Wireless Lab Design Project Report

Mason Nixon and Matthew Humphry

Auburn University
Auburn Alabama, United States

men0004@auburn.edu

mzh0013@auburn.edu

***Abstract*— This document contains all the details of our project for Wireless Lab. We first provide and introductions to our project followed by relevant technical information about Xbee and Arduino including how to set them up and write code for them. Finally we discuss possible applications of our design.**

***Keywords*— Arduino, Xbee, hardware, technology, and wireless.**

I. Introduction (MN)

Wireless communications date back to Guglielmo Marconi's and Nikola Tesla's experiments in the late 19th century. Communication through the air can be thought of as the peak of human communication technology since we can potentially transmit wireless signals around the world and beyond. For our simple demonstration of a wireless communications link, we propose to take two Xbee Pro 50mW wireless modules and create a Zigbee (protocol) wireless link between a PC and a sensor.

Although the hardware is stated to have a range of 2 miles (~3200m) outdoors in a line-of-sight environment and up to 300ft (~90m) in an indoor environment, our link will only be used for communication over a short distance of about 100ft (~33m). As noted above, the transmit power will be 50mW (+17dBm). The RF Data rate will be 250kbps. The operating frequency is 2.4GHz. This part is FCC approved under FCC prt 15.247 and FCC ID: MCQ-XBEEPRO2.3 We will use the Atmel328 microcontroller to interface with the Xbee module and we will use a 15cm Infrared distance sensor for our data stream (or some other sensor with an equivalent data stream).

Since the primary goal of wireless communication is the transmission of information through the channel, this project is an all-encompassing exhibition. Our project will demonstrate the ability to set up a wireless link and transmit data effectively over the channel.

II. Technical information on Xbee and Arduino (MH)

A. Arduino Microcontroller

"Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in created interactive objects of environments" [IV].

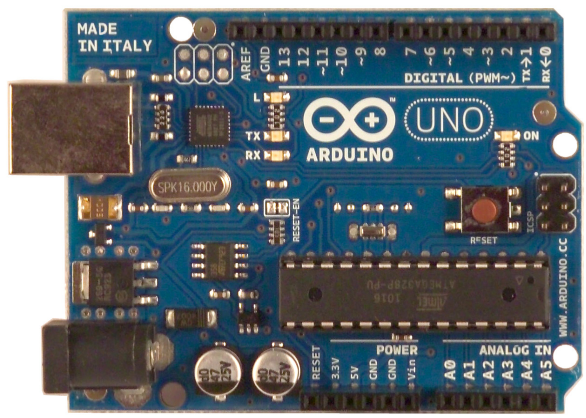

Fig. 1 Arduino Uno Microcontroller [IV]

Power

The Arduino Uno can be powered with the USB connection or with an external power supply such as a battery or AC-DC power adapter. When powering the board from an external power supply the power pins are as follows:

- **VIN:** The input voltage to the Arduino board.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board.
- **3V3:** A 3.3-volt supply generated by the on-board regulator.
- **GND:** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library)

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial 0 (RX) and 1 (TX):** Used to receive and transmit serial data.
- **External Interrupts 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provides 10 bits of resolution. By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I2C:** A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

B. Xbee RF Module

“The Xbee family of embedded RF modules provides OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the Xbee can substitute one Xbee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-to-market” [1].

Fig. 2 Xbee RF Module [I].

Performance
- **RF Data Rate:** 250kbps
- **Indoor/Urban Range:** 300 feet
- **Outdoor/RF Line-of-Sight Range:** 1 mile
- **Transmit Power:** 50 mW (+17 dBm) / Int’l 10 mW (+10 dBm)

Features
- **Serial Data Interface:** 3.3V CMOS UART
- **Frequency Band:** 2.4 GHz
- **Serial Data Rate:** 1200 bps / 1.2Mbps
- **ADC Inputs:** 4 10bit ADC Inputs
- **Digital IO:** 10

Networking and Security
- **Encryption:** 128 bit AES
- **Reliable-Packet-Delivery:** Retries/Acknowledgements
- **IDS and Channels:** PAN ID, 64-bit IEEE MAC, 13 Channels

Power Requirements
- **Supply Voltage:** 3.0 – 3.4VDC
- **Transmit Current:** 295 mA at 3.3 VDC
- **Receive Current:** 45 mA at 3.3 VDC

III. Xbee Setup (MN)

The setup of the Xbee units can be performed using the manufacturer’s program called X-CTU. A comprehensive User’s guide to all Xbee models is available via Digi’s website, although the following is setup for this specific model, the Xbee Pro Series 2. Some requirements for use include: Only Windows 98 and higher operating systems and only 32 bit systems. Fortunately, drivers for the Xbee are available for Mac OSX (32 bit), Linux (Unspecified version, 32 bit only), and Windows 98+ (32 and 64 bit). After installing the software, utilizing a 3.3V voltage regulation circuit and a USB to FTDI circuit, we were able to connect the Xbee to a computer to program it. Beginning in the PC Settings tab in the primary window of the X-CTU program, select the appropriate COM port that the Xbee is connected to. Next, one must check the connection and current settings by selecting Test/Query to continue. The default, out-of-the box settings are: Baud: 9600, Flow Control: None, Data bits: 8, Parity: None, Stop bits: 1. If these values result in an error from Test/Query, you may have to reset the Xbee to pair with it.

