International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Programming a Microcontroller based Wireless Message Display

Duru Chinedu
Lecturer, Department of Electronic Engineering,
University of Nigeria, Nsukka
Enugu State, Nigeria.

Ochonu Regina
Graduate, Electronic Engineering
University of Nigeria, Nsukka
Enugu State, Nigeria.

Okoronkwo C. Onyinye
Graduate, Electronic Engineering
University of Nigeria, Nsukka
Enugu State, Nigeria.

***Abstract* — Communication is the activity of conveying information through the exchange of thoughts, messages, or information, as by speech, visuals, signals, written, or behavior. Communication requires a sender, a message and a recipient, although the receiver does not have to be present or aware of the sender's intent to communicate at the time of communication; thus, communication can occur across vast distances in time and space. The communication process is complete once the receiver understands the sender's message. This project is designed to develop a Wireless Message Display using PIC Microcontroller and the Embedded C Programming Language. It can be used basically to communicate information in various places and to various people. This Wireless Message Display offers flexibility and control of information to its users worldwide. The information is transmitted over GSM networks. The project involves the use of MikroC IDE, Proteus simulation software, Embedded C language and AT Commands. SMS are sent via a mobile Phone from any location to a SIM card placed in a GSM Modem. The Modem receives the SMS and sends it serially to the USART module of the microcontroller through a DB9 serial cable. Depending on the programming of the microcontroller, a valid SMS is displayed on an LCD screen and all invalid SMS are ignored.**

Keywords: GSM Modem, PIC Microcontroller, LCD, UART, AT Commands, Embedded C, MikroC.

I. INTRODUCTION

Different technologies have continued to shape how we communicate and share information. From the development of wireless hand-held devices to development of robots, software engineering continues to play a vital role in achieving automations in systems that require minimal downtime and remote control or update. While a large number of programming languages may be available to choose from in implementing such systems, it is important to critically evaluate the underlying architectures of these choice components and leverage on suitable development environment that can ensure optimal speed and minimal resource usage. Low level languages such as embedded C and other procedural languages remain top choices when developing systems where optimal use of resources is desired.

In our choice of architecture of leveraging on PIC Microcontroller (16F877A) an Electrically Erasable Programmable Memory microcontroller to achieve automation in the remote update and display of information on an electronic noticeboard. Code implementations provided in this work, show a logical attempt towards usage of instructions set of the PIC microcontroller for controlling flow of data from the serial terminals of the Universal Asynchronous Receiver-Transmitter (UART); a GSM module that provides data to be display on the electronic noticeboard with other commands. Such messages received from the UART terminal are however controlled over secure lines to prevent unauthorized usage.

II. LITERATURE REVIEW

A. Microcontroller

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. They are designed for embedded applications and they are usually used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes.

Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. Typically, microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assemblers are used respectively to convert high-level language and assembler language codes into a compact machine code for storage in the microcontroller's memory. For this project, the MikroC compiler is used and the codes are written in C Programming Language. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory or program memory that may be field-alterable flash or erasable read-only memory. Manufacturers

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

have often produced special versions of their microcontrollers in order to help the hardware and software development of the target system.

Microcontrollers usually contain from several to dozens of general purpose input/output pins (GPIO). GPIO pins are software configurable to either an input or an output state. When GPIO pins are configured to an input state, they are often used to read sensors or external signals. Configured to the output state, GPIO pins can drive external devices such as LEDs or motors. Also, they contain several modules such as; ADC (Analogue to Digital Converter), USART, Timers, etc. used for different purposes. [1][2]

PIC Microcontrollers: PIC is a family of modified Harvard architecture microcontrollers made by Microchip Technology. [3] PICs are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

B. Serial Comunication

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels

Many serial communication systems were originally designed to transfer *data over relatively large distances through some sort of data cable.*

Serial data communication uses two methods; synchronous and asynchronous.

1. Synchronous method transfers a block of data at a time. Data can be transmitted both ways at a time (full Duplex).
2. Asynchronous method transfers a single byte at a time. Data can be transmitted one way at a time (half Duplex).

