Wireless Remote Control System

Adrian Tigauan*, Eugen Lupu**

*Technical University of Cluj Napoca,Communications Dept., Cluj Napoca, Romania, adrian_tigauan@yahoo.com
** Technical University of Cluj Napoca,Communications Dept., Cluj Napoca, Romania, Eugen.Lupu@com.utcluj.ro

Abstract—**This paper presents the design of a wireless remote control system based on the ZigBee communication protocol. Gathering data from sensors or performing control tasks through wireless communication is advantageous in situations in which the use of cables is impractical. An Atmega328 microcontroller (from slave device) is used for gathering data from the sensors and transmitting it to a coordinator device with the help of the XBee modules.**

The ZigBee standard is suitable for low-cost, low-data-rate and low-power wireless networks implementations. The XBee-PRO module, designed to meet ZigBee standards, requires minimal power for reliable data exchange between devices over a distance of up to 1600m outdoors.

A key component of the ZigBee protocol is the ability to support networking and this can be used in a wireless remote control system. This system may be employed e.g. to control temperature and humidity (SHT11 sensor) and light intensity (TSL230 sensor) levels inside a commercial greenhouse.

I. INTRODUCTION

Wireless sensor networks is a promising and fast growing technology which has numerous applications in the military and the civilian domains (medical, industrial, home network). They have attracted a lot of research attention recently and have passed from the research projects to the real world commercial applications. Modern sensor networks are bidirectional and allow controlling the remote system based on measurements from the sensors. [1]

A wireless sensor network is basically built from a number of autonomous sensor nodes (can be from a few to hundreds or even thousands) which cooperate and communicate between them. [6]

There are several standards for wireless communication but only a few are compatible with wireless sensor networks. The most significant wireless standards are ZigBee, Bluetooth and Wi-Fi (WLAN). Each standard is practical in its own application types. A brief comparison is made between these standards in the following table. [1]

	ZigBee	**Wi-Fi(WLAN)**	**Bluetooth**
Typical Range	10-100 meters	50-100 meters	10-100 meters
Networking Topology	Ad-hoc, peer to peer, star or mesh	Point to hub	Ad-hoc, very small networks
Operating Frequency	868 MHz (Europe) 900-928 MHz (North America), 2.4 GHz (worldwide)	2.4 and 5 GHz	2.4 GHz
Complexity (Device and application impact)	Low	High	High
Power Consumption (Battery option and life)	Very low (low power is a design goal)	High	Medium
Typical Applications	Industrial control and monitoring, sensor networks, building automation, home control and automation, environmental monitoring	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets, mouses

Table 1: Shows a comparison between three wireless communication protocols [5]

From these standards for wireless communication the ZigBee standard is adequate for use in wireless sensor networks.

II. ZIGBEE PROTOCOL

A. *Specifications of the ZigBee standard*

The ZigBee standard has been created by the ZigBee Alliance, made up many member companies, from semiconductor industry and software developers to original equipment manufacturers (OEMs). It is a standard designed for low-cost, low-power and low-data-rate networking. Devices based on ZigBee operate in the industrial, scientific and medical (ISM) radio bands: 868 MHz in Europe, 915 MHz in North America and 2.4 GHz worldwide. [2]

The standard operates on the IEEE 802.15.4 physical radio specification. The specification describes the physical layer (PHY) protocol functions and interactions with the medium access control layer (MAC) layer and also defines the minimum hardware-level requirements, such as the receiver sensitivity and the transmitter output power. The modulations used in IEEE 802.15.4 are BPSK (binary phase shift keying), ASK (amplitude shift keying) and O-QPSK (offset quadrature phase shift keying). [2]

One of the benefits of ZigBee is the low cost which allows for it to be widely deployed in wireless monitoring and control. ZigBee devices can activate (pass from sleep mode to active mode) in a very short time (15 msec) so they can sleep most of the time. This makes it possible to have a battery life lasting for years.

B. *XBee Module*

The XBee and XBee-Pro RF modules, made by Digi International, were engineered to meet up the specifications of the IEEE 802.15.4 standard which is compliant with the ZigBee standard. The modules are suitable for low-cost and low-power wireless sensor networks. They assure a reliable communication between them and use minimum power. [3]

From the three typical frequency bands these module operate in ISM 2.4 GHz band.

