATA
        ANL     A,#0FH          ;MASK TOP PART OF CHAR.
        SWAP    A
        MOV     DPTR,#8000H
        MOVX    @DPTR,A         ;DISPLAY UPPER PART OF CHAR.
        INC     DPL
        MOV     A,R4
        ANL     A,#0F0H
        SWAP    A
        MOVX    @DPTR,A         ;DISPLAY LOWER PART OF CHAR.
        CPL     P1.2            ;CLOCK 'LOW'
        DEC     RAM_L
        MOV     A,RAM_L
        CJNE    A,#0FFH,LL13

        DEC     RAM_H
        MOV     R0,#0FFH
LL13:
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
        DJNZ    CNTR1,LL3       ;? END OF DISPLAY
        MOV     A,R0
        CJNE    A,#0FFH,LL14
        INC     RAM_H
        MOV     R0,#00H
l114:
        MOV     CNTR1,#78H
        CJNE    R6,#01H,LL4     ;HAS INT. OCCURED
        CJNE    R7,#0EFH,LL10   ;? IS BACKSPACE KEY PRESSED
        LCALL   BACKSPACE
        INC     R1
        CJNE    R1,#15H,LL21
        MOV     R1,#00H
        INC     SCRNCNTR
LL21:
        MOV     A,CNTRC
        CJNE    A,#14H,LL16     ;? 20 CHAR. ENTERED
        MOV     A,RAMVAL
        SUBB    A,#06H
        MOV     RAMVAL,A
        JNC     LL16
        DEC     RAM_H
LL16:
        LJMP    LL9
LL10:
        CJNE    R7,#0F8H,LL5    ;? IS ENTER KEY PRESSED
        MOV     CNTRC,#00H      ;RESET CHAR. COUNT REG.
        MOV     DPL,NR_L
        MOV     DPH,NR_H
        MOV     A,#00H
        MOVX    @DPTR,A         ;WRITE END OF PROG.COMMAND
        INC     DPL
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        INC     DPL
```

```
        MOV     A,#0DH
        MOVX    @DPTR,A
        RET
LL5:
        CJNE    R7,#0F0H,LL22   ;? IS SPEC. FUNC. KEY
                                ;PRESSED
        LJMP    LL9
LL22:
        CJNE    R7,#0F1H,LL23   ;? IS PROG. KEY PRESSED
        LJMP    LL9
LL23:
        CLR     C
        MOV     A,R7
        SUBB    A,#0F8H
        JC      LL17            ;EXCLUDE    FUNC. KEYS
        LJMP    LL9
LL17:
        CJNE    R1,#00H,LL19
        LJMP    LL20
LL19:
        MOV     R6,#00H         ;RESET INT. REG.
        MOV     A,R7
        MOV     R7,#00H
        CLR     C
        SUBB    A,#0C0H
        MOV     B,#06H
        MUL     AB
        MOV     OFFSET,A
        MOV     DPTR,#KEYTABLE
        MOV     A,DPH
        ADD     A,B
        MOV     DPHSAVE,A
LL7:
        MOV     DPTR,#KEYTABLE
        MOV     DPH,DPHSAVE
        MOV     A,OFFSET
        MOVC    A,@A+DPTR
        MOV     DPH,NR_H
        MOV     DPL,NR_L
        MOVX    @DPTR,A
        INC     NR_L
        MOV     A,NR_L
        JNZ     LL12
        INC     NR_H
LL12:
        INC     OFFSET
        MOV     A,OFFSET
        CJNE    A,#00H,LL6
        INC     DPHSAVE
LL6:
        INC     CNTRL
        MOV     A,CNTRL
        CJNE    A,#06H,LL7      ;? WAS 6 POSITIONS SAVED
        DJNZ    R1,LL15
        MOV     R1,#14H
        DJNZ    SCRNCNTR,LL15
        INC     SCRNCNTR
LL20:
        MOV     R5,#01H
        MOV     DPTR,#SCNT14
        LCALL   STATIC1
        MOV     R5,#00H
        MOV     R1,#00H
```

```
LL15:
            MOV     OFFSET,#00H
            MOV     CNTRL,#00H
            INC     CNTRC
            MOV     A,CNTRC
            CJNE    A,#15H,LL8
            MOV     A,RAMVAL
            ADD     A,#06H
            MOV     RAMVAL,A
            JNC     LL11
            INC     RAM_H
LL11:
            DEC     CNTRC
LL8:
            LJMP    LL9
;-------------------------------------------------
;       ENTER DATA ON SCREEN SINGLE LINE MODE
;       BIG LETTERS MORE THAN ONE SCREEN
;-------------------------------------------------
ENTERROTATEBIG:
            MOV     CNTR1,#78H
            MOV     CNTRC,#00H
            MOV     CNTRL,#00H
            MOV     RAMVAL,#0EFH
            MOV     DPTR,#7600H
            MOV     R1,#0AH
            MOV     SCRNCNTR,#0AH
LLL18:
            MOV     A,#00H
            MOVX    @DPTR,A         ;CLEAR DATA SEGMENT PROG. 10
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#78H,LLL18    ;? ALL LOCATIONS CLEARED
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
LLL9:
            MOV     DPH,NR_H
            MOV     DPL,NR_L        ;PROG. DATA START
            MOV     R0,#40H
            INC     DPTR            ;SET CURSOR ADRESS
LLL1:
            MOV     A,R0
            MOVX    @DPTR,A         ;WRITE CURSOR TO DISPLAY
            INC     DPTR
            INC     DPTR            ;2 X BECAUSE CURSOR AT
                                            ;BOTTOM
            INC     CNTRL           ;CHAR LINE COUNT
            MOV     A,CNTRL
            CJNE    A,#0AH,LLL1     ;? END OF BIG CURSOR

            MOV     CNTRL,#00H      ;RESET LETTER LINE CNTR
            MOV     A,#00H
            MOVX    @DPTR,A         ;DISPLAY 2 BLANK LINES
            INC     DPL
            MOVX    @DPTR,A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
LLL4:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H         ;RESET SHIFT REG.
            SETB    P1.0            ;SERIAL DATA =ONE
```

```
        SETB    P1.1            ;ENABLE REG'S
        SETB    P1.2            ;CLOCK 'HIGH'
        CPL     P1.0            ;SERIAL DATA=ZERO
        LJMP    LLL2
LLL3:
        SETB    P1.2            ;CLOCK 'HIGH'
LLL2:
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR         ;FETCH DATA
        MOV     DPTR,#8001H
        MOVX    @DPTR,A         ;DISPLAY LOWER PART OF
                                        ;CHAR.
        DEC     RAM_L
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        MOV     DPTR,#8000H
        MOVX    @DPTR,A         ;DISPLAY UPPER PART OF
                                        ;CHAR.
        CPL     P1.2            ;CLOCK 'LOW'
        DEC     RAM_L
        MOV     A,RAM_L
        CJNE    A,#0FFH,LLL13   ;CHECK FOR CARRY
        DEC     RAM_H
        MOV     R0,#0FFH
LLL13:
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A          ;CLEAR DISPLAY DATA
        DJNZ    CNTR1,LLL3       ;? END OF DISPLAY
        MOV     A,R0
        CJNE    A,#0FFH,LLL14
        INC     RAM_H
        MOV     R0,#00H
LLL14:
        MOV     CNTR1,#78H
        CJNE    R6,#01H,LLL4     ;HAS INT. OCCURED
        CJNE    R7,#0EFH,LLL10   ;? IS BACKSPACE KEY
                                        ;PRESSED
        MOV     CNTRL,#18H
        LCALL   BIGBACKSPACEROT
        INC     R1
        CJNE    R1,#0BH,LLL21
        MOV     R1,#00H
        INC     SCRNCNTR
LLL21:
        MOV     A,CNTRC
        CJNE    A,#0AH,LLL16     ;? 10 CHAR. ENTERED
        MOV     A,RAMVAL
        SUBB    A,#18H
        MOV     RAMVAL,A
        JNC     LLL16
        DEC     RAM_H
LLL16:
        LJMP    LLL9
LLL10:
        CJNE    R7,#0F8H,LLL5    ;? IS ENTER KEY PRESSED
        MOV     CNTRC,#00H       ;RESET CHAR. COUNT REG.
        MOV     DPL,NR_L
        MOV     DPH,NR_H
```

```
        MOV     A,#00H
        MOVX    @DPTR,A
        INC     DPTR
        MOV     A,#0AH
        MOVX    @DPTR,A          ;WRITE END OF PROG.
                                      ;COMMAND
        RET
LLL5:
        CJNE    R7,#0F0H,LLL22   ;? IS SPEC. FUNC. KEY
                                      ;PRESSED
        LJMP    LLL9
LLL22:
        CJNE    R7,#0F1H,LLL23   ;? IS PROG. KEY PRESSED
        LJMP    LLL9
LLL23:
        CLR     C
        MOV     A,R7
        SUBB    A,#0F8H
        JC      LLL17            ;EXCLUDE FUNC. KEYS
        LJMP    LLL9
LLL17:
        CJNE    R1,#00H,LLL19
        LJMP    LLL20
LLL19:
        MOV     R6,#00H          ;RESET INT. REG.
        MOV     A,R7
        MOV     R7,#00H
        CLR     C
        SUBB    A,#0C0H          ;SETUP DATA FROM KEYBOARD
        MOV     B,#18H
        MUL     AB               ;SETUP VALUE FOR OFFSET
        MOV     OFFSET,A
        MOV     DPTR,#BIGLETTERS
        MOV     A,DPH
        ADD     A,B
        MOV     DPHSAVE,A
LLL7:
        MOV     DPTR,#BIGLETTERS
        MOV     DPH,DPHSAVE
        MOV     A,OFFSET
        MOVC    A,@A+DPTR
        MOV     DPH,NR_H
        MOV     DPL,NR_L
        MOVX    @DPTR,A
        INC     NR_L
        MOV     A,NR_L
        JNZ     LLL12
        INC     NR_H
LLL12:
        INC     OFFSET
        MOV     A,OFFSET
        CJNE    A,#00H,LLL6
        INC     DPHSAVE
LLL6:
        INC     CNTRL
        MOV     A,CNTRL
        CJNE    A,#18H,LLL7      ;? WAS 24 POSITIONS SAVED
        DJNZ    R1,LLL15         ;CHAR COUNT
        MOV     R1,#0AH
        DJNZ    SCRNCNTR,LLL15   ;SCREEN COUNT
        INC     SCRNCNTR
LLL20:
        MOV     R5,#01H
```

```
                MOV     DPTR,#SCNT14      ;'END OF PROGRAM
                                                  WORKSPACE'
                LCALL   STATIC1
                MOV     R5,#00H
                MOV     R1,#00H
LLL15:
                MOV     OFFSET,#00H
                MOV     CNTRL,#00H
                INC     CNTRC
                MOV     A,CNTRC
                CJNE    A,#0BH,LLL8
                MOV     A,RAMVAL
                ADD     A,#18H
                MOV     RAMVAL,A
                JNC     LLL11
                INC     RAM_H
LLL11:
                DEC     CNTRC
LLL8:
                LJMP    LLL9

;-------------------------------------------------
;       SAVE THE PROGRAM STATUS AT THE RIGHT ADDRESSES
;-------------------------------------------------
PROGSTATUS:
                MOV     A,PROGNUM
                CJNE    A,#01H,EE1
                MOV     DPH,#40H          ;START ADD. OF PROG. NR1
                LCALL   WRSCNMODE
                MOV     DPTR,#40F8H
                MOV     A,SCRNCNTR
                MOVX    @DPTR,A           ;WRITE SCREEN COUNT TO
                                                ;FILE
                RET
EE1:
                CJNE    A,#02H,EE2
                MOV     DPH,#44H          ;START ADD. OF PROG. NR2
                LCALL   WRSCNMODE
                MOV     DPTR,#44F8H
                MOV     A,SCRNCNTR
                MOVX    @DPTR,A           ;WRITE SCREEN COUNT TO
                                                ;FILE
                RET
EE2:
                CJNE    A,#03H,EE3
                MOV     DPH,#48H          ;START ADD. OF PROG. NR3
                LCALL   WRSCNMODE
                MOV     DPTR,#48F8H
                MOV     A,SCRNCNTR
                MOVX    @DPTR,A           ;WRITE SCREEN COUNT TO
                                                ;FILE
                RET
EE3:
                CJNE    A,#04H,EE4
                MOV     DPH,#4CH          ;START ADD. OF PROG. NR4
                LCALL   WRSCNMODE
                MOV     DPTR,#4CF8H
                MOV     A,SCRNCNTR
                MOVX    @DPTR,A           ;WRITE SCREEN COUNT TO
                                                FILE
                RET
EE4:
                CJNE    A,#05H,EE5
```

```
            MOV     DPH,#52H        ;START ADD.OF PROG. NR5
            LCALL   WRSCNMODE
            MOV     DPTR,#52F8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A         ;WRITE SCREEN COUNT TO
                                          FILE
            RET
EE5:
            CJNE    A,#06H,EE6
            MOV     DPH,#58H        ;START ADD. OF PROG. NR6
            LCALL   WRSCNMODE
            MOV     DPTR,#58F8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A         ;WRITE SCRN COUNT TO FILE
            RET
EE6:
            CJNE    A,#07H,EE7
            MOV     DPH,#5EH        ;START ADD. OF PROG. NR7
            LCALL   WRSCNMODE
            MOV     DPTR,#5EF8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A         ;WRITE SCRN COUNT TO FILE
            RET
EE7:
            CJNE    A,#08H,EE8
            MOV     DPH,#66H        ;START ADD. OF PROG. NR8
            LCALL   WRSCNMODE
            MOV     DPTR,#66F8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A
            RET
EE8:
            CJNE    A,#09H,EE9
            MOV     DPH,#6EH
            LCALL   WRSCNMODE
            MOV     DPTR,#6EF8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A
            RET
EE9:
            MOV     DPH,#76H        ;START ADD. OF PROG. NR0
            LCALL   WRSCNMODE
            MOV     DPTR,#76F8H
            MOV     A,SCRNCNTR
            MOVX    @DPTR,A         ;WRITE SCREEN COUNT TO
                                          FILE
            RET
;-------------------------------------
;       BACKSPACE ROUTINE
;-------------------------------------
BACKSPACE:
            MOV     A,CNTRC
            JZ      BACKSP4
            MOV     DPH,NR_H
            MOV     A,NR_L
            ADD     A,#06H
            JNC     BACKSP2
            INC     DPH
BACKSP2:
            MOV     DPL,A
            MOV     CNTRL,#0CH
BACKSP1:
            DEC     DPL
```

