http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (9): 1662-1667, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.1662.1667

A Real-Time Electric-Power Monitoring and Wireless Transmission System Based on the Internet of Things

[1,2]Wei Wei, [1]Xiang Li, [3]Bin Zhou, [1]Hongfang Zhou, [4]Peiyi Shen, [1]Lei Wang and [1]Jing Zhang
[1]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]Shannxi Key Laboratory For Network Computing and Security Technology,
[3]College of Science, Xi'an University of Science and Technology, Xi'an 710054, China
[4]National School of Software, Xidian University, Xi'an 710071, People's Republic of China

Abstract: Wireless Radio Frequency (RF) technology, which can take the place of cable, has been widely used in many current fields as one of information technologies with widest horizon of development. The problems caused by the environment restrictions and the unfavorable conditions for cable wiring have been solved because of the emergence of the wireless. Its three important characteristics and advantages are low costing, low power consumption and peer-to-peer communication. During the design procedure, the system is divided into three major modules according to the functional structure, namely central control module, wireless transceiver module and computer dealing software, The central control module uses Sygnal company's low power enhanced 51 MCU C8051F310 as CPU, mainly to complete the data acquisition and processing. The wireless transceiver module is made up of TI CC1101 wireless transceiver chip, which mainly realizes wireless data transmission and The computer dealing software real-timely displays and deals with the data through visual, friendly programming interface, based on C#.NET. Final achievement of the system design is two compact communication devices, one of them connects with PC through RS-232 interface, System is operated through the graphical user interface based on windows, which is very convenient to operate. Debugging result indicated that data acquisition, processing and transmission could be achieved in wireless or wired way under the control of PC and it visually displays the results to the user, satisfies request of the wireless power monitoring data communication system, reaches the technical index.

Key words: Wireless transmission, serial communication, C8051F310, CC1101, C#.NET

INTRODUCTION

The communication frequency bands of ISM (Industrial Scientific Medical) are opened for the agencies of industry, science, medicine, no authorization is needed. Using them within a certain transmission power (generally less than 1 w) and no interference to other frequency bands are permitted.

The ISM frequency bands of 315, 433 and 868 MHz are commonly used in domestic. And the chips of these frequency bands are mainly principally for wireless data transmitting and receiving. Most of the existing applications in the wireless data acquisition and wireless monitoring area are running on these frequency bands, as well as the WSN (wireless sensor network) (Krzysztofik, 2009).

The data rate on these frequency bands is generally between 1.2 ~ 20 kbps. Error-correction, acknowledgment, retransmission are mainly used solve the interference and conflict problems. And there is no better solution temporarily without using frequency hopping technology when continuous interference occurs. But because of low band, the penetrating ability is relatively stronger and the communication distance is relatively farther (Li *et al*., 2012).

CHIPS CONFIGURATION

MCU configuration of data acquisition node: In this system, the data acquisition node packs data gathered by sensors and sends the packets to the base station through the radio. A series of operations like these will be done after a request of the upper computer sent by the base station is received. The ADC, SPI, I/O, clock and WTD (watchdog) of C8051F310 need to be configured as follows; (1) Crossbar enable (XBARE (XBR 1.6) = 1), (2) I/O register of output signal need to be configured as push-pull mode (PnMDOUT = 1), (3) Tri-line (SI, SO, SCLK) SPI enable (SPI0E(XBR 0.1) = 1), the three lines and Csn (chip enable port) added compose four-line SPI communication to CC1101, (4) Internal oscillator enable, frequency undivided (OSCICN = 0x83) and (5) WTD closed (PCA0MD & = ~0×40).

