CHAPTER 3

HARDWARE DEVELOPMENT OF MICRO CONTROLLER BASED SMART CARD ATTENDANCE TERMINAL SYSTEM
CHAPTER 3

HARDWARE DEVELOPMENT OF MICROCONTROLLER BASED SMART CARD ATTENDANCE TERMINAL SYSTEM

In the present study, a microcontroller based Smart Card Attendance Terminal system is designed and constructed for the employee’s attendance performance analysis using Personal Computer. The principle of working of the system can be described conceptually with the help of the block diagram as follows, and shown in figure 3.1.

3.1. HARDWARE DESCRIPTION

The Block diagram of the smart card instrumentation attendance terminal system using personal computer and its schematic diagram is shown in figure 3.1 and 3.2 respectively.

The Microcontroller based attendance terminal system, which is designed and constructed in the present study, consists of the following functional units. They are

3.1.1 Smart card
3.1.2 Smart card Reader System
3.1.3 Real time clock
3.1.4 EEPROM
3.1.5 RS-232
3.1.6 Liquid Crystal Display
3.1.7 Microcontroller (89c51)
3.1.8 Power supply
3.1.9 Personal Computer

The constructional details of the each individual unit are presented below with the help of schematic diagram and the photographs of the Microcontroller based Attendance Terminal System is shown in Photographs 2, 3 and 4.
Figure 3.2: Block diagram of Microcontroller based Smart card Attendance Terminal system with PC
Figure 3.2: Schematic diagram of the smart card instrumentation Attendance Terminal system
Photograph 4: Microcontroller Based Smart Card Attendance Terminal System
Photograph 2: Hardware of Microcontroller Based Smart Card Attendance Terminal System

Photograph 3: Complete System of MC-PC Based Smart Card Attendance Terminal System
3.2.1. SMART CARD – SLE 4442

In the present study the smart card SLE4442 [1] is used for the analysis of employee's attendance performance of the organization, have the following features. The SLE 4442 consists of EEPROM main memory and a bit protection memory with PROM functionality. The schematic diagram of smart card shown in figure 3.3.

Figure 3.3: circuit details of smart card SLE4442

Feature of SLE 4442

- Byte-wise addressing

- 256 x 8-bit EEPROM organization

- Irreversible byte-wise write protection of lowest 32 addresses (Byte 0 ... 31)

- 32 x 1-bit organization of protection memory

- Two-wire link protocol

- End of processing indicated at data output

- Answer-to-Reset acc. to ISO standard 7816-3

- Programming time 2.5 ms per byte for both erasing and writing

- Minimum of 104 write/erase cycles
• Data retention for minimum of ten years1)

• Contact configuration and serial interface in accordance with ISO standard 7816 (synchronous transmission)

• Data can only be changed after entry of the correct 3-byte programmable security code (security memory)

SLE 4442 Pin Configuration

<table>
<thead>
<tr>
<th>VCC</th>
<th>C1</th>
<th>C5</th>
<th>GND</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST</td>
<td>C2</td>
<td>C6</td>
<td>N.C.</td>
</tr>
<tr>
<td>CLK</td>
<td>C3</td>
<td>C7</td>
<td>I/O</td>
</tr>
<tr>
<td>N.C.</td>
<td>C4</td>
<td>C8</td>
<td>N.C.</td>
</tr>
</tbody>
</table>

Figure 3.4 Pin Diagram of a Smart Card

Pin Definitions and Functions


The main memory is erased and written byte by byte. When erased, all 8 bits of a data byte are set to logical one. When written, the Information in the individual EEPROM cells is, according to the input data, altered bit by bit to logical zeros (logical AND between the old and the new data in the EEPROM). Normally a data change consists of an erase and write procedure. It depends on the contents of the data byte in the main memory and the new data byte whether the EEPROM is really erased and/or
written. If none of the 8 bits in the addressed byte requires a zero to-one transition the erase access will be suppressed. Vice versa the write access will be suppressed if no one-to-zero transition is necessary.

Figure 3.5 Internal Memory Diagram of a Smart Card SLE4442

The write and the erase operation take at least 2.5 ms each. Each of the first 32 bytes can be irreversibly protected against data change by writing the corresponding bit in the protection memory. Each data byte in this address range is assigned to one bit of the protection memory and has the same address as the data byte in the main memory, which it is assigned to. Once written the protection bit cannot be erased (PROM).