```
            MOV     A,DPL
            CJNE    A,#0FFH,BACKSP3
            DEC     DPH
BACKSP3:
            MOV     A,#00H
            MOVX    @DPTR,A
            DJNZ    CNTRL,BACKSP1
            MOV     NR_L,DPL
            DEC     CNTRC
BACKSP4:
            MOV     R6,#00H
            RET
;-------------------------------------
;       BIG LETTERS BACKSPACE ROUTINE
;-------------------------------------
BIGBACKSPACE:
            MOV     A,CNTRC
            JZ      BIGSPACE6
            CJNE    A,#0AH,BIGSPACE4
            MOV     CNTRL,#0CH
            MOV     DPH,NR_H
            MOV     DPL,NR_L
            LJMP    BIGSPACE5
BIGSPACE4:
            MOV     CNTRL,#18H
            MOV     DPH,NR_H
            MOV     A,NR_L
            ADD     A,#0CH          ;SETUP POSITION FROM
            JNC     BIGSPACE2       ; WHERE TO CLEAR FROM
            INC     DPH
BIGSPACE2:
            MOV     DPL,A
            LJMP    BIGSPACE5
BIGSPACE1:
            POP     DPH
            POP     DPL
BIGSPACE5:
            DEC     DPL
            MOV     A,DPL
            CJNE    A,#0FFH,BIGSPACE3
            DEC     DPH
BIGSPACE3:
            MOV     A,#00H
            MOVX    @DPTR,A
            PUSH    DPL
            PUSH    DPH
            MOV     A,DPL
            ADD     A,#78H
            MOV     DPL,A
            MOV     A,#00H
            MOVX    @DPTR,A
            DJNZ    CNTRL,BIGSPACE1
            POP     DPH
            POP     DPL
            MOV     NR_L,DPL
            MOV     A,DPL
            ADD     A,#78H
            MOV     DPL,A
            DEC     CNTRC
BIGSPACE6:
            MOV     R6,#00H
            RET
```

```
;-------------------------------------------------------
;       BACKSPACE ROUTINE FOR ENTERROTATE
;               BIG LETTERS
;-------------------------------------------------------
BIGBACKSPACEROT:
            MOV     A,CNTRC
            JZ      BIGSPACEROT6
            CJNE    A,#0AH,BIGSPACEROT4
            MOV     CNTRL,#18H
            MOV     DPH,NR_H
            MOV     DPL,NR_L
            LJMP    BIGSPACEROT5
BIGSPACEROT4:
            MOV     CNTRL,#30H
            MOV     DPH,NR_H
            MOV     A,NR_L
            ADD     A,#18H          ;SETUP POSITION FROM
            JNC     BIGSPACEROT2    ; WHERE TO CLEAR FROM
            INC     DPH
BIGSPACEROT2:
            MOV     DPL,A
BIGSPACEROT5:
            DEC     DPL
            MOV     A,DPL
            CJNE    A,#0FFH,BIGSPACEROT3
            DEC     DPH
BIGSPACEROT3:
            MOV     A,#00H
            MOVX    @DPTR,A
            DJNZ    CNTRL,BIGSPACEROT5
            MOV     NR_H,DPH
            MOV     NR_L,DPL
            DEC     CNTRC
BIGSPACEROT6:
            MOV     R6,#00H
            RET
;---------------------------------------
;       WRITE DISP & SCREEN MODES TO
;               THE ALLOCATED PROGRAMS
;---------------------------------------
WRSCNMODE:
            MOV     DPL,#0F0H
            MOV     A,SCRNCNTR
            ADD     A,DPH               ;TO SELECT WHAT SCREEN
            MOV     DPH,A               ; TO STORE MODES
            MOV     A,MODEDISP
            MOVX    @DPTR,A             ;WRITE DISPLAY MODE
            INC     DPL                 ; 2 LINES OR 1 LINE
            MOV     A,MODESCRN
            MOVX    @DPTR,A             ;WRITE SCREEN MODE
            INC     DPL
            MOV     A,CLRSCRN           ;WRITE CLEAR MODE
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,DISPLAY           ;WRITE PROGRAM MODE
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,PIC
            MOVX    @DPTR,A             ;WRITE PICTURE MODE
            RET
```

```
;---------------------------------------
;       DISPLAY SELECTED PROGRAM
;       IN THE CORRECT SCREEN MODES
;---------------------------------------
RUNPROG:
            MOV     VALUE,#00H
            MOV     AUTO,#00H
            MOV     R7,#00H
            MOV     DPH,RAM_H
            MOV     DPL,#0F0H
            MOVX    A,@DPTR
            CJNE    A,#0FDH,FF44        ;DOES PROGRAM EXIST ?
            LJMP    FF36
FF44:
            CJNE    A,#0FCH,FF45
            LJMP    FF36
FF45:
            CJNE    A,#0FFH,FF46
            LJMP    FF36
FF46:
            RET
FF36:
            MOV     PROG_H,RAM_H
            MOV     PROG_L,RAM_L
FF8:
            MOV     RAM_H,PROG_H
            MOV     RAM_L,PROG_L
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
            MOV     A,PROG_L
            ADD     A,#0F8H         ;SETUP ADD. FOR SCREEN CNTR
            MOV     DPL,A
            MOV     DPH,PROG_H
            MOVX    A,@DPTR          ;FETCH SCREEN COUNT
            MOV     SCRNCNTR,A
            MOV     DPL,#0F3H
            MOVX    A,@DPTR          ;LOAD AUTO/SPECIAL MODE
            MOV     DISPLAY,A
            INC     SCRNCNTR
            INC     SCRNCNTR
FF7:
            DJNZ    SCRNCNTR,FF6     ;? LAST SCREEN OF PROGRAM
                                            SAVED
            LJMP    FF8
FF37:
            MOV     DPTR,#AUTODISP
            MOV     A,AUTO
            MOVC    A,@A+DPTR
            INC     AUTO
            MOV     R7,AUTO
            CJNE    R7,#07H,FF42     ;? START DISP MODES AGAIN
            MOV     AUTO,#00H
FF42:
            LJMP    FF39
FF38:
            MOV     DPTR,#AUTODISP
            MOV     A,AUTO
            MOVC    A,@A+DPTR
            INC     AUTO
            MOV     R7,AUTO
            CJNE    R7,#07H,FF43
            MOV     AUTO,#00H
```

```
FF43:
            LJMP    FF40
FF6:
            MOV     DPL,#0F0H       ;SETUP ADD. OF DISP. MODE
            MOV     DPH,RAM_H
            MOVX    A,@DPTR         ;FETCH DISPLAY MODE
            MOV     MODEDISP,A
            CJNE    A,#0FDH,FF16    ;? SINGLE LINE
            LJMP    FF3

;DUAL LINE DISPLAY MODE
;----------------------
FF16:
            MOV     A,DISPLAY
            CJNE    A,#0FFH,FF37
            INC     DPL
            MOVX    A,@DPTR
FF39:
            CJNE    A,#0C3H,FF2     ;? STATIC SCREEN

;STATIC SCREEN MODE
            LCALL   RUN_STATIC      ;DISPLAY SCREEN IN STATIC
                                         MODE
            INC     RAM_H
            CJNE    R7,#0FAH,FF21
            RET
FF21:
            LJMP    FF7
FF2:
            CJNE    A,#0CBH,FF4     ;? TOP & BOTTOM SHIFT

;SHIFT IN FROM TOP & BOTTOM MODE
            LCALL   RUN_TOPBOTTOM
            INC     RAM_H
            CJNE    R7,#0FAH,FF22
            RET
FF22:
            LJMP    FF7
FF4:
            CJNE    A,#0E9H,FF5     ;? SHIFT FROM RIGHT

;SHIFT IN FROM RIGHT, DISPLAY STATIC
            LCALL   RUN_RIGHTSTAT
            INC     RAM_H
            CJNE    R7,#0FAH,FF23
            RET
FF23:
            LJMP    FF7
FF5:
            CJNE    A,#0C8H,FF12    ;? STARBURST

;SHIFT IN FROM RIGHT STARBURST
            LCALL   CLEAR_WORKSPACE
            LCALL   RUN_STARBURST
            INC     RAM_H
            CJNE    R7,#0FAH,FF24
            RET
FF24:
            LJMP    FF7
FF12:
            CJNE    A,#0EAH,FF14
```

```
;SHIFT IN FROM LEFT,DISPLAY STATIC
            LCALL   RUN_LEFTSTAT
            INC     RAM_H
            CJNE    R7,#0FAH,FF25
            RET
FF25:
            LJMP    FF7
FF14:
            CJNE    A,#0D0H,FF17

;SHIFT IN FROM CENTRE ,DISPLAY STATIC
            LCALL   RUN_TOPBOTTOM
            INC     RAM_H
            CJNE    R7,#0FAH,FF26
            RET
FF26:
            LJMP    FF7
FF17:
            CJNE    A,#0D1H,FF19

;SHIFT IN FROM LEFT AND RIGHT TO CENTRE
            LCALL   LR_CENTRE
            INC     RAM_H
            CJNE    R7,#0FAH,FF27
            RET
FF27:
            LJMP    FF7
FF19:
            CJNE    A,#0D2H,FF3

;SHIFT FROM CENTRE TOWARDS LEFT AND RIGHT
            LCALL   CENTRE_LR
            INC     RAM_H
            CJNE    R7,#0FAH,FF28
            RET
FF28:
            LJMP    FF7
FF1:
            LJMP    FF38

;SINGLE LINE DISPLAY MODE
;------------------------
FF3:
            MOV     A,DISPLAY
            CJNE    A,#0FFH,FF1
            INC     DPL
            MOVX    A,@DPTR
FF40:
            CJNE    A,#0C3H,FF9

;STATIC SCREEN MODE
            LCALL   SRUNSTATIC
            INC     RAM_H
            CJNE    R7,#0FAH,FF29
            RET
FF29:
            LJMP    FF7
FF9:
            CJNE    A,#0CBH,FF10

;SHIFT IN FROM TOP & BOTTOM
            LCALL   SRUNTOPBOTTOM
            INC     RAM_H
```

```
            CJNE    R7,#0FAH,FF30
            RET
FF30:
            LJMP    FF7
FF10:
            CJNE    A,#0E9H,FF11

;SHIFT IN FROM RIGHT, DISPLAY STATIC
            LCALL   SRUNRIGHTSTAT
            INC     RAM_H
            CJNE    R7,#0FAH,FF31
            RET
FF31:
            LJMP    FF7
FF11:
            CJNE    A,#0EAH,FF13

;SHIFT IN FROM LEFT,DISPLAY STATIC
            LCALL   SRUNLEFTSTAT
            INC     RAM_H
            CJNE    R7,#0FAH,FF32
            RET
FF32:
            LJMP    FF7
FF13:
            CJNE    A,#0D0H,FF15

;SHIFT IN FROM CENTRE,DISPLAY STATIC
            LCALL   SRUNTOPBOTTOM
            INC     RAM_H
            CJNE    R7,#0FAH,FF33
            RET
FF33:
            LJMP    FF7
FF15:
            CJNE    A,#0D1H,FF18
, SINGLE LINE
            LCALL   LRCENTRE
            INC     RAM_H
            CJNE    R7,#0FAH,FF34
            RET
FF34:
            LJMP    FF7
FF18:
            CJNE    A,#0D2H,FF20

;SHIFT IN FROM CENTRE TO LEFT AND RIGHT
            LCALL   CENTRELR
            INC     RAM_H
            CJNE    R7,#0FAH,FF35
            RET
FF35:
            LJMP    FF7
FF20:
            LJMP    FF8
;---------------------------------------
;       CLEAR ROTATE PROGRAM
;---------------------------------------
ROTOVERWRITE:
            MOV     DPTR,#7600H         ;CLEAR PROGRAM NR10
RAMC:
            MOV     A,#00H
            MOVX    @DPTR,A
```

```
            INC     DPTR
            MOV     A,DPH
            CJNE    A,#80H,RAMC
            RET
;---------------------------------------
;       CLEAR WORKSPACE
;---------------------------------------
CLEAR_WORKSPACE:
            MOV     A,#00H
            MOV     DPTR,#4FFFH
WORK:
            MOVX    @DPTR,A
            DJNZ    DPL,WORK
            MOVX    @DPTR,A
            RET
;---------------------------------------
;       CLEAR PROGRAM DATA
;---------------------------------------
CLEARPROG:
            MOV     DPL,#0F8H   ;LOAD SCRN CNTR
            MOV     DPH,RAM_H
            MOVX    A,@DPTR             ;FETCH SCRN TOTAL
            MOV     R4,A
            MOV     DPL,RAM_L
CLR1:
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,DPL
            CJNE    A,#00H,CLR1         ;CHECK FOR END OF SCRN
            INC     DPH
            DEC     R4
            CJNE    R4,#0FFH,CLR1   ;PROGRAM CLEARED ALL SCRN'S
            RET

;*************************************************************;
DISPLAY MODES
;*************************************************************;
STATIC RUN DISPLAY MODE
;---------------------------------------
RUN_STATIC:
            MOV     MODESCRN,R7
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R4,#0FFH
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A               ;CLEAR DISPLAY LINES
            LJMP    DISPDUAL            ;DISPLAY CHAR STATIC
;---------------------------------------
;       DISPLAY CHARACTERS STATIC
;---------------------------------------
DISPDUAL:
GG3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H             ;RESET SHIFT REG.'S
            SETB    P1.0                ;SERIAL DATA = 'HIGH'
            SETB    P1.1                ;ENABLE REG.'S
            SETB    P1.2                ;CLOCK 'HIGH'
            CPL     P1.0                ;SERIAL = 'ZERO'
            LJMP    GG1
```

```
GG2:
            SETB    P1.2                ;CLOCK 'HIGH'
GG1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR             ;FETCH DATA
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            ADD     A,#78H
            MOV     DPH,RAM_H
            MOV     DPL,A
            MOVX    A,@DPTR             ;FETCH DATA
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,GG4    ;? DUAL OR SINGLE LINE
            RL      A                   ;SHIFT FOR DUAL
GG4:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A             ;DISPLAY LOWER DATA LINE
            CPL     P1.2                ;CLOCK 'LOW'
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A                ;CLEAR DISPLAY DATA
            DJNZ    CNTR1,GG2           ;? AT END OF DISPLAY
            MOV     CNTR1,#78H          ;RESET LINE CNTR
            CJNE    R6,#01H,GG5         ;? HAS INT OCCURED
            MOV     R6,#00H             ;RESET INT. REG.
            RET
GG11:
            MOV     DPTR,#AUTOCLR
            MOV     A,CLEAR
            MOVC    A,@A+DPTR
            INC     CLEAR
            MOV     R7,CLEAR
            CJNE    R7,#05H,GG12
            MOV     CLEAR,#00H
GG12:
            LJMP    GG13
GG14:
            LJMP    GG10
GG5:
            DJNZ    R4,GG3
GG17:
            MOV     A,DISPLAY
            CJNE    A,#0FFH,GG11
            MOV     DPH,RAM_H
            MOV     DPL,#0F2H
            MOVX    A,@DPTR
            JZ      GG14
GG13:
            CJNE    A,#0CBH,GG6                ;? 'T' KEY PRESSED
            LCALL   CLRF_TOP
GG6:
            CJNE    A,#0C8H,GG7         ;? 'B' KEY PRESSED
            LCALL   CLRF_BOT
            RET
GG7:
            CJNE    A,#0D0H,GG8         ;? 'C' KEY PRESSED
            LCALL   CLRF_CNR
            RET
```

