

WIRELESS PERSONAL AREA NETWORK NODE DESIGN AND SIMULATION OF ALCOHOL SENSOR USING ZIGBEE TRANSCIEVER MODULE

Rajesh Singh¹

¹HEAD, Robotics Institute R&D, Assistant Professor,
University of petroleum and Energy studies, Dehradun, INDIA

Akanksha²

Student EE branch 3rd year
University of petroleum and Energy studies, Dehradun, INDIA

Shashank Mishra³

Student EE branch 3rd year
University of petroleum and Energy studies, Dehradun, INDIA

Ankit Joshi⁴

Student EE branch 3rd year
University of petroleum and Energy studies, Dehradun, INDIA

Abstract-- In a current scenario, beverage industry is one of the fastest growing industry governing Indian as well as the global market. Naturally, the revenues generated by these industries are of great importance to the economy. Thus detecting leakage in this beverage industry becomes very important, as the leakage may cause fire in the industry which may be hazardous to life and property. To overcome that problem we can introduce a wireless personal area network that is capable to detect the alcohol leakage in industry with the help of a hardware node and send it wirelessly. On the other hand the information send by the transmitter module can be directly received on a computer which is capable to receive this information and send reverse message to the customer. This process is not only beneficial to the industry but also prevents threats to life and property. Further, this can be used to analyze the breath of drivers and can reduce the accidents due to drunk driving. It can also be used to check the concentration of alcohol in blood and hence it can be used in hospitals as well. This is a cost effective yet an efficient system which can be implemented in all mentioned sector which can be economically beneficial and it being eco friendly

it doesn't have an adverse impact on the environment..

Index Terms— AVR Microcontroller, Wireless personal area Network (WPAN), ZigBee module.

I. INTRODUCTION

This wireless network comprises of autonomous devices using sensors to effectively monitor physical or environmental parameters which play a important role in different industrial locations. These may include various parameters such as temperature, sound, vibration, pressure, motion or pollutants,

at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor network cover various industrial and civilian application in different areas, including industrial process monitoring, control and automation, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

A sensor node is made of a certain number of pods. Each pod usually contains

- one or more sensor leading to one or more data channel,
- a processing unit such as a micro-controller or microprocessor,
- a two-way communication component such as a radio and antenna (radio ranges are typically limited by government spectrum requirements; unlicensed bands will allow for communication of a few hundred yards in unobstructed areas, although line of site is not a requirement),
- an energy source such as a battery coupled with a solar cell,
- a package to protect components against sometimes harsh environment,

Here we have designed a network with its hardware design.

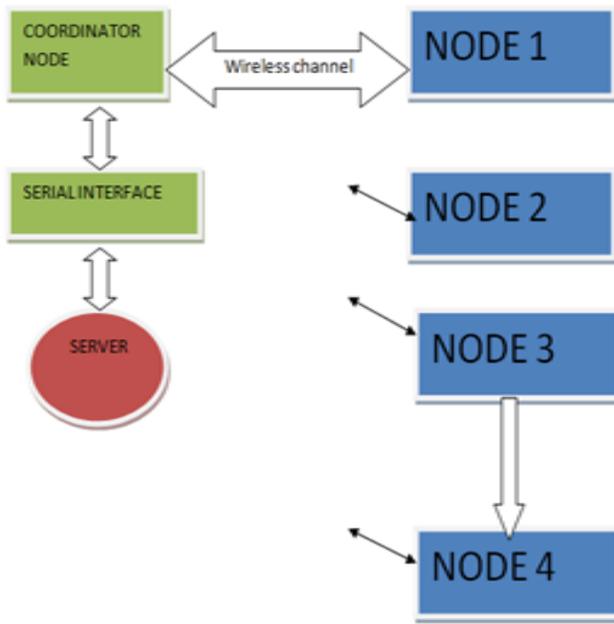


Fig (1): Block Diagram Representation of LPG monitoring system

The above figure shows the block Diagram representation of alcohol monitoring system. Here the coordinator node has to receive the data which is transmitted by any mentioned node which may be from any site in the beverage industry. Then data is fed with the help of serial interface into the PC and information logged on through HyperTerminal. If the level of alcohol is greater than a predefined safe or acceptable value and then message will be transmitted reverse through coordinator node and again receives by the corresponding nodes and it will be displayed back to the various device such as(LCD) or a computer itself.