The two modules, XBee and XBee-Pro have similar features. The main difference between them is the transmission power: XBee-Pro uses a power of 63mW while XBee uses only 1mW. This makes the XBee-Pro have a greater power requirement (250mA at 3.3V for transmission) than the ordinary XBee (45mA at 3.3V for transmission). The gain of XBee-Pro over XBee is the superior range for transmission. XBee-Pro can send data over a distance of up to 90m indoors (or in an urban environment) and up to 1600m outdoors while the XBee can only transmit at a distance of up 30m indoor and up to 90m outdoors. [3]

In my project I have chosen to use XBee-Pro because of its better range and because power consumption is not an essential aspect. In this paper the two modules will be named generically XBee as most of the application specific characteristics are the same.

Both modules have a RF data rate of 250.000bps and a serial data rate ranging from 1200bps to 250 kbps. These modules interface with a host device through a logic-level asynchronous serial port. The host must have a logic and voltage compatible UART or a level translator has to be introduced between the XBee and host. [3]

The modules operate by default in transparent mode which makes them act as a serial line replacement. No configuration to the module has to be made in this mode and all the data received through the input pin of the transmitter will reappear on the data output pin of the receiver.

III. SYSTEM DESIGN

A wireless sensors network is made up from a series of independent sensor nodes.

Sensor nodes work together to send data to the coordinator device. Putting together a wireless sensors network involves that the nodes meet the requirements of a given application. When choosing hardware components requirements of the application influence cost, size and energy consumption of the nodes. [1]

A sensor node is a device containing sensors and actuators, a microcontroller (a microprocessor), memory, a radio transceiver and a power supply.

Sensors take measurements of physical conditions such as temperature, pressure, vibration, motion and sound or check the presence of chemical substances. Sensors working together with actuators provide the actual interface to the physical world because they can observe and modify the parameters of the environment. [6]

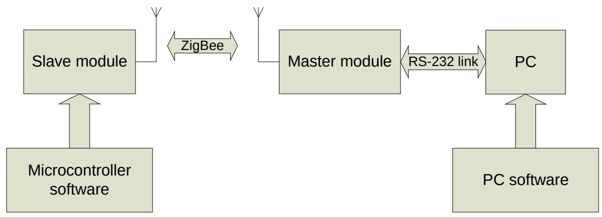

Figure 1: Shows the block schematic of the application

Memory is needed for storing programs and data. The microcontroller is used to process the relevant data and to execute code. The power supply can be a battery or some form of energy harvesting. Sensor nodes communicate between them through the wireless medium using a radio transceiver and a wireless communication protocol. [1]

These components provide sensing, data computation, control and communication capabilities. Network protocols turn a collection of sensor nodes in a sensor network.

In my application I only used two nodes leaving the network design for future developments. The block schematic of the system created is shown in figure 1. The system has both a hardware component and a software component.

IV. HARDWARE IMPLEMENTATION

Master/slave is a model for a communication protocol, which involves one device or process (known as the master) controlling one or more devices or processes (known as slaves). After the master/slave association was set up, the direction of control will always be from the master to the slaves.

In the system created the hardware component has two parts corresponding to this model: a Master module and a Slave module.

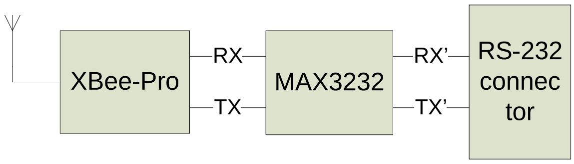

Figure 2: Shows a simplified schematic of the Master module

The slave device will be placed in the field and will measure humidity, temperature and light intensity.

This data will be transmitted over the air, using ZigBee, to the master device. The master device passes the data to the PC where it will be processed in real time.

The Master module is made up of a XBee module and a Max3232 integrated circuit. This device receives data through the XBee module and transmits it through the serial port to the PC. The Max3232 is a logic level translator assuring an electrical interface between the ZigBee communication module and the RS-232 serial port. Max3232 is circuit similar to the well-known Max232 but operates with a power supply ranging from 3V to 5.5V. The XBee device was described earlier in this paper.

The Slave device is assembled from an ATmega328 microcontroller, a XBee module and two sensors (SHT11 and TSL230).