```
GG8:
            CJNE    A,#0C3H,GG9         ;? 'S' KEY PRESSED
            LCALL   CLRF_SDS
            RET
GG9:
            CJNE    A,#0F0H,GG10        ;?IS SPECIAL KEY PRESSED
            MOV     DPH,RAM_H
            MOV     DPL,#0F4H
            MOVX    A,@DPTR
            CJNE    A,#0C4H,GG15
            MOV     DPTR,#PUCMAN1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#PUCMAN2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#PUCMAN3
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
GG15:
            CJNE    A,#0CCH,GG16
            MOV     DPTR,#DUCK1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#DUCK2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#DUCK3
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
GG16:
            CJNE    A,#0D4H,GG10
            MOV     DPTR,#GRADER1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#GRADER2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#GRADER1
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
GG10:
            RET
;-----------------------------------
;       CHARACTERS SHIFT IN FROM
;       TOP AND BOTTOM (2 LINES)
;-----------------------------------
RUN_TOPBOTTOM:
            MOV     MODESCRN,A      ;SAVE CENTRE OR TOP&BOT MODE
            MOV     CNTR1,#78H      ;DISP. LENGTH   *******
            MOV     CNTR3,#07H      ;DISP CHAR STEP BY STEP
            MOV     RAMVAL,#77H
            MOV     LOOP1,#07H      ;SHIFT TOP CHAR. IN ACC
            MOV     LOOP2,#07H      ;SHIFT BOTTOM CHAR. IN ACC
            MOV     LOOP3,#0FH      ;DELAY PART OF CHAR
            MOV     LOOP4,#20H      ;TIME DELAY
            MOV     R4,#0FFH        ;SCREEN DELAY
```

```
        MOV     DPTR,#8000H     ;SET SCAN LINES
        MOV     A,#00H                  ; OF DISPLAY TO
        MOVX    @DPTR,A         ;  ALL ZERO'S
        INC     DPL             ;NEXT PORT
        MOVX    @DPTR,A
HH3:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H         ;RESET SHIFT REGISTERS
        SETB    P1.0            ;SERIAL DATA = 'ONE'
        SETB    P1.1            ;ENABLE REGISTERS
        SETB    P1.2            ;CLOCK  'HIGH'
        CPL     P1.0            ;SERIAL DATA = 'ZERO'
        LJMP    HH1
HH2:
        SETB    P1.2            ;CLOCK 'HIGH'
HH1:
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR         ;FETCH DATA TOP CHAR.
        MOV     R1,MODEDISP
        CJNE    R1,#0FFH,HH11
        RL      A
HH11:
        MOV     LOOP1,CNTR3
REP1:
        MOV     R1,MODESCRN
        CJNE    R1,#0D0H,HH7    ;? IS MODE = TOP&BOTTOM
        CLR     C
        RLC     A               ;SET UP TOP CHAR.
        LJMP    HH9
HH7:
        CLR     C
        RRC     A
HH9:
        DJNZ    LOOP1,REP1
        MOV     DPTR,#8000H
        MOVX    @DPTR,A          ;DISPLAY UPPER CHAR.
        MOV     A,RAM_L
        ADD     A,#78H
        MOV     DPH,RAM_H
        MOV     DPL,A
        MOVX    A,@DPTR          ;FETCH DATA
        MOV     LOOP2,CNTR3
REP2:
        MOV     R1,MODESCRN
        CJNE    R1,#0D0H,HH8     ;? IS MODE = TOP&BOTTOM
        CLR     C
        RRC     A                ;SET UP BOTTOM CHAR.
        LJMP    HH10
HH8:
        CLR     C
        RLC     A
HH10:
        DJNZ    LOOP2,REP2
        MOV     DPTR,#8001H
        MOVX    @DPTR,A             ;DISPLAY LOWER CHAR.
        CPL     P1.2                ;CLOCK 'LOW'
        DEC     RAM_L
        MOV     A,#00H
        MOV     DPTR,#8000H
        LCALL   DELAY               ;DELAY
        MOVX    @DPTR,A             ;CLEAR DATA
        INC     DPL
```

```
            MOVX    @DPTR,A
            DJNZ    CNTR1,HH2           ;? END OF DISPLAY
            MOV     CNTR1,#78H
            DJNZ    CNTR3,HH3
AGAIN:
            DJNZ    LOOP4,AGAIN
            LJMP    DISPDUAL
;-----------------------------------------------
;START OF SUBROUTINES
;-----------------------------------------------
DELAY:
            PUSH    ACC
            MOV     A,#28H
DEL:
            DEC     A
            JNZ     DEL
            POP     ACC
            RET
;-----------------------------------
;       CHARACTERS SHIFT IN FROM RIGHT
;       OF SCREEN (2 LINES) THEN STATIC
;-----------------------------------
RUN_RIGHTSTAT:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     CNTR2,#00H
            MOV     CNTR3,#77H
            MOV     RAM_L,#00H

            MOV     DPTR,#8000H                 ;SET DATA LINES
            MOV     A,#00H              ; OF DISPLAY TO
            MOVX    @DPTR,A             ;  ALL ZERO'S
            INC     DPL                         ;NEXT PORT
            MOVX    @DPTR,A

;DISPLAY CHARACTERS
JJ3:
            MOV     P1,#00H           ;RESET SHIFT REGISTERS
            SETB    P1.0                ;SERIAL DATA = 'ONE'
            SETB    P1.1                ;ENABLE REGISTERS
            SETB    P1.2                ;CLOCK  'HIGH'
            CPL     P1.0                ;SERIAL DATA = 'ZERO'
            LJMP    JJ1
JJ2:
            SETB    P1.2                ;CLOCK 'HIGH'
JJ1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR             ;FETCH DATA
            MOV     DPTR,#8000H
            MOVX    @DPTR,A             ;DISPLAY UPPER CHAR.
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPH,RAM_H
            MOV     DPL,A
            MOVX    A,@DPTR             ;FETCH DATA
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,JJ11
            RL      A
JJ11:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A             ;DISPLAY LOWER CHAR.
            CPL     P1.2                ;CLOCK 'LOW'
```

```
            LCALL   DELAY               ;DELAY
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A             ;CLEAR DATA
            INC     DPL
            MOVX    @DPTR,A
            MOV     A,RAM_L
            JZ      JJ9
            DEC     RAM_L
            LJMP    JJ2
JJ9:
            MOV     R0,CNTR3            ;LOAD BLANKS CNTR
JJ10:
            SETB    P1.2                ;CLOCK  'HIGH'
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A             ;WRITE 'ZERO'S' TO DISPLAY
            CPL     P1.2                ;CLOCK LOW
            LCALL   DELAY1
            DJNZ    R0,JJ10             ;? AT END OF DISPLAY
            INC     CNTR2               ;RAM_L CNTR
            MOV     RAM_L,CNTR2
            DJNZ    CNTR3,JJ3         ;? IS BLANK CNTR ZERO
            MOV     R4,#0FFH        ;SET REG FOR SCREEN DELAY
            LJMP    DISPDUAL
;-----------------------------------------------
;     ELECTRONIC MESSAGE DISPLAY
;          CHAR. SHIFT IN FROM LEFT OF SCREEN
;                        DUAL LINE STATIC CHARACTERS
;                              !!!starburst!!!
;-----------------------------------------------
RUN_STARBURST:
            MOV     RAM_H,NR_H
            MOV     RAM_L,NR_L
            MOV     CNTR1,#78H          ;DISPLAY LINE CNTR
            MOV     CNTR2,#77H          ;STARBURST DATA CNTR
            MOV     RAMVAL,#77H
            MOV     RVAL,#77H
            MOV     R4,#0FFH            ;DELAY
            MOV     DPTR,#8000H
            MOV     A,#00H              ; OF DISPLAY TO
            MOVX    @DPTR,A             ;  ALL ZERO'S
            INC     DPL                 ;NEXT PORT
            MOVX    @DPTR,A

;DISPLAY CHARACTERS
;-------------------------------------------------

KK9:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR             ;FETCH TOP CHAR. FROM RAM
            MOV     DPTR,#7F77H         ;TOP WORKSPACE
            MOVX    @DPTR,A             ;SHIFT DATA TO WORKSPACE
            MOV     A,RAM_L
            ADD     A,#78H              ;SETUP FOR BOTTOM CHAR.
            MOV     DPL,A
            MOV     DPH,RAM_H
            MOVX    A,@DPTR             ;FETCH DAT FOR BOTTOM CHAR.
            MOV     DPTR,#7FEFH         ;BOTTOM WORKSPACE
            MOVX    @DPTR,A
```

```
KK3:
        MOV     WORKVAL,#77H  ;ADD. CNTR OF WORKSPACE DATA
        MOV     P1,#00H         ;RESET SHIFT REGISTERS
        SETB    P1.0            ;SERIAL DATA = 'ONE'
        SETB    P1.1            ;ENABLE REGISTERS
        SETB    P1.2            ;CLOCK  'HIGH'
        CPL     P1.0            ;SERIAL DATA = 'ZERO'
        LJMP    KK1
KK2:
        SETB    P1.2            ;CLOCK 'HIGH'
KK1:
        MOV     DPH,#7FH
        MOV     DPL,WORKVAL
        MOVX    A,@DPTR                 ;FETCH DATA
        MOV     DPTR,#8000H
        MOVX    @DPTR,A             ;DISPLAY UPPER CHAR.
        MOV     A,WORKVAL
        ADD     A,#78H
        MOV     DPH,#7FH
        MOV     DPL,A
        MOVX    A,@DPTR           ;FETCH DATA
        MOV     R1,MODEDISP
        CJNE    R1,#0FFH,KK10
        RL      A
KK10:
        MOV     DPTR,#8001H
        MOVX    @DPTR,A           ;DISPLAY LOWER CHAR.
        CPL     P1.2              ;CLOCK 'LOW'
        DEC     WORKVAL
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A           ;CLEAR DATA
        INC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,KK2         ;? END OF DISPLAY

        MOV     DPH,#7FH
        MOV     DPL,RVAL          ;UPPER PART OF CHAR.
        MOVX    A,@DPTR
        MOV     R0,A              ;SAVE DATA OF WORKSPACE
        MOV     A,#00H
        MOVX    @DPTR,A       ;CLEAR DATA IN WORKSPACE
        DEC     DPL               ;SHIFT ONE POSITION
        MOV     A,R0
        MOVX    @DPTR,A     ;SETUP UPPER DATA FOR NEXT RUN
        MOV     A,RVAL
        ADD     A,#78H            ;LOWER PART OF CHAR.
        MOV     DPL,A
        MOVX    A,@DPTR           ;SAVE DATA OF WORKSPACE
        MOV     R0,A
        MOV     A,#00H
        MOVX    @DPTR,A        ;CLEAR DATA IN WORKSPACE
        DEC     DPL
        MOV     A,R0
        MOVX    @DPTR,A    ;SETUP LOWER DATA FOR NEXT RUN
        DEC     RVAL           ;SETUP NEXT SHIFT POSITION
        MOV     WORKVAL,#77H    ;RESET ADD. FOR DISPLAY
        MOV     CNTR1,#78H       ;RESET DISPLAY LINE CNTR
        DJNZ    RAMVAL,KK3          ;?END OF MESSAGE
        DEC     CNTR2
        MOV     RAMVAL,CNTR2
        INC     RAM_L
        MOV     RVAL,#77H
```

```
                MOV     A,CNTR2
                CJNE    A,#00H,DOWN

KK8:

                MOV     CNTR1,#78H
                MOV     RAMVAL,#77H
                MOV     R4,#0FFH
                MOV     DPTR,#7F00H
KK4:
                MOV     A,#00H
                MOVX    @DPTR,A
                INC     DPTR
                MOV     A,DPL
                CJNE    A,#0FFH,KK4             ;? IS WORKSPACE CLEARED
                LJMP    DISPDUAL
DOWN:
                LJMP    KK9

;-----------------------------------
;               SHIFT IN FROM LEFT
;               DUAL LINE DISPLAY
;               THEN STATIC
;-----------------------------------
RUN_LEFTSTAT:
                MOV     CNTR1,#78H
                MOV     CNTR2,#76H
                MOV     RAMVAL,#77H
SS4:
                MOV     RAM_L,RAMVAL
                MOV     P1,#00H
                SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                CPL     P1.0
                MOV     R0,CNTR2
                LJMP    SS1
SS2:
                SETB    P1.2
SS1:
                MOV     A,#00H
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                INC     DPL
                MOVX    @DPTR,A
                CPL     P1.2
                DEC     CNTR1
                LCALL   DELAY1
                DJNZ    R0,SS2
SS3:
                SETB    P1.2
                MOV     DPH,RAM_H
                MOV     DPL,RAM_L
                MOVX    A,@DPTR
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                MOV     A,RAM_L
                ADD     A,#78H
                MOV     DPH,RAM_H
                MOV     DPL,A
                MOVX    A,@DPTR
                MOV     R1,MODEDISP
                CJNE    R1,#0FFH,SS5
```

```
            RL      A
SS5:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,SS3
            MOV     CNTR1,#78H
            DJNZ    CNTR2,SS4
            MOV     R4,#0FFH
            LJMP    DISPDUAL
;-----------------------------------
;               DISPLAY DUAL LINES
;               CREATED FROM LEFT AND RIGHT
;               TOWARDS THE CENTRE
;-----------------------------------
LR_CENTRE:
            MOV     CNTR1,#3BH
            MOV     CNTR2,#01H
            MOV     CNTR3,#76H
            MOV     RAMVAL,#77H

            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
WW3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     R4,#02H
WW1:
            MOV     R0,CNTR2
WW2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPH,RAM_H
            MOV     DPL,A
            MOVX    A,@DPTR
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,WW6
            RL      A
WW6:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
```

```
                MOV     A,#00H
                MOVX    @DPTR,A
                DEC     DPL
                MOVX    @DPTR,A
                DJNZ    R0,WW2
                DJNZ    R4,WW4
                MOV     R4,#02H
                INC     CNTR2
                DJNZ    CNTR1,WW3
                MOV     CNTR1,#78H
                MOV     R4,#0FFH
                LJMP    DISPDUAL
WW4:
                MOV     R0,CNTR3
                DEC     CNTR3
                DEC     CNTR3
WW5:
                SETB    P1.2
                MOV     A,#00H
                MOVX    @DPTR,A
                INC     DPL
                MOVX    @DPTR,A
                CPL     P1.2
                LCALL   DELAY1
                DEC     RAM_L
                DJNZ    R0,WW5
                LJMP    WW1
;-------------------------------------
;               DISPLAY DUAL LINES
;               CREATED FROM THE CENTRE
;               TOWARDS LEFT AND RIGHT
;-------------------------------------
CENTRE_LR:
                MOV     CNTR1,#3BH
                MOV     CNTR2,#02H
                MOV     CNTR3,#3BH
                MOV     RAMVAL,#77H
XX3:
                MOV     RAM_L,RAMVAL
                MOV     P1,#00H
                SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                CPL     P1.0
                MOV     R4,#02H
XX1:
                MOV     R0,CNTR3
XX2:
                SETB    P1.2
                MOV     A,#00H
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                INC     DPL
                MOVX    @DPTR,A                 ;CLEAR DISPLAY DATA
                CPL     P1.2
                DEC     RAM_L
                LCALL   DELAY1
                DJNZ    R0,XX2
                DJNZ    R4,XX5
                MOV     R4,#02H
                INC     CNTR2
                INC     CNTR2
                DEC     CNTR3
```