There are special IC chips made by many manufactures for serial communications, they include;

- USART (Universal Synchronous Asynchronous Receiver-Transmitter)
- UART (Universal Asynchronous Receiver-Transmitter). Which is the protocol being used in this project as will be explained later. [4]

C. UART

UART is an integrated circuit used for serial communications that contains a receiver (serial-to-parallel converter) and a transmitter (parallel-to-serial converter), each clocked separately. The parallel side of a UART is often connected to the bus of a computer. When the computer writes a byte to the transmit data register of a UART, the UART will start to transmit it on the serial line.

UART are often included in microcontrollers. Several modern integrated circuits now come with a UART that can communicate synchronously. Such a device is called a USART (universal synchronous / asynchronous receiver / transmitter). [5]

D. GSM MODEM

GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone.

When a GSM modem is connected to a computer (in this case, a Microcontroller), it allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages. A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, [6]

E. MikroC IDE

MikroC is one of the powerful and easy to use software for programming microcontrollers in embedded C. It is the best solution for developing code for microcontrollers. It is designed to provide the user with the easiest possible solution for developing applications for embedded systems, without compromising performance or control.

Applications can be developed quickly and easily using mikroC because it provides a simple Windows OS-based point-and-click environment for developing applications, highly advanced IDE, powerful compiler with advanced SSA optimizations, lots of hardware and software libraries, comprehensive documentation, and plenty of ready-to-run examples. [33]

F. Embedded C Programming Language

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions, etc. [7]

Advantages

- It is small and reasonably simpler to learn, understand, program and debug.
- Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms.
- C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.

- As C combines functionality of assembly language and features of high level languages, C is treated as a 'middle-level computer language' or 'high level assembly language'
- It is fairly efficient
- It supports access to I/O and provides ease of management of large embedded projects.
- Java also used in many embedded systems but Java programs require Java Virtual Machine (JVM), which consume lot of resources. Hence it is not used for smaller embedded devices. [7]

G. PROTEUS LITE

Proteus is software for microprocessor simulation, schematic capture, and printed circuit board (PCB) design. It is developed by Lab Center Electronics.

System components

- ISIS Schematic Capture - a tool for entering designs.
- PROSPICE Mixed mode SPICE simulation - industry standard SPICE3F5 simulator combined with a digital simulator.
- ARES PCB Layout - PCB design system with automatic component placer, rip-up and retry auto-router and interactive design rule checking.
- VSM (Virtual System Modeling) - lets co-simulate embedded software for popular microcontrollers alongside hardware design.
- System Benefits Integrated package with common user interface and fully context sensitive help. [7]

H. AT COMMANDS:

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems are also supported by GSM/GPRS modems and mobile phones. Besides this common AT command set, GSM/GPRS modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

Note that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS. However, some books and web sites use them interchangeably as the name of an AT command.

AT commands used for the configuration and setup of the modem are shown in table 1 below;

S/NO	AT COMMANDS USED	FUNCTIONS
1	AT + CMGF = 1	Select SMS message format in text mode
2	AT + CMGS	Send SMS message
3	AT + CNMI = 2,2,0,0,0	New SMS message indications
4	AT + IPR = 9600	Baud rate configuration of GSM modem

Table 1: list of AT commands used in the system and their functions

III. SYSTEM DESIGN

A. PROGRAM FLOWCHART KEY:

VALIDITY1: when the first two characters of an SMS is '##' and its last character is '*'

VALIDITY2: when the first two characters of an SMS is '###' and its last character is '*'

N: represents the locations in EEPROM of microcontroller. Increments by 80 since we used a 20x4 LCD, an SMS can have only a maximum of 80 characters (including spacing between words).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