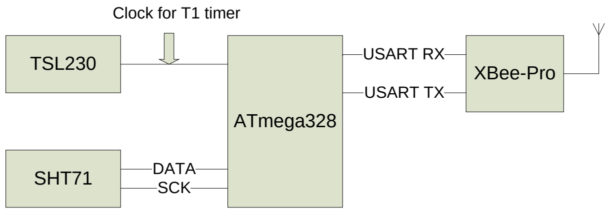

Figure 3: Shows a simplified schematic of the Slave device

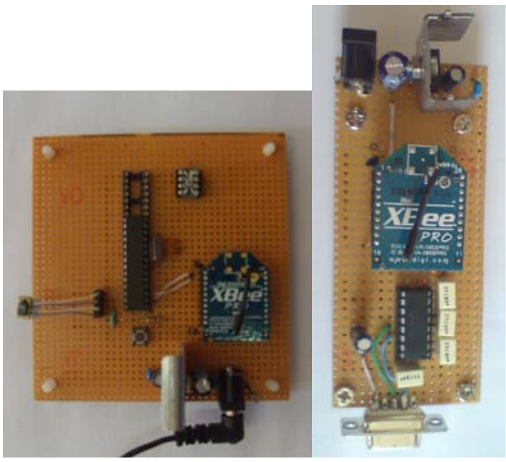

Figure 4: Actual photographs of the slave and master devices

The SHT11 sensor integrates sensing elements and signal processing and also provides a digital output. A capacitive sensor element is used for measuring relative humidity while temperature is determined using a band-gap sensor. Both sensors are coupled to a 14 bit analog to digital converter and a serial interface circuit.

The SHT11 communicates with the microcontroller through a 2-wire serial interface. The first signal of the interface is a serial clock which is an input for the sensor. This line is used for synchronizing the communication between the microcontroller and the SHT11. The serial data signal is used for transferring data in and out of the sensor so it is a bidirectional communication line. [7]

The TSL230 is a configurable light-to-frequency converter which combines a photodiode with a current-to-frequency converter on a single chip. The output from this sensor will be a pulse train with the frequency directly proportional to light intensity. This sensor measures irradiance which is the power of electromagnetic radiation per unit area incident on a surface.

The sensitivity of the photodiode can be adjusted using two of the sensor's digital pins. The output frequency given by the TSL230 can be divided by a value which is also set through two digital pins. Based on the settings made the irradiance in microwatt/cm^2 is ten times the frequency in kilohertz. [8]

ATmega328 is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. It has 32K bytes of programmable flash. The controller uses a 8 MHz external crystal oscillator. The ATmega328 reads data from the sensors, processes the data and transmits it at regular intervals to the Master device using the XBee module. The microcontroller is programmed to communicate on a 2-wire serial interface with the SHT11 and to measure the frequency of the pulse train received from the TSL230 using a 8-bit and a 16-bit timer. It uses the programmable serial USART to communicate with the XBee. [4]

V. Software Implementation

The software implementation also has three parts: a software application for the ATmega328 microcontroller, the parameters for the XBee modules and a PC software application.

The XBee module works by default in transparent mode in which it acts as a replacement for the serial line. In order to modify the parameters of the module, the module has to enter AT mode. In this mode you can change parameters such as: baud rate, transmission power, networking and security settings, power management modes, data encoding, etc.

The software application on the ATmega328 was created using AVR Studio 4 and the WinAVR GCC compiler.

The software on the microcontroller has the following functions:

- Collect data from the sensors at regular intervals
- Process the data for obtaining the values for temperature, humidity and light intensity
- Create data packets using a certain packet format
- Send data packets to the master module

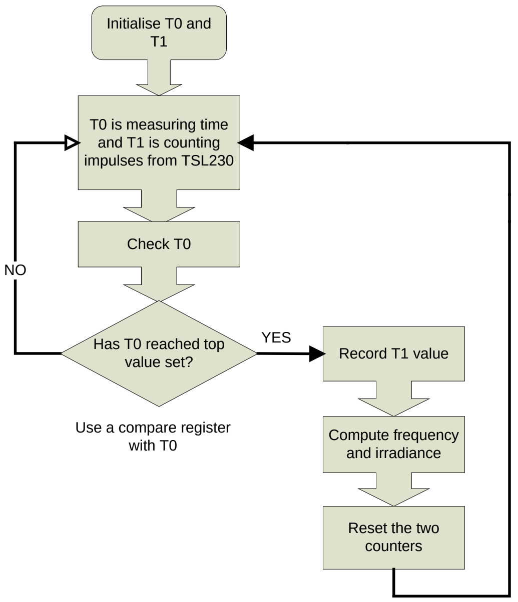

Figure 5: The algorithm for determining frequency and light intensity

For reading the SHT11 sensor the interface specifications have to be met. The interface is composed of two serial lines (one for data and one for the clock). Proper communication with the sensor assumes that the signals are strictly within certain timing limits.

Measuring the relative humidity or temperature involves a relatively simple procedure. The controller issues a command for temperature or relative humidity and then has to wait for the measurement to complete. Time for the measurement to complete depends on the measuring precision. The controller waits for the sensor to signal that the data is ready and then starts reading the data.

Gathering data using the TSL230 sensor raises more problems because the physical parameter to be determined (light intensity) is proportional to the frequency of the output signal. So a frequency measurement has to be carried on. The diagram in figure 5 shows the algorithm used for calculating frequency and light intensity.

