# Secured smart ATM transaction

**C. R. Balamurugan, K. Ramash Kumar, A. Thirumalai**
Karpagam College of Engineering Coimbatore, India

**Article Info**

***Article history:***

Received Des 2, 2018
Revised Jan 21, 2019
Accepted Feb 3, 2019

***Keywords:***

Atm
Savings
Tax
Transaction

**ABSTRACT**

The objective of this paper is to reduce the service tax during mobile transactions. To improve the security and to make the process easy and less time consuming this process is rendered with the help of GSM (Global System for Mobile communication), finger print sensors, PIC16F877A microcontroller and aadhaar number.

***Corresponding Author:***

C. R. Balamurugan,
Karpagam College of Engineering Coimbatore, India.
Email: crbalain2010@gmail.com

## 1. INTRODUCTION

Secured smart ATM transaction is newly innovated process from the conventional ATM transaction method. Card and pin number is used as an accessing key for money transaction in the ordinary ATM. But in SECURED SMART ATM TRANSACTION both card and the pin is eliminated instead of that Fingerprint authentication and GSM signal control is used. During normal money transaction process the sender who sends the money should transact the amount of money from his account to receivers account [1]. Then receiver need to withdraw the cash from the ATM. But in Smart ATM Transaction sender need not required to transact money to receiver instead of that the sender can just provide an account accessing permission to withdraw limited amount of money from his/her account directly. By this, the processing steps of money transaction is reduced to half and also nearly 50% of service tax can be reduced which is highly useful for the large organization to distribute the salary to their numerous employees. Since the adhaar number and fingerprint of every user is scanned initially the cyber crime can greatly avoided. To improve the security, less cost consuming and less time consuming, we have innovated the SECURED SMART ATM TRANSACTION process [2].

## 2. PIC MICROCONTROLLER BLOCK DIAGRAM

Circumstances that we find ourselves in today in the field of microcontrollers had their beginnings in the development of technology of integrated circuits. This development has made it possible to store hundreds of thousands of transistors into one chip. That was a prerequisite for production of microprocessors, and the first computers were made by adding external peripherals such as memory, input-output lines, timers and other. Further increasing of the volume of the package resulted in creation of integrated circuits. These integrated circuits contained both processor and peripherals. That is how the first chip containing a microcomputer, or what would later be known as a microcontroller came about. Memory is part of the microcontroller whose function is to store data. For a certain input we get the contents of a certain addressed memory location and that's all. Two new concepts are brought to us: addressing and memory location.

Memory consists of all memory locations, and addressing is nothing but selecting one of them. This means that we need to select the desired memory location on one hand, and on the other hand we need to wait for the contents of that location. Besides reading from a memory location, memory must also provide for writing onto it. This is done by supplying an additional line called control line. We will designate this line as R/W (read/write). Control line is used in the following way: if r/w=1, reading is done, and if opposite is true then writing is done on the memory location [3].

As we have separate lines for receiving and sending, it is possible to receive and send data (info.) at the same time [4]. So called full-duplex mode block which enables this way of communication is called a serial communication block. Unlike the parallel transmission, data moves here bit by bit, or in a series of bits what defines the term serial communication comes from. After the reception of data we need to read it from the receiving location and store it in memory as opposed to sending where the process is reversed. In order for this to work, we need to set the rules of exchange of data. These rules are called protocol. Data goes from memory through the bus to the sending location, and then to the receiving unit according to the protocol. Block Diagram of PIC microcontroller as shown in Figure 1.

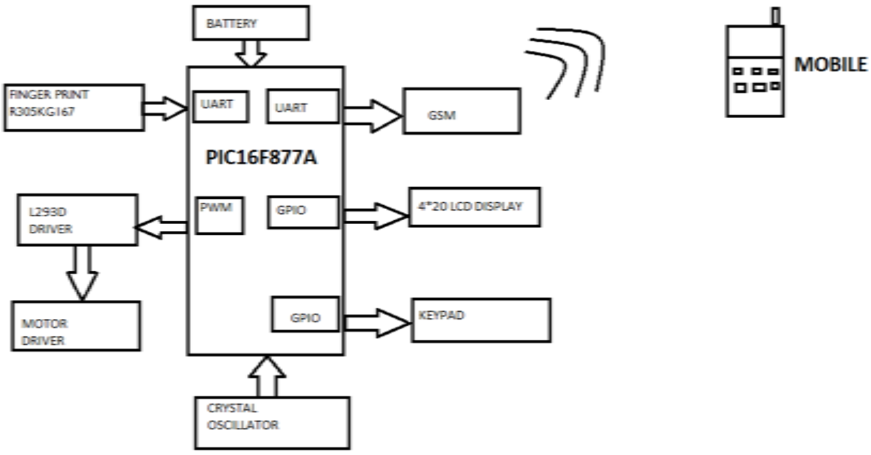


Figure 1. Block diagram of PIC microcontroller

### 2.1. Timers

The Timer0 module timer/counter has the following features:

a. 8-bit timer/counter
b. Readable and writable
c. 8-bit software programmable prescaler
d. Internal or external clock select
e. Interrupt on overflow from FFh to 00h
f. Edge select for external clock