B. DECISION TABLE

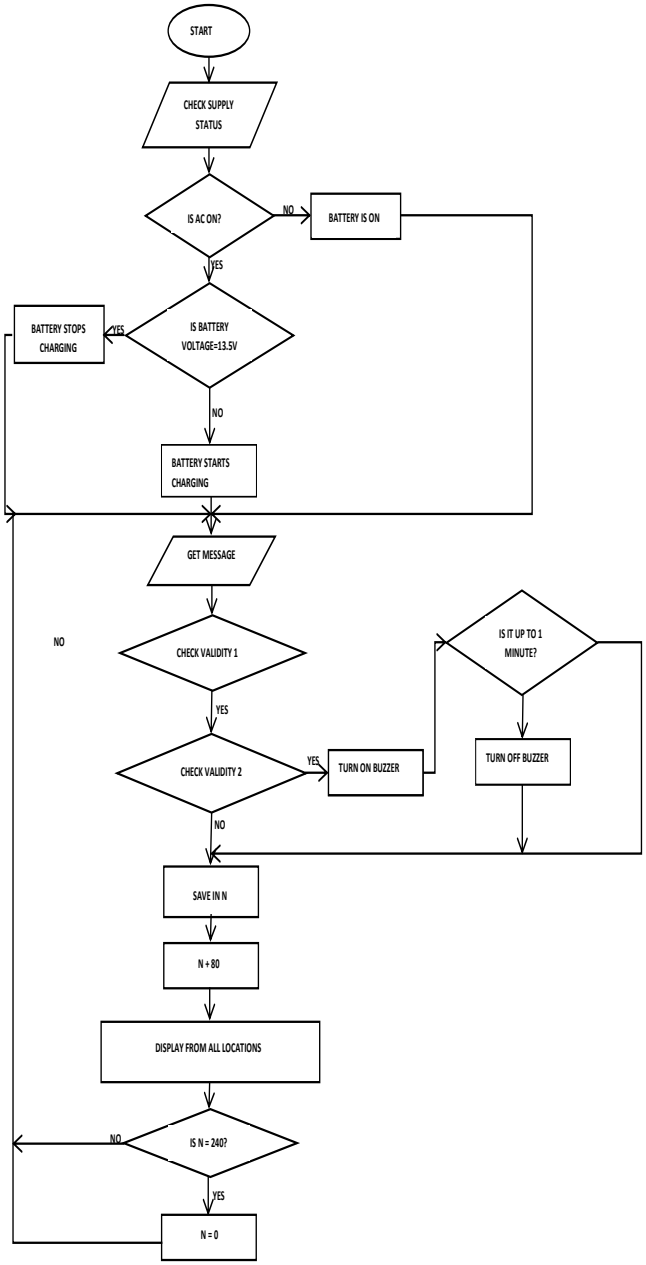

Figure1: Program Flowchart

S/N	CONDITION STUB	CONDITION ENTRY				
1	MSG Received begins with "##" & ends with "*"	N	N	–	Y	Y
2	MSG Received begins with "###" & ends with "*"	Y	N	Y	Y	Y
3	MSG Received <=3	N	Y	Y	Y	Y
4	AC relay at N/O (normally open) mode	Y	N	N	N	N
5	Battery voltage ==3V	N	Y	N	N	N
	ACTION STUB	ACTION ENTRY				
1	Replace the first saved MSG			X		X
2	Save & display MSG		X	X	X	X
3	Buzzer ON				X	X
4	Buzzer OFF		X	X		
5	AC supply ON	X	X	X		
6	Battery supply ON				X	X
7	Battery is charging				X	X
8	Battery stops charging	X	X	X		

Table 2: Decision table of the conditions system and the adequate action of the system

List of symbols used in the decision table, their meaning and their function;
Y[yes] used if the condition is satisfied.
N[no] used if the condition is not satisfied.
– [hyphen] used if the condition is not relevant to the rule
X denotes the required action to take in response to a given condition

C. PERFORMANCE ANALYSIS OF DIFFERENT NETWORK PROVIDERS

An analysis of four different GSM network providers in Nigeria, namely; MTN, GLO, ETISALAT, and AIRTEL were carried out as follows;
Over a period of eight hours per day for four consecutive days, one subscriber number on each of the networks mentioned above is dialed or called. This was also done on three other days (on UNN Convocation day, on a UNN post UTME day and on a rainy day). The results obtained is recorded which could be one of the following;

- Available (A), i.e. the call successfully connected
- Not Available (NA)
- Network Busy (NtB)
- Number Busy (NB)
- Switched Off (SO)
- Does Not Exist (DNE)

For the six days;

- ✓ Day 1 = Saturday
- ✓ Day 2 = Sunday
- ✓ Day 3 = Monday
- ✓ Day 4 = Tuesday
- ✓ Day 5 = A rainy day
- ✓ Day 6 = Post UTME day
- ✓ Day 7 = Convocation day

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

From the data obtained, we generated a table consisting of availability and time for all the days the analysis was carried out for each of the network providers, as seen below;