The SLE 4442 provides a security code logic, which controls the write/erase access to the memory. For this purpose the SLE 4442 contains a 4-byte security memory.
with an Error Counter EC (bit 0 to bit 2) and 3 bytes reference data. These 3 bytes as a whole are called Programmable Security Code (PSC). After power on the whole memory, except for the reference data, can only be read. Only after a successful comparison of verification data with the internal reference data the memory has the identical access functionality of the SLE 4432 until the power is switched off. After three successive unsuccessful comparisons the Error Counter blocks any subsequent attempt, and hence any possibility to write and erase.

Transmission Protocol

The transmission protocol [4] is a two-wire link protocol between the interface device IFD and the integrated circuit IC [5]. It is identical to the protocol type "S = A". All data changes on I/O are initiated by the falling edge on CLK. The transmission protocol consists of the 4 modes.

Operational modes:
- Reset and Answer-to-Reset
- Command Mode
- Outgoing Data Mode
- Processing Mode

The I/O pin is open drain and therefore requires an external pull up resistor to achieve a high level [6].

Reset and Answer-to-Reset

Answer-to-Reset takes place according to ISO standard [7] 7816-3 (ATR). The reset can be given at any time during operation. In the beginning, the address counter is
set to zero together with a clock pulse and the first data bit (LSB) is output to I/O when RST is set from level H to level L. Under a continuous input of additional 31 clock pulses the contents of the first 4 EEPROM addresses is read out. The 33rd clock pulse switches I/O to high impedance Z and finishes the ATR procedure.

| Answer-to-Reset (Hex) | Byte1 DO7...DO0 | Byte2 DO15...DO8 | Byte3 DO23...DO16 | Byte4 DO31...DO24 |

Reset and Answer-to-Reset

![Pulse-rate diagrams of RESET, CLOCK, DATA signal for ATR](image)

**Figure 3.6:** Pulse-rate diagrams of RESET, CLOCK, DATA signal for ATR

**Command Mode**

After the Answer-to-Reset the chip waits for a command. Every command begins with a start condition, includes a 3 bytes long command entry followed by an additional clock pulse and ends with a stop condition.

- **Start condition:** Falling edge on I/O during CLK in level H
- **Stop condition:** Rising edge on I/O during CLK in level H
After the reception of a command there are two possible modes:

- Outgoing data mode for reading
- Processing mode for writing and erasing

**Outgoing Data Mode**

In this mode the IC sends data to the IFD. The first bit becomes valid on I/O after the first falling edge on CLK. After the last data bit an additional clock pulse is necessary in order to set I/O to high impedance Z and to prepare the IC for a new command entry. During this mode any start and stop condition is discarded.

**Processing Mode**

In this mode the IC processes internally. The IC has to be clocked continuously until I/O, which was switched to level L after the first falling edge of CLK, is set to high impedance level Z. Any start and stop condition is discarded during this mode.

The RST line is low during the modes mentioned above. If RST is set to high during the CLK low level any operation is aborted and I/O is switched to high impedance Z (Break).

**Command Format:** Each command consists of three bytes.

<table>
<thead>
<tr>
<th>MSB</th>
<th>CONTROL</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>B B B B B B B A A A A A A</td>
<td>D D D D D D D D</td>
<td></td>
</tr>
</tbody>
</table>

Beginning with the control byte LSB is transmitted first.
RSI is low level
Command IFD sets I/O | Outgoing Data —► IC sets I/O
to level L . to level Z
CLK
I/O

Start from IFD
Stop from IFD
Start of Processing
End of Processing

Figure 3.7 Pulse rate diagram of outgoing Date

Figure 3.8: Command data format
Command Mode

Table 1: The list of SLE 4442 provides 7 commands.

<table>
<thead>
<tr>
<th>BYTE1 CONTROL</th>
<th>BYTE2 ADDRESS</th>
<th>BYTE3 DATA</th>
<th>OPERATION</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>B7 B6 B5 B4 B3 B2 B1 B0 A7-A0</td>
<td>D7-D0</td>
<td>No effect</td>
<td>READ MAIN MEMORY</td>
<td>Outgoing data</td>
</tr>
<tr>
<td>0 0 1 1 0 0 0 0</td>
<td>Address</td>
<td>No effect</td>
<td>UPDATE MAIN MEMORY</td>
<td>Processing</td>
</tr>
<tr>
<td>0 0 1 1 1 0 0 0</td>
<td>Address</td>
<td>Input Data</td>
<td>READ PROTECTION MEMORY</td>
<td>Outgoing data</td>
</tr>
<tr>
<td>0 0 1 1 0 1 0 0</td>
<td>No effect</td>
<td>No effect</td>
<td>WRITE PROTECTION MEMORY</td>
<td>Processing</td>
</tr>
<tr>
<td>0 0 1 1 1 1 0 0</td>
<td>Address</td>
<td>Input Data</td>
<td>READ SECURITY MEMORY</td>
<td>Outgoing data</td>
</tr>
<tr>
<td>0 0 1 1 0 0 0 1</td>
<td>No effect</td>
<td>No effect</td>
<td>UPDATE SECURITY MEMORY</td>
<td>Processing</td>
</tr>
<tr>
<td>0 0 1 1 1 0 0 1</td>
<td>Input Data</td>
<td>Input Data</td>
<td>COMPARE VERIFICATION DATA</td>
<td>Processing</td>
</tr>
</tbody>
</table>