```
            DJNZ    CNTR1,XX3
            MOV     CNTR1,#78H
            MOV     R4,#0FFH
            LJMP    DISPDUAL
XX5:
            MOV     R0,CNTR2
XX4:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPH,RAM_H
            MOV     DPL,A
            MOVX    A,@DPTR
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,XX6
            RL      A
XX6:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    R0,XX4
            LJMP    XX1
;************************************************************;
SINGLE LINE DISPLAY MODES
;************************************************************;
DISPLAY A SINGLE LINE IN
;               CENTRE OF THE SCREEN
;               IN STATIC MODE
;-----------------------------------
SRUNSTATIC:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R4,#0FFH
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A                 ;CLEAR DISPLAY DATA
            LJMP    DISPSINGLE
;-----------------------------------------
;       DISPLAY CHARACTERS STATIC SINGLE LINE
;-----------------------------------------
DISPSINGLE:
NN3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            LJMP    NN1
```

```
NN2:
            SETB    P1.2
NN1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R0,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,R0
            ANL     A,#0F0H
            SWAP    A
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,NN2
            MOV     CNTR1,#78H
            CJNE    R6,#01H,NN4
            MOV     R6,#00H
            RET
NN10:
            MOV     DPTR,#AUTOCLR
            MOV     A,CLEAR
            MOVC    A,@A+DPTR
            INC     CLEAR
            MOV     R7,CLEAR
            CJNE    R7,#05H,NN11
            MOV     CLEAR,#00H
            LJMP    NN11
NN9:
            RET
NN4:
            DJNZ    R4,NN3
            MOV     A,DISPLAY
            CJNE    A,#0FFH,NN10
            MOV     DPH,RAM_H
            MOV     DPL,#0F2H
            MOVX    A,@DPTR
            JZ      NN9
NN11:
            CJNE    A,#0CBH,NN5
            LCALL   CLRFTOP
            RET
NN5:
            CJNE    A,#0C8H,NN6
            LCALL   CLRFBOT
            RET
NN6:
            CJNE    A,#0D0H,NN7
            LCALL   CLRFCNR
            RET
NN7:
            CJNE    A,#0C3H,NN8
            LCALL   CLRFSDS
```

```
            RET
NN8:
            CJNE    A,#0F0H,NN9
            MOV     DPH,RAM_H
            MOV     DPL,#0F4H
            MOVX    A,@DPTR
            CJNE    A,#0C4H,NN12
            MOV     DPTR,#PUCMAN1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#PUCMAN2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#PUCMAN3
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
NN12:
            CJNE    A,#0CCH,NN13
            MOV     DPTR,#DUCK1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#DUCK2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#DUCK3
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
NN13:
            CJNE    A,#0D4H,NN11
            MOV     DPTR,#GRADER1
            MOV     PIC1_H,DPH
            MOV     PIC1_L,DPL
            MOV     DPTR,#GRADER2
            MOV     PIC2_H,DPH
            MOV     PIC2_L,DPL
            MOV     DPTR,#GRADER1
            MOV     PIC3_H,DPH
            MOV     PIC3_L,DPL
            LCALL   GRAFICS
            RET
;-----------------------------------------------
;               DISPLAY A SINGLE LINE
;               IN THE CENTRE OF THE
;               SCREEN SHIFT IN FROM
;               TOP AND BOTTOM
;-----------------------------------------------
SRUNTOPBOTTOM:
            MOV     MODESCRN,A
            MOV     CNTR1,#78H
            MOV     CNTR3,#07H
            MOV     RAMVAL,#77H
            MOV     LOOP1,#07H
            MOV     LOOP2,#07H
            MOV     LOOP3,#0FH
            MOV     LOOP4,#20H
            MOV     R4,#0FFH
            MOV     DPTR,#8000H
            MOV     A,#00H
            MOVX    @DPTR,A
```

```
        INC     A
        MOVX    @DPTR,A
QQ3:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
        LJMP    QQ1
QQ2:
        SETB    P1.2
QQ1:
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        MOV     R0,A
        ANL     A,#0FH
        SWAP    A
        MOV     LOOP1,CNTR3
QQ7:
        MOV     R1,MODESCRN
        CJNE    R1,#0D0H,QQ12
        CLR     C
        RLC     A
        LJMP    QQ13
QQ12:
        CLR     C
        RRC     A
QQ13:
        DJNZ    LOOP1,QQ7
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,R0
        ANL     A,#0F0H
        SWAP    A
        MOV     LOOP2,CNTR3
QQ8:
        MOV     R1,MODESCRN
        CJNE    R1,#0D0H,QQ10
        CLR     C
        RRC     A
        LJMP    QQ11
QQ10:
        CLR     C
        RLC     A
QQ11:
        DJNZ    LOOP2,QQ8
        MOV     DPTR,#8001H
        MOVX    @DPTR,A
        CPL     P1.2
        DEC     RAM_L
        MOV     A,#00H
        LCALL   DELAY1
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,QQ2
        MOV     CNTR1,#78H
        DJNZ    CNTR3,QQ3
QQ9:
        DJNZ    LOOP4,QQ9
        MOV     R4,#0FFH
```

```
            LJMP    DISPSINGLE
;-----------------------------------------------
;               DISPLAY A SINGLE LINE
;               IN CENTRE OF THE SCREEN
;               SHIFT IN FROM RIGHT TO
;               LEFT THEN STATIC
;-----------------------------------------------
SRUNRIGHTSTAT:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     CNTR2,#00H
            MOV     CNTR3,#77H
            MOV     RAM_L,#00H
            MOV     DPTR,#8000H
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
;DISPLAY CHARACTERS
RR3:
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            LJMP    RR1
RR2:
            SETB    P1.2
RR1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R0,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,R0
            ANL     A,#0F0H
            SWAP    A
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            MOV     A,RAM_L
            JZ      RR9
            DEC     RAM_L
            LJMP    RR2
RR9:
            MOV     R0,CNTR3
RR10:
            SETB    P1.2
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            CPL     P1.2
            LCALL   DELAY1
            DJNZ    R0,RR10
```

```
            INC     CNTR2
            MOV     RAM_L,CNTR2
            DJNZ    CNTR3,RR3
            MOV     R4,#0FFH
            LJMP    DISPSINGLE
;-----------------------------------------------
;       SHIFT IN FROM LEFT OF THE
;       SCREEN THEN DISPLAY STATIC
;-----------------------------------------------
SRUNLEFTSTAT:
            MOV     CNTR1,#78H
            MOV     CNTR2,#76H
            MOV     RAMVAL,#77H
TT4:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     R0,CNTR2
            LJMP    TT1
TT2:
            SETB    P1.2
TT1:
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     CNTR1
            LCALL   DELAY1
            DJNZ    R0,TT2
TT3:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R0,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,R0
            ANL     A,#0F0H
            SWAP    A
            INC     DPL
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,TT3
            MOV     CNTR1,#78H
            DJNZ    CNTR2,TT4
            MOV     R4,#0FFH
            LJMP    DISPSINGLE
```

```
;-------------------------------------
;               DISPLAY A SINGLE LINE
;               CREATED FROM LEFT AND
;               RIGHT TOWARDS THE CENTRE
;-------------------------------------
LRCENTRE:
            MOV     MODESCRN,A
            MOV     CNTR1,#3BH
            MOV     CNTR2,#01H
            MOV     CNTR3,#76H
            MOV     RAMVAL,#77H

            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A          ;CLEAR DISPLAY DATA
YY3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     R4,#02H
YY1:
            MOV     R0,CNTR2
YY2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R3,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,R3
            ANL     A,#0F0H
            SWAP    A
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    R0,YY2
            DJNZ    R4,YY4
            MOV     R4,#02H
            INC     CNTR2
            DJNZ    CNTR1,YY3
            MOV     CNTR1,#78H
            MOV     R4,#0FFH
            LJMP    DISPSINGLE
YY4:
            MOV     R0,CNTR3
            DEC     CNTR3
            DEC     CNTR3
YY5:
            SETB    P1.2
```

```
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            CPL     P1.2
            LCALL   DELAY1
            DEC     RAM_L
            DJNZ    R0,YY5
            LJMP    YY1
;-----------------------------------
;       DISPLAY SINGLE LINES
;       CREATED FROM THE CENTRE
;       TOWARDS LEFT AND RIGHT
;-----------------------------------
CENTRELR:
            MOV     MODESCRN,A
            MOV     CNTR1,#3BH
            MOV     CNTR2,#02H
            MOV     CNTR3,#3BH
            MOV     RAMVAL,#77H
ZZ3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     R4,#02H
ZZ1:
            MOV     R0,CNTR3
ZZ2:
            SETB    P1.2
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A                 ;CLEAR DISPLAY DATA
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            DJNZ    R0,ZZ2
            DJNZ    R4,ZZ5
            MOV     R4,#02H
            INC     CNTR2
            INC     CNTR2
            DEC     CNTR3
            DJNZ    CNTR1,ZZ3
            MOV     CNTR1,#78H
            MOV     R4,#0FFH
            LJMP    DISPSINGLE
ZZ5:
            MOV     R0,CNTR2
ZZ4:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R1,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,R1
```

```
            ANL     A,#0F0H
            SWAP    A
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    R0,ZZ4
            LJMP    ZZ1
;-----------------------------------------------
;ROTATE SMALL LETTERS CONTINUES THROUGH
;       A SINGLE LINE FROM RIGHT
;       TO LEFT ON THE SCREEN
;-----------------------------------------------
RUN_ROTATE:
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
            MOV     R7,#00H             ;CLEAR INT. REG.
MM11:
            MOV     RAM_H,NR_H
            MOV     RAM_L,NR_L
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     CNTR2,#00H
            MOV     CNTR3,#77H
            MOV     R3,#00H
            MOV     DPTR,#8000H
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A             ;SET DISP. LINES = ZERO

;DISPLAY CHARACTERS
MM3:
            MOV     P1,#00H                 ;RESET SHIFT REG'S
            SETB    P1.0                    ;SERIAL DATA = ONE
            SETB    P1.1                    ;ENABLE REG'S
            SETB    P1.2                    ;CLOCK 'HIGH'
            CPL     P1.0                    ;SERIAL DATA 'LOW'
            LJMP    MM1
MM2:
            SETB    P1.2                    ;CLOCK 'HIGH'
MM1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR                 ;FETCH DATA
            CJNE    A,#0DH,MM20             ;? END OF DATA ROW
            MOV     A,#00H
            MOV     R3,#0FFH
MM20:
            MOV     R4,A
            ANL     A,#0FH
            SWAP    A
            MOV     DPTR,#8000H
            MOVX    @DPTR,A         ;DISPLAY UPPER PART OF CHAR.
            INC     DPL
            MOV     A,R4
            ANL     A,#0F0H
            SWAP    A
```

```
        MOVX    @DPTR,A       ;DISPLAY LOWER PART OF CHAR.
        CPL     P1.2                    ;CLOCK 'LOW'
        LCALL   DELAY
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A             ;CLEAR DISPLAY DATA
        MOV     A,RAM_L
        JZ      MM9
        DEC     RAM_L
        LJMP    MM2
MM9:
        MOV     R0,CNTR3            ;LOAD BLANKS CNTR
MM10:
        SETB    P1.2                ;CLOCK 'HIGH'
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A             ;CLEAR DISPLAY DATA
        CPL     P1.2                ;CLOCK 'LOW'
        LCALL   DELAY1
        DJNZ    R0,MM10             ;? AT END OF DISPLAY

        INC     CNTR2               ;RAM_L CNTR
        MOV     RAM_L,CNTR2
        DJNZ    CNTR3,MM3           ;? IS BLANK CNTR 'ZERO'
        MOV     R1,#00H         ;RESET BLANK INDICATION REG.
        MOV     CNTR3,#01H          ;SET BLANK CNTR
        MOV     RAM_H,NR_H
MM8:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
        LJMP    MM5
MM6:
        SETB    P1.2
MM5:
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        CJNE    A,#0DH,MM7
        LJMP    MM14
MM7:
        MOV     R4,A
        ANL     A,#0FH
        SWAP    A
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOV     A,R4
        ANL     A,#0F0H
        SWAP    A
        MOVX    @DPTR,A
        CPL             P1.2
        DEC     RAM_L
        MOV     A,RAM_L
        CJNE    A,#0FFH,MM12
        DEC     RAM_H
```

```
        MOV     R0,#0FFH
MM12:
        LCALL   DELAY1
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,MM6               ;? END OF DISPLAY
        MOV     CNTR1,#78H              ;RESET CNTR1
        CJNE    R7,#0FAH,MM19   ;? ESCAPE KEY PRESSED
        MOV     R7,#00H
        RET
MM19:
        MOV     A,R0
        CJNE    A,#0FFH,MM13
        INC     RAM_H
        MOV     R0,#00H
MM13:
        MOV     A,R1                    ;LOAD BLANK REG.
        CJNE    A,#0FFH,MM16    ;? IS BLANK SEQUENCE SET
        LJMP    MM14
MM16:
        MOV     A,RAMVAL
        CLR     C
        ADD     A,#01H      ;SETUP RAMVAL FOR NEXT CICLE
        MOV     RAMVAL,A
        JNC     MM4
        INC     RAM_H
MM4:
        LJMP    MM8
MM14:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
        MOV     R2,CNTR3
MM15:
        SETB    P1.2                    ;CLOCK HIGH
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A         ;WRITE ZERO'S TO DISPLAY
        CPL     P1.2             ;CLOCK 'LOW'
        DEC     CNTR1             ;DISPLAY LINE CNTR
        MOV     A,RAM_L
        CJNE    A,#0FFH,MM18
        DEC     RAM_H
MM18:
        MOV     R1,#0FFH            ;SET BLANK REG
        LCALL   DELAY1
        DJNZ    R2,MM15         ;? BLANKS TO WRITE TO DISP.
        INC     CNTR3
        MOV     A,CNTR3
        CJNE    A,#78H,MM17
        LJMP    MM11
MM17:
        DEC     RAM_L
        LJMP    MM6
```

```
;-----------------------------------------------
;ROTATE BIG LETTERS CONTINUES THROUGH
;       A SINGLE LINE FROM RIGHT
;       TO LEFT ON THE SCREEN
;-----------------------------------------------
RUN_ROTATEBIG:
            MOV     NR_H,RAM_H
            MOV     NR_L,RAM_L
            MOV     R7,#00H         ;CLEAR INT. REG.
MMM11:
            MOV     RAM_H,NR_H
            MOV     RAM_L,NR_L
            MOV     CNTR1,#78H
            MOV     RAMVAL,#0EFH
            MOV     CNTR2,#00H
            MOV     CNTR3,#77H              ;BLANKS CNTR
            MOV     R3,#00H
            MOV     DPTR,#8000H
            MOV     A,#00H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;SET DISP. LINES = ZERO

;DISPLAY CHARACTERS
MMM3:
            MOV     P1,#00H         ;RESET SHIFT REG'S
            SETB    P1.0            ;SERIAL DATA = ONE
            SETB    P1.1            ;ENABLE REG'S
            SETB    P1.2            ;CLOCK 'HIGH'
            CPL     P1.0            ;SERIAL DATA 'LOW'
            LJMP    MMM1
MMM2:
            SETB    P1.2            ;CLOCK 'HIGH'
MMM1:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR         ;FETCH DATA
            CJNE    A,#0AH,MMM20    ;? END OF DATA ROW
            MOV     A,#00H
            MOV     R3,#0FFH
MMM20:
            MOV     DPTR,#8000H
            MOVX    @DPTR,A      ;DISPLAY UPPER PART OF CHAR.
            INC     RAM_L
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     DPTR,#8001H
            MOVX    @DPTR,A      ;DISPLAY LOWER PART OF CHAR.
            DEC     RAM_L
            CPL     P1.2            ;CLOCK 'LOW'
            LCALL   DELAY
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
            MOV     A,RAM_L
            JZ      MMM9
            DEC     RAM_L
            DEC     RAM_L
            LJMP    MMM2
```