Now, to change settings on the Xbee, click on the Modem Configuration tab and select Read. Here, things like Baud Rate, I/O pin configurations, and all other settings can be made. It is necessary to ensure that each Xbee is transmitting on the same operating channel on the Zigbee band and are using the same PAN ID (ours was 1337). After making changes, select Write and the Xbee will be loaded with the new parameters. Also, for the Xbee Pro Series 2 it is necessary to designate one Xbee as Coordinator and the other as an End-point device in the Function Set drop-down menu. This is so that one Xbee (the Coordinator) will setup the network for the communication between multiple End-points and Routers. Also, it should be noted that settings may be changed via terminal, such as a HyperTerminal application, Putty, or by the Terminal tab on the X-TCU software.

In addition to being able to adjust settings in a terminal, one may also view the data stream from the Xbee. Simply Test/Query in the PC Settings tab, type '+++' without quotes into the terminal, and then after receiving the message 'OK' the data stream on the Xbee will appear in the terminal. Using this, we were able to confirm transmission and reception of data, although our final design incorporates a microcontroller to receive the data.

IV. ARDUINO SETUP AND CODING (MH)

The setup of the Arduino is very simple and straightforward. A comprehensive users guide and step-by-step instructions to set up your Arduino is available from the Arduino website. Before usining your Arduino, you will first need to download the Arduino environment and possibly drivers if using an older board. If you are using the Arduino Uno or Mga2560 development boards, you will not have to download any drivers. Once you have the Arduino environment installed on your computer, you can then plug your Arduino into your computer and begin programming.

The Arduino programming language is a derivative of the C programming language. Arduino provides multiple example programs when you download the Arduino environment and you can also reference a list of Arduino commands from the Arduino website. The basic structure of a program is very simple. First there is the void setup() method, where you would initialize any variables or make any other declarations. The other part of the Arduino program is the void loop() method. This is where you will put the main execution commands for your program.

Once you have written code for your Arduino, there are only a few steps remaining to execute your program. First from the tools menu, you will have to select your Arduino board from the drop down list of boards. After the board is selected you will have to select your Serial port from the tools menu. These steps ensure that your code is appropriate for your board and that it will be written to the correct place. The final step is to upload the program to your Arduino by clicking the Upload to I/O Board button at the top of the programming environment. Once the program is uploaded to the Arduino it will continue to execute your code until another code is loaded or you remove the power supply to your Arduino.

For our project, we programmed one Arduino to read the output of an IR sensor. First we declare that we will read data from the IR sensor on pin 1. We then set the serial output rate to 9600. The main loop reads in data from the IR sensor and writes to the Serial Output whether an object is detected or not depending on the value from the IR sensor. It then waits half a second and repeats. The serial output is broadcasted via its XBee dongle to the other Arduino board. The receiving Arduino uses the NewSoftSerial library that was downloaded from the Arduino website. This library allows us to set input pins 2 and 3 to be a serial port. We set the serial rate to 9600 and begin reading the output of the transmitting Arduino. We print the results that can be monitored from the Serial Monitor on the Arduino programming environment.

V. APPLICATIONS OF WIRELESS SENSOR NETWORKS AND THE ZIGBEE PROTOCOL (MN)

The most obvious application for a Xbee-derived network is WPAN (Wireless Personal Area Networks) or a wireless sensor network. Decreased cost of installation and maintenance of wires are some of the upfront benefits. One must also assess the benefit of utilizing mesh networks to route around failed sensor nodes or nodes with interference due to machinery [VI]. There is an inherent benefit that the topology of a sensor network is free to change. Some obvious disadvantages of a wireless sensor network are the limited power from batteries, computational power, and onboard memory. These however, can be overcome by aggregating data from multiple sensors and only performing simple calculations at the site of the sensor to relieve the network of higher overhead due to too much data being transmitted. As noted by Garg, sensor networks must have "built-in tradeoff mechanisms that give the end-user the option of prolonging network lifetime at the cost of lower throughput or higher transmission delay."

Although there are some disadvantages to wireless sensor networks, the benefits can far outweigh the costs when considering for instance that one has the possibility of connecting up to 65,536 devices to the network while still retaining a flexible topology. Other advantages include lower cost, solution flexibility, use of frequencies outside of the bands needing FCC authorization [VII].