II. HARDWARE DEVELOPMENT

Hardware of this particular wireless area network basically constitute of two parts: Transmitting or Slave End node design and receiving or coordinator node design.

(a) *Slave end node design*-Components of the slave system is given below

(1) *Power supply modules*-This module is basically designed to achieved 5V,500mA.This consists of a transformer,IN4007 diodes, capacitor 1000uF,7805 regulator,330 ohm resistance, LED as indicator.

(2) *Embedded microcontroller*-There is a whole wide range of microcontroller available in the market. But this particular project is developed using AVR series of microcontroller (ATMEGA16) because of its inbuilt ADC port and its variable frequency. ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz, allowing the system designed to optimize power

consumption versus processing speed. Further it also minimizes the cost of this personal area network.

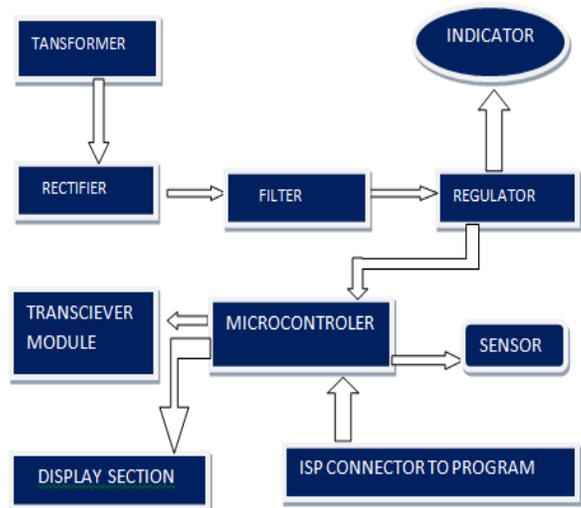


Fig (2.0): Block diagram of the transmitting module

(3) *Zigbee module*- This is a single chip low power radio frequency transceiver module, which can facilitate the OEM designers to design their remote control applications in remote control in the quickest way. These modules are based on IC CC2500 by Texas Instrument. It works at frequency of 2.4GHz. The main operating parameters and the 64-byte transmit/receive FIFOs of CC2500 can be controlled with the help of an SPI interface. In a typical system, the CC2500 can be used together with a microcontroller and some passive components.

(4) *Alcohol gas sensor (analog output)* - This alcohol sensor is suitable for detecting alcohol concentration in breath. It has a high sensitivity and fast response time. Sensor provides an analog output based on alcohol concentration. It can also be used as a blood alcohol concentration checker. It has a current consumption of less than or at the maximum of 180mA.

(5) *Display module*- The LCD(liquid crystal display) unit receives character codes (8 bits per character) from a microprocessor or microcomputer, latches the codes to its display data RAM (80-byte) DD RAM for storing 80 characters, transforms each character code into a 5 ´ 7 dot-matrix character pattern, and displays the characters on its LCD screen.

Finally, the hardware model of transmitting end is simulated using Proteus software (trial). Data which is sensed by the alcohol sensor will be processed by the controller and displayed through LCD and corresponding data also send using TX (15) pin of microcontroller. The running data on TX line is displayed using Virtual terminal in Proteus software. Various indicators like Multimeters, LCD data Display, variable data meter and virtual data terminal are shown in figure (2.1).

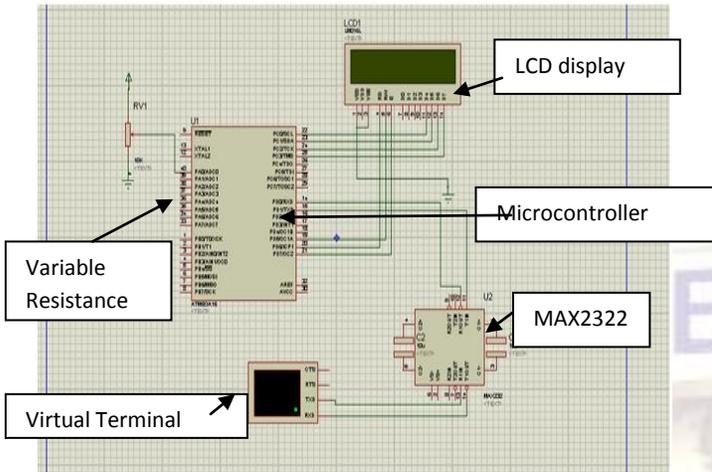


Fig (2.1) Simulation model of transmitting using Proteus trial software

(b) Coordinator node design -The hardware of this node can be developed using following components:

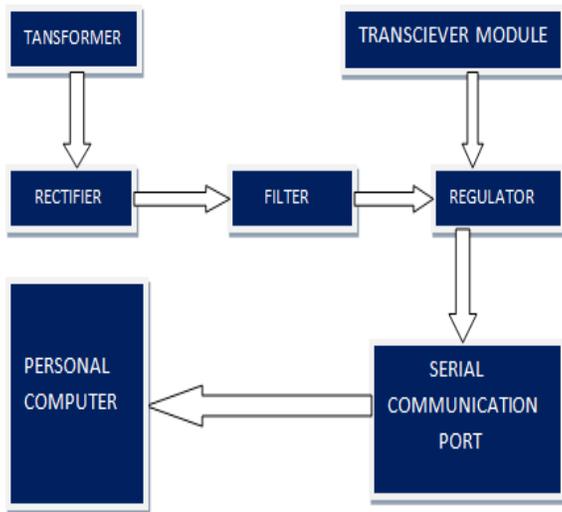


Fig (2.2) :Block diagram of Data collecting section (Receiving Section)

Block diagram of the transmitting module

- (1) Power supply unit-It is same as describe described above.
- (2) Zigbee module-Same as receiving module used in Node1 to Node 5 design.

(3) MAX 232(level converter)-MAX232 is a dual driver/receiver IC that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply [2]. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. This can be made to work with the help of a few capacitors attached to it.

(4) USB to serial cable-To interfaces the coordinator node with the other nodes.

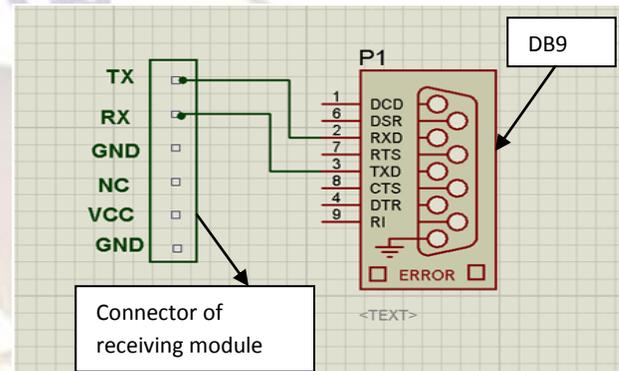


Fig (2.3) : Simulation model of receiving end using Proteus trial software

III. SOFTWARE DEVELOPMENT

Microcontroller, when it is used to operate as a wireless network involves following steps:

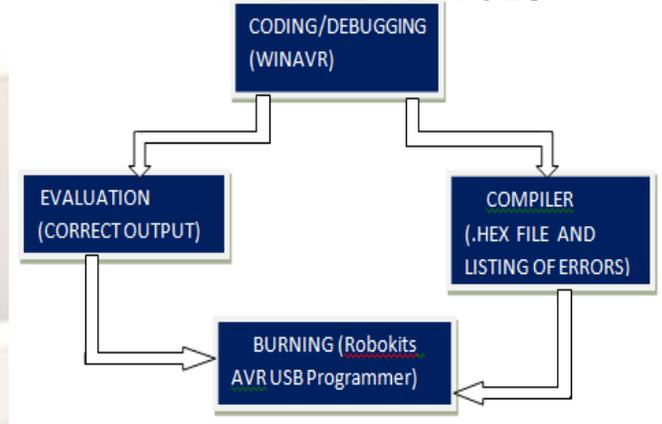


Fig (3.1): Steps for software development

(1) Coding / Debugging- Coding or debugging is one in a high-level language (such as c or java). Compiler for a high level language helps to reduce production time. To program the microcontrollers WinAVR [2] was used using C language. The source code has been commented to facilitate any occasional future improvement and maintenance. WinAVR is a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors hosted on the Windows platform. It includes the GNU GCC compiler for C and C++. WinAVR contains all the tools for developing on

the AVR. This includes AVR-gcc (compiler), AVR-gdb (debugger) etc.