```
MMM9:
            MOV     R0,CNTR3        ;LOAD BLANKS CNTR
MMM10:
            SETB    P1.2            ;CLOCK 'HIGH'
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A         ;WRITE BLANKS
            INC     DPL
            MOVX    @DPTR,A         ;CLEAR DISPLAY DATA
            CPL     P1.2            ;CLOCK 'LOW'
            LCALL   DELAY1
            DJNZ    R0,MMM10        ;? AT END OF DISPLAY

            INC     CNTR2           ;RAM_L CNTR
            INC     CNTR2
            MOV     RAM_L,CNTR2
            DJNZ    CNTR3,MMM3      ;? IS BLANK CNTR 'ZERO'
            MOV     R1,#00H       ;RESET BLANK INDICATION REG.
            MOV     CNTR3,#01H      ;SET BLANK CNTR
            MOV     RAM_H,NR_H
MMM8:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            LJMP    MMM5
MMM6:
            SETB    P1.2
MMM5:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            CJNE    A,#0AH,MMM7
            MOV     R2,CNTR3
            LJMP    MMM15
MMM7:
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            DEC     RAM_L
            MOV     A,RAM_L
            CJNE    A,#0FFH,MMM21
            DEC     RAM_H
MMM21:
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            CJNE    A,#0AH,MMM23
            LJMP    MMM14
MMM23:
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            MOV     A,RAM_L
            CJNE    A,#0FFH,MMM12
            DEC     RAM_H
            MOV     R0,#0FFH
MMM12:
            LCALL   DELAY1
            MOV     A,#00H
            MOV     DPTR,#8000H
```

```
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A          ;CLEAR DISPLAY
            DJNZ    CNTR1,MMM6       ;? END OF DISPLAY
            MOV     CNTR1,#78H       ;RESET CNTR1
            CJNE    R7,#0FAH,MMM19   ;? ESCAPE KEY PRESSED
            MOV     R7,#00H
            RET
MMM19:
            MOV     A,R0
            CJNE    A,#0FFH,MMM13
            INC     RAM_H
            MOV     R0,#00H
MMM13:
            MOV     A,R1             ;LOAD BLANK REG.
            CJNE    A,#0FFH,MMM16    ;? IS BLANK SEQUENCE SET
            LJMP    MMM14
MMM16:
            MOV     A,RAMVAL
            CLR     C
            ADD     A,#02H        ;SETUP RAMVAL FOR NEXT CICLE
            MOV     RAMVAL,A
            JNC     MMM4
            INC     RAM_H
MMM4:
            LJMP    MMM8
MMM14:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
            MOV     R2,CNTR3
MMM15:
            SETB    P1.2             ;CLOCK HIGH
            MOV     A,#00H
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A          ;WRITE ZERO'S TO DISPLAY
            CPL     P1.2             ;CLOCK 'LOW'
            DEC     CNTR1            ;DISPLAY LINE CNTR
MMM18:
            MOV     R1,#0FFH         ;SET BLANK REG
            LCALL   DELAY1
            DJNZ    R2,MMM15      ;? BLANKS TO WRITE TO DISP.
            INC     CNTR3     ;DISPLAY ONE BLANK MORE NEXT RUN
            MOV     A,CNTR3
            CJNE    A,#78H,MMM17     ;? SCREEN FULL OF BLANKS
            LJMP    MMM11            ;START DISPLAY OVER
MMM17:
            DEC     RAM_L
            DEC     RAM_L
            LJMP    MMM6       ;CONTINUE WITH DISPLAY SEQUENCE
;****************************************
;       CLEAR SCREEN MODES
;----------------------------------------
;       CLEAR SCREEN FROM THE TOP
;               DUAL LINES
;----------------------------------------
CLRF_TOP:
            MOV     CNTR1,#78H
```

```
        MOV     RAMVAL,#77H
        MOV     R0,#0FFH
        MOV     R4,#0FFH
        MOV     LOOP1,#04H
        MOV     A,#00H
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        INC     DPL
        MOVX    @DPTR,A
        LJMP    TOP4
TOP1:
        MOV     A,R4
        CLR     C
        RLC     A
        MOV     R4,A
TOP4:
        MOV     A,R0
        CLR     C
        RLC     A
        MOV     R0,A
TOP3:
        MOV     RAM_L,RAMVAL
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
TOP2:
        SETB    P1.2
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        ANL     A,R0
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,RAM_L
        ADD     A,#78H
        MOV     DPL,A
        MOV     DPH,RAM_H
        MOVX    A,@DPTR
        MOV     R1,MODEDISP
        CJNE    R1,#0FFH,TOP5
        RL      A
TOP5:
        ANL     A,R4
        MOV     DPTR,#8001H
        MOVX    @DPTR,A
        CPL     P1.2
        DEC     RAM_L
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,TOP2
        MOV     CNTR1,#78H
        DJNZ    LOOP1,TOP3
        MOV     LOOP1,#04H
        MOV     A,R0
        JNZ     TOP4
        MOV     XA,R4
        JNZ     TOP1
        RET
```

```
;-------------------------------------
;       CLEAR SCREEN FROM THE BOTTOM
;                DUAL LINES
;-------------------------------------
CLRF_BOT:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R0,#0FFH
            MOV     R4,#0FFH
            MOV     LOOP1,#04H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            LJMP    BOT4
BOT1:
            MOV     A,R0
            CLR     C
            RRC     A
            MOV     R0,A
BOT4:
            MOV     A,R4
            CLR     C
            RRC     A
            MOV     R4,A
BOT3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     XP1.0
BOT2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            ANL     A,R0
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPL,A
            MOV     DPH,RAM_H
            MOVX    A,@DPTR
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,BOT5
            RL      A
BOT5:
            ANL     A,R4
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,BOT2
            MOV     CNTR1,#78H
            DJNZ    LOOP1,BOT3
```

```
            MOV     LOOP1,#04H
            MOV     A,R4
            JNZ     BOT4
            MOV     A,R0
            JNZ     BOT1
            RET
;-------------------------------
;       CLEAR SCREEN FROM THE CENTRE
;               DUAL LINES
;-------------------------------
CLRF_CNR:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R0,#0FFH
            MOV     R4,#0FFH
            MOV     LOOP1,#04H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
CNR1:
            MOV     A,R4
            CLR     C
            RLC     A
            MOV     R4,A
            MOV     A,R0
            CLR     C
            RRC     A
            MOV     R0,A
CNR3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
CNR2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            ANL     A,R0
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPL,A
            MOV     DPH,RAM_H
            MOVX    A,@DPTR
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,CNR5
            RL      A
CNR5:
            ANL     A,R4
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
```

```
            MOVX    @DPTR,A
            DJNZ    CNTR1,CNR2
            MOV     CNTR1,#78H
            DJNZ    LOOP1,CNR3
            MOV     LOOP1,#04H
            MOV     A,R0
            JNZ     CNR1
            RET
;---------------------------------------
;       CLEAR SCREEN FROM TOP AND BOTTOM
;                DUAL LINES
;---------------------------------------
CLRF_SDS:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R0,#0FFH
            MOV     R4,#0FFH
            MOV     LOOP1,#04H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
SDS1:
            MOV     A,R4
            CLR     C
            RRC     A
            MOV     R4,A
            MOV     A,R0
            CLR     C
            RLC     A
            MOV     R0,A
SDS3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
SDS2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            ANL     A,R0
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,RAM_L
            ADD     A,#78H
            MOV     DPL,A
            MOV     DPH,RAM_H
            MOVX    A,@DPTR
            MOV     R1,MODEDISP
            CJNE    R1,#0FFH,SDS5
            RL      A
SDS5:
            ANL     A,R4
            MOV     DPTR,#8001H
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
```

```
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,SDS2
            MOV     CNTR1,#78H
            DJNZ    LOOP1,SDS3
            MOV     LOOP1,#04H
            MOV     A,R0
            JNZ     SDS1
            RET
;-------------------------------------
;       CLEAR SCREEN FROM THE TOP
;               SINGLE LINE
;-------------------------------------
CLRFTOP:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R0,#0FFH
            MOV     R4,#0FFH
            MOV     LOOP1,#04H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
            LJMP    TOPS4
TOPS1:
            MOV     A,R4
            CLR     C
            RLC     A
            MOV     R4,A
TOPS4:
            MOV     A,R0
            CLR     C
            RLC     A
            MOV     R0,A
TOPS3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
TOPS2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R1,A
            ANL     A,#0FH
            SWAP    A
            ANL     A,R0
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOV     A,R1
            ANL     A,#0F0H
            SWAP    A
            ANL     A,R4
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
```

```
                MOV     A,#00H
                MOVX    @DPTR,A
                DEC     DPL
                MOVX    @DPTR,A
                DJNZ    CNTR1,TOPS2
                MOV     CNTR1,#78H
                DJNZ    LOOP1,TOPS3
                MOV     LOOP1,#04H
                MOV     A,R0
                JNZ     TOPS4
                MOV     A,R4
                JNZ     TOPS1
                RET
;-----------------------------------
;       CLEAR SCREEN FROM THE BOTTOM
;               SINGLE LINE
;-----------------------------------
CLRFBOT:
                MOV     CNTR1,#78H
                MOV     RAMVAL,#77H
                MOV     R0,#0FFH
                MOV     R4,#0FFH
                MOV     LOOP1,#04H
                MOV     A,#00H
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                INC     DPL
                MOVX    @DPTR,A
                LJMP    BOTS4
BOTS1:
                MOV     A,R0
                CLR     C
                RRC     A
                MOV     R0,A
BOTS4:
                MOV     A,R4
                CLR     C
                RRC     A
                MOV     R4,A
BOTS3:
                MOV     RAM_L,RAMVAL
                MOV     P1,#00H
                SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                CPL     P1.0
BOTS2:
                SETB    P1.2
                MOV     DPH,RAM_H
                MOV     DPL,RAM_L
                MOVX    A,@DPTR
                MOV     R1,A
                ANL     A,#0FH
                SWAP    A
                ANL     A,R0
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                MOV     A,R1
                ANL     A,#0F0H
                SWAP    A
                ANL     A,R4
                INC     DPL
                MOVX    @DPTR,A
```

```
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,BOTS2
            MOV     CNTR1,#78H
            DJNZ    LOOP1,BOTS3
            MOV     LOOP1,#04H
            MOV     A,R4
            JNZ     BOTS4
            MOV     A,R0
            JNZ     BOTS1
            RET
;-----------------------------------
;       CLEAR SCREEN FROM THE BOTTOM
;               SINGLE LINE
;-----------------------------------
CLRFCNR:
            MOV     CNTR1,#78H
            MOV     RAMVAL,#77H
            MOV     R0,#0FFH
            MOV     R4,#0FFH
            MOV     LOOP1,#04H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A
CNRS1:
            MOV     A,R4
            CLR     C
            RLC     A
            MOV     R4,A
            MOV     A,R0
            CLR     C
            RRC     A
            MOV     R0,A
CNRS3:
            MOV     RAM_L,RAMVAL
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0
CNRS2:
            SETB    P1.2
            MOV     DPH,RAM_H
            MOV     DPL,RAM_L
            MOVX    A,@DPTR
            MOV     R1,A
            ANL     A,#0FH
            SWAP    A
            ANL     A,R0
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            MOV     A,R1
            ANL     A,#0F0H
            SWAP    A
            ANL     A,R4
            INC     DPL
```

```
                MOVX    @DPTR,A
                CPL     P1.2
                DEC     RAM_L
                LCALL   DELAY1
                MOV     A,#00H
                MOVX    @DPTR,A
                DEC     DPL
                MOVX    @DPTR,A
                DJNZ    CNTR1,CNRS2
                MOV     CNTR1,#78H
                DJNZ    LOOP1,CNRS3
                MOV     LOOP1,#04H
                MOV     A,R0
                JNZ     CNRS1
                RET
;-----------------------------------
;       CLEAR SCREEN FROM THE BOTTOM
;               SINGLE LINE
;-----------------------------------
CLRFSDS:
                MOV     CNTR1,#78H
                MOV     RAMVAL,#77H
                MOV     R0,#0FFH
                MOV     R4,#0FFH
                MOV     LOOP1,#04H
                MOV     A,#00H
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                INC     DPL
                MOVX    @DPTR,A
SDSS1:
                MOV     A,R4
                CLR     C
                RRC     A
                MOV     R4,A
                MOV     A,R0
                CLR     C
                RLC     A
                MOV     R0,A
SDSS3:
                MOV     RAM_L,RAMVAL
                MOV     P1,#00H
                SETB    P1.0
                SETB    P1.1
                SETB    P1.2
                CPL     P1.0
SDSS2:
                SETB    P1.2
                MOV     DPH,RAM_H
                MOV     DPL,RAM_L
                MOVX    A,@DPTR
                MOV     R1,A
                ANL     A,#0FH
                SWAP    A
                ANL     A,R0
                MOV     DPTR,#8000H
                MOVX    @DPTR,A
                MOV     A,R1
                ANL     A,#0F0H
                SWAP    A
                ANL     A,R4
                INC     DPL
                MOVX    @DPTR,A
```

```
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,SDSS2
            MOV     CNTR1,#78H
            DJNZ    LOOP1,SDSS3
            MOV     LOOP1,#04H
            MOV     A,R0
            JNZ     SDSS1
            RET
;---------------------------------------
;       CLEAR THE SCREEN WITH A PUCMAN
;---------------------------------------
GRAFICS:
            MOV     CNTR1,#78H          ;DISP. LINE COUNTER
            MOV     CNTR2,#00H          ;PUCMAN ADDRESS COUNTER
            MOV     CNTR3,#78H        ;SCREENS COMPLETED COUNTER
            MOV     CNTR4,#09H          ;PUCMAN MOUTH COUNTER
            MOV     RAMVAL,#77H
            MOV     A,#00H
            MOV     DPTR,#8000H
            MOVX    @DPTR,A
            INC     DPL
            MOVX    @DPTR,A             ;CLEAR DISPLAY DATA
GRAF4:
            MOV     RAM_L,RAMVAL      ;RELOAD START OF RAM
            MOV     R2,CNTR2
            MOV     P1,#00H
            SETB    P1.0
            SETB    P1.1
            SETB    P1.2
            CPL     P1.0               ;SETUP DISPLAY FOR START
GRAF1:
            SETB    P1.2
            MOV     A,R2
            MOV     DPH,PIC1_H
            MOV     DPL,PIC1_L
            MOVC    A,@A+DPTR                 ;FETCH DATA
            ANL         A,#0FFH
            MOV     DPTR,#8000H
            MOVX    @DPTR,A                   ;DISPLAY TOP LINE
            MOV     A,R2
            ADD     A,#78H
            MOV     DPH,PIC1_H
            MOV     DPL,PIC1_L
            MOVC    A,@A+DPTR                 ;FETCH DATA
            ANL     A,#0FFH
            MOV     DPTR,#8001H
            MOVX    @DPTR,A                 ;DISPLAY BOTTOM LINE
            CPL     P1.2
            DEC     RAM_L
            DEC     CNTR1
            DEC     R2                  ;SETUP DATA FOR NEXT RUN
            LCALL   DELAY1                ;DELAY ROUTINE
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            MOV     A,R2
```

