GSM BASED
BANKING SECURITY SYSTEM

A PROJECT REPORT

Submitted by

KARTHICK RAGUNATH. J 97805106026
JENISH DEV. M 97805106021
FELIX. J 97805106018
VINOTH JOHN RAJ. K 97805106060

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

UDAYA SCHOOL OF ENGINEERING, VELLAMODI

ANNA UNIVERSITY : CHENNAI 600 025
GSM technology can provide a sophisticated theft alert system for bank locker system. The embedded I/O unit automates the inner door and entry door. The inner door always kept open. There are two modes in this project one is normal mode and another one is security mode. In normal mode an authorized person can open the locker key and he can close the entry door. At that time GSM never send the message to the required person. If any person tries to open the locker key in security mode, the inner door will be closed automatically and SMS is transferred to the required person’s hand phone. after identifying the theft an authorized person can automate the inner door through the SMS.

The GSM module is connected with the microcontroller through serial port. Using ‘AT’ commands the SMS is transferred to the GSM module. The GSM module converts the digital information into airborne signals. Through GSM network the SMS is transferred to the required person’s hand phone. This system offers better solution for the Bank security system and also it will help you to track the intruder.

The embedded microcontroller used here is 89C51 microcontroller. Since, this microcontroller has in-built peripherals it is called as embedded peripheral the microcontroller has flash memory
CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

This Project focuses onto implement GSM (Global System for Mobile Communication) based Banking Security System. This system is implemented using an embedded microcontroller. The embedded microcontroller used here is 89C51.

Actually, the aim of the project is to implement an Automatic Banking Security System. Security is the protection of something valuable to ensure that it is not stolen, lost, or altered. GSM Based bank safety locker security system provides more reliability and restricts the unauthorized person who is trying to enter into bank. The microcontroller circuitry indicates the theft to the required person and alarm is ON.

Primarily, the system functions with the help of different technologies like the Global Positioning System (GPS), traditional cellular network such as Global System for Mobile Communications (GSM) and other radio frequency medium. Today GSM fitted Banks, cars; ambulances, fleets and police vehicles are
common sights on the roads of developed countries. GSM based bank safety locker security system is simple and costs less. When a GSM based bank safety locker security system is installed in a Bank & to enter the unauthorized person means the message will be transferred to a predefined number.

The functional units of our projects are
GSM module
Stepper Motor
LCD Display
Micro Controller 89c51

1.2. GSM MODULE

The GSM module consist of Wireless CPU, SIM card holder and power LED. It helps to transmit and receive the SMS with UART.

1.3. STEPPER MOTOR

The type of Stepper motor we used here is Brushless shaft. This helps in smooth rotation. These motors are used to control the Doors.

1.4. LCD DISPLAY
A liquid crystal display is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. This is used to display the functioning mode of the microcontroller.

1.5. MICROCONTROLLER 89c51

The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, a four-priority-level, nested interrupt structure, an enhanced UART on-chip oscillator and timing circuits. The added features of 89c51 make it a powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.
The embedded microcontroller used here is 89C51 microcontroller. Since, this microcontroller has in-built peripherals it is called as embedded
controller. The 89C51 microcontroller is a derivative of 8051 microcontroller whose architecture and instructions are same as 8051 microcontroller with some added facilities.

GSM technology can provide a sophisticated theft alert system for bank locker system. The embedded I/O unit automates the inner door and entry door. The inner door always kept open. There are two modes in this project one is normal mode and another one is security mode. In normal mode an authorized person can open the locker key and he can close the entry door. At that time GSM never send the message to the required person. If any person tries to open the locker key in security mode, the inner door will be closed automatically and SMS is transferred to the required person’s hand phone. after identifying the theft an authorized person can automate the inner door through the SMS.

The GSM module is connected with the microcontroller through serial port. Using ‘AT’ commands the SMS is transferred to the GSM module. The GSM module converts the digital information into airborne signals. Through GSM network the SMS is transferred to the required person’s hand phone. This system offers better solution for the Bank security system and also it will help you to track the intruder.

2.2. GSM MODULE

GSM has been the backbone of the phenomenal success in mobile telecom over the last decade. Now, at the dawn of the era of true broadband services, GSM continues to evolve to meet new demands. GSM is an open, non-proprietary system that is constantly evolving. One of its great strengths is the international roaming capability. This gives consumers seamless and same standardized same number contactability in more than 212 countries. This has been a vital driver in growth, with around 300 million GSM subscribers
currently in Europe and Asia. In the Americas, today's 7 million subscribers are set to grow rapidly, with market potential of 500 million in population, due to the introduction of GSM 800, which allows operators using the 800 MHz band to have access to GSM technology too. GSM satellite roaming has extended service access to areas where terrestrial coverage is not available.

GSM differs from first generation wireless systems in that it uses digital technology and time division multiple access transmission methods. Voice is digitally encoded via a unique encoder, which emulates the characteristics of human speech. This method of transmission permits a very efficient data rate/information content ratio.

Cellular mobile communication is based on the concept of frequency reuse. That is, the limited spectrum allocated to the service is partitioned into, for example, N non-overlapping channel sets, which are then assigned in a regular repeated pattern to a hexagonal cell grid. The hexagon is just a convenient idealization that approximates the shape of a circle (the constant signal level contour from an omni directional antenna placed at the center) but forms a grid with no gaps or overlaps. The choice of N is dependent on many tradeoffs involving the local propagation environment, traffic distribution, and costs. The propagation environment determines the interference received from neighboring co-channel cells, which in turn governs the reuse distance, that is, the distance allowed between co-channel cells (cells using the same set of frequency channels).

The cell size determination is usually based on the local traffic distribution and demand. The more the concentration of traffic demand in the area, the smaller the cell has to be sized in order to avail the frequency set to a smaller number of roaming subscribers and thus limit the call blocking
probability within the cell. On the other hand, the smaller the cell is sized, the more equipment will be needed in the system as each cell requires the necessary transceiver and switching equipment, known as the base station subsystem (BSS), through which the mobile users access the network over radio links. The degree to which the allocated frequency spectrum is reused over the cellular service area, however, determines the spectrum efficiency in cellular systems. That means the smaller the cell size, and the smaller the number of cells in the reuse geometry, the higher will be the spectrum usage efficiency. Since digital modulation systems can operate with a smaller signal to noise (i.e., signal to interference) ratio for the same service quality, they, in one respect, would allow smaller reuse distance and thus provide higher spectrum efficiency. This is one advantage the digital cellular provides over the older analogue cellular radio communication systems. It is worth mentioning that the digital systems have commonly used sectored cells with 120-degree or smaller directional antennas to further lower the effective reuse distance. This allows a smaller number of cells in the reuse pattern and makes a larger fraction of the total frequency spectrum available within each cell. Currently, research is being done on implementing other enhancements such as the use of dynamic channel assignment strategies for raising the spectrum efficiency in certain cases, such as high uneven traffic distribution over cells.