Timer mode is selected by clearing bit T0CS (OPTION_REG<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register. Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. The prescales is mutually exclusively shared between the Timer0 module and the watchdog timer. The prescaler is not readable or writable.

### 2.1.1 Timer0 Interrupt

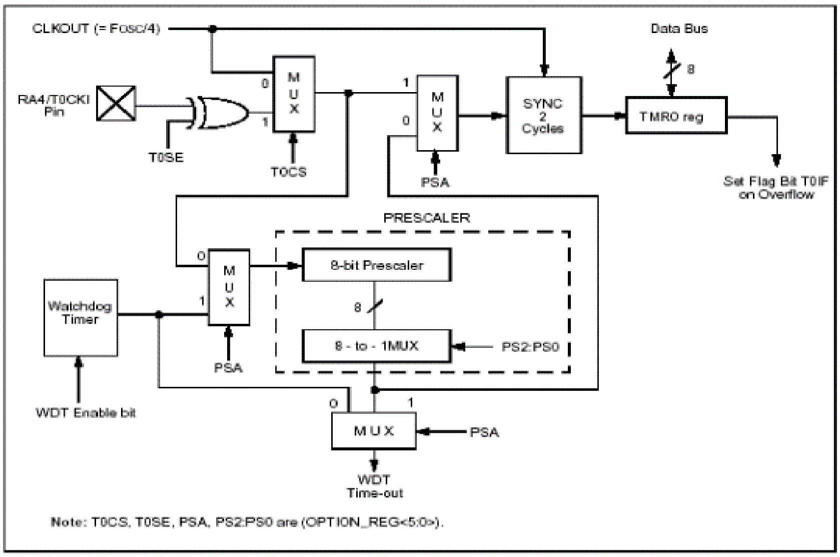
The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut off during SLEEP.

### 2.1.2 Pre-scalar

A pre-scalar assignment for the Timer0 module means that there is no prescaler for the watchdog timer, and vice-versa. The PSA and PS2:PS0 bits. (OPTION_REG<3:0>) determine the prescaler assignment and prescale ratio. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g. CLRF 1, MOVWF 1, BSF 1, x....etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

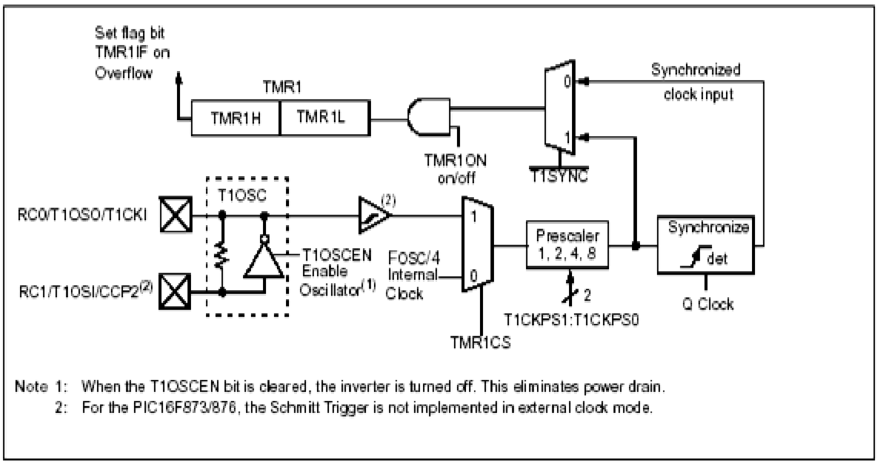
### 2.1.3 Timer1 Module

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).Timer1 can operate in one of two modes 1) As a timer, 2) As a counter. The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). In timer mode, Timer1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>). Block diagram of the TIMER0/WDT Pre-scalar as shown in Figure 2. Timer1 module as shown in Figure 3.



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

Figure 2. Block diagram of the TIMER0/WDT Pre-scalar



Note 1: When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.
2: For the PIC16F873/876, the Schmitt Trigger is not implemented in external clock mode.