(2) *Compiling*- After compiling the program, it is converted to machine level language in the form of o's and ans1's. This file is called as the Hex file and is saved with the extension (.Hex). The compiler also generates errors in the program which should be removed for proper execution of the program.

(3) *Burning*- Burning the machine language (hex) file into the microcontroller's program memory is achieved with a dedicated programmer, which attaches to a PC's peripheral. PC's serial port has been used for the purpose. For this purpose Ponyprog programmer was used to burn the machine language file into the microcontroller's program memory. Ponyprog is serial device programmer software with a user-friendly GUI framework available for Windows 95/98/ME/NT/2000/XP and Intel Linux. Its purpose is reading and writing every serial device. It supports I²C Bus, Micro wire, SPI EEPROM, and the Atmel AVR and Microchip PIC microcontroller. The microcontrollers were programmed in approximately two seconds with a high speed-programming mode. The program memory, which is of Flash type, has, just like the EEPROM, a limited lifespan. On AVR microcontroller family it may be reprogrammed up to a thousand times without any risk of data corruption. Atmega16 Programmer (ISP) which is used to burn the program into AVR microcontrollers.

(4) *Evaluation*- If the system performs as desired by the user and performs all the tasks efficiently and effectively the software development phase is over and the project is ready to be installed in any of the industrial sites as a personal area network. If not, the entire process is repeated again to rectify the errors. One of the difficulties of programming microcontrollers is the limited amount of resources the programmer has to deal with. In PCs resources such as RAM and processing speed are basically limitless when compared to microcontrollers. In contrast to a PC, the code on microcontrollers should be as low on resources as possible, but being cost effective and power efficient makes it a better option.

In the programming of the proposed system is used the following .c and .h file

(1) *lcd.c* - This c file contains the code for control of functionality of the attached LCD module. The code controls the initialization of the LCD, data writing on the LCD, and also the movement, characteristics and location of the cursor. It offers the facility to write data on the LCD character-by-character or string-wise. The command set used in the software is based on the command set used in the LCD based on Hitachi HD44780 ICs. This file contains InitLCD (), LCDClear (), LCDWriteString () and LCDWriteInit ().

(2) *lcd.h*- This header file contains all the constant variable values and names of the subroutines used by various files used in the software. It clearly indicates which variable can be used as a global variable and which of the subroutines can be used across the software files.

(c) *Adc.c*- This file contains the code for controlling the ADC of ATMEGA'S. This contains two major functions InitADC(), uint16_t ReadADC(uint8_t ch)

Initialization of ADC:

```
InitADC()
{
    ADMUX=(1<<REFS0);// For Aref=AVcc;
    ADCSRA=(1<<ADEN)|(7<<ADPS0);
}
```

Read data from ADC:

```
uint16_t ReadADC(uint8_t ch)
{
    //Select ADC Channel ch must be 0-7
    ADMUX&=~(1<<0);
    ADMUX&=~(1<<1);
    ADMUX&=~(1<<2);
    ADMUX&=~(1<<3);
    ADMUX&=~(1<<4);
    ch=ch&0b00011111;
    ADMUX|=ch;
    ADCSRA|=(1<<ADSC); //Start Single conversion
    while(!(ADCSRA & (1<<ADIF))); //Wait for
    conversion to complete
```

```
    ADCSRA|=(1<<ADIF); //Clear ADIF by writing one
    to it
    return(ADC);
}
```

(d) *Usart_lib.c* - This file contains the code for controlling the USART of ATMEGA'S. This contains three major functions USARTInit (), USARTReadChar () and USARTWriteChar ().

