### Sensitive Buildings

Instructor: Rob Faludi

# Plan for Today

- Final Project Update
- Gateway Basics
- ConnectPort Overview
- iDigi Overview
- XBee Internet Gateway
- Readings & Assignments

## Final Project Update

## Final Project Suggested Timeline

- Week 8: Make observations, select your idea and finalize your group
- Week 9: Build a prototype and test it locally. Observe the results.
- Week 10: Revise prototype and test it on site. Observe the results.
- Week 11: Create initial pilot and test it locally. Observe the results.
- Week 12: Revise pilot into a final candidate and test it on site.
- Week 13: Add robustness to your final installation. Create a presentation and demo that tells its story well
- Week 14: Final Presentations

## Gateway Basics

## Types of Gateways

- Bridging
- Routing
- Transformation
  - aggregation
  - filtration
  - applications

### Protocols

- Ethernet
- WiFi
- Bluetooth
- GSM
- Twitter
- SQL
- Mail

- FTP
- SMS
- Telephone
- Chat
- Speech
- MIDI
- everything else!

## Simple Serial Methods





### Computer as Gateway



```
# select (r.w.e) returns a tupple of the sockets that are actually readable, writeable
rlist, wlist, xlist = select(rlist, wlist, [])
if sd in rlist:
    try:
        # Receive from the socket:
        print "receiving data"
        payload, src addr = sd.recvfrom(72)
       print 'Source: ' + src addr[0] +' sent: ' + payload
    except Exception, e:
        print '* receive failed *'
       print e
if sd in wlist:
    if (time.clock() - lastReguest > reguestInterval):
        lastRequest = time.clock()
        try:
            # Send to the socket:
            print "sending request".
            print requestString
            count = sd.sendto(requestString, 0, (monitor_addr, 0xe8, 0, 0x11))
                ## Slice off count bytes from the buffer,
                ## useful for if this was a partial write:
                # payload = payload[count:]
        except Exception, e: #general exception handler
            print '* send failed *'
            print type(e)
            print e
```

import java.io.\*; // this is the input/output library needed for data streams
import java.net.\*; // this is the network library needed for sockets

String host; int port; Socket mySocket; DataInputStream myInputStream; DataOutputStream myOutputStream; byte myDataIn, myDataOut;

// declare Socket

// declare data input stream. This will run within a socket, bringing data into Java
// declare data output stream. This will run within a socket, sending data out from Java
// declare some variables to store the data we're sending and receiving

### **Dedicated Gateways**

- lower power use
- always on
- cheaper,
- smaller,
- more stable,
- sometimes...

## Hacked



## Manufactured



## Gateway Examples



## BlueWay System Diagram





I spent about 10 minutes on this diagram ... can you tell?





### **ConnectPort Basics**



### **ConnectPort X2 Configuration and Management**



#### Home

### Configuration

- Network XBee Network
- System
- Remote Management
- Security

### Applications

Python

#### Management

Connections Event Logging

### Administration

File Management Backup/Restore Update Firmware Factory Default Settings System Information Reboot

### Logout

lome	
Getting Started	
Tutorial Not sure wh	nat to do next? This Tutorial can help.
System Summary	
Model:	ConnectPort X2
Ethernet MAC Address:	00:40:9D:38:05:71
Ethernet IP Address:	10.0.1.100
Description:	None
Contact:	None
Location:	None
Device ID:	0000000-0000000-00409DFF-FF380571





#### Home

### Configuration

- Network XBee Network
- System
- Remote Management
- Security

### Applications

Python

### Management

Connections Event Logging

### Administration

- File Management Backup/Restore Update Firmware Factory Default Settings System Information Reboot
- Logout

			ricip
Network Configurat	tion		
▼ Ethernet IP Settings			
Obtain an IP address a	utomatically using	DHCP *	
Use the following IP ad	dress:		
* IP Address:	10.0.1.100		
* Subnet Mask:	255.255.255.0		
Default Gateway:	10.0.1.1		
<ul> <li>Enable AutoIP address</li> <li>* Changes to DHCP, IP add</li> </ul>	assignment ress, and Subnet M	ask may effect your browser connection.	
Apply			_
Network Services Settings	5		
Advanced Network Setting	gs		