2.2.1. GSM SPECIFICATION

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Wavecom</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM (Flash)</td>
<td>16Mb</td>
</tr>
<tr>
<td>RAM</td>
<td>2Mb</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>3.1 – 4.5 V</td>
</tr>
<tr>
<td>Receiving Frequency</td>
<td>925 – 960 MHz</td>
</tr>
</tbody>
</table>
2.2.2. GSM BLOCK DIAGRAM

2.2.3. GSM NETWORK

A GSM network is composed of several functional entities, whose functions and interfaces are specified. The GSM network can be divided into three broad parts.

The Mobile Station is carried by the subscriber.
The Base Station Subsystem controls the radio link with the Mobile Station. The Network Subsystem, the main part of which is the Mobile services Switching Center (MSC), performs the switching of calls between the mobile users, and between mobile and fixed network users.

The MSC also handles the mobility management operations. Not shown is the Operations and Maintenance Center, which oversees the proper operation and setup of the network. The Mobile Station and the Base Station Subsystem communicate across the Um interface, also known as the air interface or radio link. The Base Station Subsystem communicates with the Mobile services Switching Center across the A interface.

2.2.3.1. Mobile Station:

Mobile Equipment (ME) such as hand portable and vehicle mounted unit. Subscriber Identity Module (SIM), which contains the entire customer related information (identification, secret key for authentication, etc.). The SIM is a small smart card, which contains both programming and information. The A3 and A8 algorithms are implemented in the Subscriber Identity Module (SIM). Subscriber information, such as the IMSI (International Mobile Subscriber
Identity), is stored in the Subscriber Identity Module (SIM). The Subscriber Identity Module (SIM) can be used to store user-defined information such as phonebook entries. One of the advantages of the GSM architecture is that the SIM may be moved from one Mobile Station to another. This makes upgrades very simple for the GSM telephone user. The use of SIM card is mandatory in the GSM world, whereas the SIM (RUIM) is not very popular in the CDMA world.

2.2.3.2. Base Station Subsystem (BSS):

All radio-related functions are performed in the BSS, which consists of base Station controllers (BSCs) and the base transceiver stations (BTSs).

2.2.3.3. Base Transceiver Station (BTS):

The Base Transceiver Station (BTS) contains the equipment for transmitting and receiving of radio signals (transceivers), antennas, and equipment for encrypting and decrypting communications with the Base Station Controller (BSC). A group of BTSs are controlled by a BSC. Typically a BTS for anything other than a picocell will have several transceivers (TRXs), which allow it to serve several different frequencies and different sectors of the cell (in the case of sectorised base stations). A BTS is controlled by a parent BSC via the Base Station Control Function (BCF). The BCF is implemented as a discrete unit or even incorporated in a TRX in compact base stations. The BCF provides an Operations and Maintenance (O&M) connection to the Network Management System (NMS), and manages operational states of each TRX, as well as software handling and alarm collection.
2.2.3.4. Base Station Controller (BSC):

The BSC controls multiple BTSs and manages radio channel setup, and handovers. The BSC is the connection between the Mobile Station and Mobile Switching Center. The Base Station Controller (BSC) provides, classically, the intelligence behind the BTSs. Typically a BSC has 10s or even 100s of BTSs under its control. The BSC handles allocation of radio channels, receives measurements from the mobile phones, controls handovers from BTS to BTS. A key function of the BSC is to act as a concentrator where many different low capacity connections to BTSs become reduced to a smaller number of connections towards the Mobile Switching Center (MSC) (with a high level of utilisation). Overall, this means that networks are often structured to have many BSCs distributed into regions near their BTSs which are then connected to large centralised MSC sites.

The BSC is undoubtedly the most robust element in the BSS as it is not only a BTS controller but, for some vendors, a full switching center, as well as an SS7 node with connections to the MSC and SGSN. It also provides all the required data to the Operation Support Subsystem (OSS) as well as to the performance measuring centers. A BSC is often based on a distributed computing architecture, with redundancy applied to critical functional units to ensure availability in the event of fault conditions. Redundancy often extends beyond the BSC equipment itself and is commonly used in the power supplies and in the transmission equipment providing the A-ter interface to PCU. The databases for all the sites, including information such as carrier frequencies, frequency hopping lists, power reduction levels, receiving levels for cell border calculation, are stored in the BSC.
2.2.3.5. Network Switching Subsystem (NSS):

Network Switching Subsystem is the component of a GSM system that carries out switching functions and manages the communications between mobile phones and the Public Switched Telephone Network. It is owned and deployed by mobile phone operators and allows mobile phones to communicate with each other and telephones in the wider telecommunications network. The architecture closely resembles a telephone exchange, but there are additional functions which are needed because the phones are not fixed in one location. There is also an overlay architecture on the GSM core network to provide packet-switched data services and is known as the GPRS core network. This allows mobile phones to have access to services such as WAP, MMS, and Internet access. All mobile phones manufactured today have both circuit and packet based services, so most operators have a GPRS network in addition to the standard GSM core network.

2.2.3.6. Mobile Switching Centre (MSC):

The Mobile Switching Centre or MSC is a sophisticated telephone exchange, which provides circuit-switched calling, mobility management, and GSM services to the mobile phones roaming within the area that it serves. This means voice, data and fax services, as well as SMS and call divert. In the GSM mobile phone system, in contrast with earlier analogue services, fax and data information is sent directly digitally encoded to the MSC. Only at the MSC is this re-coded into an "analogue" signal. There are various different names for MSCs in different context, which reflects their complex role in the network, all
of these terms though could refer to the same MSC, but doing different things at different times.

A Gateway MSC is the MSC that determines which visited MSC the subscriber who is being called is currently located. It also interfaces with the Public Switched Telephone Network. All mobile to mobile calls and PSTN to mobile calls are routed through a GMSC. The term is only valid in the context of one call since any MSC may provide both the gateway function and the Visited MSC function, however, some manufacturers design dedicated high capacity MSCs which do not have any BSCs connected to them. These MSCs will then be the Gateway MSC for many of the calls they handle.