```
        CJNE    A,#0FFH,GRAF1   ;? END OF PUCMAN DATA FILE
        MOV     R4,#00H
        LJMP    GRAF2
GRAF6:
        MOV     RAM_L,RAMVAL
        MOV     R2,CNTR2
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
GRAF7:
        SETB    P1.2
        MOV     A,R2
        MOV     DPH,PIC2_H
        MOV     DPL,PIC2_L
        MOVC    A,@A+DPTR
        ANL       A,#0FFH
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,R2
        ADD     A,#78H
        MOV     DPH,PIC2_H
        MOV     DPL,PIC2_L
        MOVC    A,@A+DPTR
        ANL     A,#0FFH
        MOV     DPTR,#8001H
        MOVX    @DPTR,A
        CPL     P1.2
        DEC     RAM_L
        DEC     CNTR1
        DEC     R2
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        MOV     A,R2
        CJNE    A,#0FFH,GRAF7
        MOV     R4,#01H
        LJMP    GRAF2
GRAF8:
        MOV     RAM_L,RAMVAL
        MOV     R2,CNTR2
        MOV     P1,#00H
        SETB    P1.0
        SETB    P1.1
        SETB    P1.2
        CPL     P1.0
GRAF9:
        SETB    P1.2
        MOV     A,R2
        MOV     DPH,PIC3_H
        MOV     DPL,PIC3_L
        MOVC    A,@A+DPTR
        ANL     A,#0FFH
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,R2
        ADD     A,#78H
        MOV     DPH,PIC3_H
        MOV     DPL,PIC3_L
        MOVC    A,@A+DPTR
```

```
        ANL     A,#0FFH
        MOV     DPTR,#8001H
        MOVX    @DPTR,A
        CPL     P1.2
        DEC     RAM_L
        DEC     CNTR1
        DEC     R2
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        MOV     A,R2
        CJNE    A,#0FFH,GRAF9
        MOV     R4,#02H

GRAF2:
        MOV     R1,MODEDISP
        CJNE    R1,#0FDH,GRAF16 ;? DISTINGUISH BETWEEN
                                        DISP. MODES
GRAF15:
        SETB    P1.2
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L  ;DISPLAY DATA TO BE CLEARED
        MOVX    A,@DPTR
        MOV     R0,A
        ANL     A,#0FH
        SWAP    A
        MOV     DPTR,#8000H
        MOVX    @DPTR,A                         ;TOP LINE
        MOV     A,R0
        ANL     A,#0F0H
        SWAP    A
        MOV     DPTR,#8001H
        MOVX    @DPTR,A                 ;BOTTOM LINE
        CPL     P1.2
        DEC     RAM_L
        LCALL   DELAY1
        MOV     A,#00H
        MOVX    @DPTR,A
        DEC     DPL
        MOVX    @DPTR,A
        DJNZ    CNTR1,GRAF15            ;? END OF DISPLAY
        MOV     CNTR1,#78H
        INC     CNTR2
        DJNZ    CNTR3,GRAF5              ;? END OF SCREENS
        RET
GRAF16:
        SETB    P1.2
        MOV     DPH,RAM_H
        MOV     DPL,RAM_L
        MOVX    A,@DPTR
        MOV     DPTR,#8000H
        MOVX    @DPTR,A
        MOV     A,RAM_L
        ADD     A,#78H
        MOV     DPH,RAM_H
        MOV     DPL,A
        MOVX    A,@DPTR
        CJNE    R1,#0FFH,GRAF17
        RL      A
GRAF17:
        MOV     DPTR,#8001H
```

```
            MOVX    @DPTR,A
            CPL     P1.2
            DEC     RAM_L
            LCALL   DELAY1
            MOV     A,#00H
            MOVX    @DPTR,A
            DEC     DPL
            MOVX    @DPTR,A
            DJNZ    CNTR1,GRAF16
            MOV     CNTR1,#78H
            INC     CNTR2
            DJNZ    CNTR3,GRAF5
            RET
GRAF5:
            CJNE    R4,#00H,GRAF10
            DJNZ    CNTR4,GRAF14
            MOV     CNTR4,#09H
GRAF12:
            LJMP    GRAF6
GRAF10:
            CJNE    R4,#01H,GRAF11
            DJNZ    CNTR4,GRAF12
            MOV     CNTR4,#09H
            MOV     A,R3
            JNZ     GRAF14
GRAF13:
            INC     R3
            LJMP    GRAF8
GRAF11:
            DJNZ    CNTR4,GRAF13
            MOV     CNTR4,#09H
            LJMP    GRAF6
GRAF14:
            MOV     R3,#00H
            LJMP    GRAF4
            RET
;************************************************************
;                       SCREEN TABLES AND DATA TABLES
;************************************************************

SCNT1:  DB      '**** WELCOME TO ******** TECHSIGNS *****'
SCNT2:  DB      'RECALL SAVED PROGRAM RUN/YES PROGRAM/NO '
SCNT3:  DB      '  SELECT PROGRAM        NUMBER 1 TO 10   '
SCNT4:  DB      'ENTER FROM KEYBOARD   THE NEXT SCREEN    '
SCNT5:  DB      'CHOOSE DISPLAY MODE    DUAL OR SINGLE    '
SCNT6:  DB      ' SELECT SCREEN MODE   S/T/C/L/R/I/O/B    '
SCNT7:  DB      'SELECT NEXT SCREEN  ENTER/Y PROGRAM/SAVE'
SCNT8:  DB      'SELECT LETTER SIZE      BIG OR SMALL     '
SCNT9:  DB      '***** WARNING ******  EXISTING PROGRAM   '
SCNT10: DB      ' OVERWRITE PROGRAM   Y/YES V/VIEW N/NO   '
SCNT11: DB      ' SELECT CLEAR MODE  N/T/B/C/S/SPEC.FUNC.'
SCNT12: DB      'EASY KEY MODE >ENTERSPECIAL KEYS>SPECIAL'
SCNT13: DB      ' AUTO MODE > ENTER  SPECIAL MODE>SPECIAL'
SCNT14: DB      '** END OF PROGRAM ****** WORKSPACE *****'
SCNT15: DB      '  SELECT GRAPHICS       NUMBER 1 TO 5    '
SCNT16: DB      '  SELECT PROGRAM        NUMBER 1 TO 9    '
SCNT17: DB      '########################################'

PUCMAN1:
      DB 0C0H,0F0H,0F8H,0FCH,0FCH,0E6H,0E6H,0FEH,0FEH,0FCH,0FCH,0F8H
      DB      0F0H,0C0H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
      DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
      DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

```
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB      03H,0FH,1FH,3FH,3FH,7FH,7FH,7FH,7FH,3FH,3FH,1FH
        DB      0FH,03H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

PUCMAN2:
        DB      00H,10H,38H,7CH,0FCH,0E6H,0E6H,0FEH,0FEH,0FCH,0FCH
        DB      0F8H,0F0H,0C0H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB      00H,08H,1CH,3EH,3FH,7FH,7FH,7FH,7FH,3FH,3FH,1FH
        DB      0FH,03H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

PUCMAN3:
        DB      00H,00H,08H,1CH,3CH,66H,0E6H,0FEH,0FEH,0FCH,0FCH,0F8H
        DB      0F0H,0C0H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB      00H,00H,10H,38H,3CH,7EH,7FH,7FH,7FH,3FH,3FH,1FH
        DB      0FH,03H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

```
DUCK1:
        DB
08H,08H,1CH,9EH,0FFH,0FFH,0FFH,0EEH,0E0H,0E0H,0E0H,0E0H
        DB
0E0H,0C0H,0C0H,20H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB                  00H,00H,1CH,13H,17H,1FH,1FH,0FH,0FH,0FH,0FH,97H
        DB
0A3H,41H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

DUCK3:
        DB
00H,00H,08H,88H,0DCH,0FEH,0FFH,0FFH,0FFH,0EEH,0E0H,0E0H
        DB
0E0H,0C0H,0C0H,20H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB
00H,00H,00H,03H,0C7H,8FH,3FH,0FH,0FH,0FH,0FH,0FH
        DB                  0BH,29H,18H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

DUCK2:
        DB
00H,08H,08H,9CH,0DEH,0FFH,0FFH,0FFH,0EEH,0E0H,0E0H,0E0H
        DB
0E0H,0C0H,0C0H,20H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB                  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

```
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB
00H,00H,00H,03H,67H,8FH,4FH,3FH,0BFH,4FH,0FH,07H
        DB              03H,01H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

GRADER1:
        DB
00H,00H,00H,00H,00H,0C0H,0FCH,0C2H,0C0H,0C0H,0F8H,0FCH
        DB
0E6H,0C3H,0C3H,0C3H,0C3H,0FFH,0FFH,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB
81H,7EH,3CH,26H,3BH,55H,0D6H,0BAH,0D6H,82H,0C6H,82H
        DB
0C6H,92H,0FEH,92H,0C6H,55H,3BH,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

GRADER2:
        DB
00H,00H,00H,00H,00H,0C0H,0FCH,0C2H,0C0H,0C0H,0F8H,0FCH
        DB
0E6H,0C3H,0C3H,0C3H,0C3H,0FFH,0FFH,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

        DB
81H,7EH,3CH,26H,3BH,6DH,0AAH,0D6H,0AAH,0C6H,82H,0C6H
        DB
82H,0EEH,92H,0EEH,82H,6DH,3BH,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

```
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB              00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

KEYTABLE:
        ;COL0
        DB      7EH,11H,11H,11H,7EH,00H ;A
        DB      3EH,41H,49H,49H,3AH,00H ;G
        DB      7FH,02H,0CH,02H,7FH,00H ;M
        DB      26H,49H,49H,49H,32H,00H ;S
        DB      00H,42H,7FH,40H,00H,00H ;1
        DB      18H,14H,12H,7FH,10H,00H ;4
        DB      01H,71H,09H,05H,03H,00H ;7
        DB      08H,08H,3EH,08H,08H,00H ;+
        ;COL1
        DB      7FH,49H,49H,49H,36H,00H ;B
        DB      7FH,08H,08H,08H,7FH,00H ;H
        DB      7FH,04H,08H,10H,7FH,00H ;N
        DB      01H,01H,7FH,01H,01H,00H ;T
        DB      42H,61H,51H,49H,46H,00H ;2
        DB      27H,45H,45H,45H,39H,00H ;5
        DB      36H,49H,49H,49H,36H,00H ;8
        DB      3EH,51H,49H,45H,3EH,00H ;0
        ;COL2
        DB      3EH,41H,41H,41H,22H,00H ;C
        DB      00H,41H,7FH,41H,00H,00H ;I
        DB      3EH,41H,41H,41H,3EH,00H ;O
        DB      3FH,40H,40H,40H,3FH,00H ;U
        DB      21H,41H,45H,4BH,31H,00H ;3
        DB      3CH,4AH,49H,49H,30H,00H ;6
        DB      06H,49H,49H,29H,1EH,00H ;9
        DB      14H,08H,3EH,08H,14H,00H ;*
        ;COL3
        DB      7FH,41H,41H,22H,1CH,00H ;D
        DB      20H,40H,41H,3FH,01H,00H ;J
        DB      7FH,09H,09H,09H,06H,00H ;P
        DB      1FH,20H,40H,20H,1FH,00H ;V
        DB      00H,60H,60H,00H,00H,00H ;.
        DB      00H,50H,30H,00H,00H,00H ;,
        DB      00H,66H,66H,00H,00H,00H ;:
        DB      02H,01H,51H,09H,06H,00H ;?
        ;COL4
        DB      7FH,49H,49H,49H,41H,00H ;E
        DB      7FH,08H,14H,22H,41H,00H ;K
        DB      3EH,41H,51H,21H,5EH,00H ;Q
        DB      3FH,40H,38H,40H,3FH,00H ;W
        DB      07H,08H,70H,08H,07H,00H ;Y
        DB      36H,49H,55H,22H,50H,00H ;&
        DB      23H,13H,08H,64H,62H,00H ;PERSENTAGE
        DB      24H,2AH,7FH,28H,12H,00H ;$
        ;COL5
        DB      7FH,09H,09H,09H,01H,00H ;F
        DB      7FH,40H,40H,40H,40H,00H ;L
        DB      7FH,09H,19H,29H,46H,00H ;R
        DB      63H,14H,08H,14H,63H,00H ;X
        DB      61H,51H,49H,45H,43H,00H ;Z
        DB      08H,14H,22H,41H,00H,00H ;<
        DB      00H,41H,22H,14H,08H,00H ;>
        DB      00h,00H,00H,00H,00H,00H ;BACKSPACE
        ;COL6
        DB      00H,00H,00H,00H,00H,00H ;SPECIAL FUNCTION
```

```
        DB      00H,00H,00H,00H,00H,00H ;PROGRAM
        DB      00H,00H,00H,00H,00H,00H ;SPACE
        DB      40H,40H,40H,40H,40H,00H ;_
        DB      00H,00H,5FH,00H,00H,00H ;!
        DB      20H,10H,08H,04H,02H,00H ;/
        DB      14H,14H,14H,14H,14H,00H ;=
        DB      08H,08H,08H,08H,08H,00H ;-
        ;COL7
        DB      00H,00H,00H,00H,00H,00H ;ENTER
        DB      00H,00H,00H,00H,00H,00H ;RUN
        DB      00H,00H,00H,00H,00H,00H ;ESCAPE
        DB      00H,00H,00H,00H,00H,00H ;PREVIOUS SCREEN
        DB      00H,00H,00H,00H,00H,00H ;BIG
        DB      00H,00H,00H,00H,00H,00H ;SMALL
        DB      00h,00H,00H,00H,00H,00H ;SINGLE
        DB      00h,00H,00H,00H,00H,00H ;DUAL

STATICTABLE:
        DB      00H,00H,00H,00H,00H,00H ;SPACE
        DB      00H,00H,5FH,00H,00H,00H ;!
        DB      00H,00H,00H,00H,00H,00H
        DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;Ü
        DB      24H,2AH,7FH,28H,12H,00H ;$
        DB      23H,13H,08H,64H,62H,00H ;PERSENTAGE
        DB      36H,49H,55H,22H,50H,00H ;&
        DB      00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H
        DB      14H,08H,3EH,08H,14H,00H ;*
        DB      08H,08H,3EH,08H,08H,00H ;+
        DB      00H,50H,30H,00H,00H,00H ;,
        DB      08H,08H,08H,08H,08H,00H ;-
        DB      00H,60H,60H,00H,00H,00H ;.
        DB      20H,10H,08H,04H,02H,00H ;/
        DB      3EH,51H,49H,45H,3EH,00H ;0
        DB      00H,42H,7FH,40H,00H,00H ;1
        DB      42H,61H,51H,49H,46H,00H ;2
        DB      21H,41H,45H,4BH,31H,00H ;3
        DB      18H,14H,12H,7FH,10H,00H ;4
        DB      27H,45H,45H,45H,39H,00H ;5
        DB      3CH,4AH,49H,49H,30H,00H ;6
        DB      01H,71H,09H,05H,03H,00H ;7
        DB      36H,49H,49H,49H,36H,00H ;8
        DB      06H,49H,49H,29H,1EH,00H ;9
        DB      00H,00H,00H,00H,00H,00H
        DB      00H,56H,36H,00H,00H,00H ;KOMMA PUNT
        DB      08H,14H,22H,41H,00H,00H ;<
        DB      14H,14H,14H,14H,14H,00H ;=
        DB      00H,41H,22H,14H,08H,00H ;>
        DB      02H,01H,51H,09H,06H,00H ;?
        DB      00H,00H,00H,00H,00H,00H
        DB      7EH,11H,11H,11H,7EH,00H ;A
        DB      7FH,49H,49H,49H,36H,00H ;B
        DB      3EH,41H,41H,41H,22H,00H ;C
        DB      7FH,41H,41H,22H,1CH,00H ;D
        DB      7FH,49H,49H,49H,41H,00H ;E
        DB      7FH,09H,09H,09H,01H,00H ;F
        DB      3EH,41H,49H,49H,3AH,00H ;G
        DB      7FH,08H,08H,08H,7FH,00H ;H
        DB      00H,41H,7FH,41H,00H,00H ;I
        DB      20H,40H,41H,3FH,01H,00H ;J
        DB      7FH,08H,14H,22H,41H,00H ;K
        DB      7FH,40H,40H,40H,40H,00H ;L
```