### **XBee Configuration**

### Network View of the XBee Devices

Node ID	Network Address	Extended Address	Node Type	Product Type
	[fffe]!	00:13:a2:00:40:31:7c:80!	router	
	[fffe]!	00:13:a2:00:40:31:f9:f5!	router	
	[51e9]!	00:13:a2:00:40:30:d0:22!	router	Unspecified
GORDIE	[d21c]!	00:13:a2:00:40:30:cf:e3!	router	Unspecified
QUIET	[7b76]!	00:13:a2:00:40:30:d0:0e!	router	Unspecified
RECEPTION	[f43e]!	00:13:a2:00:40:30:cf:dc!	router	Unspecified
ROB	[fffe]!	00:13:a2:00:40:31:f9:ee!	router	Unspecified
ZIG Coordinator	[0000]!	00:13:a2:00:40:54:ae:03!	coordinator	X2 Gateway
Clear list before	e performing refresh			
Refresh				
Firmware Update				

#### Configuration

	riles	
Jpload Fi	les	
Upload Py	thon programs	
Upload Fi	le: Choose File	no file selected
Upload		
Manago P	iloc	
Action	File Name	Size
	zigbee.py	1147 bytes
	python.zip	129910 bytes
	xig.py	3802 bytes
	url_libs.zip	47321 bytes
	base64.py	11261 bytes
	mains abundant mut	17638 bytes
	mimetypes.py	
	email.zip	155588 bytes
	email.zip quopri.py	155588 bytes 6969 bytes

### **Python Configuration**

- Python Files
- ▼ Auto-start Settings

Specify python programs to be run when the device boots.

### Enable Auto-start command line (specify program filename to execute and any arguments)

1	

X	В	e	e	(	2	0	n	f	ic	1	I	r	a	t	1	0	ī	1
		-			_	-					-							

Extended Address: 00:13:a2:00:40:30:cf:dc! Product Type: Unspecified Firmware Version: 0x2241

### Basic Settings

Extended PAN ID (ID):	0x0000000000aaaa 8 hex bytes
	Setting to 0 allows a random extended PAN ID to be used.
	Note: Changing the PAN ID may make this node inaccessible.
Node Identifier (NI):	RECEPTION
Discover Timeout (NT):	60 tenths of second (1-255)
Scan Channels (SC):	0x1ffe hex (0xffff=all channels)
Scan Duration (SD):	3 (0-7)
dvanced Radio Settings	
ransmit Power Level (PL):	Maximum (4)
Allows Join Time (NJ):	255 seconds (0-255. 255=always)
Broadcast Hops (BH):	0 (0-30, 0=maximum)
RSSI PWM (P0):	Enable RSSI PWM
RSSI Timer (RP):	40 tenths of second (0-255)
Associate LED (D5):	LED Blinks When Associated
erial Interface Settings	
Baud Rate (BD):	9600

```
0 0
```

```
Trying 128.122.151.101...
Connected to zigbeegate.itp.tsoa.nyu.edu.
Escape character is '^]'.
login: root
password:
#> python
>>> import zigbee
>>> nodes = zigbee.getnodelist()
>>> for node in nodes:
      print "%12s %12s %8s %12s" % (node.label, node.type, node.addr_short, node
. . .
.addr_extended)
. . .
                   router [d21c]] [00:13:02:00:40:30:cf:e3]]
      GORDITE
```