TIME (IN DAYS)	AVAILABILITY (NO. OF TIMES AVAILABLE PER DAY)			
	MTN	GLO	AIRTEL	ETISALAT
DAY1(SATURDAY)	7	8	3	6
DAY2(SUNDAY)	7	1	3	2
DAY3(MONDAY)	7	3	1	5
DAY4(TUESDAY)	6	2	1	5
DAY5(RAINY DAY)	2	4	4	2
DAY6(POST UTME)	3	4	3	3
DAY7 (CONVOC.)	2	5	2	5

Table 3: Network parameter analysis each day for the four network providers

Thus, plotting the graph of AVAILABILITY against TIME (in days), using table 3, we obtained the following curves;

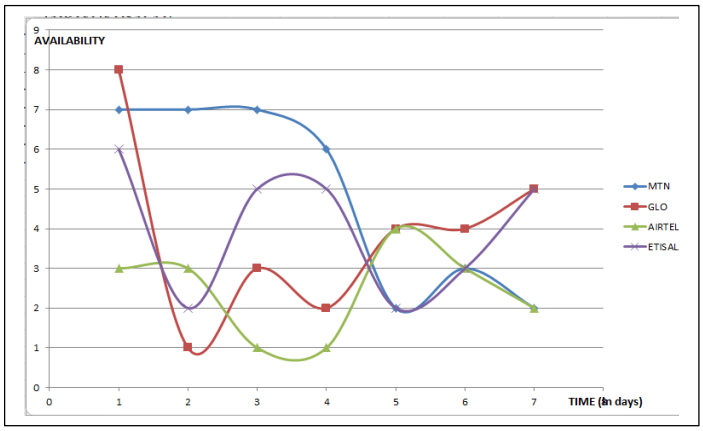

Figure2: Graph of Availability against Time

INFERENCES FROM THE GRAPH

As shown in the graph of figure 2;

- MTN is more available for most of the days than the other networks providers, followed by ETISALAT, GLO and AIRTEL, in that order.
- AIRTEL numbers were mostly switched off compared to the other networks. Reason is because, most subscribers use the AIRTEL network basically for browsing the internet (data communication) due to cheap data plans offered by the network. Hence, only switched on when there is a need to surf the internet.
- The MTN network experience more traffic congestion on a rainy day, convocation day and post UTME day, this is possibly because most GSM users use their MTN numbers as their primary phone numbers. Hence, when several calls (traffic) are generated by subscribers at a given period of time, congestion sets, hence more calls are lost.

Also, the MTN network has a wider coverage range than the other networks.

Hence, from the inferences made above, MTN SIM card was used in our system for the GSM MODEM.

IV. SYSTEM IMPLIMENTATION

The steps taken in the Implementation process are;

1 Circuit development and design [8]
2 Writing and developing basic code for message display using MikroC IDE
3 Circuit simulation on Proteus Lite
4 GSM MODEM testing and configuration using a PC
5 Programming the Microcontroller with the basic codes written in step 2 above.
6 Code simulation on Proteus Lite.
7 Adding an alerting unit to the circuit [8]
8 Writing code to control the beeping of the buzzer in the alerting unit and repeat step 6.
9 Incorporating a 12v rechargeable battery (a backup power source) and repeat step 3
10 Developing a charging circuit for the battery and repeat step 6
11 Writing code to control charging of battery and then repeat step 6
12 Simulating circuit with the new code
13 Reprogramming the microcontroller
14 Final testing of circuit

A. CODE DEVELOPMENT AND CIRCUIT SIMULATION

The MikroC IDE was used to develop (write and compile) the embedded C code used to program the microcontroller. The code was to implement the major functionality of the system, which is to receive, save and display valid SMS sent to the system from any mobile phone.

The designed circuit and the compiled code were then simulated on Proteus and all necessary corrections were made.

B. GSM MODEM TESTING AND CONFIGURATION USING A PC

The GSM modem was connected to a personal computer using a USB to serial converter; this allows the PC to use the GSM modem to interconnect over the mobile network. The USART terminal of the MikroC library is the software application environment that facilitated this testing, configuration and set-ups.

How to test and configure GSM modem using USART terminal on MikroC library

1) Using USB to Serial converter, connect the GSM modem to the PC with the driver installed on the PC.
2) On the MikroC IDE development environment, click on tools and select USART Terminal
3) Select the COM port number assigned to the MODEM (as seen in "ports" slot of the PC's device manager) and baud rate. Also select the data format and the new line settings.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

4) On the send text box, type 'AT'. The modem should replay with 'OK' which is seen at the receive text box.
5) If replay is received then type then configure modem with the AT Commands discussed in section 3.1.2, otherwise exit and go back to step 2.