Initialization of USART:

This function will initialize the USART.

```
void USARTInit(uint16_t ubrr_value)
{
    UBRR= ubrr_value; //Set Baud rate
    UCSRC=(1<<URSEL)|(3<<UCSZ0); // Set Frame Format
    UCSRB=(1<<RXEN)|(1<<TXEN); //Enable The receiver
    and transmitter
}
```

Reading From The USART :

This function will read data from the USART.

```
char USARTReadChar()
{
    while(!(UCSRA & (1<<RXC))) //Wait until a data is
    available
    {
        //Do nothing
    }
    return UDR; //Now USART has got data from host and is
    available is buffer
}
```

Writing to USART:

```
void USARTWriteChar(char data)
{
while(!(UCSRA & (1<<UDRE))) //Wait until the transmitter
is ready
{
//Do nothing
}
UDR=data; //Now write the data to USART buffer
}
```

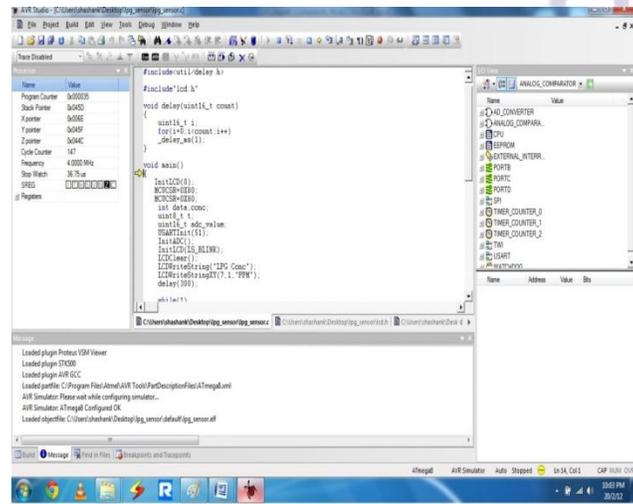


Fig (3.2): snapshot C coding for proposed system using of AVRstudio4



Fig (3.3): view of hardware AVR programmer (ISP) by Robokits

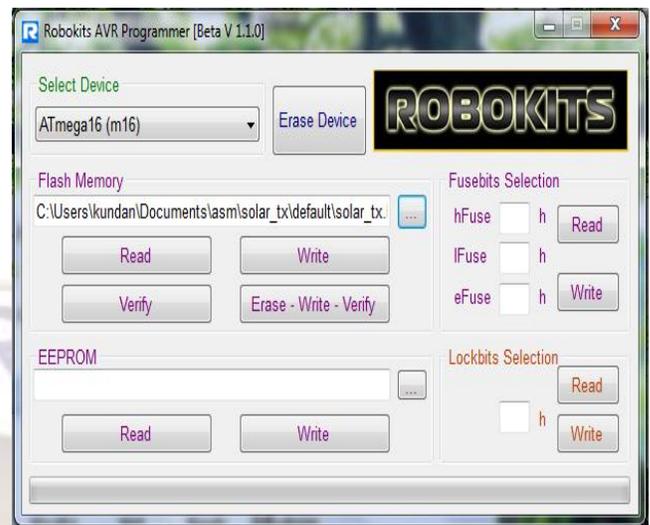


Fig (3.4): snapshot of robokits AVR programmer window to erase, write and verify .h file generated by AVRstudio4 to target

View of system

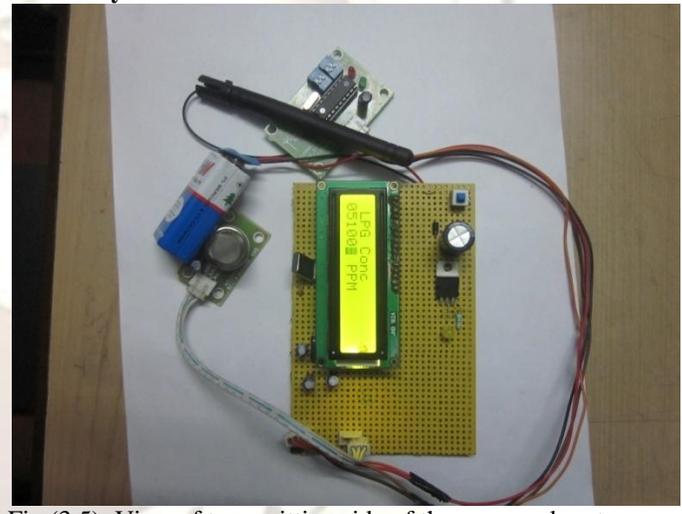


Fig (3.5): View of transmitting side of the proposed system.