3.2.2.SMART CARD READER SYSTEM (TERMINAL)

Smart card by itself is useless, it requires a reader. Generally “smart card readers” are referred as all smart card enabled terminals. Smart card readers have an ability to read and write. Readers [8] often are called the read-write unit as it can read as well as write to the card. Smart card reader read and writes as long as the smart card supports it and the process access conditions have been fulfilled. Smart card readers come in a variety of form factors, with varying levels of mechanical and logical sophistication.
Some examples include readers integrated into vending machines, handheld battery operated readers with a small LCD screen, readers integrated into GSM mobile phones, and readers attached to microcontroller, personal computers. Mechanically, readers have various options, including whether the user must insert/remove the card versus automated insertion/ejection mechanisms, sliding contacts versus landing contacts.

3.2.3. REAL TIME CLOCK (DS1307)

Real time clock DS1307 [9] counts seconds, minutes, hour’s date of the month, month, day of the week and year with leap year compensation valid up to 2100.

![Figure 3.9: Circuit diagram of Real Time Clock (DS1307)](image)

Real Time clock DS1307 have the following features are

- 2-wire serial interface
- 56 bytes nonvolatile RAM for data storage.
- Programmable square wave output signal
- Automatic power-fail detect & switch circuitry
• Consumes less than 500nA in battery backup mode with oscillator running.

• Optional industrial temperature range -40 degree C to +85 degree C available for DS1307 and DS1308

• DS1307 available in 8-pin DIP or SOIC

The DS1307 Serial real time clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via 2-wire bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hours or 12-hours format with AM/PM indicator. The DS1307 has a built in power sense circuit which detects power failure and automatically switches to the battery supply.

OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When VCC falls below 1.25x VBAT the device terminates an access in progress & resets the device address counter. Inputs to the device from an out of tolerance system. When VCC falls below VBAT the device switches into a low current battery backup mode. Upon power up, the device switches from battery to VCC when VCC is greater than VBAT +0.2v and recognizes input when VCC is greater than
1.25 x into VBAT. The block diagram in below fig shows the main elements of serial real
time clock.

![Internal Block Diagram & Address Map of Real Time Clock DS1307](image)

**Signal Descriptions**

VCC, GND-DC power is provided to the device on these pins. Vcc is the +5v
input. When 5v is applied within normal limits, the device is fully accessible and data
can be written and read. When a 3v battery is connected to the device and VCC is below
1.25 x VBAT, reads and writes are inhibited. However, the timekeeping function continues unaffected by the low input voltages. As VCC falls below VBAT and RAM and timekeeper are switched over to the external power supply (normal 3.0 v DC) at VBAT.

**VBAT:**

Battery input for any standard 3-volt lithium cell or other energy source. Battery voltage must be held between 2.0 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25x VBAT nominal. A lithium battery with 48mA hr or greater will back up the DS1307 for more than 10 years in the absence of power at 25 degree C.

**SCL (serial clock input):**

SCL is used to synchronize data movement on the serial interface.

**SDA (serial Data Input/Output)**

SDA is the input/out pin for the 2-wire serial interface. The SDA pin is open drain, which requires an external pull up resistor.

**SQW/OUT (Square wave/Output Driver)**

When enabled, the SQWE bit set to 1 the SQU/OUT pin outputs one of the square wave frequencies (1Hz, 4Hz, 8Hz, 32Hz). The SQWE/OUT pin is open drain, which requires an external pull up resistor.

**X1, X2:** Connections for a standard 32.768KHz quartz crystal. The internal oscillator
circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 Pf.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in fig 3.10. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3fh. During a multi-byte access, when the address pointer reaches 3fh, the end of ram space, it wraps around to locations 00h, the beginning of the clock space.