The Visited MSC is the MSC where a customer is currently located. The VLR associated with this MSC will have the subscriber's data in it. The Anchor MSC is the MSC from which a handover has been initiated. The Target MSC is the MSC toward which a Handover should take place. An MSC Server is a part of the redesigned MSC concept starting from 3GPP Release 5.

2.2.4.FREQUENCY BAND USAGE:

Since radio spectrum is a limited resource shared by all users, a method must be devised to divide up the bandwidth among as many users as possible. The method chosen by GSM is a combination of Time- and Frequency-Division Multiple Access (TDMA/FDMA). The FDMA part involves the division by frequency of the (maximum) 25 MHz bandwidth into 124 carrier frequencies spaced 200 kHz apart. One or more carrier frequencies are assigned to each base station. Each of these carrier frequencies is then divided in time, using a TDMA scheme. The fundamental unit of time in this TDMA scheme is called a burst period and it lasts 15/26 ms (or approx. 0.577 ms). Eight burst periods are grouped into a TDMA frame (120/26 ms, or approx. 4.615 ms), which forms the
basic unit for the definition of logical channels. One physical channel is one burst period per TDMA frame.

Channels are defined by the number and position of their corresponding burst periods. All these definitions are cyclic, and the entire pattern repeats approximately every 3 hours. Channels can be divided into dedicated channels, which are allocated to a mobile station, and common channels, which are used by mobile stations in idle mode. A traffic channel (TCH) is used to carry speech and data traffic. Traffic channels are defined using a 26-frame multiframe, or group of 26 TDMA frames. The length of a 26-frame multiframe is 120 ms, which is how the length of a burst period is defined (120 ms divided by 26 frames divided by 8 burst periods per frame). Out of the 26 frames, 24 are used for traffic, 1 is used for the Slow Associated Control Channel (SACCH) and 1 is currently unused. TCHs for the uplink and downlink are separated in time by 3 burst periods, so that the mobile station does not have to transmit and receive simultaneously, thus simplifying the electronics. In addition to these full-rate TCHs, there are also half-rate TCHs defined, although they are not yet implemented. Half-rate TCHs will effectively double the capacity of a system once half-rate speech coders are specified (i.e., speech coding at around 7 kbps, instead of 13 kbps). Eighth-rate TCHs are also specified, and are used for signalling. In the recommendations, they are called Stand-alone Dedicated Control Channels (SDCCH).
Organization of bursts, TDMA frames, and multiframes for speech and data

GSM is a digital system, so speech which is inherently analog, has to be
digitized. The method employed by ISDN, and by current telephone systems for
multiplexing voice lines over high speed trunks and optical fiber lines, is Pulse
Coded Modulation (PCM). The output stream from PCM is 64 kbps, too high a
rate to be feasible over a radio link. The 64 kbps signal, although simple to
implement, contains much redundancy. The GSM group studied several speech
coding algorithms on the basis of subjective speech quality and complexity
(which is related to cost, processing delay, and power consumption once
implemented) before arriving at the choice of a Regular Pulse Excited -- Linear
Predictive Coder (RPE--LPC) with a Long Term Predictor loop. Basically,
information from previous samples, which does not change very quickly, is
used to predict the current sample. The coefficients of the linear combination of
the previous samples, plus an encoded form of the residual, the difference
between the predicted and actual sample, represent the signal. Speech is divided
into 20 millisecond samples, each of which is encoded as 260 bits, giving a total
bit rate of 13 kbps. This is the so-called Full-Rate speech coding. Recently, an
Enhanced Full-Rate (EFR) speech-coding algorithm has been implemented by
some North American GSM1900 operators. This is said to provide improved speech quality using the existing 13 kbps bit rate.

2.2.5. WORKING

The GSM module is connected with the controller. As the controller is keep on monitoring the doors and locker key, when the door get opened, the microcontroller sends the command “AT” to initiate the module. Now the module sends an sms as “Theft Occurred” to the already fed mobile number. Thus the information is passed from the module to the Authorized person. Whenever it receives the correct password from the mobile, it will inform the microcontroller to open the door.

2.2.6. FEATURES

Performance - Fast with high real throughput
Integrity - Secure controlled data transfer
Network Access - Quick and consistent
Contestation Control - Avoid conflicts and collisions
Installation - Simple quick installation
Frequency Choice - Choice of RF bands to suit different terrains
Network Diagnostics - For ease of maintenance and cost saving
2.3. STEPPER MOTOR

A stepper motor is a brushless, synchronous electric motor that can divide a full rotation into a large number of steps, for example, 200 steps. Thus the motor can be turned to a precise angle.

2.3.1. DRIVER CIRCUIT

![Stepper Motor Diagram]

\[
V = 4 \times \text{Motor Voltage} \\
R_s = 3 \times (\text{Motor Resistance / phase}) \\
\text{Suitable for Slow RPM}
\]
2.3.2 FUNDAMENTALS OF OPERATION

Stepper motor operate much differently from normal DC motors, which simply spin when voltage is applied to their terminals. Stepper motor effectively have multiple “Toothed” electromagnets arranged around a central metal gear. To make the motor shaft turn, first one electromagnet is given power, which makes the gear’s teeth magnetically attracted to the electromagnets teeth. When the gear’s teeth are thus aligned to the first electromagnet, they are slightly offset from the electromagnet. So when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with next one, and from there the process is repeated. Each of those slight rotation is called a “Step”.

The top electromagnet (1) is charged, attracting the topmost four teeth of a sprocket.
The top electromagnet (1) is turned off, and the right electromagnet (2) is charged, pulling the nearest four teeth to the right. This results in a rotation of 3.6°.

The bottom electromagnet (3) is charged; another 3.6° rotation occurs.
The left electromagnet (4) is enabled, rotating again by 3.6°. When the top electromagnet (1) is again charged, the teeth in the sprocket will have rotated by one tooth position; since there are 25 teeth, it will take 100 steps to make a full rotation.

2.3.3. APPLICATIONS

Computer controlled stepper motors are one of the most versatile forms of positioning systems, particularly when digitally controlled as part of a servosystem. Stepper motors are used in floppy disk drives, flatbed, scanners, printers, plotters and many more devices note that the hard drives no longer use stepper motors to position the read/write heads instead utilizing a voice coil and servo feedback in head positioning.

2.4. LIQUID CRYSTAL DISPLAY
A liquid crystal display is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is prized by engineers because it uses very small amounts of electric power and is therefore suitable for use in battery powered electronic devices.

