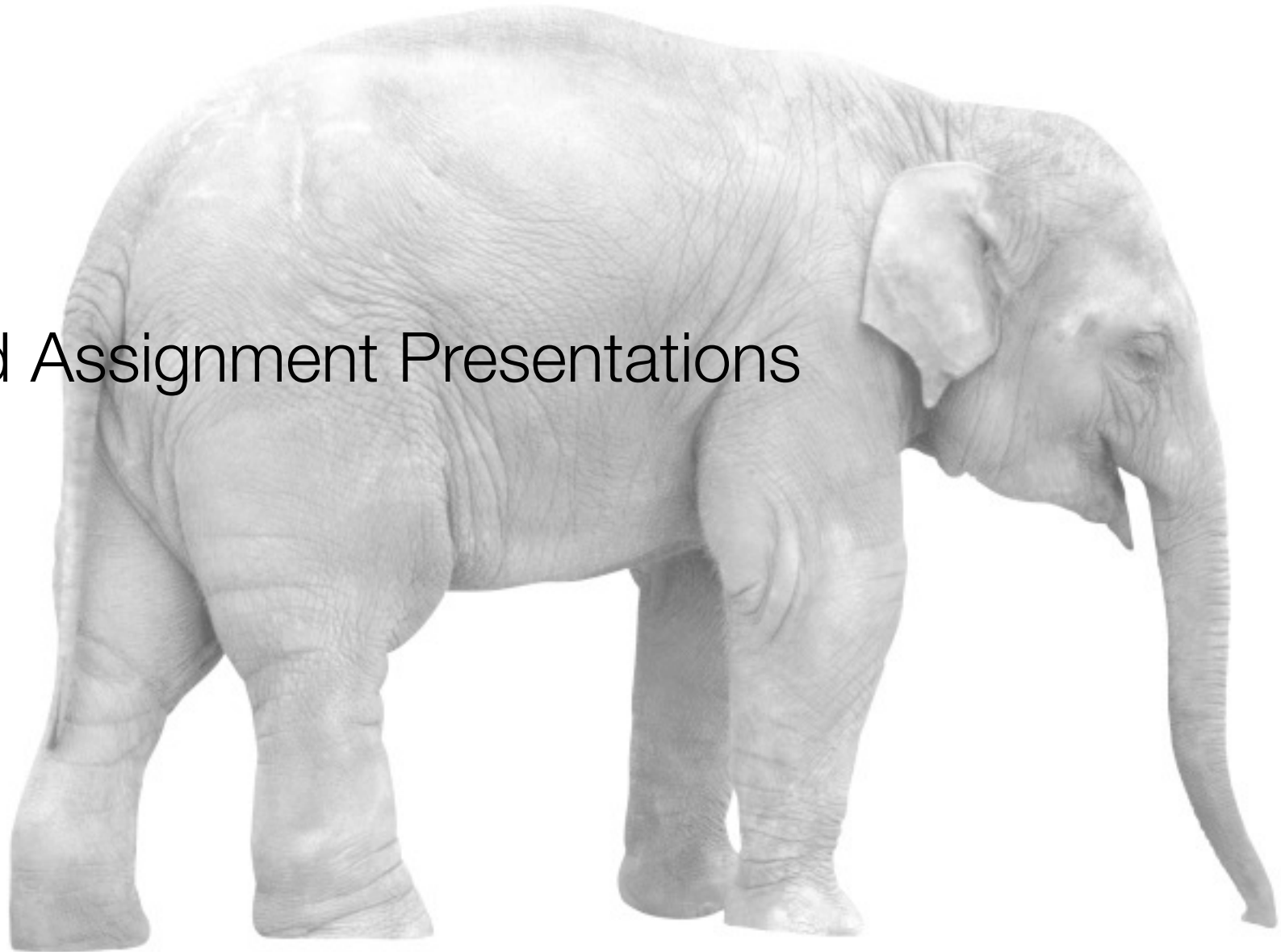# Sociable Objects Workshop

Instructor: Rob Faludi

Plan for Today

- Card Assignment Presentations

- Gateway Basics

- ConnectPort Overview

- ConnectPort Demo

- Readings & Assignments

Card Assignment Presentations
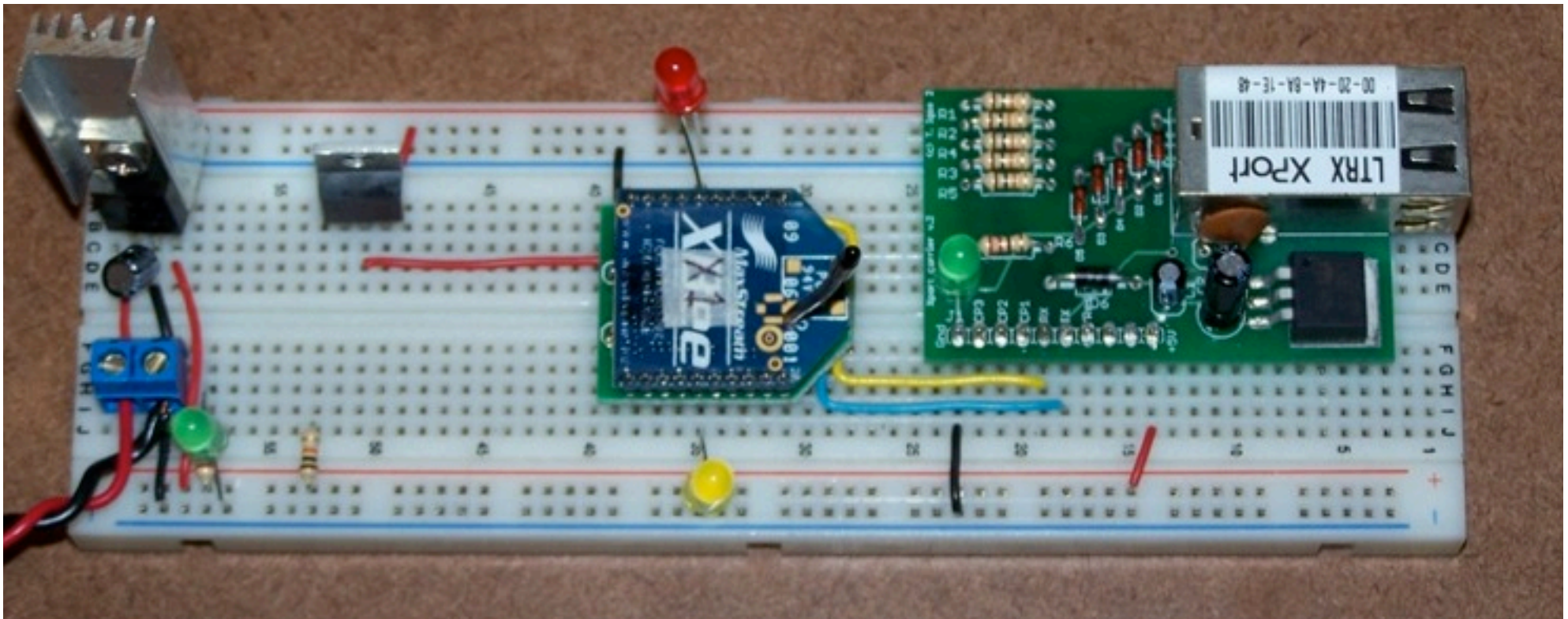
Gateway Basics

Types of Gateways

- Bridging

- Routing

- Transformation

    - aggregation
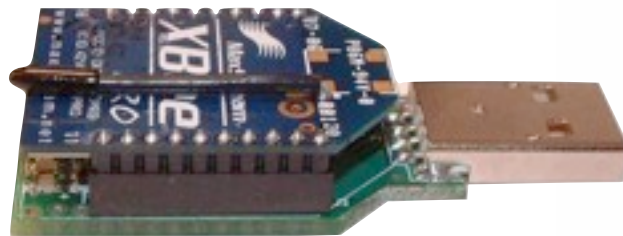
    - filtration

    - applications

Protocols

- Ethernet

- WiFi

- Bluetooth

- GSM

- Twitter

- SQL

- Mail

- FTP

- SMS

- Telephone

- Chat

- Speech

- MIDI

- everything else!