3.2.4. EEPROM (AT24C256)

Serial memory devices offer significant advantages over parallel devices in applications. In addition to requiring less board space, serial devices allow microcontroller I/O pins to be conserved.

![Figure3.11: Circuit for Serial Memory EEPROM AT24C256](image)

The AT24C256 [10] shown in figure is serial electrically erasable and programmable read only memory (EEPROM) organized as 32,768 words of 8-bits each. This is interfaced to the controller over an I2C bus. I2C bus is 2 wire serial data SDA & serial clock SCL carry the information between the devices connected to the same bus.
Each device is recognized by a unique address and can operate as either transmitter or receiver depending on the function of the device. Another way of looking at this configuration is as a master and slave relation.

A master can initiate a data transfer on the bus and generate the clock signals to permit the transfer. On the other hand any device addressed at that time is considered a slave.

Serial data is bi-directional line and Serial clock SCL is unidirectional, both connected to a positive supply voltage via pull-up resistor. When the bus is free, both lines are high. Data on the I2C bus can be transferred at rates of up to 1kbts in the fast mode.

An I2C master transmission proceeds as follows

1. The master polls the bus to see if it is in use.
2. The master generates start condition on the bus.
3. The master broadcasts the slave address & expects an acknowledge (ACK) from the addressed slave.
4. The master transmits 0 or more bytes of data, expecting an ACK following each byte.
5. The master generates stop condition and releases the bus.
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter</td>
<td>the devices which send data to the bus</td>
</tr>
<tr>
<td>Receiver</td>
<td>the devices which received data from the bus</td>
</tr>
<tr>
<td>Master</td>
<td>the devices, which initiate a transfer, generates clock signal and terminates a transfer</td>
</tr>
<tr>
<td>Slave</td>
<td>the device addressed by a master</td>
</tr>
<tr>
<td>Multi-master</td>
<td>more than one master can attempt to control the bus at the same time without corrupting the message.</td>
</tr>
</tbody>
</table>

A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition. The master always generates START & STOP conditions. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after STOP condition.

All communication on the I2C between the start and stop conditions, including addressing and data, takes place as an 8-bit data value followed by an acknowledge bit. Every byte put on SDA line must be 8-bit long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte is followed by an acknowledge bit. Data is transferred with the most significant bit (MSB). If a slave can’t receive or
transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL low to force the master into a wait state. Data transfer then continuous when the slave is ready for another byte of data and release clock line SCL. Data transfer with acknowledges is obligatory. The acknowledge related clock pulse is generated by the master. The transmitter release the SDA line (HIGH) during the acknowledge clock pulse.

The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. Of course, setup and hold times must also be taken into account. Usually, a receiver which has been addressed is obligated to generate acknowledge after each byte has been received, except when the message starts with a BUS address. When a slave doesn't acknowledge the slave address, data line must be left high by the slave. The master can then generate either a stop condition to abort the transfer, or a repeated START condition to start a new transfer. If a slave receiver does acknowledge the slave address but some time late in the transfer cannot receive any more data byte, the master must again abort the transfer. This is indicated by the slave generating the not acknowledge on the first byte to follow. The slave leaves the data line HIGH and the master generates a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked
out of the slave the slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

Master generates their own clock on the SCL line to transfer message on the I2C bus. Data is only valid during the high period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedure to take place.

After the start condition a slave address is sent. This address is 7 bits long followed by an eight bit, which is a data direction bit (R/W) - a ‘zero’ indicates a transmission (WRITE), a ‘one’ indicates a request for data (READ). A data transfer is always terminated by a STOP condition. Generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition and address

Now the address of the device EEPROM is A0h used for initiating SDA & SCL lines to read and store data from smart card reader system. Data on the I2C bus can be transferred at rates of up to 1 Kbits in the fast mode. In this design we used EEPROM to store user id number, time which they inserted the card.

3.2.5 SERIAL COMMUNICATION (RS-232)

Microcontroller transfer the data in two ways as parallel and serial. In parallel data transfers, often 8 or more lines (wire conductors) are used to transfer data to device that is only a few away. Example of parallel transfers is printers. Each uses cable with many wire strips. Although in such cases a lot of data can be transferred in a short
amount of time by using many wires in parallel, the distance cannot be great. To transfer to a device located many meters away, the serial method is used.