C. PROGRAMMING THE MICROCONTROLLER

EEPROM Programmer was used to program the microcontroller taking the following steps;

- Connect the programmer to the PC
- Plug in the microcontroller to the programmer and check if it has been detected
- If detected, erase previous content of the microcontroller
- Import code from MikroC
- Write code to microcontroller
- Check to ensure that the microcontroller is not blank
- Exit and unplug microcontroller from the programmer

D. EMBEDDED C CODE

Below are the codes burnt into the microcontroller to achieve a wireless message display.

```
// variable declarations
unsigned int i, read1, p1, READ, row, n;
double analog_voltage;
double digital_voltage;
unsigned char col = 1;
unsigned char disp, track = 0;
volatile char RecdFlag, rec, recd, data_segment, k1, w1, k, j, t, l;
volatile char ComingFlag, BUZZER, BUZ;
volatile char Counter;
volatile char tagRX[80];
// LCD module connections
sbit LCD_RS at RD4_bit;
sbit LCD_EN at RD5_bit;
sbit LCD_D4 at RD0_bit;
sbit LCD_D5 at RD1_bit;
sbit LCD_D6 at RD2_bit;
sbit LCD_D7 at RD3_bit;
sbit LCD_RS_Direction at TRISD4_bit;
sbit LCD_EN_Direction at TRISD5_bit;
sbit LCD_D4_Direction at TRISD0_bit;
sbit LCD_D5_Direction at TRISD1_bit;
sbit LCD_D6_Direction at TRISD2_bit;
sbit LCD_D7_Direction at TRISD3_bit;
// End LCD module connections

// Monitoring when new message is received in the GSM Modem
void interrupt() {
  unsigned char rxByte;
  rxByte = UART1_Read();  // Read character received from USART
  if (ComingFlag == 0 && rxByte == 0x23)  // 0X23 is #
  {
    ComingFlag = 2;
    Counter = 0;
    rxByte = 0;
    track = 1;
  } else if (ComingFlag == 1) {
    if (rxByte == 0x2A) {  // star
      ComingFlag = 0;
      RecdFlag = 1;
      rxByte = 0x20;  // 0x20 = EMPTY SPACE
      track = 0;
    }
    if (Counter < 80) {
      tagRX[Counter] = rxByte;
    }
    Counter++;
  } else {
    Counter = 0;
  }

  if (track == 1 && rxByte == 0x23)  // second #
  {
    Counter = 0;
    Track = 2;
    rxByte = 0;
    ComingFlag = 1;
    BUZZER = 0;
  }
  if (track = = 2 && rxByte = = 0x23)  // third #
  {
    Counter = 0;
    track = 0;
    rxByte = 0;
    ComingFlag = 1;
    BUZZER = 1;
  }
  if (BUZZER == 1 && ComingFlag != 0) {
    BUZ = 1;
  }
   RCIF_bit = 0;  //  Clear  UART  receive interrupt flag
}

// Monitoring Battery voltage levels
void monitor_battery() {  //  function  to monitor battery voltage level
  digital_voltage = ADC_Read(1);
  analog_voltage = (5 * digital_voltage) / 1023;
  if (analog_voltage >= 3) {
    RB5_bit = 1;
  } else {
    RB5_bit = 0;
  }
}

// Saving New Message received on EEPROMof the Microcontroller
void memory_write() {  // function to save new messages received to EEPROM
  t = 0;
  for (j = k; j < k + 80; j++) {
    rec = tagRX[t];
    EEPROM_Write(j, rec);
    monitor_battery();
    t++;
  }
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

```
  data_segment++;
  if (data_segment == 4) {
    data_segment = 1;
  }
  EEPROM_Write(0xff, data_segment);
}

