Solar Power Mesh Network

Jeff Ruvinsky, Marek Smaciarz, Charles Currier, James Atkins

ITMT 492 & TIMT 593

Dr. Dan Tomal & Prof. Jeremy Hajek

Abstract

Have you ever needed to keep track of different temperatures within an environment like a greenhouse? We made a mesh network using Xbees that can transmit data, such as temperature data, to each other and back to the coordinator. This data is then displayed on a monitor using a processing sketch. Each Xbee node is encased in a custom built container that has a solar panel on one side and suction cups to stick on windows. The solar panels are used to charge the batteries that give power to the Xbees. We programmed our Xbees to display temperature in Fahrenheit.

Table of Contents

Introduction

Xbees are like Bluetooth sensors. They are also like Wi-Fi. But Xbees are neither of those things. They are radio modules that communicate with each other and can go much further than Bluetooth. They can also be set up like any topology including point to point, star and mesh. Xbees are very fascinating and you can use them for so many great things. What we used them for is monitoring temperature data within a broad distance. Xbees can reach approximately 10 to 20 meters and we had one coordinator, 1 router and one endpoint. The router and endpoint collect data and send it to the coordinator which is in charge of all the data that passes in its network. We keep our Xbees constantly charged using rechargeable batteries that are powered by a solar panel. For the software aspect of our project, we used a processing sketch that collects the data and displays it on the monitor. Processing is a language developed by architecture students. Its goal is to be a simple and efficient language that lets you easily display things. It is like a descendant of java with benefits.

Hardware

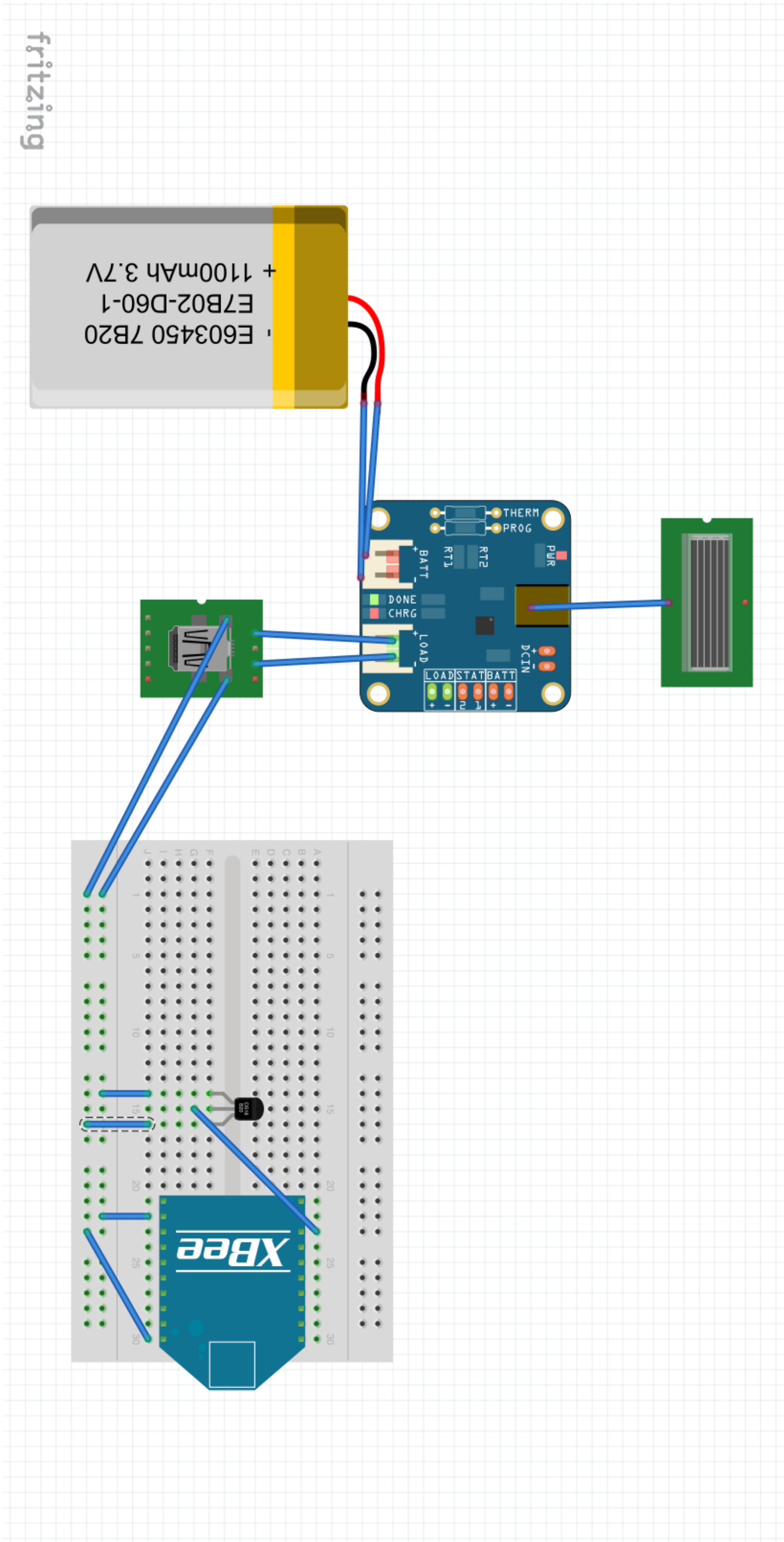
Hardware, verticality, code, and a few imaginations are key to turn this simple set up into a complex powerhouse. As stated before, using this device for a HVAC temperature system but also expanding on the system is extremely doable. With this prototype system, the group has created a 3D printed plastic case that is able to be expanded upon.