Figure 3. Timer1 module

## 3. MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc [5-7]. The MSSP module can operate in one of two modes:

a. Serial Peripheral Interface (SPI)
b. Inter-Integrated Circuit (I2C)

### 3.1. SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

a. Serial Data Out (SDO)
b. Serial Data In (SDI)
c. Serial Clock (SCK)

Additionally, a fourth pin may be used when in a slave mode of operation: Slave Select (SS). To enable the serial port, MSSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON registers, and then set bit SSPEN. Figure shows the block diagram of the MSSP module when in SPI mode. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

a. SDI is automatically controlled by the SPI module
b. SDO must have TRISC<5> cleared
c. SCK (Master mode) must have TRISC<3> cleared
d. SCK (Slave mode) must have TRISC<3> set
e. SS must have TRISA<5> set

### 3.2. Master Mode

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave is to broadcast data by the software protocol. In master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>).

### 3.3. Slave Mode

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the interrupt flag bit SSPIF (PIR1<3>) is set. While in slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications. While in sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from sleep [8].

### 3.4. MSSP I2C Operation

The MSSP module in I2C mode fully implements all master and slave functions (including general call support) and provides interrupts-on-start and stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing. Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I2C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>). The SSPCON register allows control of the I2C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I2C modes to be selected:

a. I2C Slave mode (7-bit address)
b. I2C Slave mode (10-bit address)
c. I2C Master mode, clock = OSC/4 (SSPADD +1)

The SSPSTAT register gives the status of the data transfer. SSPBUF is the register to which the transfer data is written to or read from. In receive operations; the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set.

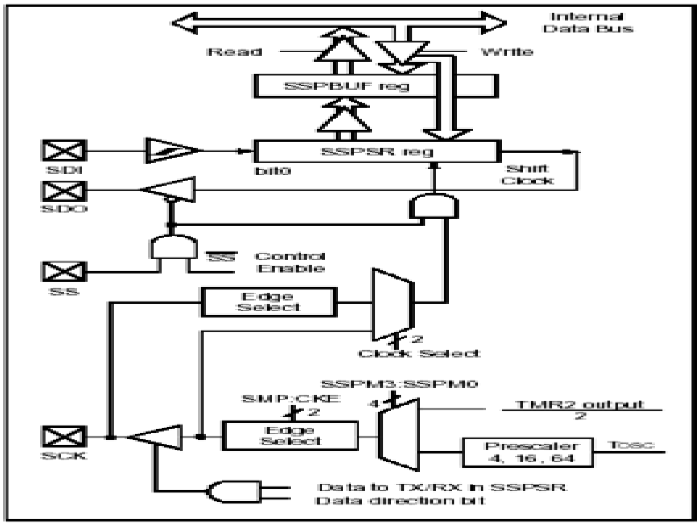


Figure 4. MSSP Block diagram

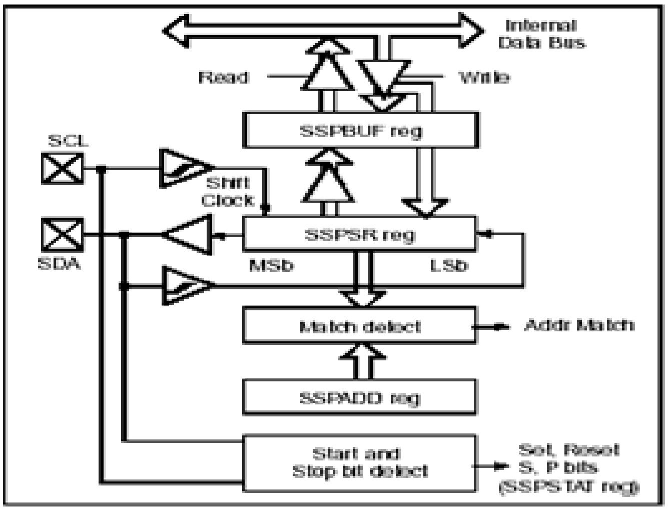


Figure 5. I2C Slave mode block diagram

### 3.5 Master Mode

Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the MSSP module is disabled. In master mode, the SCL and SDA lines are manipulated by the MSSP hardware. The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

a. START condition
b. STOP condition
c. Data transfer byte transmitted/received
d. Acknowledge transmit
e. Repeated Start

## 4. GLOBAL SYSTEM MOBILE COMMUNICATION

If you are in Europe, Asia or Japan and using a mobile phone then most probably you must be using GSM technology in your mobile phone.GSM stands for Global System for Mobile Communication and is an open, digital cellular technology used for transmitting mobile voice and data services. The GSM emerged from the idea of cell-based mobile radio systems at Bell Laboratories in the early 1970s. The GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard. The GSM standard is the most widely accepted standard and is implemented globally. The GSM is a circuit-switched system that divides each 200 kHz channel into eight 25 kHz time-slots. GSM operates in the 900 MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. The GSM is owns a market share of more than 70 percent of the world's digital cellular subscribers. The GSM makes use of narrowband Time Division Multiple Access (TDMA) technique for transmitting signals. The GSM was developed using digital technology. It has an ability to carry 64 kbps to 120 Mbps of data rates. Presently GSM support more than one billion mobile subscribers in more than 210 countries throughout of the world. The GSM provides basic to advanced voice and data services including Roaming service. Roaming is the ability to use your GSM phone number in another GSM network.

