International Journal of Scientific & Engineering Research, Volume ŀ Ŀ ,
ISSN 2229-5518


# TWO WAY WIRELESS DATA MESSAGING SYSTEM IN RURAL AREAS USING ZIGBEE

A.NAVANEETH REDDY[1], D.L.V.S.SRIKANTH[1], T.MANOJ NAGA KUMAR[1], S.SAI ROHITH[2]

Abstract— In this paper, the data can be transmitted to and received from remote ZigBee communication device. Data Security is primary concern for every communication system. There are many ways to provide security data that is being communicated. However, what if the security is assured irrespective of the hackers are from the noise. This Project describes a design of effective security for data communication by designing standard algorithm for encryption and decryption. Zigbee is a PAN technology based on the IEEE 802.15.4 standard. Unlike Bluetooth or wireless USB devices, ZigBee devices have the ability to form a mesh network between nodes. Meshing is a type of daisy chaining from one device to another. This technique allows the short range of an individual node to be expanded and multiplied, covering a much
Larger area. The source information is generated by a keypad and this will be encrypted and is sent to destination through Zigbee modules. The receiving system will check the data and decrypt according to a specific algorithm and displays on the LCD. This project uses regulated 5V, 500mA power supply. 7805 three terminal voltage regulator is used for voltage regulation. Bridge type full wave rectifier is used to rectify the ac out put of secondary of 230/12V step down transformer.

Index Terms— Sensors, Micro Radio, Geographical Packet Routing, Embedded Systems, Keil, Radio Frequency Control Robot, Network.



## 1 INTRODUCTION

Embedded systems are electronic devices that incorporate microprocessors with in their implementations. The main purposes of the microprocessors are to simplify the system design and provide flexibility. Having a microprocessor in the device means that removing the bugs, making modifications, or adding new features are only matters of rewriting the software that controls the device [1]. Or in other words embedded computer systems are electronic systems that include a microcomputer to perform a specific dedicated application. The computer is hidden inside these products. Embedded systems are ubiquitous. Every week millions of tiny computer chips come pouring out of factories finding their way into our everyday products.

Embedded systems are self-contained programs that are embedded within a piece of hardware. Whereas a regular computer has many different applications and software that can be applied to various tasks, embedded systems are usually set to a specific task that cannot be altered without physically manipulating the circuitry. Another way to think of an embeddedsystem is as a computer system that is created with optimal efficiency, thereby allowing it to complete specific functions as quickly as possible [2].

[1] A.NAVANEETH REDDY *is currently pursuing batchelors degree program in* electronics and communication engineering *in* Laki reddy Bali reddy college of engineering, India, *PH-09703309470.*
*E-mail:* navaneethredy35@gmail.com
[2] D.L.V.S.SRIKANTH *is currently pursuing batchelors degree program in* electronics and communication engineering *in* Laki reddy Bali reddy college of engineering, India.
[3] T.MANOJ NAGA KUMAR *is currently pursuing batchelors degree program in* electronics and communication engineering *in* Laki reddy Bali reddy college of engineering, India,
[4} S.SAI ROHITH *is currently pursuing batchelors degree program in* electronics and electrical engineering *in* V.R.Siddhartha College of engineering, India

Embedded systems technologies are usually fairly expensive due to the necessary development time and built in efficiencies, but they are also highly valued in specific industries. Smaller businesses may wish to hire a consultant to determine what sort of embedded systems will add value to your organization.

PLATFORM*:*

There are many different CPU architectures used in embedded designs. This in contrast to the desktop computer market, which as of this writing (2003) is limited to just a few competing architectures, mainly the Intel / AMD x86, and the Apple / Motorola / IBM PowerPC, used in the Apple Macintosh.

One common configuration for embedded systems is the *system on a chip*, an application-specific integrated circuit, for which the CPU was purchased as intellectual property to add to the IC's design [3].

TOOLS*:*

Like a typical computer programmer, embedded sys-

International Journal of Scientific & Engineering Research Volume ƙǾ(ǾǾÌ ǾǾ , ÈÙÊÏ ı l ƳƙK
ISSN 2229-5518

tem designers use compilers, assemblers and debuggers to develop embedded system. Those software tools can come from several sources. Software companies that specialize in the embedded market Ported from the GNU software development tools. Sometimes, development tools for a personal computer can be used if the embedded processor is a close relative to a common PC processor. Embedded system designers also use a few software tools rarely used by typical computer programmers [4]. Some designers keep a utility program to turn data files into code, so that they can include any kind of data in a program. Most designers also have utility programs to add a checksum or CRC to a program, so it can check.