Corresponding Author: Wei Wei, School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

MCU configuration of base station: The communication between base station and the data acquisition node is full-duplex mode, as well as the upper computer and base station. The base station transfers some requests and commands of the upper computer to the data acquisition node and it unpacks and sends the received data to the PC through a serial port. So, the SPI, UART, I/O, clock, timer and WTD of C8051F310 need to be configured as follows; (1) Crossbar enable (XBARE (XBR 1.6) = 1), (2) I/O register of output signal need to be configured as push-pull mode (PnMDOUT = 1), (3) PnSKIP register configuration, (4) URAT0 enable (URAT0E (XBR0.0) = 1), (5) URAT0 reception permitted (SCON0 = 0×10), (6) Timer1, an 8-bit auto-reloadable timer/counter is used to be baud rate (9600 baud) generator. Frequency divided by twelve (TMOD = 0×20, CKCON = 0×00, TH1 = 0×96), (7) Tri-line SPI enable (SPI0E(XBR 0.1) = 1), (8) Open Timer1 (TR1 = 1), global interrupt (EA = 1) and URAT0 interrupt (ES0 = 1), (9) Internal oscillator enable, frequency undivided (OSCICN = 0×83) and (10) WTD closed (PCA0MD & = ~0×40).

CC1101 initial configuration: CC1101 need to be initially configured before normal operation, like setting working frequency, channel and communication rate. CC1101 gets configured by MCU through 4-line compatible SPI interface (SI, SO, SCLK and CSn). It is a full-duplex synchronous serial communication interface. And CSn is chip selection pin, SI is digital transmission pin for data input, SO for output. SCLK provides a synchronous clock, data is to be written or read at the rising edge.

To begin the configuration, a header file, designated "CC1101 h" is needed, while it cannot be found in IDE and has to be defined all by myself. The initial definitions throughout the header file almost contain all the registers of CC1101 and transform actual addresses into mnemonic codes, which is convenient to configure the chip and realize corresponding functions.

Now it is time to initialize CC1101. C8051F310 reads and writes the registers of CC1101 by SPI interface. A function named halSpiWriteReg () is used to uni-writes all the status registers of CC1101. Address of a register must be written before writing data to the address and then CC1101 do corresponding operation.

Program execution: Firstly, CSn pin is pulled to low-level, thus CC1101 entry SPI mode. Next, the program calls halSpiWriteReg () to write address information and then write data. Finally sets CSn pin high-level, closes the SPI mode, see Code 1.

We can write-through CC1101 by setting the continuous-writing register of CC1101. halSpiStrobe () is for command input, we just need to input a command byte and then a register configuration will be completed, see Code 2.

Code 1: halSpiWriteBurstReg

```
void halSpiWriteBurstReg(unsigned char addr, unsigned char *buffer,
unsigned char count)
{
        unsigned char i, temp;
         temp = addr / WRITE_BURST;
         CS_CC1101 = 0;
         while (GDO0);
        SpiReadWrite(temp);   // write address information
        for (i = 0; i < count; i ++)
        {SpiReadWrite(buffer[i]); }   // write data
        CS_CC1101 = 1;
}
```

Code 2: halSpiStrobe

```
void halSpiStrobe(unsigned char strobe)
{
        CS_CC1101 = 0;
         while (GDO0);
        SpiReadWrite(strobe);   // input a command byte
        CS_CC1101 = 1;
}
```

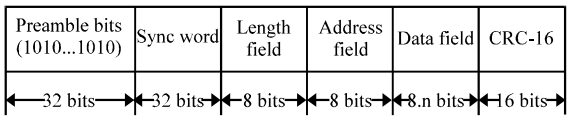

Fig. 1: Wireless packet format

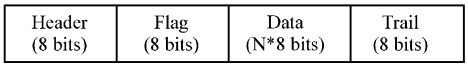

Fig. 2: Data field format

CC1101 supports multiple packet formats which can be chosen according to the need. The system communication module set four-bytes preamble bits; 32 bits sync word, 30 must matches when receiving data; the first byte after the sync word of length-variable packet is length field, 255 bytes at most; open address check and CRC check. The packet format is shown in Fig. 1 (Lou *et al*., 2011) and the data field format is shown in Fig. 2.