A GSM digitizes and compresses data, then sends it down through a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. Why GSM?

The GSM study group aimed to provide the followings through the GSM:

Improved spectrum efficiency.
International roaming.
Low-cost mobile sets and base stations (BSs)
Features
Specification Summary of GSM Cell Phone System
Multiple Access Technology
FDMA / TDMA
Duplex Technique FDD
Uplink frequency band 933 - 960 MHz(basic 900 MHz band only)
Downlink frequency band 890 - 915 MHz(basic 900 MHz band only)
Channel spacing 200 kHz
Modulation GMSK
Speech coding Various - Original was RPE-LTP/13
Speech channels per RF channel 8
Channel data rate 270.833 kbps
Frame duration 4.615 mS

A GSM network consists of several functional entities whose functions and interfaces are defined. The GSM network can be divided into following broad parts.

The Mobile Station(MS)
The Base Station Subsystem (BSS)
The Network Switching Subsystem (NSS)
The Operation Support Subsystem(OSS)
Following is the simple architecture diagram of GSM Network.

The added components of the GSM architecture include the functions of the databases and messaging systems:

Home Location Register (HLR)
Visitor Location Register (VLR)
Equipment Identity Register (EIR)
Authentication Center (AuC)
SMS Serving Center (SMS SC)
Gateway MSC (GMSC)
Chargeback Center (CBC)
Trans-coder and Adaptation Unit (TRAU)

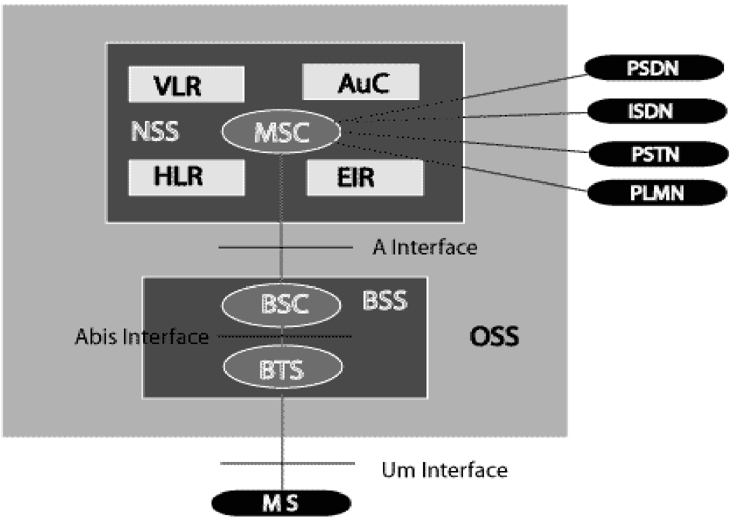


Figure 6. GSM

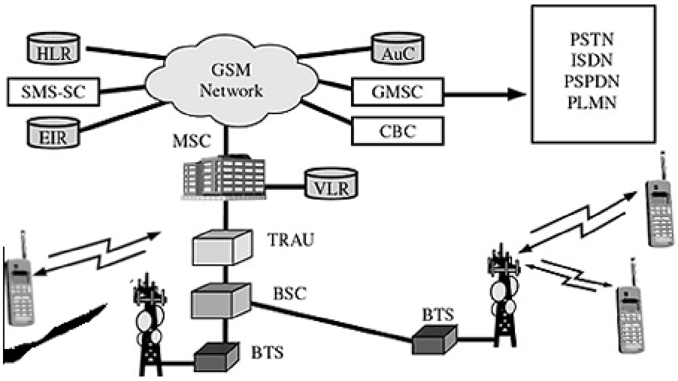


Figure 7. GSM Network along with added elements

### 4.1. GSM Network

The MS and the BSS communicate across the Um interface, also known as the air interface or radio link. The BSS communicates with the Network Service Switching center across the A interface.

#### 4.1.1. GSM Network Areas

In a GSM network, the following areas are defined

#### 4.1.2. Cell

Cell is the basic service area: one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.