```
        DB      7FH,02H,0CH,02H,7FH,00H ;M
        DB      7FH,04H,08H,10H,7FH,00H ;N
        DB      3EH,41H,41H,41H,3EH,00H ;O
        DB      7FH,09H,09H,09H,06H,00H ;P
        DB      3EH,41H,51H,21H,5EH,00H ;Q
        DB      7FH,09H,19H,29H,46H,00H ;R
        DB      26H,49H,49H,49H,32H,00H ;S
        DB      01H,01H,7FH,01H,01H,00H ;T
        DB      3FH,40H,40H,40H,3FH,00H ;U
        DB      1FH,20H,40H,20H,1FH,00H ;V
        DB      3FH,40H,38H,40H,3FH,00H ;W
        DB      63H,14H,08H,14H,63H,00H ;X
        DB      07H,08H,70H,08H,07H,00H ;Y
        DB      61H,51H,49H,45H,43H,00H ;Z

BIGLETTERS:
        ;COL 0
        DB      0F8H,7FH,0FCH,7FH,06H,06H,06H,06H,06H,06H,06H,06H,06H
        DB      06H,06H,06H,0FCH,7FH,0F8H,7FH,00H,00H,00H,00H    ;A
        DB      0F8H,1FH,0FCH,3FH,06H,60H,06H,60H,86H,61H,86H,61H,86H
        DB      61H,86H,61H,0CH,3FH,08H,1EH,00H,00H,00H,00H      ;G
        DB      0FEH,7FH,0FEH,7FH,0CH,00H,18H,00H,30H,00H,30H,00H,18H
        DB      00H,0CH,00H,0FEH,7FH,0FEH,7FH,00H,00H,00H,00H    ;M
        DB      78H,10H,0FCH,30H,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB      61H,86H,61H,0CH,3FH,08H,1EH,00H,00H,00H,00H      ;S
        DB      00H,00H,00H,00H,08H,60H,0CH,60H,0FEH,7FH,0FEH,7FH,00H
        DB      60H,00H,60H,00H,00H,00H,00H,00H,00H,00H,00H      ;1
        DB      80H,07H,0C0H,07H,60H,06H,30H,06H,18H,06H,0CH,06H,0FEH
        DB      7FH,0FEH,7FH,00H,06H,00H,06H,00H,00H,00H,00H     ;4
        DB      06H,00H,06H,00H,06H,00H,06H,07EH,06H,07FH,86H,01H,0C6H
        DB  00H,66H,00H,36H,00H,1EH,00H,00H,00H,00H,00H      ;7
        DB      80H,01H,80H,01H,80H,01H,80H,01H,0F8H,1FH,0F8H,1FH,80H
        DB  01H,80H,01H,80H,01H,80H,01H,00H,00H,00H,00H      ;+

        ;COL 1
        DB      0FEH,7FH,0FEH,7FH,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,86H,61H,0FCH,3FH,78H,1EH,00H,00H,00H,00H     ;B
        DB      0FEH,7FH,0FEH,7FH,80H,01H,80H,01H,80H,01H,80H,01H,80H
        DB  01H,80H,01H,0FEH,7FH,0FEH,7FH,00H,00H,00H,00H    ;H
        DB      0FEH,7FH,0FEH,7FH,30H,00H,60H,00H,0C0H,00H,80H,01H,00H
        DB  03H,00H,06H,0FEH,7FH,0FEH,7FH,00H,00H,00H,00H    ;N
        DB      06H,00H,06H,00H,06H,00H,06H,00H,0FEH,7FH,0FEH,7FH,06H
        DB  00H,06H,00H,06H,00H,06H,00H,00H,00H,00H,00H      ;T
        DB      18H,70H,1CH,78H,06H,6CH,06H,66H,06H,63H,86H,61H,0C6H
        DB  60H,66H,60H,3CH,60H,18H,60H,00H,00H,00H,00H      ;2
        DB      0FEH,19H,0FEH,39H,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,86H,61H,06H,3FH,06H,1EH,00H,00H,00H,00H      ;5
        DB      78H,1EH,0FCH,3FH,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,86H,61H,0FCH,3FH,78H,1EH,00H,00H,00H,00H     ;8
        DB      0F8H,1FH,0FCH,3FH,06H,66H,06H,63H,86H,61H,0C6H,60H,66H
        DB  60H,36H,60H,0FCH,3FH,0F8H,1FH,00H,00H,00H,00H    ;0

        ;COL2
        DB      0F8H,1FH,0FCH,3FH,06H,60H,06H,60H,06H,60H,06H,60H,06H
        DB  60H,06H,60H,0CH,30H,08H,10H,00H,00H,00H,00H      ;C
        DB      00H,00H,00H,00H,06H,60H,06H,60H,0FEH,7FH,0FEH,7FH,06H
        DB  60H,06H,60H,00H,00H,00H,00H,00H,00H,00H,00H      ;I
        DB      0F8H,1FH,0FCH,3FH,06H,60H,06H,60H,06H,60H,06H,60H,06H
        DB  60H,06H,60H,0FCH,3FH,0F8H,1FH,00H,00H,00H,00H    ;O
        DB      0FEH,1FH,0FEH,3FH,00H,60H,00H,60H,00H,60H,00H,60H,00H
        DB  60H,00H,60H,0FEH,3FH,0FEH,1FH,00H,00H,00H,00H    ;U
        DB      06H,18H,06H,38H,06H,60H,06H,60H,86H,61H,0C6H,61H,0E6H
        DB  61H,0B6H,61H,1EH,3FH,0EH,1EH,00H,00H,00H,00H     ;3
```

```
        DB      0F0H,1FH,0F8H,3FH,0CH,63H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,86H,61H,0CH,3FH,08H,1EH,00H,00H,00H,00H     ;6
        DB      78H,10H,0FCH,30H,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,0C6H,30H,0FCH,1FH,0F8H,0FH,00H,00H,00H,00H  ;9
        DB
80H,01H,0B0H,0DH,0F0H,0FH,0E0H,07H,0F8H,1FH,0F8H,1FH,0E0H
        DB  07H,0F0H,0FH,0B0H,0DH,80H,01H,00H,00H,00H,00H   ;*

        ;COL 3
        DB      0FEH,7FH,0FEH,7FH,06H,60H,06H,60H,06H,60H,06H,60H,06H
        DB  60H,06H,60H,0FCH,3FH,0F8H,1FH,00H,00H,00H,00H   ;D
        DB      00H,18H,00H,38H,06H,60H,06H,60H,0FEH,3FH,0FEH,1FH,06H
        DB  00H,06H,00H,00H,00H,00H,00H,00H,00H,00H,00H     ;J
        DB      0FEH,7FH,0FEH,7FH,06H,03H,06H,03H,06H,03H,06H,03H,06H
        DB  03H,06H,03H,0FCH,01H,0F8H,00H,00H,00H,00H,00H   ;P
        DB      0FEH,07H,0FEH,0FH,00H,18H,00H,30H,00H,60H,00H,60H,00H
        DB  30H,00H,18H,0FEH,0FH,0FEH,07H,00H,00H,00H,00H   ;V
        DB      00H,00H,00H,00H,00H,70H,00H,70H,00H,70H,00H,00H,00H
        DB  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H     ;.
        DB      00H,00H,00H,00H,00H,0B8H,00H,0F8H,00H,78H,00H,00H,00H
        DB  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H     ;,
        DB      00H,00H,00H,00H,38H,0B8H,38H,78H,38H,38H,00H,00H,00H
        DB  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H     ;DUBBELPUNT
        DB      18H,00H,1CH,00H,06H,00H,06H,00H,06H,6EH,06H,6FH,86H
        DB  01H,0C6H,00H,7CH,00H,38H,00H,00H,00H,00H,00H    ;?

        ;COL 4
        DB      0FEH,7FH,0FEH,7FH,86H,61H,86H,61H,86H,61H,86H,61H,86H
        DB  61H,86H,61H,06H,60H,06H,60H,00H,00H,00H,00H     ;E
        DB      0FEH,7FH,0FEH,7FH,80H,01H,0C0H,03H,60H,06H,30H,0CH,18H
        DB  18H,0CH,30H,06H,60H,02H,40H,00H,00H,00H,00H     ;K
        DB      0F8H,1FH,0FCH,3FH,06H,60H,06H,60H,06H,64H,06H,6CH,06H
        DB  78H,06H,30H,0FCH,7FH,0F8H,4FH,00H,00H,00H,00H   ;Q
        DB      0FEH,1FH,0FEH,3FH,00H,60H,00H,60H,0C0H,3FH,0C0H,3FH,00H
        DB  60H,00H,60H,0FEH,3FH,0FEH,1FH,00H,00H,00H,00H   ;W
        DB      3EH,00H,7EH,00H,0C0H,00H,80H,01H,00H,7FH,00H,7FH,80H
        DB  01H,0C0H,00H,7EH,00H,3EH,00H,00H,00H,00H,00H    ;Y
        DB      60H,1CH,0F0H,3EH,98H,63H,98H,63H,0F0H,67H,60H,6CH,00H
        DB  38H,00H,38H,00H,6CH,00H,46H,00H,00H,00H,00H     ;&
        DB      00H,20H,1CH,18H,1CH,0CH,1CH,06H,00H,03H,80H,01H,0C0H
        DB  38H,60H,38H,30H,38H,18H,00H,00H,00H,00H,00H     ;PERSENTAGE
        DB      70H,0CH,0F8H,1CH,8CH,31H,8CH,31H,0FEH,7FH,0FEH,7FH,8CH
        DB  31H,8CH,31H,38H,1FH,30H,0EH,00H,00H,00H,00H     ;$

;COL 5
        DB      0FEH,7FH,0FEH,7FH,86H,01H,86H,01H,86H,01H,86H,01H,06H
        DB  00H,06H,00H,06H,00H,06H,00H,00H,00H,00H,00H     ;F
        DB      0FEH,7FH,0FEH,7FH,00H,60H,00H,60H,00H,60H,00H,60H,00H
        DB  60H,00H,60H,00H,60H,00H,60H,00H,00H,00H,00H     ;L
        DB      0FEH,7FH,0FEH,7FH,86H,01H,86H,01H,86H,03H,86H,07H,86H
        DB  0DH,86H,19H,0FCH,70H,78H,60H,00H,00H,00H,00H    ;R
        DB      1EH,78H,3EH,7CH,60H,06H,0C0H,03H,80H,01H,80H,01H,0C0H
        DB  03H,60H,06H,3EH,7CH,1EH,78H,00H,00H,00H,00H     ;X
        DB      06H,78H,06H,6CH,06H,66H,06H,63H,86H,61H,0C6H,60H,66H
        DB  60H,36H,60H,1EH,60H,0EH,60H,00H,00H,00H,00H     ;Z
        DB      00H,00H,00H,00H,00H,00H,00H,01H,80H,03H,0C0H,06H,60H
        DB  0CH,30H,18H,18H,30H,0CH,60H,00H,00H,00H,00H     ;<
        DB      04H,40H,0CH,60H,18H,30H,30H,18H,60H,0CH,0C0H,06H,80H
        DB  03H,00H,01H,00H,00H,00H,00H,00H,00H,00H,00H     ;>
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

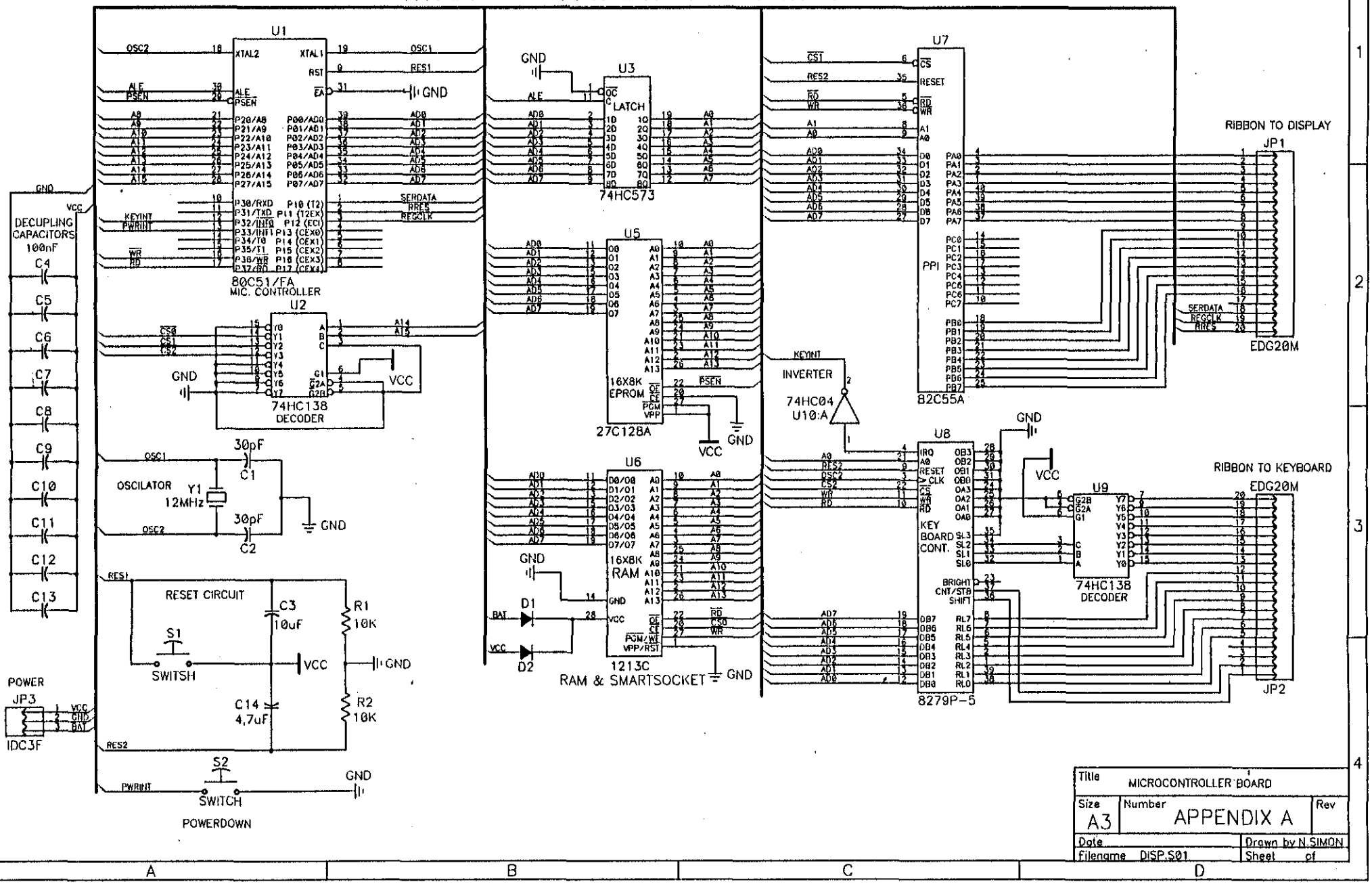
```
;COL 6
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,60H,00H,60H,00H,60H,00H,60H,00H,60H,00H,60H,00H
        DB    60H,00H,60H,00H,60H,00H,60H,00H,00H,00H,00H       ;_
        DB      00H,00H,00H,00H,00H,00H,00H,00H,0FEH,73H,0FEH,77H,0FEH
        DB    73H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H       ;!
        DB      00H,30H,00H,18H,00H,0CH,00H,06H,00H,03H,80H,01H,0C0H
        DB    00H,60H,00H,30H,00H,18H,00H,00H,00H,00H,00H       ;/
        DB      60H,06H,60H,06H,60H,06H,60H,06H,60H,06H,60H,06H,60H
        DB    06H,60H,06H,60H,06H,60H,06H,00H,00H,00H,00H       ;=
        DB      80H,01H,80H,01H,80H,01H,80H,01H,80H,01H,80H,01H,80H
        DB    01H,80H,01H,80H,01H,80H,01H,00H,00H,00H,00H       ;-

;COL 7
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
        DB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H

AUTODISP:
        DB      0C3H,0CBH,0E9H,0EAH,0D0H,0D1H,0D2H

AUTOCLR:
        DB      00H,0CBH,0C8H,0D0H,0C3H
END
```

APPENDIX C.