Enclosed in the plastic case, there is dedicated places for primary equipment. This primary layer includes a solar panel, a minty boost USB charger. Connected to the USB charger is a 3.3w brick battery, similar to the lithium ion batteries that we have in our cell phones. From there, the power supply is up scaled to around 5W, and wired to the next layer. With this Layer, it encloses a breadboard that is open for quick and easy modifications. On our prototype, we have three of these cases, with a light sensor, Xbee's wireless radios that will forward the information to the main computer.

The case is molded for expansion, the first layer has specific spaces for the USB charger, the 5v converter, and a back enclosure for the solar panel. Next layer, the breadboard, enclosed in an open bracket that is used for modifications. With prototype, we have used the Xbee's with temperature sensors; however, the prototype can be upgraded with Geiger counters, luminescence sensors, or humidity sensors which can be added to the breadboard, allowing for verticality.

With cost being a question, it is really simple, with the solar panels we are using free solar light we get everyday… for the most part being in Chicago. With 3D printing boxes being relatively cheap, to the point of around just a few cents. All in all the overall cost for one unit is around fifty dollars, but also being very expandable.

Schematics


fritzing
1100mAh 3.7V
E7B02-D60-1
E603450 7B20
THERM
PROG
BATT
RT1
RT2
PWR
DONE
CHRG
LOAD
DCIN
LOAD
STAT
BATT
XBee

The Processing Code

```
import processing.serial.*;

Serial myPort;            // The serial port

float tempC;
float tempF;
int yDist;
PFont font12;
PFont font24;
float[] tempHistory = new float[100];

int addr1;
int addr2;
int addr3;
int addr4;

void setup() {

  //setup fonts for use throughout the application
  font12 = loadFont("Verdana-12.vlw");
  font24 = loadFont("Verdana-24.vlw");

  //set the size of the window
  size(550, 250);


  //fill tempHistory with default temps
  for(int index = 0; index<100; index++)
        tempHistory[index] = 0;

  // I know that the first port in the serial list on my mac
  // is always my  FTDI adaptor, so I open Serial.list()[0].
  // On Windows machines, this generally opens COM1.
  // Open whatever port is the one you're using.
  String portName = Serial.list()[1];
  myPort = new Serial(this, portName, 9600);
}

void draw() {

   if (myPort.available() >= 21) { // Wait for coordinator to recieve full XBee frame
        if (myPort.read() == 0x7E) { // Look for 7E because it is the start byte
        for (int i = 1; i<19; i++) { // Skip through the frame to get to the unique 32 bit address
        //get each byte of the XBee address
```

```
        if(i == 8) { addr1 = myPort.read(); }
        else if (i==9) { addr2 = myPort.read(); }
        else if (i==10) { addr3 = myPort.read(); }
        else if (i==11) { addr4 = myPort.read(); }
        else { int discardByte = myPort.read(); } //else throwout byte we don't need it
        }
        int analogMSB = myPort.read(); // Read the first analog byte data
        int analogLSB = myPort.read(); // Read the second byte
        float volt = calculateXBeeVolt(analogMSB, analogLSB);//Convert analog values to