#### 4.1.3. Location Area

A group of cells form a Location Area. This is the area that is paged when a subscriber gets an incoming call. Each Location Area is assigned a Location Area Identity (LAI). Each Location Area is served by one or more BSCs.

#### 4.1.4. MSC/VLR Service Area

The area covered by one MSC is called the MSC/VLR service area.

#### 4.1.5. PLMN

The area covered by one network operator is called PLMN. A PLMN can contain one or more MSCs. Specifications for different Personal Communication Services (PCS) systems vary among the different PCS networks. The GSM specification is listed below with important characteristics [9].

### 4.1.6. Modulation

Modulation is a form of change process where we change the input information into a suitable format for the transmission medium. We also changed the information by demodulating the signal at the receiving end. The GSM uses **Gaussian Minimum Shift Keying** (GMSK) modulation method.

### 4.1.7. Location Area Identity (LAI)

Each LA of an PLMN has its own identifier. The Location Area Identifier (LAI) is also structured hierarchically and internationally unique as follows: Country Code (CC) 3 decimal places. Mobile Network Code (MNC): 2 decimal places. Location Area Code (LAC): maximum 5 decimal places or, maximum twice 8 bits coded in hexadecimal (LAC < FFFF).

### 4.1.8. Temporary Mobile Subscriber Identity (TMSI)

The VLR, which is responsible for the current location of a subscriber, can assign a temporary mobile subscriber identity (TMSI) which has only local significance in the area handled by the VLR. It is stored on the network side only in the VLR and is not passed to the HLR. Together with the current location area, TMSI allows a subscriber to be identified uniquely and it can consist of upto 4x8 bits.

### 4.1.9. Local Mobile Subscriber Identity (LMSI)

The VLR can assign an additional searching key to each mobile station within its area to accelerate database access. This unique key is called the Local Mobile Subscriber Identity (LMSI). The LMSI is assigned when the mobile station registers with the VLR and is also sent to the HLR. An LIMSI consists of four octets (4x8 bits).

### 4.1.10. Cell Identifier (CI)

Within an LA, the individual cells are uniquely identified with a cell identifier (CI), maximum 2 x 8 bits. Together with the global cell identity (LAI+CI) calls are thus also internationally defined in a unique way. The operation of the GSM system can be understood by studying the sequence of events that takes place when a call is initiated from the Mobile Station.

### 4.1.11. Call from Mobile Phone to PSTN:

When a mobile subscriber makes a call to a PSTN telephone subscriber, the following sequence of events takes place:

a. The MSC/VLR receives the message of a call request.
b. The MSC/VLR checks if the mobile station is authorized to access the network. If so, the mobile station is activated. If the mobile station is not authorized, service will be denied.
c. MSC/VLR analyzes the number and initiates a call setup with the PSTN.

## 5. FINGER PRINT SENSOR

R305 Fingerprint Module is a serial fingerprint scanner which can be directly connected to the PC’s com port. R305 Fingerprint Sensor can easily be connected to any controller via MAX232 IC. This Fingerprint scanner is capable of storing and comparing the fingerprint and accordingly giving the desired output. Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching (the matching can be 1:1 or 1: N). When enrolling, user needs to enter the finger two times. The system will process the two time finger images, generate a template of the finger based on processing results and store the template. When matching, user enters the finger through optical sensor and system will generate a template of the finger and compare it with templates of the finger library. For 1:1 matching, system will compare the live finger with specific template designated in the Module; for 1: N matching, or searching, system will search the whole finger library for the matching finger. In both circumstances, system will return the matching result, success or failure.

a. Basic Power: 8-12v AC/DCInterface: RS232.
b. Matching Mode: 1:1 and 1:N
c. Baud rate: 9600 – 115200. Default: 57600.
d. Storage Capacity: 256.
e. Average Search Time: <1sec
f. Image Acquire Time: <0.5sec.

**Features**

a. Integrated image collecting and algorithm chip together, ALL-in-One
b. Fingerprint reader can conduct secondary development, can be embedded into a variety of end products
c. Low power consumption, low cost, small size, excellent performance
d. Professional optical technology, precise module manufacturing techniques
e. Good image processing capabilities, can successfully capture image up to resolution 500 dpi

**Specifications**

a. Fingerprint sensor type: Optical
b. Sensor Life: 100 million times
c. Static indicators: 15KVBacklight: bright green
d. Interface: USB1.1/UART(TTL logical level)
e. RS232 communication baud rate: 4800BPS~115200BPS changeable
f. Dimension: 55*32*21.5mm
g. Image Capture Surface 15—18(mm)
h. Verification Speed: 0.3 sec
i. Scanning Speed: 0.5 sec
j. Character file size: 256 bytes
k. Template size: 512 bytes
l. Storage capacity: 250
m. Security level: 5 (1,2,3,4,5(highest))
n. False Acceptance Rate (FAR) :0.0001%
o. False Rejection Rate (FRR): 0.1%
p. Resolution 500 DPI
q. Voltage :3.6-6.0 VDC
r. Working current: Typical 90 mA, Peak 150mA
s. Matching Method: 1: N
t. Operating Environment Temperature: -20 to 45° centigrades