	GONDIE	router	Lactel.	Looi toi dei o	of for solution of the solutio	
F	RECEPTION	router	[f43e]!	[00:13:a2:0	0:40:30:cf:dc]!	
	ROB	router	[fffe]!	[00:13:a2:0	0:40:31:f9:ee]!	
		router	[51e9]!	[00:13:a2:0	0:40:30:d0:22]!	
		router	[fffe]!	[00:13:a2:0	0:40:31:7c:80]!	
	QUIET	router	[7b76]!	[00:13:a2:0	0:40:30:d0:0e]!	
ZIG	Coordinator	coordinat	or [0000	0]! [00:13:a	Z:00:40:54:ae:03]	]!
>>>						
~~~						

Exploring a Mesh Network

### XBee ZigBee Node Indicators

- ATNI Node Indicator
- ATND Node Discovery
- ATDN Destination Node

- Also:
  - ATDB signal strength in DBm
  - AT%V Voltage

## Transmitting Data

• Read a list of all nodes on the network using ATND

MY<CR> SH<CR> SL<CR> NI<CR> (Variable length) PARENT\_NETWORK ADDRESS (2 Bytes)<CR> DEVICE\_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS<CR> (1 Byte: Reserved) PROFILE\_ID<CR> (2 Bytes) MANUFACTURER\_ID<CR> (2 Bytes) <CR>

• Set the Destination Node using ATDN

### Internet Interlude

- IP addresses
- ports
- sockets
- layers physical, transport, application
- telnet demo

### Telnet

- into XIG
- ssh into itp
- daytime
- web server
- mail server

### ConnectPort via Telnet

## Command Line

- telnet
- port 23 is default
- hostname or IP address needed

• telnet xig.faludi.com 25

## ConnectPort via HTTP

### HTTP

- web access
- port 80 is default

• <u>http://xig.faludi.com</u>

## ConnectPort via iDigi

# iDigi

- web access
- developer.idigi.com
- user itp pass <generic>

• http://developer.idigi.com

XBee Internet Gateway

To use the gateway you need to:

Switch the PAN to AAAA: ATID AAAA

Set your radio to 115200 baud: ATBD 7 (optional)

Set the destination address to zero: ATDH0 and ATDL0

When you're ready to go, attach an XBee to an Arduino's serial port, then send a URL and you'll get back the response. For example to send your request from Arduino:

Serial.println("<u>http://www.faludi.com/test.html</u>");

And to read the response back:

if (Serial.available()) {

```
char inChar = Serial.read();
```

```
print ( inChar );
```

}

Some useful things to know:

• currently supported URL formats (items in [] are optional):

http://host/path[:port]

https://host/path[:port]

ftp://[username:password@]host/path[:port]

- sending help will get the current help file from the gateway
- baud rates lower than 115200 will work if the results you're getting are brief
- the software is still under development so bug reports are useful

### COMMANDS:

help or xig://help: displays the help file quit or xig://quit: quits program abort or xig://abort: aborts the current session time or xig://time: prints the time in ISO format

http://host/path: retrieves a URL https://host/path: retrieves a secure URL http://user:pass@host/path: retrieves a URL using username and password https://user:pass@host/path: retrieves via SSL using username and password udp://host:port: initiate UDP session to remote server and port number (note: session will end only by using xig://abort)

iomode is accessed by sending raw I/O packets to the gateway push mode is accessed by sending remote AT for a radio from iDigi XBee Internet Gateway Demo

## Send a request

Serial.println("http://itp.nyu.edu/~raf275/testpage.html");

### Seek a character

if (Serial.available() > 0) {

if (Serial.read() == 'A') {

//do something

}

}

### Send a value

Serial.println("<u>http://faludi.com/testpage.php</u>?value=137");

### Read an ASCII decimal value

```
if (Serial.available() >= 3) {
```

}

```
position1 = Serial.read() - 48;
```

```
position2 = Serial.read() - 48;
```

```
postion3 = Serial.read() - 48;
```

```
value = position1 *100 + position2 * 10 + position3
```

```
// using a buffer would be more sophisticated
```

### Read a phrase

```
char buffer[128], result[128];
int count = 0;
```

```
if (Serial.available() > 0) {
```

}

```
buffer[count] = Serial.read();
count++;
if (buffer[count] == '\r') {
    strcopy(result, buffer);
    count = 0;
```

// additional code would be added to make this work well

## Readings and Assignments

- Readings
  - Building Wireless Sensor Networks, Chapter 7
  - ThinkCSpy:

How to Think Like a Computer Scientist, Learning with Python <a href="http://openbookproject.net/thinkCSpy">http://openbookproject.net/thinkCSpy</a>

- Assignments
  - Final Project Prototypes