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the Industry standard 80C51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller, which provides a highly flexible and cost-effective solution to many, embedded control applications [5].

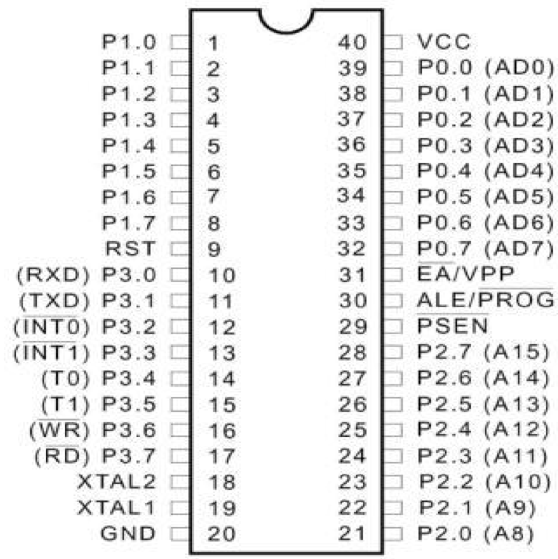


FIGURE: 1(A). MICROCONTROLLER

FIGURE: 1(B). PIN CONFIGURATION OF PORT 1

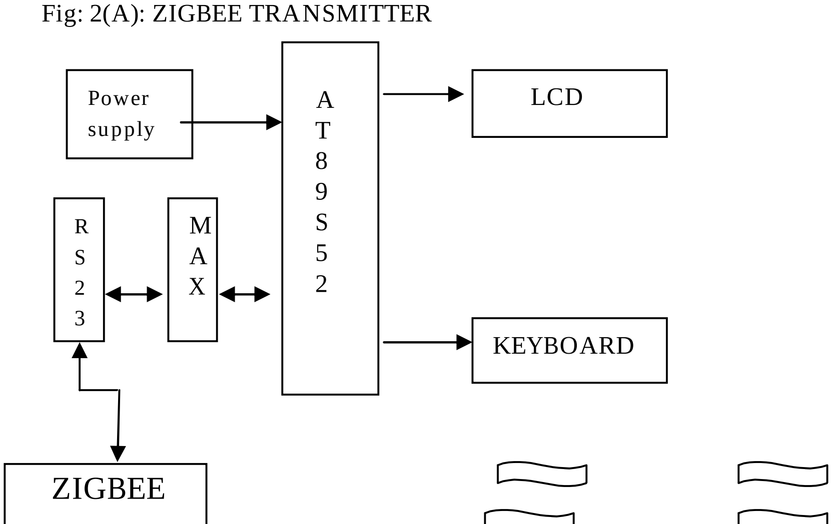
| Port Pin | Alternate Functions |
|---|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.5 | MOSI (used for In-System Programming) |
| P1.6 | MISO (used for In-System Programming) |
| P1.7 | SCK (used for In-System Programming) |

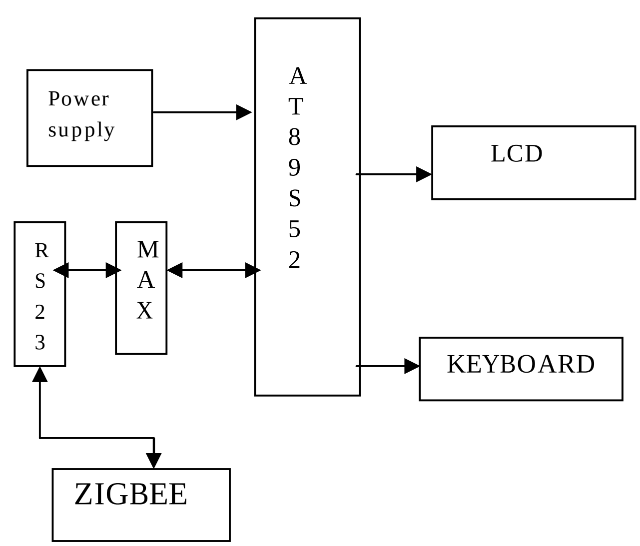
FIGURE:1(C).PIN CONFIGURATION OF PORT 2

| Port Pin | Alternate Functions |
|---|---|
| P3.0 | RXD (serial input port) |
| P3.1 | TXD (serial output port) |
| P3.2 | $\overline{INT0}$ (external interrupt 0) |
| P3.3 | $\overline{INT1}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | $\overline{WR}$ (external data memory write strobe) |
| P3.7 | $\overline{RD}$ (external data memory read strobe) |