voltage values
        println(indentifySensor(addr1,addr2,addr3,addr4)); //get identity of XBee and print it
        print("Temperature in F: ");
        println(String.format("%.1f", calculateTempF(volt))); //calculate temperature value from
voltage value

//refresh the background to clear old data
        background(123);

        //draw the temp rectangle
        colorMode(RGB, 160);  //use color mode sized for fading
        stroke (0);
        rect (49,19,22,162);
        //fade red and blue within the rectangle
        for (int colorIndex = 0; colorIndex <= 160; colorIndex++)
        {
        stroke(160 - colorIndex, 0, colorIndex);
        line(50, colorIndex + 20, 70, colorIndex + 20);
        }

        //write reference values
        fill(0,0,0);
        textFont(font12);
        textAlign(RIGHT);
        text("257 F", 45, 25);
        text("-40 F", 45, 187);

        //draw triangle pointer
        yDist = int(calculateTempF(volt));
        stroke(0);
        triangle(75, yDist + 20, 85, yDist + 15, 85, yDist + 25);

        //write the temp in C and F
        fill(0,0,0);
        textFont(font24);
        textAlign(LEFT);
        //text(str(int(tempC)) + " C", 115, 37);
```

```
        //tempF = ((tempC*9)/5) + 32;
        //text(calculateTempF(volt));
        text(indentifySensor(addr1,addr2,addr3,addr4), 115 , 95);
        text("Temperature in F: " + String.format("%.1f", calculateTempF(volt)) + " F", 115,
135);
        }
}

}
//Function takes in the XBee address and returns the identity of the Xbee that sent the
temperature data
String indentifySensor(int a1, int a2, int a3, int a4) {
  int rout1[] = {64, 159, 115, 24}; //Arrays are the 32 bit address of the two XBees routers
  int end1[] = {64, 139, 174, 61};
  int end2[] = {64, 166, 42, 119};
  if(a1==rout1[0] && a2==rout1[1] && a3==rout1[2] && a4==rout1[3]) { //Check if Sensor 1
        return "Temperature from router - sensor 1"; } //temp data is from XBee one
  else if(a1==end1[0] && a2==end1[1] && a3==end1[2] && a4==end1[3]) {//Check if Sensor 2
        return "Temperature from end - sensor 2"; } //temp data is from XBee two
  else if(a1==end2[0] && a2==end2[1] && a3==end2[2] && a4==end2[3]) {//Check if Sensor 2
        return "Temperature from end - sensor 3"; } //temp data is from XBee two
  else { return "I don't know this sensor"; }  //Data is from an unknown XBee
}

//this function calculates temp in F from temp sensor
float calculateTempF(float v1) {
 float temp = 0;
 //calculate temp in C, .75 volts is 25 C. 10mV per degree
 if (v1 < .75) { temp = 25 - ((.75-v1)/.01); } //if below 25 C
 else if (v1 == .75) {temp = 25; }
 else { temp = 25 + ((v1 -.75)/.01); } //if above 25
 //convert to F
 temp =((temp*9)/5) + 32;
 return temp;
}

//This function takes an XBee analog pin reading and converts it to a voltage value
float calculateXBeeVolt(int analogMSB, int analogLSB) {
  int analogReading = analogLSB + (analogMSB * 256); //Turn the two bytes into an integer
value
  float volt = ((float)analogReading / 1023)*1.23; //Convert the analog value to a voltage value
  return volt;
}
```
(ForceTronics, 2014), (Pscmpf, 2008).

Business Development Plan

Abstract.