Each pixel of an LCD consists of a layer of perpendicular molecules aligned between two transparent electrodes and two polarizing filters, the axes of polarity of which are perpendicular to each other with no liquid crystal between the polarizing crystals.

The surfaces of the electrodes that are in contact with the liquid crystal material are treated so as to align the liquid crystal molecules in a particular direction this treatment typically consists of a thin polymer layer that is unidirectionally rubbed using a cloth.

Before applying the electric field, the orientation of the liquid crystal molecules is determined by the alignment at the surfaces. In a twisted pneumatic device, the surface alignment directions at the two electrodes are perpendicular and so the molecules arrange themselves in a helical structure or twist. Because the liquid crystal material is birefringent, light passing through the liquid crystal, allowing it to pass through the second polarized filter.

When a voltage is applied across the electrodes, torque acts to align the liquid crystal molecules parallel to the electric fields, distorting the helical structures. This reduces the rotation of the polarization of the incident light, and the device appears grey. If the
applied voltage is the polarization of the incident light is not rotated and it passes through the crystal layer.

With a twisted pneumatic liquid crystal device it is usual to operated the device between crossed polarizes, such that it appears bright with no applied voltage. With this setup, the dark voltage-on state is uniform. The device can be operated between parallel polarizes, in which case the bright and dark states are reversed.

Both the liquid crystal material and alignment layer material contain ionic components. If an electric field of one particular polarity is applied for a long period of time, this ionic material is attracted to the surfaces and degrades the device performance. This is avoided by applying either an alternating current, or reversed by the polarity of the electric field as the device is addressed.
MICROCONTROLLER
AND SERIAL
COMMUNICATION

2.5. MICROCONTROLLER AND SERIAL COMMUNICATION
2.5.1 MICROCONTROLLER 89C51

The microcontroller used here is P89C51RD2BN. The expansion of the part number of this microcontroller is given below.

![Diagram of 89C51 part number expansion]

Fig. 1.3 89C51 part number expansion

The P89C51RD2BN contains a non-volatile 64KB Flash program memory that is both parallel programmable and serial In-System and In-Application Programmable. In-System Programming (ISP) allows the user to download new code while the microcontroller sits in the application. In-Application Programming (IAP) means that the microcontroller fetches new program code and reprograms itself while in the system. This allows for remote programming over a modem link. A default serial loader (boot loader) program in ROM allows serial In-System programming of the Flash memory via the UART without the need for a loader in the Flash code. For In-Application Programming, the user program erases and reprograms the Flash memory by use of standard routines contained in ROM.
The device supports 6-clock/12-clock mode selection by programming a Flash bit using parallel programming or In-System Programming. In addition, an SFR bit (X2) in the clock control register (CKCON) also selects between 6-clock/12-clock mode. Additionally, when in 6-clock mode, peripherals may use either 6 clocks per machine cycle or 12 clocks per machine cycle. This choice is available individually for each peripheral and is selected by bits in the CKCON register. This device is a Single-Chip 8-Bit Microcontroller manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The instruction set is 100% compatible with the 80C51 instruction set. The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. The added features of the P89C51RD2BN make it a powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.

When the 89C51 microcontroller is connected to a crystal oscillator and is powered up, we can observe the frequency on the XTAL2 pins using the oscilloscope. The time to execute the instruction is calculated by using the following expression,

\[
T(\text{inst}) = \frac{(MC \times Cn)}{(\text{crystal})}
\]

Oscillator Frequency f

\[
\text{State 1} \quad \text{State 2} \quad \text{State 3} \quad \text{State 4} \quad \text{State 5} \quad \text{State 6}
\]
Fig. 1.4. CPU clock cycles

MC → Number of Machine Cycles for an instruction to execute and Cn is the number of clock cycles for one machine cycle. For 89C51RD2BN the number of clock cycles for one machine cycle is 12. For example, If the number of machine cycles to execute a instruction is 1 and the oscillator frequency used is 11.0592MHz, the time to execute an instruction is 1.085μs.

2.5.2. Basic Features of 89C51

- 80C51 Central Processing Unit
- On-chip Flash Program Memory with In-System Programming (ISP) and In-Application Programming (IAP) capability
- Boot ROM contains low level Flash programming routines for downloading via the UART
- Supports 6-clock/12-clock mode via parallel programmer (default clock mode after Chip Erase is 12-clock)
- 6-clock/12-clock mode Flash bit erasable and programmable via ISP
- 6-clock/12-clock mode programmable “on-the-fly” by SFR bit
- Peripherals (PCA, timers, UART) may use either 6-clock or 12-clock mode while the CPU is in 6-clock mode
- Speed up to 20 MHz with 6-clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clocks per machine cycle
- Fully static operation
- RAM expandable externally to 64 kilo bytes
- Four interrupt priority levels
- Seven interrupt sources
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock-out pin
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programmable Counter Array (PCA)
- PWM
- Capture/compare

### 2.5.2.1 Pin Description

Examining the following figure, note that of the 40 pins a total of 32 pins are set aside for the four ports P0, P1, P2 and P3, where each port takes 8 pins. The rest of the pins are designated as Vcc, GND, XTAL1, XTAL2, RST, EA,
ALE, and PSEN. Of these 8 pins, all 8051 derivatives use six of them. In other words, they must be connected in order for the system to work.

Fig. 1.5. Pin Diagram of 89C51 Microcontroller

1–8: P1.0 to P1.7 (Port 1): Each of these pins can be used as either input or output according to your needs. Port 1 is an 8-bit bi-directional I/O port with internal pull-ups on all pins. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. Each pin of Port1 has an alternate function

- **Pin 1: T2 (P1.0)** - Clock input of counter 0
Pin 2: **T2EX (P1.1)** - Timer/Counter 2Reload / Capture / Direction Control

Pin 3: **ECI (P1.2)** - External Clock Input to the PCA

Pin 4: **CEX0 (P1.3)** - External I/O for PCA module 0

Pin 5: **CEX1 (P1.4)** - External I/O for PCA module 1

Pin 6: **CEX2 (P1.5)** - External I/O for PCA module 2

Pin 7: **CEX3 (P1.6)** - External I/O for PCA module 3

Pin 8: **CEX4 (P1.7)** - External I/O for PCA module 4

**9: RST (Reset Signal):** High logical state on this input halts the MCU and clears all the registers. Bringing this pin back to logical state zero starts the program anew as if the power had just been turned on. In another words, positive voltage impulse on this pin resets the MCU. Depending on the device's purpose and environs, this pin is usually connected to the push-button, reset-upon-start circuit or a brown out reset circuit.