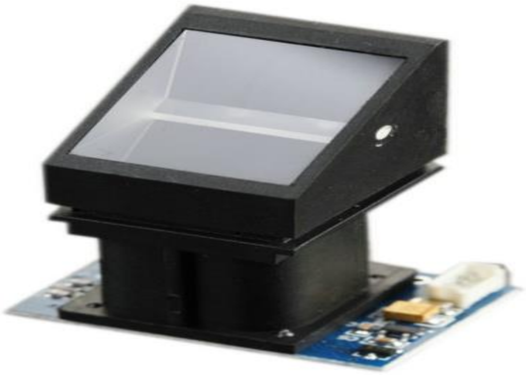
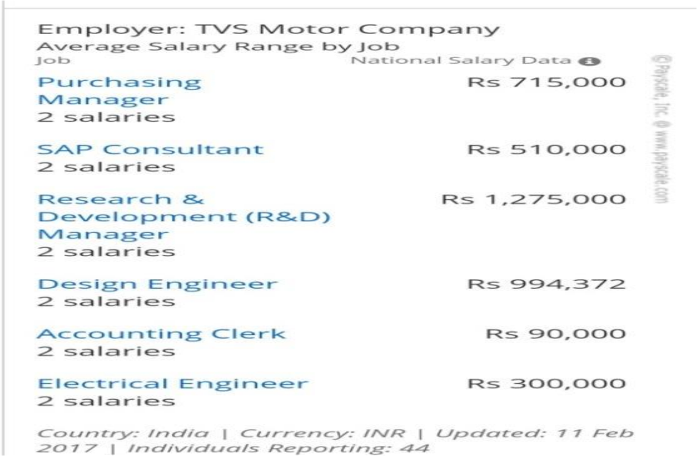


Figure 8. Finger print sensor

The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These include patterns, which are aggregate characteristics of ridges, and minutia points, which are unique features found within the patterns. It is also necessary to know the structure and properties of human skin in order to successfully employ some of the imaging technologies. Figure 9 shows salary details. Figure 10 shows hardware prototype. Figure 11 shows hardware output.



Employer: TVS Motor Company

Average Salary Range by Job

| Job | National Salary Data |
|---|---|
| Purchasing Manager<br>2 salaries | Rs 715,000 |
| SAP Consultant<br>2 salaries | Rs 510,000 |
| Research & Development (R&D) Manager<br>2 salaries | Rs 1,275,000 |
| Design Engineer<br>2 salaries | Rs 994,372 |
| Accounting Clerk<br>2 salaries | Rs 90,000 |
| Electrical Engineer<br>2 salaries | Rs 300,000 |

*Country: India | Currency: INR | Updated: 11 Feb 2017 | Individuals Reporting: 44*

Figure 9. Salary details

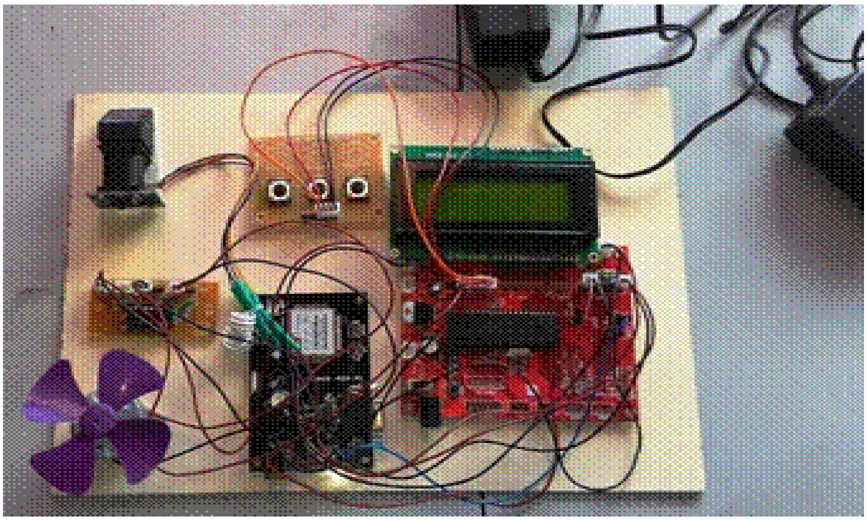
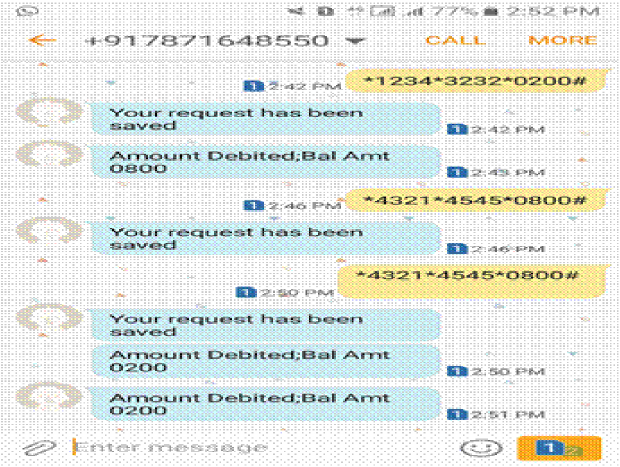