PROCESS DESCRIPTION

The transmit-receive process of CC1101: CC1101 has three working states: IDEL, RX (Receive), TX (Transmit). Transition between these states is shown in Fig. 3.

Apart from the basic chip initial configuration, sending and receiving data, C8051F310, as an microcontroller, but also interrupt on CC1101 if needed. Interrupt management program is designed for state detection and switch and execution of the corresponding

interrupt operation. Thus the state switch goes properly when system is running (Porret *et al*., 2000).

The address of CC1101 transmitting end needs to be initialized before sending started and mutually matches with the receiver's address, as well as the effective data width of. And then set MCU to delivery mode, pack the data, encode, modulate and send. It is proved that sending is successful if a pulse is detected on GD02, otherwise within a short time, failure and sending again. After the transmission is completed, wipes up the buffer and pulls the corresponding pin low, the CC1101 goes back to IDLE mode.

CC1101 entries receiving state if needed and continually detects the carrier to receive data. The receiving work gets started if a carrier in the same frequency baud is got and the address matches in the preset period of time. Then verification work and so on. At the same time, a pulse will be detected on GD02. Now the first byte will be read from the buffer, which exactly is the length of the packet. If it is the right length, read out the data from the buffer. Finally, read two state registers named RSSI (Received Signal Strength Indicator) and LQI (Link Quality Indicator). Receiving success if the MSB (Most Significant Bit) is '1'. Then back to IDLE state (Fig. 4).

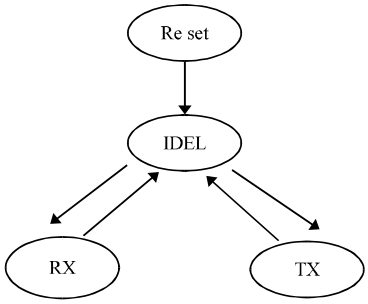

Fig. 3: CC1101 state transition diagram

Program flow of data acquisition node: The system must be initialized after the power of data acquisition node turned on, which includes the initial configuration of C8051F310 and CC1101. After that the system gets and packs all collected data by sensors and waits for upper command. Thus the packet will be sent to the base station through the air as soon as a right command is received. The system will pack another data packet and wait again (Fig. 5).

Program flow of base station: The base station system must be initialized before formal work. After series of configuration, the system starts to wait. If a data request command comes from the upper computer, the system goes into serial interrupt, reads buffer, analyses the command and gives it to data acquisition node when needed, returns the break point and waits again. When a packet comes, the base station will send it to upper computer by serial port (Fig. 6).

Serial port interrupt: Full-duplex asynchronous serial communication is adopted between base station and computer, baud rate 9600 bps. Computer