**10-17: P3.0 to P3.7 (Port 3):** As with Port 1, each of these pins can be used as universal input or output. However, each pin of Port 3 has an alternative function:

- **Pin 10: RxD(P3,0)** - Serial input for asynchronous communication
- **Pin 11: TxD(P3.1)** - Serial output for asynchronous communication
- **Pin 12: INT0(P3.2)** - Input for interrupt 0
- **Pin 13: INT1(P3.3)** - Input for interrupt 1
- **Pin 14: T0(P3.4)** - Clock input of counter 0
- **Pin 15: T1(P3.5)** - Clock input of counter 1
• Pin 16: **WR(P3.6)** - Signal for writing to external RAM memory
• Pin 17: **RD(P3.7)** - Signal for reading from external RAM memory

**18-19: XTAL2 and XTAL1 (Crystal input and output):** Input and output of internal oscillator. Quartz crystal controlling the frequency commonly connects to these pins.

**20: V_{ss}:** Ground

**21-28: P2.0 to P2.7 (Port 2):** If external memory is not present, pins of Port 2 act as universal input/output. If external memory is connected, this is the location of the higher address byte, i.e. addresses A8 – A15. It is important to note that in cases when not all the 8 bits are used for addressing the memory (i.e. memory is smaller than 64kB), the rest of the unused bits are not available as input/output.

**29: PSEN (Program Store Enable):** MCU activates this bit (brings to low state) upon each reading of byte instruction from program memory. If external ROM is used for storing the program, PSEN is directly connected to its control pins.

**30: ALE (Address Latch Enable):** Before each reading of the external memory, MCU sends the lower byte of the address register (addresses A0 – A7) to port P0 and activates the output ALE. External Chip (eg: 74HC373), memorizes the state of port P0 upon receiving a signal from ALE pin, and uses it as part of the address for memory chip. During the second part of the MCU cycle, signal on ALE is off, and port P0 is used as *Data Bus*. In this way, by adding only one integrated circuit, data from
port can be multiplexed and the port simultaneously used for transferring both addresses and data.

31: **EA (External Access Enable):** Bringing this pin to the logical state zero designates the ports P2 and P3 for transferring addresses regardless of the presence of the internal memory. This means that even if there is a program loaded in the MCU it will not be executed, but the one from the external ROM will be used instead. Conversely, bringing the pin to the high logical state causes the controller to use both memories, first the internal, and then the external (if present).

32-39: **P0.7 to P0.0 (Port 0):** Similar to Port 2, pins of Port 0 can be used as universal input/output, if external memory is not used. If external memory is used, P0 behaves as address output (A0 – A7) when ALE pin is at high logical level, or as data output (Data Bus) when ALE pin is at low logical level.

40: $V_{CC}$: Power +5V

### 2.6. Architecture of 89C51 Microcontroller

The architecture of the 8051 family of microcontrollers (8051 derivatives) is referred to as the MCS-51 architecture (Micro Controller Series – 51), or sometimes simply as MCS-51. The block diagram of 89C51 microcontroller is shown below.
**ACCUMULATOR (ACC):** Accumulator is a general-purpose register, which stores runtime results. Before performing any operation upon an operand, operand has to be stored in the accumulator. Results of arithmetical operations are also stored in the accumulator. When transferring data from one register to another, it has to go through the accumulator. Due to its versatile role, this is the most frequently used register, essential part of every MCU.

**B REGISTER:** B Register is used along with the Accumulator for multiplication and division. This B register provides temporary storage space for the result of multiplication & division operation. Instructions of multiplication and division can be applied only to operands located in registers A and B. Other instructions can use this register as a secondary accumulator (A).

**PORTS:** Term "port" refers to a group of pins on a microcontroller which can be accessed simultaneously, or on which we can set the desired combination of zeros and ones, or read from them an existing status. Ports represent physical connection of Central Processing Unit with an outside world. Microcontroller uses them in order to monitor or control other components or devices. 89C51 has 4 ports; with each port have 8-bit length. All the ports are bit and byte addressable.
**Fig. 1.6. 89C51 Architecture**

**Port 0 (P0):** Port 0 has two-fold role: If external memory is used, P0 behaves as address output (A0 – A7) when ALE pin is at high logical level, or as data output (Data Bus) when ALE pin is at low logical level, otherwise all bits of the port are either input or output. Another feature of this port comes to play when it has been designated as output. Unlike other ports, Port 0 lacks the "pull up" resistor (resistor with +5V on one end). This seemingly insignificant change has the following consequences:

- When designated as input, pin of Port 0 acts as high impedance offering the infinite input resistance with no "inner" voltage.
• When designated as output, pin acts as "open drain". Clearing a port bit grounds the appropriate pin on the case (0V). Setting a port bit makes the pin act as high impedance. Therefore, to get positive logic (5V) at output, external "pull up" resistor needs to be added for connecting the pin to the positive pole.

Therefore, to get one (5V) on the output, external "pull up" resistor needs to be added for connecting the pin to the positive pole.

**Port 1 (P1):** Port 1 is I/O port. Having the "pull up" resistor, Port 1 is fully compatible with TTL circuits. The alternate functions of Port1 are

<table>
<thead>
<tr>
<th>Pin</th>
<th>Alternate Name</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1.0</td>
<td>T2</td>
<td>Serial input</td>
</tr>
<tr>
<td>P1.1</td>
<td>T2EX</td>
<td>Serial output</td>
</tr>
<tr>
<td>P1.2</td>
<td>ECI</td>
<td>External interrupt 0</td>
</tr>
<tr>
<td>P1.3</td>
<td>CEX0</td>
<td>External interrupt 1</td>
</tr>
<tr>
<td>P1.4</td>
<td>CEX1</td>
<td>Timer 0 external input</td>
</tr>
<tr>
<td>P1.5</td>
<td>CEX2</td>
<td>Timer 1 external input</td>
</tr>
<tr>
<td>P1.6</td>
<td>CEX3</td>
<td>Signal <strong>write</strong> to external memory</td>
</tr>
<tr>
<td>P1.7</td>
<td>CEX4</td>
<td>Signal <strong>read</strong> from external memory</td>
</tr>
</tbody>
</table>
Port 2 (P2): When using external memory, this port contains the higher address byte (addresses A8–A15), similar to Port 0. Otherwise, it can be used as universal I/O port.