Wireless sensor networks based on Xbee technology consist of light-weight, small size sensor nodes, and low-energy consumption powered by solar panels. This networks have the ability to monitor, calculate and communicate data wirelessly. The key advantages for this mesh topology system is adaptability, scalability and affordability. The networks can be used in office environment to monitor temperature or over far distances to monitor weather and alarm about weather's anomalies.

What are Xbee modules and ZigBee protocol?

Xbee is a compatible radio module produced by Digi International. The first Xbee radios were introduced in 2005 by the brand MaxStream for point-to-point and star communications at over-the-air baud rates of 250 kbit/s.

ZigBee is a specification for a suite of high level communication protocols based on IEEE 802.15 standard which specifies the physical layer and media access control for low-rate wireless personal area networks.

Advantages of Zigbee versus Wi-Fi and Bluetooth

ZigBee is a low-cost, energy efficient, wireless mesh network standard. The inexpensive cost allows the technology to be widely applied in wireless control and monitoring networks. Energy efficiency allows for longer operation time with smaller solar batteries. Mesh networking provides reliability and extensive range.

Wi-Fi is whether reliable; however, it is incomparably more expensive in operation and also hardware needy than Zigbee.

Bluetooth which is significantly less expensive than Wi-Fi recently designed and introduced its mesh network for retail facilities; however, it falls short in range capabilities, up to 195 feet, and it is also more expensive than Zigbee.

Xbee technical capabilities

Platform	XBee® ZB	XBee-PRO® ZB	Programmable XBee-PRO® ZB
Performance			
RF Data Rate	250 Kbps		
Indoor/Urban Range	133 ft (40 m)	300 ft. (90 m)	
Outdoor/RF Line-of-Sight Range	400 ft (120 m)	2 miles (3200 m)/ Int'l 5000 ft (1500 m)	
Transmit Power	1.25 mW (+1 dBm) / 2 mW (+3 dBm) boost mode	63 mW (+18 dBm) / Int'l 10 mW (+10 dBm)	
Receiver Sensitivity (1% PER)	-96 dBm in boost mode	-102 dBm	
Features			
Adjustable Power	Yes		
I/O Interface	3.3V CMOS UART, ADC, DIO	3.3V CMOS UART, SPI, I2C, PWM, DIO, ADC	
Configuration Method	API or AT commands, local or over-the-air		
Frequency Band	2.4 GHz		
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)		
Serial Data Rate	1200 bps - 1 Mbps		
ADC Inputs	(4) 10-bit ADC inputs		
Digital I/O	10		
Antenna Options	Chip, Wire Whip, U.FL, RPSMA	PCB Embedded Antenna, Wire Whip, U.FL, RPSMA	
Operating Temperature & Humidity	-40° C to +85° C, 0-95% humidity non-condensing		
Programmability			
Memory	N/A		32 KB Flash / 2 KB RAM)
CPU/Clock Speed	N/A		HCS08 / Up to 50.33 MHz
Networking & Security			
Encryption	128-bit AES		
Reliable Packet Delivery	Retries/Acknowledgments		
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 channels	PAN ID, 64-bit IEEE MAC, 15 channels	
Power Requirements			
Supply Voltage	2.1 - 3.6VDC	2.7 - 3.6VDC	
Transmit Current	40 mA (@3.3 V)	205 mA	220 mA
Receive Current	38 mA / 40 mA boost mode @ 3.3VDC	47 mA	62 mA
Power-Down Current	<1 uA @ 25° C	3.5 uA @ 25° C	4 uA @ 25° C
Regulatory Approvals			
FCC (USA)	Yes		
IC (Canada)	Yes		
ETSI (Europe)	Yes		
C-TICK (Australia)	Yes		
TELEC (Japan)	Yes	Yes (int'l unit only)	

Figure 1 (Digi International Inc, n.d.)

Xbee mesh network deployment cost

The Xbees network costs varies based on the network size and desired performance.

Basic Xbee Wireless Kit’s price is $95.95 and it includes:

· 1 x XBee Shield

· 1 x XBee Explorer

· 2 x XBee Modules

· 2 x Stackable Headers 6-pin

· 2 x Stackable Headers 8-pin

Price of 1 Solar Cell Large - 2.5W is $34.95.