## 2.**BLOCK DIAGRAM:**

Fig: 2(A): ZIGBEE TRANSMITTER



International Journal of Scientific & Engineering Research Volume 5, Issue 1, January-2014
ISSN 2229-5518

## 5. 8052 Microcontroller Memory Organization:

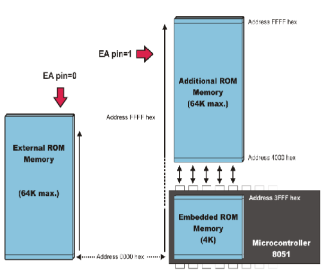
**1.Data Memory:**

As already mentioned, Data Memory is used for temporarily storing and keeping data and intermediate results created and used during microcontroller's operating. Besides, this microcontroller family includes many other registers such as: hardware counters and timers, input/output ports, serial data buffers etc. The previous versions have the total memory size of 256 locations, while for later models this number is incremented by additional 128 available registers. In both cases, these first 256 memory locations (addresses 0-FFh) are the base of the memory. Common to all types of the 8052 microcontrollers. Locations available to the user occupy memory space with addresses from 0 to 7Fh. First 128 registers and this part of RAM is divided in several block [6]**.** The first block consists of 4 banks each including 8 registers designated as R0 to R7. Prior to access them, a bank containing that register must be selected. Next memory block (in the range of 20h to 2Fh) is bit- addressable, which means that each bit being there has its own address from 0 to 7Fh. Since there are 16 such registers, this block contains in total of 128 bits with separate addresses (The 0th bit of the 20h byte has the bit address 0 and the 7th bit of the 2Fh byte has the bit address 7Fh). The third groups of registers occupy addresses 2Fh-7Fh (in total of 80 locations) and do not have any special purpose or feature [7].

**2.Program Memory:**

The oldest models of the 8052-microcontroller families did not have any internal program memory. It was added from outside as a separate chip. These models are recognizable by their label beginning with 803 (for ex. 8031 or 8032). All later models have a few Kbytes ROM embedded, Even though it is enough for writing most of the programs, there are situations when additional memory is necessary [8].

The microcontroller handle external memory depends on the pin EA logic state:




**EA=0** in this case, internal program memory is completely ignored; only a program stored in external memory is to be executed.

**EA=1**In this case, a program from built-in ROM is to be executed first (to the last location). Afterwards, reading additional memory continues the execution [9].

In both cases, P0 and P2 are not available to the user because they are used for data and address transmission. Besides, the pins ALE and PSEN are used too.

While Vcc and Vss provide +5V and ground respectively, Vee is used for controlling LCD contrast. The ASCII code to be displayed is eight bits long and is sent to the LCD either four or eight bits at a time.

| PIN | SYMBOL | I/O | DESCRIPTION |
|---|---|---|---|
| 1 | Vss | -- | Ground |
| 2 | Vcc | -- | +5V power supply |
| 3 | Vee | -- | Power supply to control contrast |
| 4 | RS | I | RS=0 to select command register<br>RS=1 to select data register |
| 5 | R/W | I | R/W=0 for write<br>R/W=1 for read |
| 6 | EN | I/O | Enable |
| 7 | DB0 | I/O | The 8-bit data bus |
| 8 | DB1 | I/O | The 8-bit data bus |
| 9 | DB2 | I/O | The 8-bit data bus |
| 10 | DB3 | I/O | The 8-bit data bus |
| 11 | DB4 | I/O | The 8-bit data bus |
| 12 | DB5 | I/O | The 8-bit data bus |
| 13 | DB6 | I/O | The 8-bit data bus |
| 14 | DB7 | I/O | The 8-bit data bus |

If four-bit mode is used, two "nibbles" of data (Sent high four bits and then low four bits with an "E" Clock pulse with each nibble) are sent to make up a full eight-bit transfer [10].