Port 3 (P3): Beside its role as universal I/O port, each pin of Port 3 has an alternate function. In order to use one of these functions, the pin in question has to be designated as input, i.e. the appropriate bit of register P3 needs to be set. By selecting one of the functions the other one is disabled. From a hardware standpoint, Port 3 is similar to Port 0. The alternate functions of Port 3 is given below

<table>
<thead>
<tr>
<th>Pin</th>
<th>Alternate Name</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3.0</td>
<td>RXD</td>
<td>Serial input</td>
</tr>
<tr>
<td>P3.1</td>
<td>TXD</td>
<td>Serial output</td>
</tr>
<tr>
<td>P3.2</td>
<td>INT0</td>
<td>External interrupt 0</td>
</tr>
<tr>
<td>P3.3</td>
<td>INT1</td>
<td>External interrupt 1</td>
</tr>
<tr>
<td>P3.4</td>
<td>T0</td>
<td>Timer 0 external input</td>
</tr>
<tr>
<td>P3.5</td>
<td>T1</td>
<td>Timer 1 external input</td>
</tr>
<tr>
<td>P3.6</td>
<td>WR</td>
<td>Signal <strong>write</strong> to external memory</td>
</tr>
<tr>
<td>P3.7</td>
<td>RD</td>
<td>Signal <strong>read</strong> from external memory</td>
</tr>
</tbody>
</table>

Data Pointer (DPTR): The Data pointer register is made up of two 8 bit registers, named DPH (Data Pointer High) and DPL (Data Pointer Low). These
registers are used to give addresses of the internal or external memory. The DPTR is under the control of program. DPTR is also manipulated as one 16 bit register, DPH & DPL are each assigned an address. The 89C51 microcontroller has additional DPTR. The dual DPTR structure is a way by which the chip will specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS (bit0 in AUXR1) that allows the program code to switch between them.

**Stack Pointer (SP)**: The stack refers to an area of internal RAM that is used in conjunction with certain opcodes to store and retrieve data quickly. The register used to access the stack is called Stack Pointer. The 8 bit stack pointer register is used by the 89C51 to hold an internal RAM address that is called then top of the stack. The stack pointer increments before storing the data on the stack. As retrieved from the stack the SP is decremented by one. The number in Stack Pointer points to the location of the last "valid" address within the Stack. With the beginning of every new routine, Stack Pointer increases by 1; upon return from routine, SP decreases by 1. Upon reset (or turning the power on), the stack pointer contains the value 07h.

**Program Counter (PC)**: Used to access code memory. Program counter always points to the address of the next instruction in memory to be executed. Upon reset (or turning the power on), the program counter resets to the starting location of the program.

**Instruction Register**: When an instruction is fetched from the Flash memory, it is loaded in the instruction register.

**Timing & Control unit**: The timing and control unit synchronizes the operation of the microcontroller and generates control signals necessary for communication between the microcontroller and the peripherals.
**Program Status Word (PSW):** The Program Status Word (PSW) register is an 8 bit register. It is also referred to as the flag register. It contains the math flags, user program flag F0, and the register select bits that identify which of the four general purpose register banks is currently in use by the program.

**Oscillator:** Oscillator circuit is used for providing a microcontroller with a clock. Clock is needed so that microcontroller could execute a program or program instructions. Stable pace provided by the oscillator allows harmonious and synchronous functioning of all other parts of MCU. The manufacturers make available 89C51 designs that can run at specified maximum and minimum frequencies, typically 1 megahertz to 33 megahertz. Minimum frequencies imply that some internal memories are dynamic and must always operate above a minimum frequency or data will be lost.

**Interrupts:** An interrupt is a signal from a device attached to a computer or from a program within the computer that causes the main program that operates the computer to stop and points out what to do next. In general, there are hardware interrupts and software interrupts. A hardware interrupt is related to the hardware of the system. For example, when an I/O operation is completed such as reading some data into the computer from a keyboard interrupt the main program. As the name implies the software interrupts related to the software of the system. It occurs when an application program terminates or requests certain services from the operating system.

**Timers/Counters:** Timers are usually the most complicated parts of a microcontroller. Physically, timer is a register whose value is continually increasing to FFFFh, and then it starts all over again: 0h, 1h, 2h, 3h, 4h...FFFFh....0h, 1h, 2h, 3h......etc. The 89C51 MCU clock employs a quartz crystal. As this frequency is highly stable and accurate, it is ideal for time measuring. Since one instruction takes 12 oscillator cycles to complete, the
math is easy. 89C51 has three Timers/Counters marked as T0, T1 & T2. Their purpose is to measure time and count external occurrences, but can also be used as clock in serial communication purpose called as, **Baud Rate**.

**Serial Port**: Serial port is used to provide communication among two devices. Serial data communication has been widely used for long distance communication because of the ease and the economy of using only one wire to transmit data. Serial port is also referred as RS232 port. RS232 is a asynchronous way of communication. Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word, which are used to synchronize the sending and receiving units.

When a word is given to the UART for Asynchronous transmissions, a bit called the **"Start Bit"** is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter.

---

TRANSMITTER USES AN INTERNAL CLOCK TO DETERMINE WHEN TO SEND EACH BIT

RECEIVER DETECTS THE FALLING EDGE OF START, THEN

USES ITS INTERNAL CLOCK TO READ THE FOLLOWING BITS
ASYNCHRONOUS TRANSMISSION SEND LSB FIRST

Fig 1. 7: Asynchronous Transmission

2.6.1. Memory Organization
RAM (Data Memory)

RAM is used for storing temporary data and auxiliary results generated during the runtime.

The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.

The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.

The Special Function Registers, SFRs, (addresses 80H to FFH) are directly addressable only.

The 768-bytes expanded RAM (ERAM, 00H – 2FFH) are indirectly accessed by move external instruction, MOVX,

![Fig 1.8: RAM Organization](image)

The Lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can be accessed by indirect addressing only. The Upper 128 bytes occupy the same address space as the SFR. That means
they have the same address, but are physically separate from SFR space. The ERAM can be accessed by indirect addressing, with EXTRAM bit in the AUXR register cleared and MOVX instructions. This part of memory is physically located on-chip, logically occupies the first 7936-bytes of external data memory. With EXTRAM = 0, the ERAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to ERAM will not affect ports P0, P3.6 (WR) and P3.7 (RD). P2 SFR is output during external addressing. The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the ERAM.