To allows compatibility among data communication equipment made by various manufacturers, an interfacing standard called RS232 [11] was set by the Electronics Industries Association (EIA) in 1960. In 1963 it was modified and called RS232A. In this we use RS232. RS232 is the most widely used serial I/O interfacing standard. This standard is used in PCs and numerous types of equipment. Its input and output voltage levels are not TTL compatible. In RS232 [12], a 1 is represented by -3 to -25v, while a 0 bit is +3v to +25v. For this reason, to connect any RS232 to microcontroller system we must use voltage converters such as MAX232 to convert the TTL logic levels to the RS232 voltage level, and vice versa. Max 232 IC chips are commonly referred to as line drivers.

Figure 3.12: Circuit for Serial Communication between Microcontroller & PC

To allows compatibility among data communication equipment made by various manufacturers, an interfacing standard called RS232 [11] was set by the Electronics Industries Association (EIA) in 1960. In 1963 it was modified and called RS232A. In this we use RS232. RS232 is the most widely used serial I/O interfacing standard. This standard is used in PCs and numerous types of equipment. Its input and output voltage levels are not TTL compatible. In RS232 [12], a 1 is represented by -3 to -25v, while a 0 bit is +3v to +25v. For this reason, to connect any RS232 to microcontroller system we must use voltage converters such as MAX232 to convert the TTL logic levels to the RS232 voltage level, and vice versa. Max 232 IC chips are commonly referred to as line drivers.
RS232 pins

Table 2 provides the pins and their labels for the RS232 cable, commonly referred to as DB9. IBM introduced the DB9 version of the serial I/O standard, which uses 9 pins.

Table 2:

<table>
<thead>
<tr>
<th>PIN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data carrier detect (DCD)</td>
</tr>
<tr>
<td>2</td>
<td>Received data (RXD)</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted data (TXD)</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal ready (DTR)</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground (GND)</td>
</tr>
<tr>
<td>6</td>
<td>Data set ready (DSR)</td>
</tr>
<tr>
<td>7</td>
<td>Request to send (RTS)</td>
</tr>
<tr>
<td>8</td>
<td>Cleat to send (CTS)</td>
</tr>
<tr>
<td>9</td>
<td>Ring indicator (RI)</td>
</tr>
</tbody>
</table>

Figure 3.13 Interconnection between Microcontroller and Personal Computer
The connection between a PC and microcontroller requires a minimum of three pins. TXD, RXD, and ground as shown in figure. Notice in that figure that the RXD and TXD pins are interchanged.

The 8951 have two pins that are used specifically for transferring receiving data serially. These two pins are called TXD and RXD and are part of the port3 group (P3.0 and P3.1). Pin11 of the 89c51(P3.1) is assigned to TXD and pin 10 (P3.0) is designated as RXD. And used for the present study for the analysis made with Personal Computer.

The RS232 has two sets of line drivers for transferring and receiving data as shown in figure. The line drivers used for TXD are called T1 and T2, while the line drivers for RXD are designated as R1 and R2. In many applications only one of each is used. For example T1 and R1 are used together for TXD and RXD of the 8051, and the second set is left unused. RS 232 [13] requires four capacitors ranging from 1 to 22uF. In this project we used these capacitors of 10 uF shown in figure.

Further there are four programmable modes for serial data communication that are chosen by setting the SMX bits in SCON. The baud rate in the 89c51 is programmable. Baud rates are determined by the mode chosen. This is done with the help of timer1. The serial port can operate in 4 modes.

Mode1 has a variable baud rate. The baud rate is generated by timer1. For this purpose, timer 1 is used in mode2 (Auto - Reload).

\[
\text{Baud rate} = \frac{K \times \text{Osc Freq}}{32 \times 12 \times [256-TH1]}
\]
If SMOD = 0 then K = 1

If SMOD = 1 then K = 2 (SMOD is in the CON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

\[
K \times \text{Osc Freq} \\
\text{TH1} = 256 - \frac{32 \times 12 \times \text{Baud rate}}{32 \times 12 \times 9600}
\]

With XTAL = 11.0592MHz and TH1 = 9600

\[
K \times 11.0592 \times 10^6 \\
\text{TH1} = 256 - \frac{32 \times 12 \times 9600}{32 \times 12 \times 9600} \\
\text{TH1} = \text{FDh}
\]

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. When timer 1 is used to set the baud rate it must be programmed in mode 2, that is 8-bit auto reload.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (i.e., ORLPCON, #80h). The address of PCON is 87H.

Programming the 89c51 [14] to transfer data serially:

In programming the 89c51 to transfer character bytes serially, the following steps must be taken.
1. The TMOD register is loaded with the value 20h, indicating the use of timer 1 in mode 2 (8-bit auto reload) to set the baud rate.