        // MAIN FUNCTION
        void main() {
            TRISB = 0;  //  make  portb output
           PORTB = 0;  // clear all bits of portb
            TRISA = 0x02;  //  make  AN1 input
           PORTA = 0;  // clear all bits in porta
                       UART1_Init(9600);  // initialiases uart1
            Delay_ms(500);  //  wait  for UART module to stabilize.
          ADC_Init();
           RCIE_bit = 1;  // PIE1(bit 5) =1, to enable d UART rciv interrupt
          GIE_bit = 1;  // INTCON (bit 7) =1, to enable all unmasked interrupt
          PEIE_bit =
               1;  // INTCON (bit 6) =1, to enable all unmasked peripheral interrupt
          Lcd_Init();  // Initialize LCD
           Lcd_Cmd(_LCD_CLEAR);         // Clear display
             Lcd_Cmd(_LCD_CURSOR_OFF);  // Cursor off
          monitor_battery();
          RecdFlag = 0;
          ComingFlag = 0;
           Counter = 0;  // Counts bytes. It is d index of d tagRx array
          BUZZER = 0;
          BUZ = 0;
              lcd_out(1, 1, "READY      TO RECEIVE");
          for (i = 0; i < 80; i++) {
              tagRX[i] = 0x20;  // =epmty space. To clear initial contents of the array
          }
          read1 =
               EEPROM_Read(0xff);  // read what is at address 0xff=location 255 of EEPROM
          if (read1 > 3) {
                EEPROM_WRITE(0XFF, 1);  // write 1 to address 255
            for (l = 0; l < 255; l++) {
                EEPROM_WRITE(l, 0x20);  // clear address 0 to 254 (<255)
              monitor_battery();
            }
          }
          while (1) {  // Endless loop
            if (BUZ == 1) {
              RB2_bit = 1;
              n++;
              if (n == 5) {
                RB2_bit = 0;
                BUZ = 0;
                n = 0;
              }
            }
            row = 1;
            col = 1;
            k1 = 0;  // assignments
            for (i = 0; i < 3; i++) {
               for (p1 = k1; p1 < k1 + 80; p1++) {
                READ = EEPROM_Read(p1);
                lcd_chr(row, col, READ);
                col++;
                if (col == 21) {
                  row++;
                  col = 1;
                  if (row == 5) {
                    row = 1;
                  }
                }
              }
              k1 = k1 + 80;
              delay_ms(500);
              monitor_battery();
    Lcd_Cmd(_LCD_CLEAR);  // Clear display
              monitor_battery();
            }
            if (RecdFlag == 1) {
               data_segment = EEPROM_Read(0xff);
              if (data_segment == 1) {
                k = 0;
                memory_write();
               } else if (data_segment == 2) {
                k = 80;
                memory_write();
               } else if (data_segment == 3) {
                k = 160;
                memory_write();
              } else {
                k = 0;
              }
              monitor_battery();
              ComingFlag = 0;
              Counter = 0;
              RecdFlag = 0;
              BUZZER = 0;
            }
          }
        }
```

V. CONCLUSION

In this work, we have attempted to provide a sample implementation of the code that drives a microcontroller based wireless message display unit. A notification message is wirelessly sent from a Mobile phone as SMS (Text Message) and received by the system which is then displayed on an LCD screen. The reliability, scalability and portability of Embedded C programming language between different platforms has made it easy for the objective of this work to be attained.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

REFERENCES

[1] Augarten, Stan (1983). "The Most Widely Used Computer on a Chip: The TMS 1000". State of the Art: A Photographic History of the Integrated Circuit (New Haven and New York: Ticknor & Fields). ISBN 0-89919-195-9. Retrieved 2009-12-23.

[2] "Atmel's Self-Programming Flash Microcontrollers". 2012-01-24. Retrieved 2014-5-25. By Odd Jostein Svendsli 2003.

[3] http://ww1.microchip.com/downloads/en/DeviceDoc/39630C.pdf

[4] Mazidi, M.A. "The 8051 Microcontroller and Embedded Systems". 2nd Edition, page 362-364.

[5] http://www.futureelectronics.com/en/Signal-Interface/uart.aspx

[6] http://www.nowsms.com/faq/what-is-a-gsm-modem

[7] http://www.engineersgarage.com/tutorials/emebedded-c-language

[8] Duru Chinedu, Ochonu Regina, Okoronkwo Onyinye, "DESIGN AND IMPLEMENTATION OF A WIRELESS NOTICE BOARD WITH INTERFACE FOR REMOTE UPDATE", *International Journal of Scientific and Engineering Research* May 2017; 8(5).