Price of 1 Polymer Lithium Ion Battery - 2000mAh is $12.95.

*Prices are based on www.sparkfun.com as of 04/26/2014.

Market research projection

Wireless Sensors and Transmitters market is growing worldwide. The below figures indicates there is a drastic change in last few years and these number increases in future (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

From 2003 to 2008: Market growth is huge in the last 5 years increased from 180 million dollars in 2005 to 380 million dollars approximately (200 % growth), (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

Future: Market forecasts reaches to 1800 million dollars by 2012 approximately (45% growth), (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

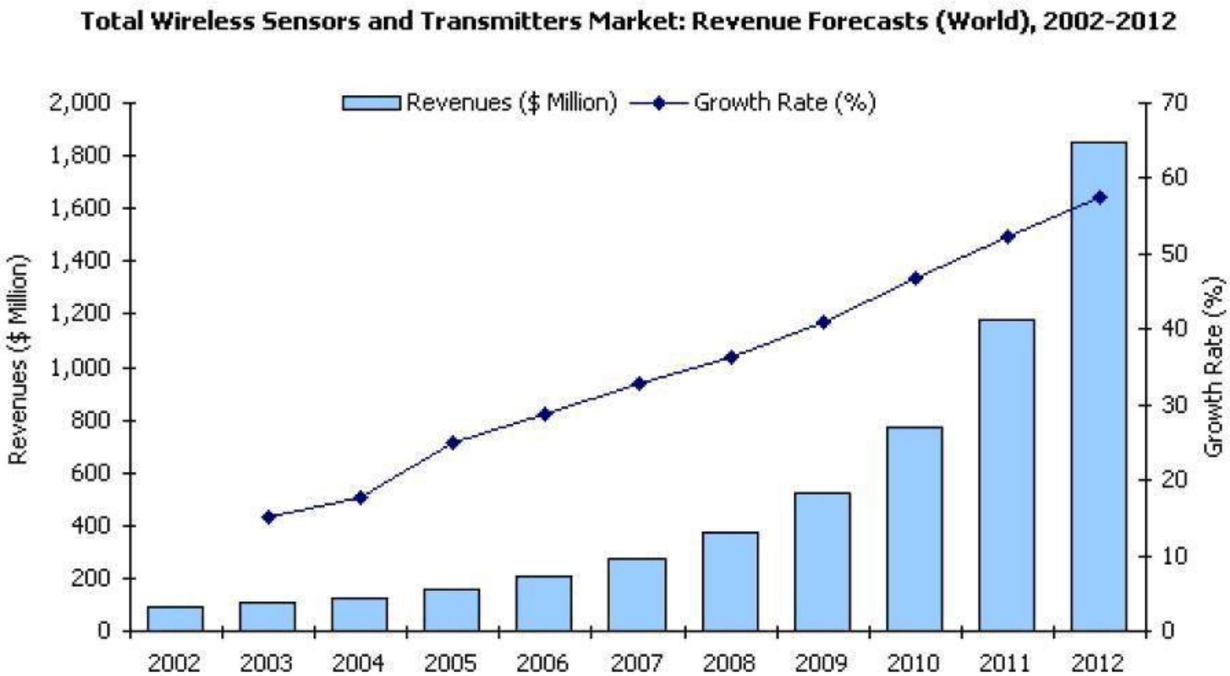


Figure 2 (Rajeshwari, Haritha, Garg, & Sateesh, 2009)

The below graph shows wireless sensors and Transmitters market in different regions North America, Europe, Asia pacific, and other countries of world (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

Major market growth in North America, Europe and Asia Pacific (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

From 2003 to 2008: In North America the market increased from 20 million dollars to 160 million dollars approximately, in Europe the market increased from 15 million dollars to 130 million dollars approximately, in Asia-Pacific the market increased from 8 million dollars to 120 million dollars approximately (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

Future: Market forecast reaches to 600 million dollars in North America, 500 million dollars in Europe and 400 million dollars in Asia pacific approximately (Rajeshwari, Haritha, Garg, & Sateesh, 2009).