Figure 10. Hardware prototype



*1234*3232*0200# 2:42 PM

Your request has been saved 2:42 PM

Amount Debited;Bal Amt 0800 2:43 PM

*4321*4545*0800# 2:46 PM

Your request has been saved 2:46 PM

*4321*4545*0800# 2:50 PM

Your request has been saved

Amount Debited;Bal Amt 0200 2:50 PM

Amount Debited;Bal Amt 0200 2:51 PM

Figure 11. Hardware output

## 6. EMBEDDED CODING

```
#include<pic.h>
#include<string.h>

#include"fingerprint.c"
#include"gsm_driver_new.c"

bank1 struct db{
        unsigned char code[5];
        unsigned char ad_no[5];
        unsigned char f_id;
}ac_holders[]={"1234","3232",0,
               "4321","4545",1,
               "5678","2345",2,
               "8765","5214",3};

bank1 unsigned char i,msg_code[5],msg_adno[5];
bank2 unsigned char msg[]="Amount Debited;Bal Amt      ";
unsigned int bal_amt=1000,amt;
bit status=0;
void main(){
        TRISB = 0x07;           //Key
        RBPU = 0;
        RB3 = 0;                        //Key Ground
        ADCON1 = 0x06;
                TRISB4 = TRISB5 = 0;    //Motor Control
        RB4 = RB5 = 0;                  //Motor Off
        initialise_lcd_io();
        initialise_lcd();
        DelayMs(200);
        initialise_sw_uart1();          //RC0,RC1 - Tx,Rx
        initialise_uart(9600);
        enable_uart_interrupt();
        DelayMs(250);
        initialise_gsm();
        lcd_goto(1);
        print_lcd("    Cardless ATM");
        delay(1);
        while(1){
                if(!RB2){
                        DelayMs(100);
                        if(!RB2){
                                RecordFingerPrints(4);
                                lcd_goto(21);
                                print_lcd("Record Completed");
                                delay(4);
                                lcd_goto(21);
                                print_lcd("                    ");
                                while(!RB2);
                        }
                }
                if(cmd_status_flag == new_msg_id){
                        cmd_status_flag = 0;
//
//                      clear_lcd_screen();
//                      DelayMs(250);
//
//                      lcd_goto(7);
//                      ldisp_no_uchar(msgno);
```

```
    //
                read_message();

    //          lcd_goto(15);
    //          ldisp_no_uint(cnt+6);
    //          cnt = 0;

    //          lcd_goto(21);
    //          print_lcd(rec_no);
    //
    //          DelayMs(250);
    //          DelayMs(250);

                lcd_goto(61);
                print_lcd(rec_msg);
                delay(2);
                lcd_goto(61);
                print_lcd("                ");
                if(rec_msg[0] == '*'){
                    for(i=0;i<4;i++){
                        msg_code[i] = rec_msg[i+1];
                    }
                    for(i=0;i<4;i++){
                        msg_adno[i] = rec_msg[i+6];
                    }
                    amt             =             ((rec_msg[11]-48)*1000)+((rec_msg[12]-
48)*100)+((rec_msg[13]-48)*10)+((rec_msg[14]-48));

    //              lcd_goto(26);
    //              print_lcd("    ");
    //              lcd_goto(26);
    //              ldisp_no_uint(amt);

                    if(amt < bal_amt){
                        status = 1;
                        send_message(rec_no,"Your request has been saved");
                        delay(2);
                    }else if(amt > bal_amt){
                        status = 0;
                        send_message(rec_no,"Insufficient Fund");
                        delay(2);
                    }
                }else{
                    send_message(rec_no,"Invalid Format");
                    delay(2);
                }
            }

            if(!RB1 && status){
                DelayMs(100);
                if(!RB1){
                    while(!RB1);
                    if(FindMatchingFinger() == 0){
                        lcd_goto(21);
                        print_lcd("Match Found       ");

                        for(i=0;i<4;i++){
                            if(        AckPacket[11]                        ==
ac_holders[i].f_id){
```

```
if(strcmp(msg_adno,ac_holders[i].ad_no)==0){
                                lcd_goto(41);
                                print_lcd("Collect Cash");
                                lcd_goto(54);
                                ldisp_no_uint(amt);
                                bal_amt = bal_amt - amt;

                                amt = bal_amt;
                                msg[23] = (amt/1000)+'0';
                                amt = amt%1000;
                                msg[24]=(amt/100)+'0';
                                amt = amt%100;
                                msg[25]=(amt/10)+'0';
                                msg[26]=(amt%10)+'0';

                                send_message(rec_no,msg);

                                RB4 = 1;
                                RB5 = 0;
                                delay(1);
                                RB4 = RB5 = 0;
                                delay(3);
                                RB4 = 0;
                                RB5 = 1;
                                delay(1);
                                RB4 = RB5 = 0;
                                delay(1);

                                delay(4);
                                lcd_goto(41);
                                print_lcd("                ");
                                lcd_goto(21);
                                print_lcd("                ");
                            }

                        }
                    }
                }else{
                    lcd_goto(41);
                    print_lcd("No Match Found     ");

                    delay(4);
                    clear_lcd_screen();
                    lcd_goto(1);
                    print_lcd("    Cardless ATM");
                }
            }
        }
    }
}
```