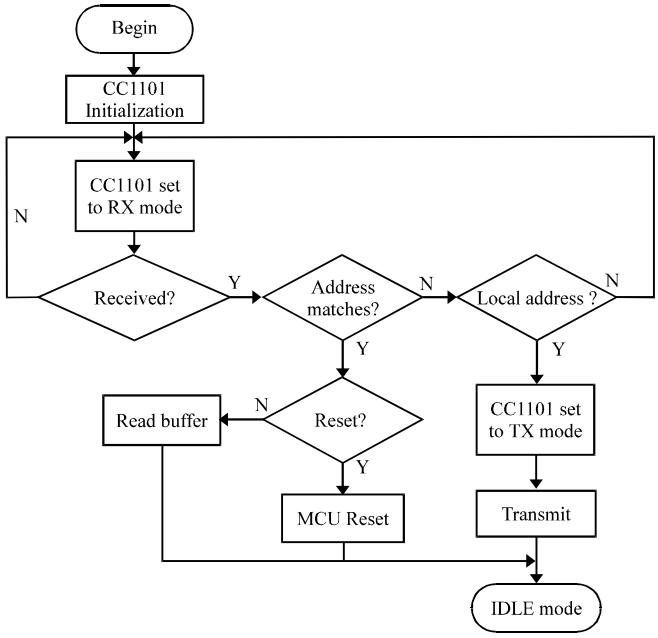

Fig. 4: Flow-chart of wireless data transmission

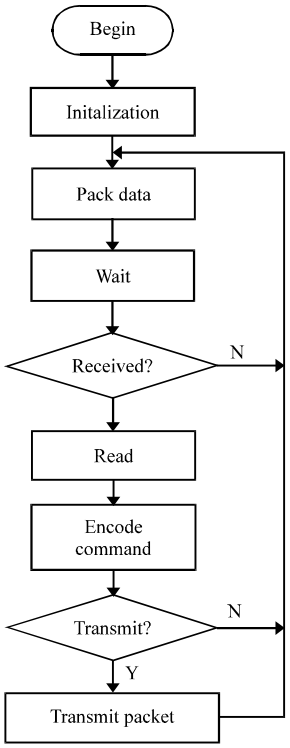

Fig. 5: Flow-chart of acquisition program

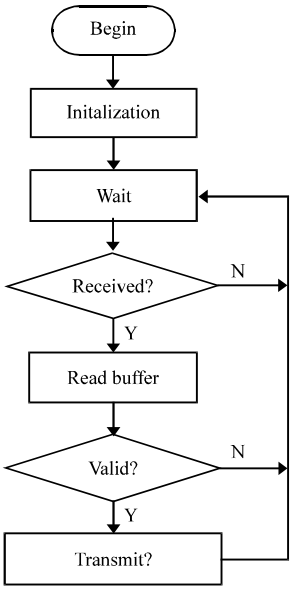

Fig. 6: Program-chart of base station program

gets all in control that the base station and acquisition node do any step as computer commands.

The base station MCU program goes into interrupt as soon as it gets a command from computer. The base station sends it to acquisition node if it is a data-required

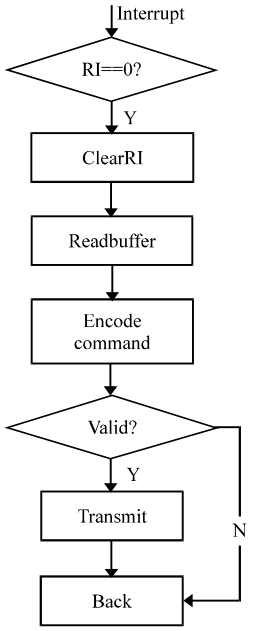

Fig. 7: Program-chart of serial port interrupt

command and then transmits the packet that comes from the acquisition node to computer. After everything is done, the program goes back to the break point (Fig. 7).

COMPUTER SOFTWARE DESIGN

The PC interface is written in C#.NET. We use the standard RS-232C to communicate between PC and the base station. It operates in full-duplex mode and no handshaking or flow controller. Dot NET has encapsulated SerialPort class for serial port resource included in System.IO.Ports namespace. The class provides the synchronous I/O and the event-driven I/O to access to pin, interrupt states and the properties of serial driver. So, 'using System.IO.Ports' must be added at the starting position in the program (Lutz and Laplante, 2003).

Communication parameters: The most commonly used parameters in serial communication are the port number and communication format (baud rate, data bits, stop bits and parity bit).

Get or set the communication port for [PortName] property, including, but not limited to all available COM ports. Noted that the returned type of the [PortName] is String. Under normal circumstances, the normal value that the [PortName] returns is COM1, COM2 and so on.

SerialPort class has four attributes corresponding to hardware, named [BaudRate], [DataBits], [StopBits] and [Parity]. [Parity] and [StopBits] are enumeration types.

[Parity] enumerates Odd, Even, None and [StopBits] None, One, Two. [DataBits] has a default value 8, [StopBits] is 1 and port number is COM1. These attributes can be set as you wish. Both sides who communicate each other should keep the same data format. Generally, when communication happens between PC and non-PC, the baud rate is indicated by the latter (Liu, 2008).