2. The TH1 is loaded with Fdh to set baud rate =9600kbs for serial data transfer (XTAL=11.0592MHZ)

3. The SCON register is loaded with t value 50H, indicating serial mode 1.

4. TR1 is set to 1 to start timer1.

5. TI is cleared.

6. The character byte to be transferred serially is written into the SBUF register.

7. The TI flag bit is monitored, to see if the character has been transferred completely.

To transfer next character repeats from 5.

Programming the 89c51 to receive data serially:

In programming the 89c51 to receive character bytes serially, the following steps must be taken.

1. The TMOD register is loaded with the value 20h, indicating the use of timer 1 in mode 2 (8-bit auto reload) to set the baud rate.

2. The TH1 is loaded with Fdh to set baud rate =9600kbs for serial data transfer (XTAL=11.0592MHZ)

3. The SCON register is loaded with t value 50H, indicating serial mode 1.

4. TR1 is set to 1 to start timer1.

5. RI is cleared.

6. The RI flag bit is monitored, to see if an entire character has been received.
7. When RI is raised, SBUF has the byte. Its contents are moved into a safe place.

To receive next character repeats from 5.

3.2.6. LIQUID CRYSTAL DISPLAY MODULE

In the present study the LCD display LM16200 (LAMPEX, Hyd) [15] used for displaying the messages required for operating the smart card system. The LAMPEX display module is a dot matrix liquid crystal display that display alphanumeric, characters and special symbols. The built-in-controller &driver provide convenient connectivity between a dot matrix LCD 4 or 8 bit microprocessors or microcontroller [16].

Figure 3.14: Interfacing circuit of LCD module (LM16200) with AT89c51
All the functions required for dot matrix liquid crystal display drive are internally provided. Internal refresh is provided by the LAMPEX. The CMOS technology makes the devices ideal for applications in hand-held portable and other powered instruments with low power consumption.

FEATURES

- Easy interface with a 4-bit or 8-bit MPU
- Built-in Dot Matrix LCD controller with font 5x7 or 5x10 dots
- Display Data Ram for 80 characters (80x8 bits)
- Characters generator ROM, which provides 16 characters with font 5x7 dots and 32 characters with font 5x10 dots
- Both display data and character generator RAMs can be read from the MPU.
- Internal automatic reset circuit at power ON.
- Built-in Oscillator circuits (No external clock required).
- Wide range of instruction functions: Clear Display, Cursor Home, Display ON/OFF, and cursor shift, Display shift.

INTERFACEING LCD display LM16200 with Microcontroller AT89C51

The data bus with MPU is available either for 8-bit operation or 4-bit 2 operations allowing the LAMPEX to be interfaced with either an 8-bit or 4-bit MPU.

I/D

When the I/D is set, the 8-bit character code is write or read to and from the DDRAM, the cursor shifts to the right by 1 character position (I/D = '1', increment) or to the left by 1 character position (I/D = 0, decrement). The address counter is incremented
(I/D = '1') or decrements (I/D = '1') by 1.

S
Shifts the entire display to the right or to the left when S is 1, to the left when I/D = 1 and to the right when I/D = 0. Thus it looks as if the cursor stands and the display moves. The display does not shift when reading from the DDRAM or when writing into or reading out from the CGRAM when s = 0.

d. Display ON/OFF control

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>D</td>
<td>C</td>
<td>B</td>
</tr>
</tbody>
</table>

Code higher order bits

D: The display is ON when D = '1' and OFF when D = '0'. Display data remains in the DDRAM. It can be displayed immediately by setting D = '1'.

B: The character indicated by the cursor blinks when B = '1'. The blink is displayed by switching between all black dots and display characters at 409.6 ms interval when Fcp or Fosc = 250 KHz. The cursor and the blink can be set to display simultaneously. (The blink interval changes according to the reciprocal of Fcp or Fosc: 409.6 x 250/270 = 379.2 ms when Fcp = 20 KHz).

e. Cursor or Display Shift

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>S/C</td>
<td>R/L</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Code higher order bits

Shifts cursor positions or display to the right or left without writing or reading
display data. This function is used to correct or search for the display. In a 2-line display the cursor moves to the 2nd line when it passes the 40th digit of the 1st line. Notice that the 1st and 2nd line displays will shift at the same time. When the displayed data is shifted repeatedly each line only moves horizontally. The 2nd line display does not shift into the 1st line position.