The TSL230 generates an impulse train which has a variable frequency. The output from the sensor is connected to the T1 timer clock pin so the impulse train acts as a clock for the T1 timer/counter. The T1 timer counts the impulses on the T1 timer clock pin. Knowing the number of impulses given in reference time we can calculate the mean frequency over that period of time. Frequency will be inversely proportional to the value stored by T1. For obtaining a reference time timer T0 is also used. Once the frequency and light intensity is computed, the data is transmitted on the USART.

The PC software application was written in C# using Visual Studio 2010. The application detects serial ports and has the ability to open them. Several parameters for the serial port can be set, such as: baud rate, parity, number of data bits and the number of stop bits. Once the serial port is open the application is ready to receive data.

SerialPort is a class from the namespace System.IO.Ports from .NET 4. The class is used to control a serial port file resource. It provides event-driven and synchronous I/O and also access to the properties of the serial driver. This class can open port connections, close port connections, set ports for communication, set the serial baud rate, read or write bytes on the serial port, etc.

Receiving data on the serial port is an event that triggers two actions. The first action is the printing of the packets received, in a text box on a new line. The second action presumes extracting values from the packets and plotting them on charts. The incoming packets have to be in the following format for the data to be extracted:

#TtempTLirradLHhumiH# as in temperature in °C, irradiance in microwatt/cm^2 and relative humidity in percent.

For plotting data on charts the Chart Controls library from .NET4 was used. Chart controls enables creating Windows Forms applications wich include charts for complex statistical or financial analysis.

The charts in my application get updated in real time.

VI. Conclusion and Results

I have managed to collect data from the two sensors. In the following screen capture the values for temperature, light intensity and humidity are plotted on three charts. These physical parameters modify in real time and the charts change accordingly.

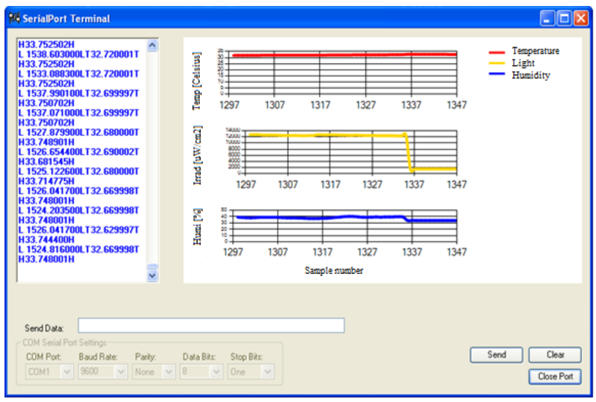

Figure 6: A screen capture of the PC software application plotting temperature, light intensity and humidity in real time

In the screen capture above the charts show that temperature is approximately 32 °C, relative humidity is about 33 % and irradiance is about 1500 uW/cm^2. The packets arriving on the serial port can be seen on the left of the figure. On the bottom of the figure we can see that data is coming in on port COM1, at a baud rate of 9600 bps.

The ZigBee communication protocol used in my system is suitable for implementing low cost and low complexity systems.

The modules built can form a WSN (wireless sensor network) which can monitor and control the temperature, humidity and lighting in a commercial greenhouse. For controlling these physical parameters additional equipment is needed such as: vents, fans, heating systems, cooling systems, lighting systems (light bulbs, fluorescent lamps), etc.

The project is under development and other improvements will be introduced such as: dynamical change of the data rate, more sensors added to the slave device, communication with multiple XBee modules and the implementation of several network topologies.

REFERENCES

[1] Holger Karl, Andreas Willig, "Protocols and architectures for wireless sensor networks", *John Wiley & Sons Ltd*, 2005

[2] Shahin Farahani," ZigBee Wireless Networks and Transceivers", *Newnes, Elsevier*, 2008

[3] "XBee®/XBee-PRO® RF Modules", Product Manual available on http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf

[4] "ATmega48A/48PA/88A/88PA/168A/168PA/328/328P Datasheet" available on

http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf

[5] The comparison of Wi-Fi, Bluetooth and ZigBee

http://www.sena.com/blog/?p=359

[6] Wireless sensor network

http://en.wikipedia.org/wiki/Wireless_sensor_network

[7] "Datasheet SHT1x (SHT10, SHT11, SHT15) Humidity and Temperature Sensor" available on

http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf

[8] "TSL230, TSL230A, TSL230B PROGRAMMABLE LIGHT-TO-FREQUENCY CONVERTERS Datasheet" available on

http://www.roboeq.com/PDF/0405001.pdf