Fig 1.9: Internal RAM Architecture
**Flash Memory (Program Memory)**

89C51 have built-in 64 kilo bytes of Flash memory. The P89C51RD2BN Flash memory augments EPROM functionality with in-circuit electrical erasure and programming. The Flash can be read and written as bytes. The Chip Erase operation will erase the entire program memory. The Block Erase function can erase any Flash block. In-system programming and standard parallel programming are both available. On-chip erase and write timing generation contribute to a user-friendly programming interface. The P89C51RD2BN Flash reliably stores memory contents even after 10,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling. The P89C51RD2BN uses a +5 V VPP supply to perform the Program/Erase algorithms.

![Fig 1.10: 89C51 Program Memory](image-url)
Special Function Registers

Special Function Register (SFR) can be seen as a sort of control panel for managing and monitoring the microcontroller. Every register and each of the belonging bits has its name, specified address in RAM and strictly defined role (e.g. controlling the timer, interrupt, serial communication, etc). Although there are 128 available memory slots for allocating SFR registers. The rest has been left open intentionally to allow future upgrades while retaining the compatibility with earlier models. This fact makes possible to use programs developed for obsolete models long ago.

2.6.2. RS-232 Level Converters

Almost all digital devices which we use require either TTL or CMOS logic levels. Therefore the first step to connecting a device to the RS-232 port is to transform the RS-232 levels back into 0 and 5 Volts. As we have already covered, this is done by RS-232 Level Converters. Two common RS-232 Level Converters are the 1488 RS-232 Driver and the 1489 RS-232 Receiver. Each package contains 4 inverters of the one type, either Drivers or Receivers. The driver requires two supply rails, +7.5 to +15v and -7.5 to -15v. As you could imagine this may pose a problem in many instances where only a single supply of +5V is present.

However the advantages of these I.C's are they are cheap. Above: (Figure 6) Pinouts for the MAX-232, RS-232 Driver/Receiver. Right: (Figure 7) Typical MAX-232 Circuit. Another device is the MAX-232. It includes a Charge Pump, which generates +10V and -10V from a single 5v supply. This I.C. also includes two receivers and two transmitters in the same
package. This is handy in many cases when you only want to use the Transmit and Receive data Lines. You don't need to use two chips, one for the receive line and one for the transmit. However all this convenience comes at a price, but compared with the price of designing a new power supply it is very cheap.

2.6.2.1 PIN CONFIGURATION

![Pin Configuration Diagram]

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Carrier Detect</td>
<td>6</td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>2</td>
<td>Received Data</td>
<td>7</td>
<td>Request to Send</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted Data</td>
<td>8</td>
<td>Clear to Send</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal Ready</td>
<td>9</td>
<td>Ring Indicator</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.7 POWER SUPPLY

Power Supply is the device that transfers electric power from a source to a load using electronic circuits. Power supplies are used in many industrial and aerospace applications and also in consumer products. Some of the requirements of power supplies are small size, lightweight, low cost, and high power conversion efficiency. In addition to these, some power supplies require the following: electrical isolation between the source and load, low harmonic distortion for the input and output waveforms, and high power factor (PF) if the
source is ac voltage. Some special power supplies require controlled direction of power flow. Typical application of power supplies is to convert utility's AC input power to a regulated voltage(s) required for the purpose of the regulation of the device.

2.7.1. LINEAR POWER SUPPLY

A linear power supply is the oldest and simplest type of power supply. In these power supplies, electrical isolation can only be provided by bulky line frequency transformers. The ac source can be rectified with a bridge rectifier to get an uncontrolled dc, and then a dc-to-dc converter can be used to get a controlled dc output. The output voltage is regulated by dropping the extra input voltage across a series transistor (therefore, also referred to as a series regulator). They have very small output ripple, theoretically zero noise, large hold-up time (typically 1–2 ms), and fast response.

Linear Power Supply Block Diagram:

The action of a transformer is such that a time-varying (AC) voltage or current is transformed to a higher or lower value, as set by the transformer turns ratio. The transformer does not add power, so it follows that the power (V × I) on either side must be constant. That is the reason that the winding with more turns has higher voltage but lower current, while the winding with less turns has lower voltage but higher current. The step down transformer converts the AC input with the higher level to some lower level.
2.7.2 MULTI OUTPUT POWER SUPPLY

It is also possible to generate multiple voltages using linear power supplies. In multi output power supply a single voltage must be converted into the required system voltages (for example, +5V, +12V and -12V) with very high power conversion efficiency.
Fig. 1.27 Multi output linear power supply
SIMULATION CODES

//******************BANK LOCKER SYSTEM USING GSM******************

//******************Password is 11253**************************

#include<reg51.h>
#include<stdio.h>
#include<string.h>

//******************PREPROCESSOR DIRECTIVES**************************
sbit security_sw=P3^2;
sbit locker=P3^3;
sbit sw1=P1^3;
sbit sw2=P1^4;
sbit red_led=P1^5;
sbit green_led=P1^6;

//******************PREPROCESSOR DIRECTIVES FOR LCD******************
sbit RS=P1^0;
sbit RW=P1^1;
sbit EN=P1^2;

//******************PREPROCESSOR DIRECTIVES FOR KEYS******************
sbit key1=P3^7;
sbit key2=P3^6;
sbit key3=P3^5;
sbit key4=P3^4;
sbit enter=P1^7;

//***************VARIABLE DECLARATION**********
unsigned char init[]= "AT";
unsigned char text[]="AT+CMGF=1";
unsigned char read[]="AT+CMGR=1";
unsigned char no[] = "AT+CMGS="9486163383";"
unsigned char sms[] = "THEFT OCCURED";
unsigned char del_all[]="AT+CMGD=1,4";
unsigned char del[]="AT+CMGD=1";
unsigned char clockwise1[4]={0x09,0x05,0x06,0x0A};
unsigned char clockwise2[4]={0x90,0x50,0x60,0xA0};
unsigned char anticlockwise1[4]={0x0A,0x06,0x05,0x09};
unsigned char anticlockwise2[4]={0xA0,0x60,0x50,0x90};
unsigned char valid_data[10],c[10],rec[80],check=0,chk_msg=0;

//******************VARIABLE DECLARATION FOR LCD*****************
unsigned char code command1[5]={0x38,0x01,0x06,0x0c,0x80};

unsigned char key[10],count;
/***********FUNCTION TO TRANSMIT THE DATA THROUGH SERIAL PORT***********/

void transmit(unsigned char array[]) {
    unsigned int i;
    TMOD=0x20;
    TH1=0xFD;
    TL1=0x00;
    SCON=0x50;
    TR1=1;
    for(i=0;array[i]!='\0';i++) {
        SBUF=array[i];
        while(TI==0);
        TI=0;
    }
    SBUF=0x0d;
    while(TI==0);
    TI=0;
}