International Journal of Scientific & Engineering Research Volume ƱȮ(ƛƛƜ̍ƼȮ , ƜƜƝƐƛ
ISSN 2229-5518

## 3.PROGRAM:

```
/*==========================*/
/*------------bt robo programming------------*/
/*------PH:89C51RD+,28/01/11-----------------*/
/*============================*/
LCD_DPort EQU P0
LCD_Rs EQU P1.5
LCD_Rw EQU P1.6
LCD_En EQU P1.7 DEELAY1 DATA 30h DELAY2 DATA 31h DELAY3 DATA 32h VStack DATA 61h FDispRFID Bit 01h
/*---------------------------------*/
org 00hjmp Power_on org 23h
      ljmp serial_ISR
/*-------------------------------------*/
org 100h Power_on: mov R0,#00h clr A
      LclearNxtRAM:mov @R0,a inc R0
      cjne R0,#0FFh,LclearNxtRAMmov R0,#00h acall LCD_Init mov TMOD,#20h mov TH1,#0FDh
      mov SCON,#50hmov IE,#90h
      setb TR1
      Call Btinitacall welcomedata call delay1sec
      mov P2,#00h
      MOV R1,#00H
      /*-------------------------------------*/ MainLoop:
      jb FDispRFID,RFIDLoop
      call
      delay500ms
      /*--------------------------------------*/
      RFIDLoop:
      clr FDispRFID
      cjne R1,#31h,backlp
      SETB P2.7 call lcdclear call displine1mov dptr,#dev1on call DISPDATA
      call delay1sec;Call Btinitacall welcomedata jmp MainLoop backlp:
      cjne R1,#32h,leftlpCLR P2.7 call lcdclear call displine1mov dptr,#dev1of call DISPDATA call delay1sec;
      Call Btinitacall welcomedata jmp MainLoop
      leftlp:cjne R1,#33h,rightlpSETB P2.6 call lcdclear call displine1mov dptr,#dev2on call DISPDATA call delay1sec;
      Call Btinitacall welcomedata jmp MainLoop rightlp:cjne R1,#34h,stop CLR P2.6
      call lcdclear
      call displine1mov dptr,#dev2of call DISPDATA call delay1sec;
      Call Btinitacall welcomedata jmp MainLoop stop:
jmp MainLoop
/****************************************/
Btinit:mov dptr,#mycmnd0
call serialsend
call delay1sec
mov dptr,#mycmnd1
call serialsend
call delay1sec
movdptr,#mycmnd2
call serialsend
call delay1sec
ret
mycmnd0:DB"**********ANDROID BASED **",0Dh,0Ah,0
mycmnd1:DB"*********CONTROL KEYS: DEVICE1OFF*********",0Dh,0Ah,0
mycmnd2:DB"*********CONTROL KEYS: DEVICE2OFF*********",0Dh,0Ah,0
/****************************************
/
serialsend:
looop:
clr a movc a,@a+dptr jz exitmov SBUF,a call delay20ms inc Dptrjmp looop exit:
ret
/*---------------------------------------*/
serial_ISR:jb RI,RX_Service jb TI,TX_Service reti
RX_Service:
clr RImov a,sbuf mov R1,a
setb FDispRFIDljmp End_isr
TX_Service: clr TI
ljmp End_isrEnd_isr: Ret

/****************************************
/
/*-----welcome data programming---------*/
/****************************************
/welcomedata:
call lcdclear
call displine1mov dptr,#mydata
call DISPDATA
call displine2mov dptr,#mydata1 call DISPDATA
retmydata : DB" ANDROID BASED ",0
mydata1: DB" DEVICE CONTROL ",0
dev1on: DB" ****DEVICE1ON**** ",0
dev1of: DB" ****DEVICE1OFF*** ",0
dev2on:
DB"****DEVICE2ON***,0dev2of: DB" ****DEVICE2OFF*** ",0/*---LCD initialization program-----------*/Init:mov a,#30hcall LCD_CMND_OUT
mov a,#38hcall LCD_CMND_OUT
mov a,#06hcall LCD_CMND_OUT
```