Fig (3.6): View of receiving side of the proposed system

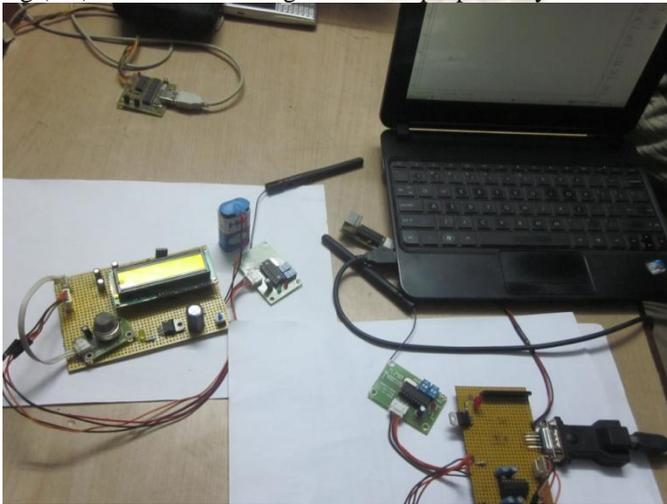


Fig (3.7): Complete View of proposed system in functioning mode

IV. CONCLUSION

In this paper, a personal area wireless network is designed using various hardware and software techniques, using simple and cost effective electronic components. This project has a wide range of application in beverage industries. Traffic police can also use this system stop drunk driving. As this is a personal area network it can also be used outside discos and park to bann the entrance of people under the influence of alcohol to stop eve teasing and various social tribulations. Being cost effective it is easy to install it anywhere. Further optimization techniques can be used to improve the features and efficiency of the arrangement and make it further more advanced.

REFERENCE

1. Singh, R. Mishra, S. Joshi, P. "Pressure Monitoring in wireless sensor network using Zigbee™ transceiver module" ICCCT – 2011 ISBN: 978-1-4577-1385-9
2. Singh, R.; Mishra, S. Temperature monitoring in wireless sensor network using Zigbee™ transceiver module" ICPCES – 2010 ISBN: 978-1-4244-8543-7
3. Young Wung Kim; Sang Jin Lee; Guk Hee Kim; Gi Joon Jeon; , "Wireless electronic nose network for real-time gas monitoring system," *Robotic and Sensors Environments, 2009. ROSE 2009. IEEE International Workshop on* , vol., no., pp.169-172, 6-7 Nov. 2009
4. Zhao Xiaoqiang; Zhang Zuhou; , "Development of Remote Waste Gas Monitor System," *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on* , vol.1, no., pp.1105-1108, 13-14 March 2010
5. Guo Xiu-cai; Zhao Jing; Zhang Xin-liang; , "The Research of Remote-Monitoring System of the Gas Based on Web," *E-Business and E-Government (ICEE), 2010 International Conference on* , vol., no., pp.2583-2586, 7-9 May 2010
6. Hua Fu; Tao Wang; Cui Yang; , "Gas monitoring system based on the multi-sensor information fusion," *Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference on* , vol., no., pp.2-930-2-933, 16-19 Aug. 2009
7. Kumar, A.; Singh, I.P.; Sud, S.K.; , "Indoor environment gas monitoring system based on the digital signal processor," *Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT '09. International*, vol., no., pp.245-249, 14-16 March 2009
8. Xianyi Qian; Xianjun Wang; , "Design of Temperature Humidity and Toxic Gas Monitoring System of Underground Garage," *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on* , vol., no., pp.1-3, 23-24 May 2009
9. Wenlian Li; Chuanqing Liu; Yang Li; Fang Xiao; , "The Design and Implementation of Monitoring System for H2S Gas Volume Fraction with Virtual Instrument," *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on* vol.2, no., pp.557-560, 19-20 Dec. 2008
10. Lei Chen; Shuang Yang; Ying Xi; , "Based on ZigBee wireless sensor network the monitoring system design for chemical production process toxic and harmful gas," *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on* , vol.4, no., pp.425-428, 24-26 Aug. 2010