## 7. COMPARISON BETWEEN NORMAL AND SMART ATM TRANSACTIONS

### 7.1. Normal Process

a. Amount delivered by the TVS organization To the Employees Rs:24,62,00,000

b. TAX paid by the organization due to this Transaction Per month Rs:3,70,000

Loss of TAX per Annum Rs:44, 31, 600

### 7.2. Smart Transaction Process:

a. TAX per Annum Rs:44,31,600
b. TAX charges are Reduced up to 50% Rs:22,15,800

## 8. CONCLUSION

This proposed work is done based on fingerprint authentication from the user by using PIC microcontroller. This process is very secured compared to other ATM transaction process. Secured smart ATM transaction is newly innovated process from the conventional ATM transaction method. Card and pin number is used as an accessing key for money transaction in the ordinary ATM. But in SECURED SMART ATM TRANSACTION both card and the pin is eliminated instead of that Fingerprint authentication and GSM signal control is used. During normal money transaction process the sender who sends the money should transact the amount of money from his account to receivers account. Then receiver need to withdraw the cash from the ATM. But in Smart ATM Transaction sender need not required to transact money to receiver instead of that the sender can just provide an account accessing permission to withdraw limited amount of money from his/her account directly.

## REFERENCES

[1] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann, Dynamic adaptive HTTP streaming of live content, in Proc. *IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM), Lucca, Italy.* 2011; 1–8.

[2] L. Shi, E. Obregon, K. W. Sung, J. Zander, and J. Bostrom, CellTV—On the benefit of TV distribution over cellular networks: A case study, *IEEE Trans. Broadcast.*, 2014; 60(1), 73–84.

[3] Y. Hu and R. Huang, Development of Monitoring System based on Raspberry Pi for Technology Rich Classroom, *Emerging Issues in Smart network, Springer,* 2015; 123–129.

[4] O. Tanaka, T. Ryu, A. Hayashida, V. G. Moshnyaga, and K. Hashimoto, A Smart wallet Design for Monitoring People with Dementia, *in Progress in Systems Engineering, Springer,* 2015; 653–659.

[5] Y. Chen and Z. Liu, Distributed Intelligent Monitoring and Control System Based on Wireless Communication chip nRF401," 2009; 278–281.

[6] PIC Micro Pascal a.k.a. PMP is a free Pascal cross compiler for PIC microcontrollers. It is intended to work with the Microchip Technology MPLAB suite http://www.ccsinfo.com/devices. For more details read Datamanhttp://dataman.com/Webpages/Programmers/Product48UXPInformation.aspx, 986-997.

[7] ManiacBug. (2013, September 29). Maniacbug GitHub. Retrieved July 15, 2014, from https://github.com/maniacbug

[8] ManiacBug. (2012, March 30). Maniacbug Explorations in Embedded. Retrieved from http://maniacbug.wordpress.com

[9] The embedded Analysis. Northern Illinois University. May 5, 1999. Retrieved December 1, 2009.Rincon, Paul (November 30, 2009). "Large Hadron Collider sets world energy record". BBC News. Retrieved November 30, 2009.