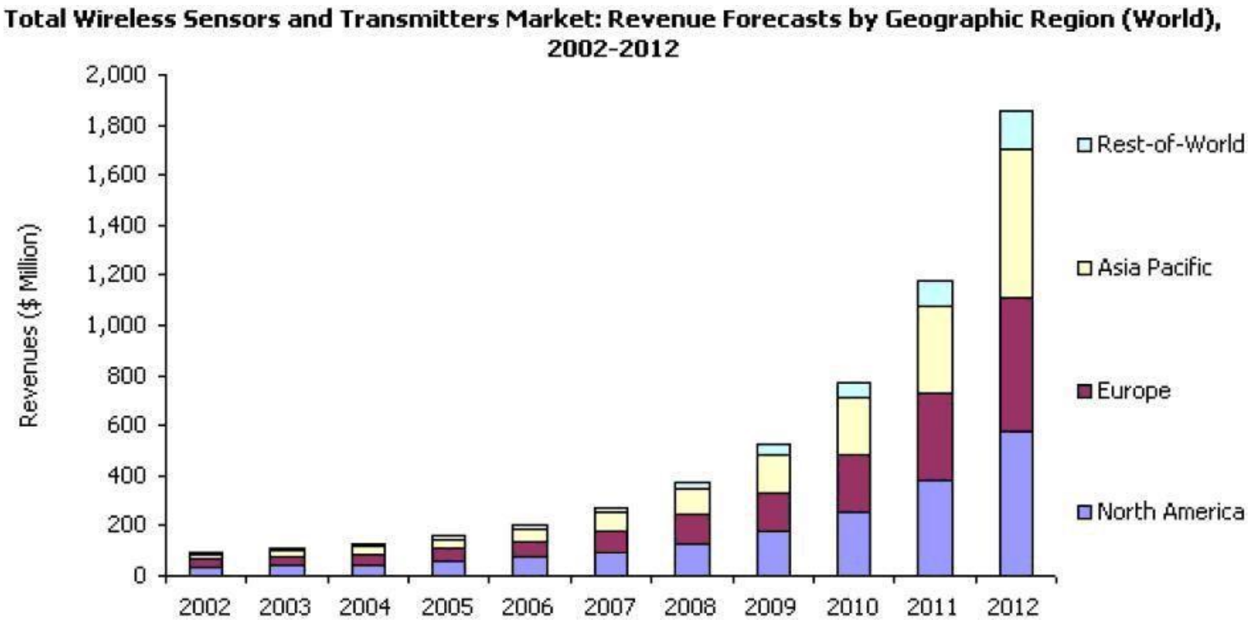


Figure 3 (Rajeshwari, Haritha, Garg, & Sateesh, 2009)