Open and close the serial port: SerialPort calls Open () method to open a serial port and Close() method to close. As a rule of thumb, [RtsEnable] need to be set as True after the serial port open, or the data read and write can be done normally.

Data read and write: For writing data, it is easy for SerialPort class to call Write () or WriteLine () method. Write () method is used to send byte arrays, characters and strings to the other side. And WriteLine () sends stings appended with a line break.

Or use ReadExisting () method to read all the current data and returns as a string.

All the methods are synchronous called except for ReadExisting () and ReadTo (). Thread is blocked until the buffer has data in or the time last more than [ReadTimeOut] and causes ReadExisting exception.

DataReceived event: DataReceived event will be triggered if more than [ReceivedBytesThreshold] bytes (characters) or an end-of-file character is received. And [ReceivedBytesThreshold] default value is 1.

Reading data in SerialPort class is far more different than writing. Because a serial will never know the data comes at what time, there are two ways to achieve the reading things: (1) Thread read in real time (2) Event triggered. Event triggered way is better to be used because the former way is not efficient enough. The trigger thing is decided by the operation system and implemented in a indirect way and it cannot transfer data with data-display widget of the main thread. For the cross-thread treatment, C# has asynchronous or synchronous delegate way, named Control.BeginInvoke or Invoke when a complete piece of data comes (Han *et al*., 2007).

Concrete implementations: The whole progress of serial communication in the dot net program is in control by creating a new SerialPort class.

The introductions of PC control program flow (Fig. 8): run the program, select port name and communication format first and open the port, click 'Run' button, a data-required command will be sent to the base station.