S/C R/L

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>DL</td>
<td>N</td>
</tr>
</tbody>
</table>

0 0 shifts the cursor position to the left (AC is decrement by one)

0 1 shifts the cursor position to the right (AC is increment by one)

1 0 shifts the entire display to the left. The cursor follows the display shift

1 1 shifts the entire display to the right. The cursor follows the display shift

Address Counter (AC) contents do not change if the only action performed is shift display.

F Function set

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>DL</td>
<td>N</td>
<td>F</td>
<td>*</td>
</tr>
</tbody>
</table>

DL: Sets interface data length

When DL='1' the data input/output to and from the MPU is carried out by means of 8 bits DB7 to DB0. When DL='0' the data input/output to and from the MPU is carried out in two steps through the 4-bits DB7 to DB4.

N: Sets character font

The 5x7 dots character font is selected when N=1' while the 1-line display mode
is selected when N='0'

F: Set character front.

The 5x7 dots character front is selected when F='0', while the 5x10 dots character front is selected when F='1' and n='0'.

Note: This instruction is to be executed at the start of the program. From this point the function set instruction cannot be executed unless the interface data length is charged i.e., software reset is performed.

<table>
<thead>
<tr>
<th>N</th>
<th>F</th>
<th>No Display Lines</th>
<th>Character Font</th>
<th>Duty Factor</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5x7 dots</td>
<td>1/8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5x10 dots</td>
<td>1/11</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>1</td>
<td>5x7 dots</td>
<td>1/16</td>
<td>can't display 2 lines with 5x10 dots character font</td>
</tr>
</tbody>
</table>

**g. Function Set**

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A5</td>
<td>A4</td>
<td>A3</td>
<td>A2</td>
<td>A1</td>
<td>A0</td>
</tr>
</tbody>
</table>

Sets CGRam address into the address counter in binary a5 to a0. in the 5x10 front mode

A5 & A4 define the CGRAM block is defined by A5-A3 while A2-A0 define the row.

**h. Set DD RAM address**

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>B6</td>
<td>B5</td>
<td>B4</td>
<td>B3</td>
<td>B2</td>
<td>B1</td>
<td>B0</td>
</tr>
</tbody>
</table>

Sets DD RAM address.
Sets the DDRAM address into the address counters in binary B6 to B0. Data then written or read from the LAMPEX pertains to the DDRAM. However, when N=‘0’ (1-line display), B6 to B0 is 00H-4FH. When N=‘1’ (2-line display), B6 to B0 is 00H-27H for the first line and 40H-67H for the second line.

I. Read Busy flag and Address

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>BF</td>
<td>C6</td>
<td>C5</td>
<td>C4</td>
<td>C3</td>
<td>C2</td>
<td>C1</td>
<td>C0</td>
</tr>
</tbody>
</table>

| Code higher order bits | Code lower order bits |

Read the busy flag (BF) that indicates the system is now internally executing a previously received instruction, BF =1 indicates the internal operation is in progress. The next instruction will not be acceptable BF goes. Check the BF status before the next write operation.

At the same time, the value of the address counter expressed in binary C6 to C0 is read. The address counter is used by both CG & DD RAM addresses. And its value is determined by the previous instruction.

j. Write data to CG or DD RAM

<table>
<thead>
<tr>
<th>RS</th>
<th>W/R</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>

| Code higher order bits | Code lower order bits |

Write binary 8 data DDDDDDDDD to the CG or DD RAM. Whether the CG or DD RAM is to be written into is determined by the previous specification of CG RAM or DD RAM address setting. After write, the address is automatically incremented or
In present system LCD interfaces supports most of the available external LCD controller, with 4 data lines and 3 control lines.

The LCD interface is used as follows:

- P0.0 for register select (LCD_RS)
- P0.1 for Enable (LCD_EN)
- P0.7 - P0.4 for data (LCD_D (3:0))

The LM16200 LCD module connected with the system, having 16 characters with 2 lines to display data of ID Number, Date and Time of the each individual employee inserted card with smart card reader system, and other messages required for the data acquisition for further analysis with personal computer.

3.2.7. Microcontroller (AT89C51)

Microcontroller tend to be utilized in many applications. The AT89c51 [17] is a low power, high performance CMOS 8-bit microcontroller with 4Kbytes of flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel’s high-density non-volatile memory technology. The on-chip flash [18] allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer [19].