/**********************DELAY FUNCTION**********************/

void delay(unsigned long int y) {
    
}
int x;
for(x=0;x<y;x++);
}

//**********FUNCTION TO AUTOMATE THE STEPPER MOTOR**********

void stepper(unsigned char array[])
{
    unsigned int i,j;
    for(i=0;i<13;i++)
    {
        for(j=0;j<4;j++)
        {
            P0=array[j];
            delay(800);
        }
    }
}

/****************************DELAY**************************

************/

void busy_check()
{
    unsigned char z;
    for(z=0;z<1000;z++);
}
//******************** LCD COMMAND
FUNCTION********************************/

void lcd_command()
{
    unsigned int i;
    RS=0;RW=0;EN=0;
    for(i=0;i<5;i++)
    {
        P2=command1[i];
        EN=1;
        busy_check();
        EN=0;
    }
}

//*************** LCD DISPLAY
FUNCTION*******************************************/

void lcd_display(unsigned char array[])
{
    unsigned char i;

    for(i=0;array[i]!="\0";i++)
    {
        RS=1;
        P2=array[i];
        EN=1;
    }
}
busy_check();
EN=0;
}
}

/**LCD SINGLE COMMAND**********/
void lcd_com(unsigned char code c)
{
    RS=0;RW=0;EN=0;
P2=c;
    EN=1;
    busy_check();
    EN=0;
}

/**LCD SINGLE COMMAND**********/
void lcd_data(unsigned char code d)
{
    RS=1;RW=0;EN=0;
P2=d;
    EN=1;
    busy_check();
    EN=0;
}
unsigned char single_rx()
{
  unsigned char ch;
  while(RI==0);
  RI=0;
  ch=SBUF;
  return(ch);
}

void compare()
{
  unsigned int i;
  unsigned char dat;
  while(single_rx()!='+');//CHECK FOR NEW MESSAGE
  ACKNOWLEDGMENT
  {
    while(single_rx()!='T');
    {
      while(single_rx()=='I')
{  
delay(1000);
transmit(read);
while(single_rx()!=':');
dat=single_rx();
while(single_rx()!=':');
while(single_rx()!=':');
while(single_rx()!='"');
dat=single_rx();
dat=single_rx();
for(i=0;rec[i-1]!=0x0D;i++)
{
    while(RI==0);
    RI=0;
    rec[i]=SBUF;
}
for(i=0;i<4;i++)
{
    valid_data[i]=rec[i];
}
valid_data[i]='\0';
delay(100);
transmit(valid_data);
if((strcmp(valid_data,"open")==0)||(strcmp(valid_data,"Open")==0))
{

stepper(clockwise2);//open the second door
transmit(del);
delay(10000);
check=0;
}
else
{
transmit(del);
delay(10000);
check=0;
}


//******************FUNCTION FOR GETTING PASSWORD
FROM THE KEYBOARD***********

void psw(void)
{
loop:
if(key1==0)
{
}
key[count]=0x31;
count=count+1;
lcd_data('*');
delay(3000);
}
else if(key2==0)
{
    key[count]=0x32;
count=count+1;
lcd_data('*');
delay(3000);
}
else if(key3==0)
{
    key[count]=0x33;
count=count+1;
lcd_data('*');
delay(3000);
}
else if(key4==0)
{
    key[count]=0x34;
count=count+1;
lcd_data('*');
delay(3000);
}
else if(enter==0)
{  
    key[count]=0x35;  
    count=count+1;  
    lcd_data('*');  
    delay(3000);  
}
else  
goto loop;

void security_mode()  
{  
    int i;  
        red_led=1;  
    lcd_com(0x80);  
    lcd_display(" Key Open In ");  
    lcd_com(0xc0);  
    lcd_display(" Security Mode ");  
    delay(10000);  
    lcd_com(0x80);  
    lcd_display(" Second door ");  
    lcd_com(0xc0);  
    lcd_display(" Closed ");  
    stepper(anticlockwise2);  //close the 2nd door  
    lcd_com(0x80);  
    lcd_display("Message sending ");
```
lcd_com(0xc0);
lcd_display(" .... ");
delay(500000);
transmit(init);
delay(5000);
transmit(text);
delay(10000);
transmit(no);
delay(10000);
transmit(sms);
delay(1000);
SBUF=0X1A;
while(TI=0);
TI=0;
lcd_com(0x80);
lcd_display(" Waiting for ");
lcd_com(0xc0);
lcd_display(" Message ");
compare();// get the control to open the second door through sms
lcd_com(0x80);
lcd_display(" Key Open In ");
lcd_com(0xc0);
lcd_display(" Security Mode ");
while(security_sw==0);
red_led=0;
```
void check_password()
{
    lcd_com(0x80);
    lcd_display(" Enter Password ");
    lcd_com(0xc0);
    lcd_display("            ");
    lcd_com(0xc6);
    do
    {
        psw();
    } while(count<5);
    key[count]='\0';
    count=0;
    if(strcmp(key,"11253")==0)
    {
        lcd_com(0x80);
        lcd_display("Allowed to Access");
        lcd_com(0xc0);
        lcd_display("    key    ");
        check=check+1;
        stepper(clockwise1);
        delay(10000);
    }
    if(strcmp(key,"11253")!=0)
    {
        lcd_com(0x80);
void main()
{
    int i, j;
    red_led = green_led = 0;
    lcd_command();
    lcd_display(" BANK LOCKER ");
    lcd_com(0xc0);
    lcd_display(" SYSTEM ");
    delay(10000);

    transmit(init);
    delay(3000);
    transmit(del_all);
    delay(3000);
    green_led = 1;
    while(1)
    {
        next:
lcd_com(0x80);
lcd_display("  BANK LOCKER   ");
lcd_com(0xc0);
lcd_display("    SYSTEM      ");

//************get the password from the keyboard to open the first door*******************
if(sw1==0)
{
    for(i=0;i<2;i++)
    {
        if(check==0)
            check_password();
        else
            goto next;
    }
} while(sw1==0);
check=0;

//******************to close the first door*****************************
if(sw2==0)
    stepper(anticlockwise1);
while(sw2==0);

//**********************check if the system in security mode*************
if(security_sw==0)
{
    if(locker==1)
security_mode();

}  
if(locker==1)
    red_led=1;
    if(locker==0)
        red_led=0;

}