International Journal of Scientific & Engineering Research Volume ƀǿǾ(ÚÚÜÌǐĦǿǾ , ÈÙÊÏıǀƔƵƘ
ISSN 2229-5518

```
mov a,#0Ch
call LCD_CMND_OUT
mov a,#01hcall LCD_CMND_OUT
RET/***********************************
********//*-------lcd command program-
ming-----------
*//*************************************
*****/LCD_CMND_OUT:call
lcd_busymov LCD_DPort,a clr
LCD_Rsclr LCD_Rw setb LCD_En
nopnopclr LCD_En
RET/***********************************
*****//*-------lcd data programming-------
----
*//*************************************
**/ LCD_DATA_OUT:call lcd_busymov
LCD_DPort,a setb LCD_Rsclr LCD_Rw
setb LCD_En
Nop
Nop
clr LCD_En
RET/***********************************
*****//*----------busy check program-
ming------
*//*************************************
**/lcd_busy:mov
LCD_DPort,#0ffhCLR LCD_Rs SETB
LCD_Rw AGAIN1:CLR LCD_EN
NOP
NOPSETB LCD_EN
JB p0.7,AGAIN1
RET
/***************************************/
/*-----Display the string&send---------*/
/***************************************/
DISPDATA:
Next_Char:
clr a
movc a,@a+dptr jz
End_Str
call LCD_DATA_OUT
inc dptr
jmp Next_charEnd_Str: RET
end
```

```
/***************************************/
/*---------display routine-------------*/
/***************************************/
lcdclear:
mov a,#01h
call LCD_CMND_OUTret displine1:
mov a,#80h
call LCD_CMND_OUTret displine2:
mov a,#0c0h call LCD_CMND_OUTret
displine1A: mov a,#8Ah
call LCD_CMND_OUT
ret
/*-------Delay routine-------------------*/
delay1sec:
mov DELAY1,#10wait2: mov DE-
LAY2,#200 wait1: mov DELAY3,#250
wait: djnz DELAY3,wait djnz DE-
LAY2,wait1djnz DELAY1,wait2 ret
delay500ms:
mov DELAY1,#5
wait22:
mov DELAY2,#183 wait12:
mov DELAY3,#250 wait0:
djnz
DELAY3,
wait0
djnz
DELAY2,
wait12
djnz
DELAY1,
wait22ret
delay20ms:
wait32: mov DELAY2,#40
wait31: mov DELAY3,#250 wait3:
djnz
DELAY3,wait31
Ret
Delay:
Mov DELAY1,same
Ret
end
```

## 4. Conclusion:

The project "TWO WAY WIRELESSDATA MESSAGING SYSTEM FOR RURAL AREAS USING ZIGBEE." is successfully tested and implemented. This project helps to track the solar energy to get a maximum efficiency. Solartracking system is a device for orienting a solar panel or concentrating a solar reflector or lens towards the sun. Thus we track a maximum solar energy. In solar cell applications, require a high degree of accuracy to ensure that the concentrated sunlight is directed precisely to the powered device. Precise tracking of the sun is achieved through systems with single tracking.

## 5. References:

[1] Ayala, Kenneth J. (1996), The 8051 Microcontroller-Architecture, Programming and Applications, Delmar Publishers, Inc. India Reprint Pernam.

[2] www.wikipedia.org

[3] Raj Kamal (2004), Embedded Systems.Architecture, Programming andDesign, International Edition, New Delhi: McGraw-Hill.

[4] Barr, Micheal (1999), Programming Embedded systems in C and C++, Sebastopol, C.A: O Reilly.

[5] www.8051projects.com

International Journal of Scientific & Engineering Research Volume ƙǾ(ÚÚÜÌƗǾ , ÈÙÊÏıl ƔK
ISSN 2229-5518

[6] Calcutt, M.C.F.J Cowan, and G.H. Parchizadeh (1998), atmega32 Micro controllers-Hardware and Software Applications, Arnold (and also John Wiley).
[7] www.embedded-computing .com
[8] Mazidi, M. Ali and J. G. Mazidi (2000), The 8051 Microcontroller and Embedded Systems, Pearson Education.
[9] www.e-insite.net/edmag/
[10] www.mcjournal.com