The circuit diagram of smart card instrumentation attendance terminal system with 89c51 is shown in fig 3.2. The control lines and data lines of the LCD (LM16200) is connected to the port 0. Then for 2-wire Serial communication i.e. for SCL and SDA
lines for serial memory, EPROM (AT24c256) and Real time clock (DS-1307) are connected to the P1.2 and P1.3 respectively. Smart card data line (CIO) is connected to the line P1.4. Reset pin of the card (CRST) is connected to P1.0, and Card Clock (CCLK) is connected to P1.1 of the Microcontroller [20]. For serial communication between microcontroller and Personal Computer, RXD line P3.0 is connected to the Rout of the RS232 and TXD line P3.1 is connected to the Tin of the RS232 and GND is connected to GND line. The detailed of 89c51 already explained in chapter 2.

3.2.8. Power supply

Most of the devices in electronic equipments require essentially a constant d.c voltage for their operation. Therefore, almost all electronic equipments include a circuit that converts a.c voltage of mains supply into D.C voltage, which is independent of changes in a.c line voltage. The circuit diagram of +5v power supply required for the present study

![Circuit Diagram of Power Supply](image)

Figure 3.15: Circuit diagram of power supply

Fig.3.15 shows a complete solid-state +5v power supply required for the present study using power transformer, a full wave rectifier with two P-N junction diodes, a filter section, a transistor series voltage regulator and a voltage divider.
The A.C. voltage to be rectified is applied to the primary of power transformer T.
two p-n junction diodes for full wave rectification. It is then followed by a filter, which
filters the unidirectional pulsating voltage of the rectifier output. A transistor series
voltage regulator 7805 regulates it. At Output of regulator we get 5v regulated power
supply.

3.2.9. Personal Computer
IBMPC/Compatible COM Ports:

In the present study a Personal Computer with the following features is analysis
the performance of the employee in organization.

- Pentium -32 MHz Intel Microprocessor
- 512MB RAM
- 40GB Hard disk drives
- 1.44MB Floppy disk drives
- Two Serial port
- One Parallel Port

Normally PC [21] has two COM ports. Both COM ports have RS232-type
connectors. Many PCs use one each of the DB25 and DB9 RS232 connectors. The COM
ports are designated as COM1 and COM2. We can connect the 89c51 serial ports to the
COM1 and COM2 ports of a PC for serial communication experiments.

Working of smart card system

The hardware details of the Microcontroller based smart card system is shown in
fig 3.1 already. The heart of the system is the microcontroller 89c51 which handles the
total process in reading the Id number with Real Time and analysis with the personal
computer have the following two tasks.

Microcontrollers having two tasks

1. Checking for presence for smart card and
2. Any request from pc comport for serial communication

Now microcontroller checking for presence of a smart card or not.

If card is present, microcontroller read the card number, Time and Date from
RTC, from DS1307, at what time they inserted the card. After reading number, Date and
Time, microcontroller store all these data into serial memory, for this purpose we use
serial EEPROM (AT24256).

MicroController checks continuously for card presence till power on.

All this process is displayed on LCD display. Here we used LM16200.LCD data
lines and controller lines are connected to the port P0.

After reading the data from the user card the data transferred and stored in
external memory. Further this data is transferred to PC using RS-232 serial
communication interfaced with PC, and analysis made for each individual, and group
of employee's attendance performance by developing a menu based software package
in visual Basic language explained in chapter 4.
REFERENCES

1. ICs for chip cards intelligent 256-byte EEPROM SLE 4432/SLE4442 data sheet 07.95
2. An introduction to smart cards, by Steve Petri, Litronic, INC
3. Introduction to Smart Cards by Sumit Dhar SLM soft India Pvt, Ltd.
4. Smart cards and smart objects communication protocols by Dennis PRACA, Anne-marie PRADEN
5. Smart card protocols and E-commerce protocols, Giampaola Bella
6. ISO/IEC-7816-3 information technology-Identification cards-Integrated circuit(s) cards with contacts-part3: Electronics signals and transmission protocols
7. Smart card standards, Garther
8. Siemens data manual for smart cards
9. DS 1307 64X8 serial real time clock semi conductors, www.Maxim-Ic.com
10. 2-wire serial EEPROMS, ATMEL Data Manual 2004 edition
12. www.maxim-ic.com/Tech support
16. LCD module with an 8051 microcontroller, Iguana labs
17. Microprocessor/Microcomputers: architecture software and systems by A.J. Khambata
18. Microprocessors and interfacing -programming and hardware by Doughias v.Hall

19. EZ31, 89c51 device programmer

20. Microcontroller: EPO applied technology series vol-8 by OAR cornillie