Wireless Building Controls Penetration Rate by Region, World Markets: 2012-2020

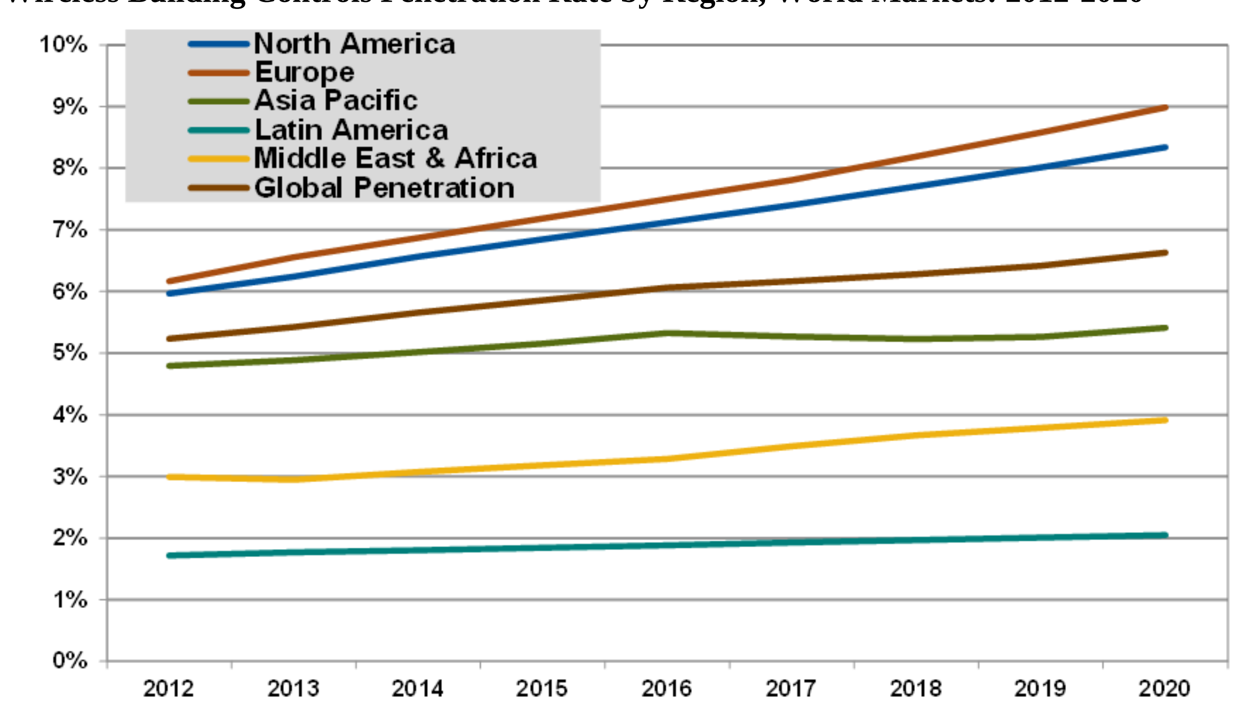


Figure 4 (Gohn, 2013)

Business Development Plan Summary

The strongest advantages of Xbee radio modules with ZigBee protocol are project adaptability, scalability, and deployment and operation cost efficient. The deployed mesh networks based on that technology includes: remote monitoring and control of wind turbines, monitoring cooking oil in restaurants, automating parking structure systems, monitoring residential wind turbines, monitoring farm silo levels, and measuring tank levels in a brewery.

Summary

Our solar powered mesh network is very efficient and cost friendly. There is little to no maintenance. We created an expandable model that can hold sensors to meet the needs of any individual. We used it for temperature but the design is versatile so virtually anything can be added including, Geiger counters, luminosity sensors, humidity sensors, anything! If you need to keep track of a large environment, then a mesh network made of Xbees is a great idea.

These sensors will be very beneficial in the future. We can integrate them with new technologies and make them do anything in our desire. One thing they can be used with is Google Glass. Google Glass has Android API and programming for it is like programming for Android. What we can do is to send information to the Google Glass and it will display all the info to the user. This is a very good idea because this way you won’t need a computer, you just monitor everything that happens on the go. This way, anyone who walks into a building can monitor whatever their network is designed to do. In the future, some things we can add to our sensors is cameras. We can use the cameras to monitor activity in the halls, so security officers with their google glass can have a constant visual on anything that is going on. Another very helpful task you can set the mesh Xbees to do is to monitor for extreme changes in environment. If there are sensitive areas that are prone to fires or earthquakes the Xbees can be used to detect the change and quickly notify the user. Solar powered mesh networks using Xbees is a very great idea and our future.

Works Cited

Digi International Inc. (n.d.). Digi. Retrieved from XBee® ZB: http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module#specs

ForceTronics. (2014, February 5). ForceTronics. Retrieved from Building a Wireless Temperature Sensor Network Part 2 : http://forcetronic.blogspot.com/2014/01/building-wireless-temperature-sensor_9.html

Gohn, B. (2013, April 17). Navigant Research. Retrieved from Wireless Building Controls Standards Emerge: http://www.navigantresearch.com/blog/wireless-building-controls-standards-emerge

Pscmpf. (2008, December 27). My Projects. Retrieved from Arduino LM35 Sensor: http://pscmpf.blogspot.com/2008/12/arduino-lm35-sensor.html

Rajeshwari, Haritha, Garg, & Sateesh. (2009, August 2009). The Dolcera Blog. Retrieved from Wireless Sensor Networks - A Market Research Report: http://blogs.dolcera.com/blog/2009/08/02/wireless-sensor-networks-a-report/