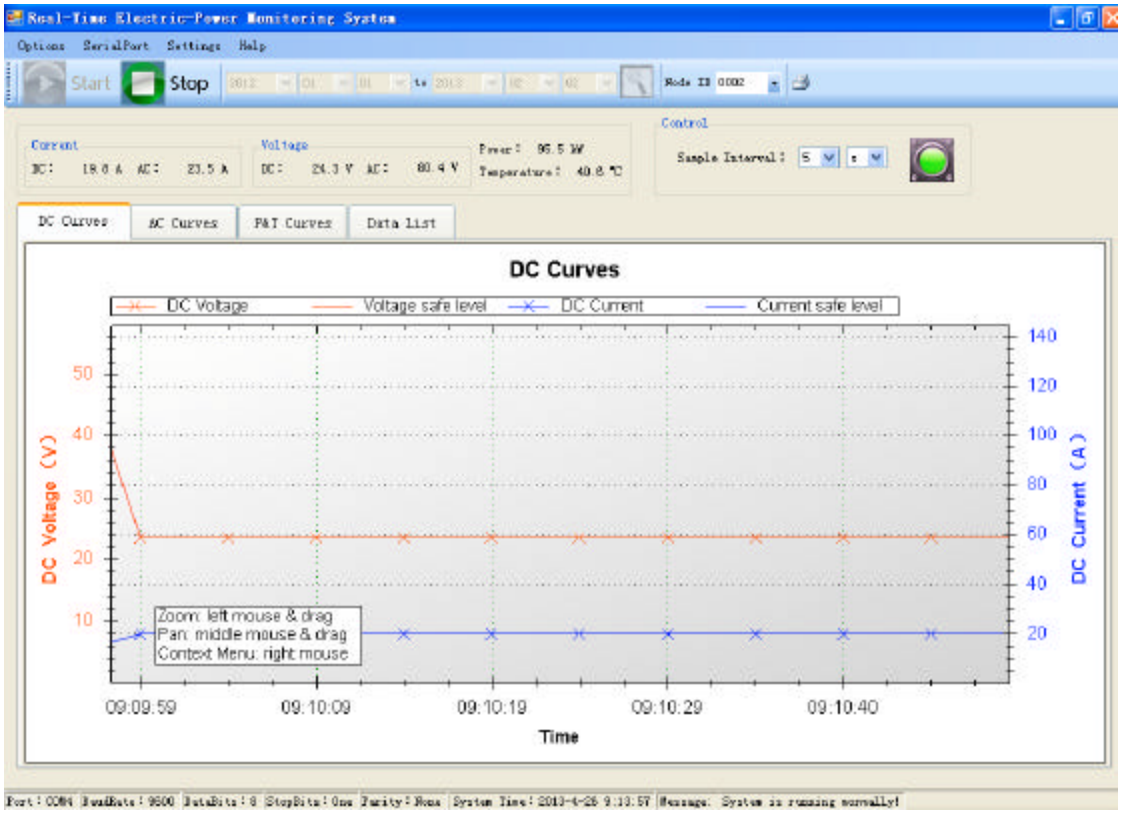

Fig. 8: Program interface

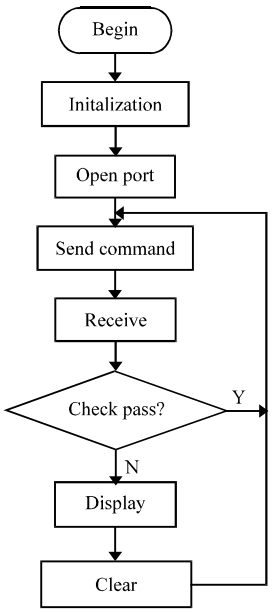

Fig. 9: Program-chart of computer

Header (8 bits)	Flag (8 bits)	Data (N*8 bits)	Check (8 bits)	Trail (8 bits)

Fig. 10: Serial-port packet format

Sum-checking is adopted when a packet (format, see Fig. 9) comes and the data will be displayed in certain sections (Fig. 10). The refresh frequency can be verified in the menu.

CONCLUSIONS

This is a wireless transmit-receive system based on wireless point-to-point communication technology, which includes two hardware modules and a PC software. The transformation between wire and wireless signal has been achieved and all the power real-time information presents to the users intuitively. The effective distance of communication reaches above 20 m and the unobstructed distance can be increased to 300 m by verifying transmit power. This system has many excellent characters, such as low cost, few hardware, simple structure, large versatility and strong scalability, stability and reliability.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. This program is supported by Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No.2013JK1139) and Supported by China Postdoctoral Science Foundation (No. 2013M542370) and supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20136118120010). And this project is also supported by NSFC Grant (Program No. 11226173, 11301414 and 61272283).

REFERENCES

Han, Z., Z. Zhao and Z. Ding, 2007. Implementation of serial communications between PC and IC read-writer based on C#. Serialport class. Sci. Mosaic, 3: 79-81.

Krzysztofik, W.J., 2009. Modified sierpinski fractal monopole for ISM-bands handset applications. IEEE Trans. Antennas Propag., 57: 606-615.

Li, B., W. Wang, Q. Yin, R. Yang, Y. Li and C. Wang, 2012. A new cooperative transmission metric in wireless sensor networks to minimize energy consumption per unit transmit distance. IEEE Commun. Lett., 16: 626-629.

Liu, L.N., 2008. Implement of serial port communication of unicode coding string based on C# language. Modern Comput., 7: 136-137.

Lou, E.H.M., E.K. Brunton, F. Kamal, A. Renggli and K. Kemp *et al*., 2011. A low power wireless data acquisition device to monitor gait patterns for children with toe walking during daily activities. J. Med. Dev., Vol. 5. 10.1115/1.4003809

Lutz, M.H. and P.A. Laplante, 2003. C# and the. NET framework: Ready for real time? IEEE Software, 20: 74-80.

Porret, A.S., T. Melly, C.C. Enz and E.A. Vittoz, 2000. A low-power low-voltage transceiver architecture suitable for wireless distributed sensors network. Proceedings of the IEEE International Symposium on Circuits and Systems, Volume 1, May 28-31, 2000, Geneva, Switzerland, pp: 56-59.