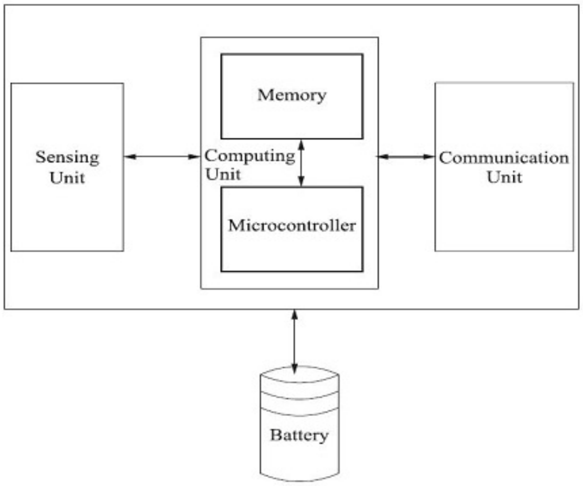

Fig. 3 A simple sensor network [V].

In regards to the Zigbee protocol, some advantages over alternatives such as Bluetooth or Wifi protocols can be described as follows. The competing systems are typically optimized for massive data transfers for, perhaps, multimedia applications. "In comparison with other wireless communication systems, the ZigBee stack can be considered to be small: 4 Kb for the simplest nodes and 32 KB for the complete stack; for example the MAC (Medium Access Control) defines only 26 primitives as against the 131 of Bluetooth's MAC." [VII] The IEEE 802.15.4 protocol allows for 250kb/s at 2.4 GHz. Also new nodes may be added to the network in as little as 30ms and sleeping nodes wake in 15ms. The medium access layer detects if a channel is free before transmission using carrier sense multiple access-collision avoidance communication (CSMA-CA) [VII]. Another advantage of Zigbee as noted by Ascariz is that "each node can act as a host or as a router, forwarding packets on behalf of other nodes. One of its advantages is that any node can send a message to another node, using as many intermediate elements as may be necessary as relays" which is important in transmission assurance to the coordinator.

The usage of wireless sensor networks in general have application in the areas of: environmental observation of pollution or weather, military monitoring, building climate and vibration monitoring, home automation, and also healthcare [VI]. Many applications for Zigbee are available in the healthcare field, particularly in chronic disease monitoring, senior activity monitoring, personal wellness, and personal fitness. Examples of sensors in this field are: glucose meter, pulse oximeter, electrocardiograph, social alarm devices (fall detector, emergency notifier) [VIII]. For home automation, intrusion detection, fire detection, climate control, appliance automation. Other notable applications are temperature and weather sensoring, seismic monitoring, habitat sensing, radiation sensing, nuclear threat sensing, weapons sensors for ships [V].

VI. CONCLUSION (MH)

In conclusion, our project was a great success. We were able to setup and program the Arduino Microcontrollers to communicate the IR sensors data wireless via Xbee dongles. In our course work here at Auburn we have discussed theoretically how a system like this would work, but this is the first opportunity we have had to actually implement a system like this. The experience has been very rewarding because we feel that we have a much better grasp of the mechanics of wireless communication. This project may be one of the most valuable things we have done here at Auburn University.

VII. REFERENCES

I. Digi International Inc, 2011. Retrieved April 5, 2011, from Digi-International Inc. www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules.

II. Nobel Lectures, 1967. Retrieved April 5, 2011, from Nobel Lectures, Physics 1901-1921, Elsevier Publishing Company, Amsterdam. http://nobelprize.org/nobel_prizes/physics/laureates/1909/marconi-bio.html

III. Pellerin, Cheryl, United States Updates Global Posistioning System Technology, 2006. Retrieved April 6, 2011 from America.gov Archive. http://www.america.gov/st/washfile-english/2006/February/20060203125928lcnirellep0.5061609.html

IV. Arduino Inc, 2011. Retrieved April 5, 2011, from Arduino Inc. http://arduino.cc/en/

V. Zainol, Abidin, 2010. Development of a PIC-Based Wireless Sensor Node Utilizing Xbee Technology. Retrieved April 4, 2011 from, IEEExplore. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05478148

VI. Garg, Vijay K. Wireless Communications & Networking, p 675-677. Morgan Kaufmann Publishers, San Francisco, 2007.

VII. Rodriguez, J. M. System for Measuring Power Supply With Zigbee Technology. Retrieved April 4, 2011 from IEEExplore. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4258186&userType=&tag=1

VIII. ZigBee. ZigBee Wireless Sensor Applications for Health Wellness and Fitness. ZigBee Alliance, March 2009.

VIII. BIOGRAPHIES

Mason Nixon – Wireless Hardware and Electrical Engineering, May 2011

Matthew Humphry – Wireless Software Engineering, May 2011