DIAGRAMS.

INDEX.

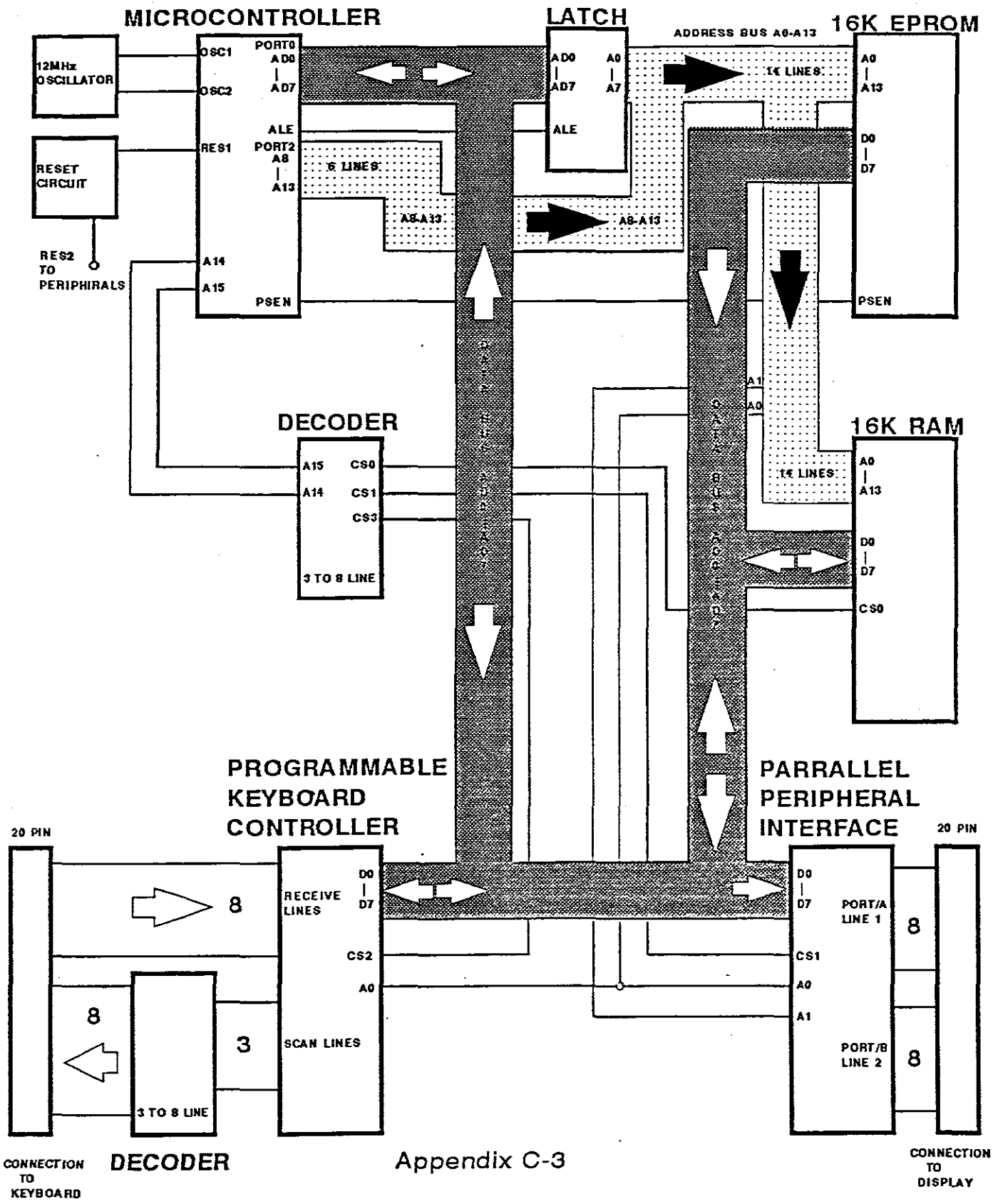
Microcontroller Board



Title	MICROCONTROLLER BOARD
Size	A3
Number	APPENDIX A
Rev	
Date	
Filename	DISP.S01
Drawn by	N.SIMON
Sheet	of

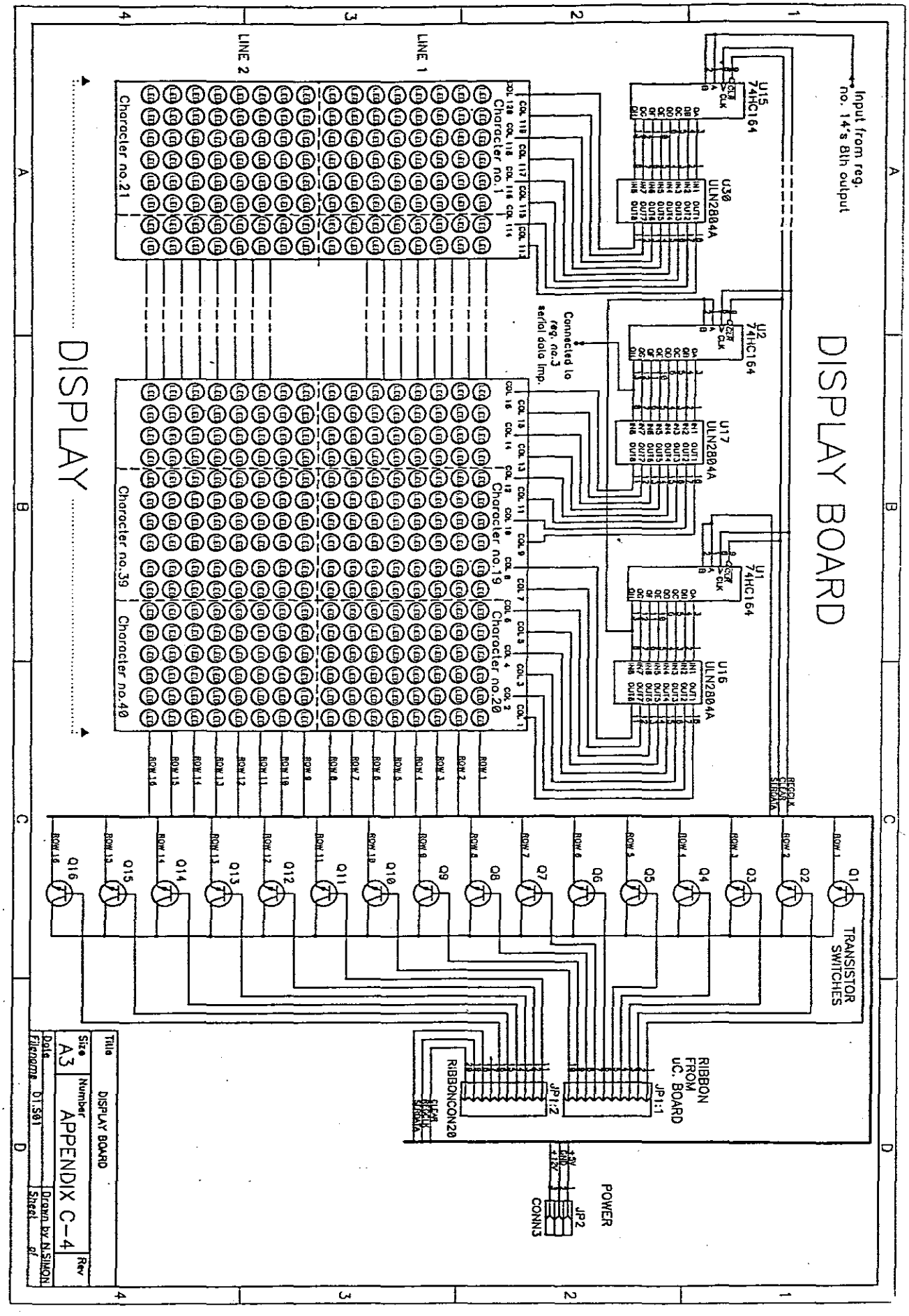
C-2

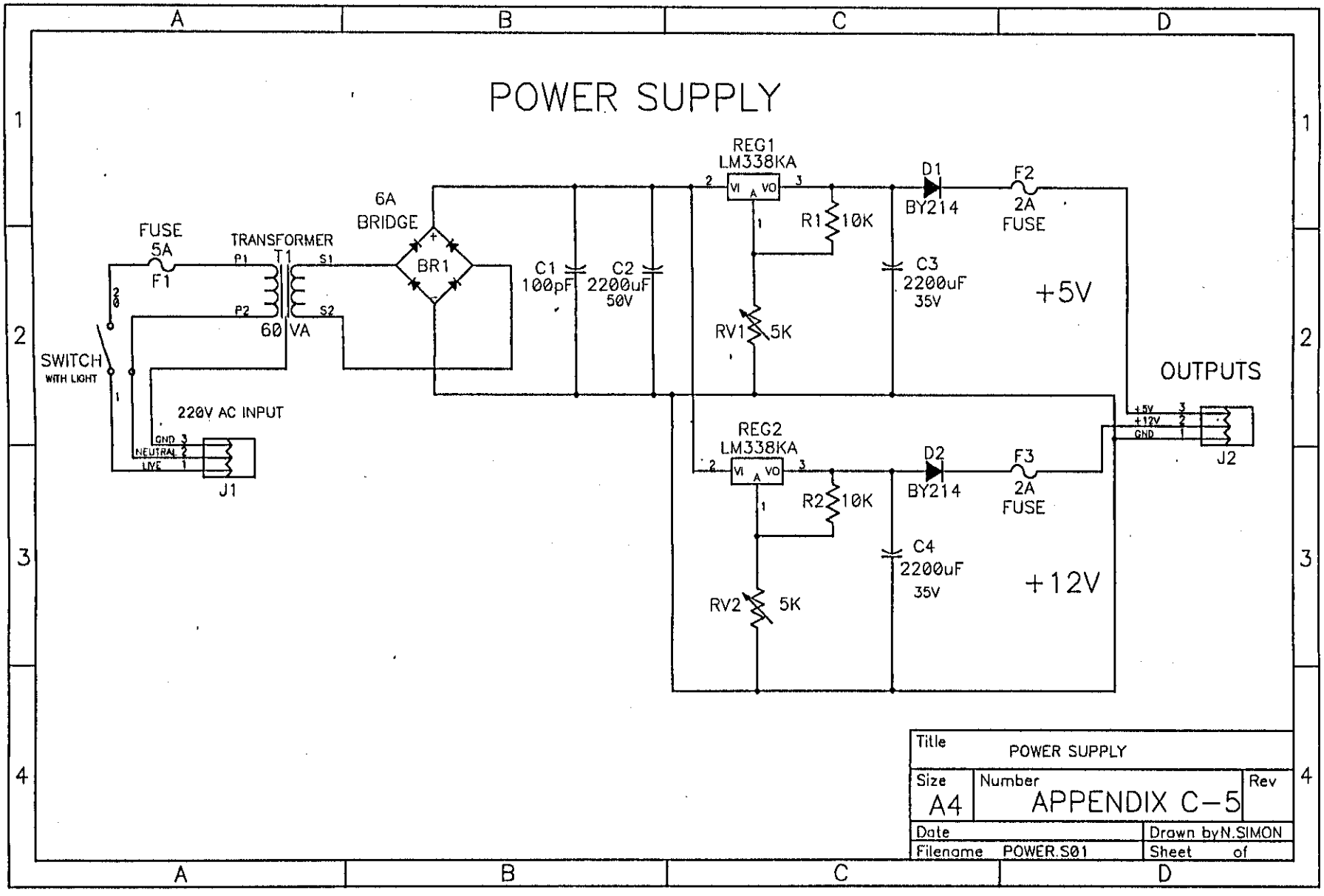
MICROCONTROLLER BOARD BLOCKDIAGRAM



DECODER

Appendix C-3


DISPLAY BOARD
Input from reg. no. 14's 8th output
U15 74HC164
U30 ULN2804A
U2 74HC164
U17 ULN2804A
U1 74HC164
U16 ULN2804A
Connected to reg. no.3 serial data Imp.
LINE 1
LINE 2
Character no.1
Character no.19
Character no.20
Character no.21
Character no.39
Character no.40
DISPLAY
TRANSISTOR SWITCHES
RIBBON FROM uC. BOARD
RIBBONCON20
POWER
CONN3
Title
DISPLAY BOARD
Size
A3
Number
APPENDIX C-4
Rev
Date
Filename
Drawn by N.SIMON
Sheet of


POWER SUPPLY
FUSE
5A
F1
TRANSFORMER
T1
P1
P2
S1
S2
60 VA
6A
BRIDGE
BR1
C1
100pF
C2
2200uF
50V
REG1
LM338KA
VI
A
VO
R1
10K
RV1
5K
D1
BY214
F2
2A
FUSE
C3
2200uF
35V
+5V
SWITCH
WITH LIGHT
220V AC INPUT
GND
NEUTRAL
LIVE
J1
OUTPUTS
+5V
+12V
GND
J2
REG2
LM338KA
R2
10K
RV2
5K
D2
BY214
F3
2A
FUSE
C4
2200uF
35V
+12V
Title
POWER SUPPLY
Size
A4
Number
APPENDIX C-5
Rev
Date
Drawn by N.SIMON
Filename
POWER.S01
Sheet
of