Simple Serial Methods

Computer as Gateway

```python
# 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() - lastRequest > requestInterval):
        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
```

```java
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;                        // declare Socket
DataInputStream myInputStream;          // declare data input stream. This will run within a socket, bringing data into Java
DataOutputStream myOutputStream;        // declare data output stream. This will run within a socket, sending data out from Java
byte myDataIn, myDataOut;               // declare some variables to store the data we're sending and receiving

void setup(){
```

Dedicated Gateways

- lower power use

- always on

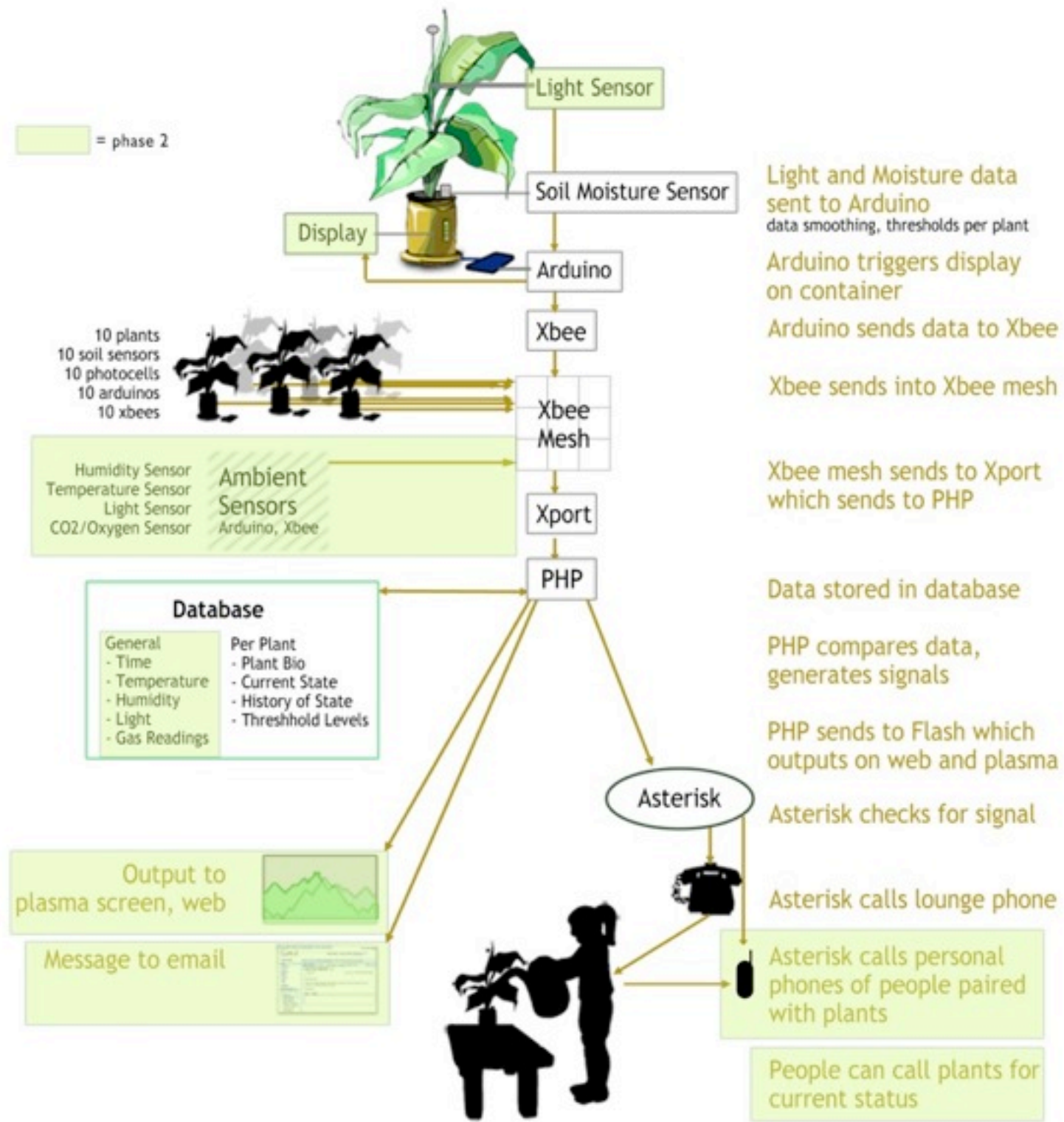- cheaper,

- smaller,
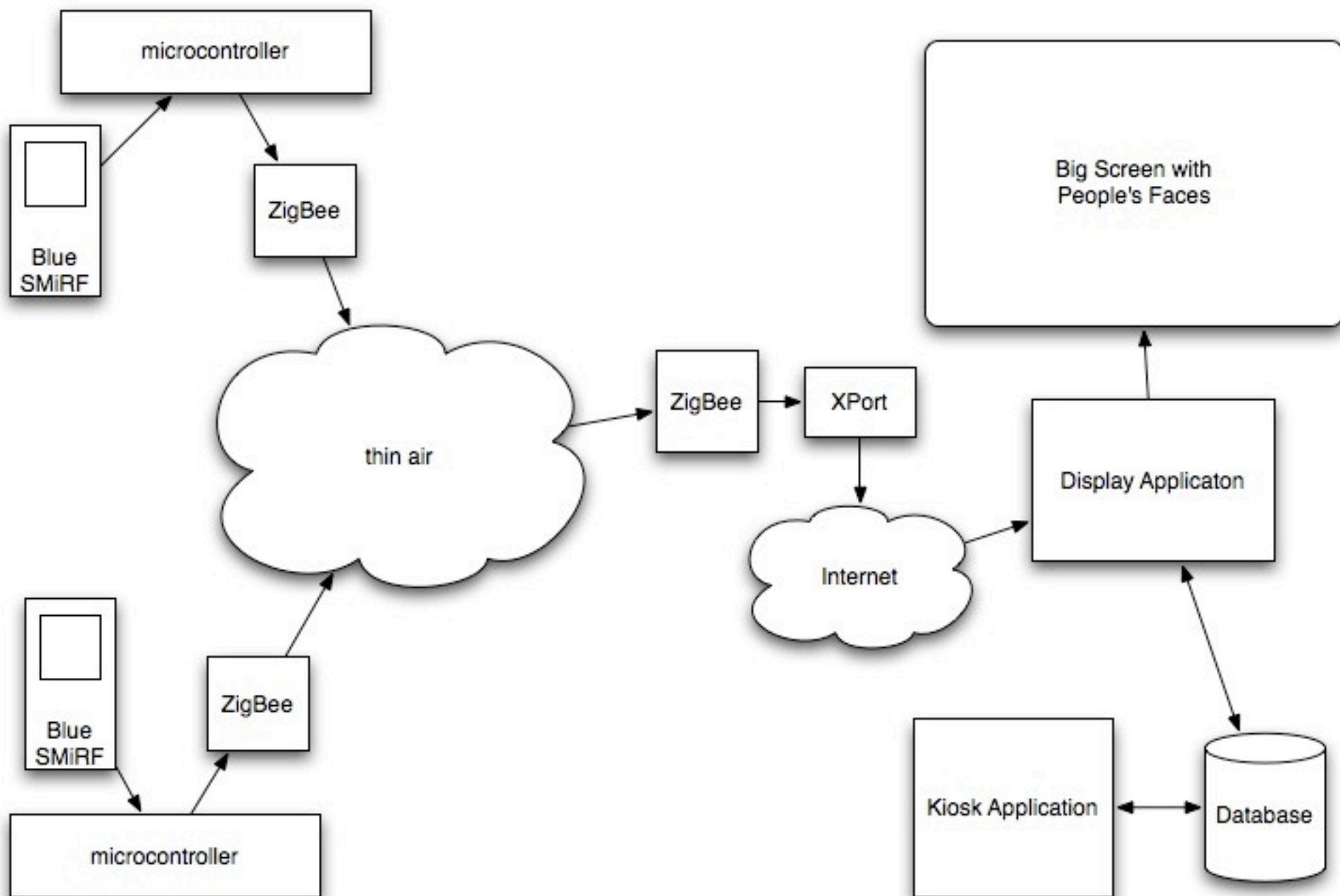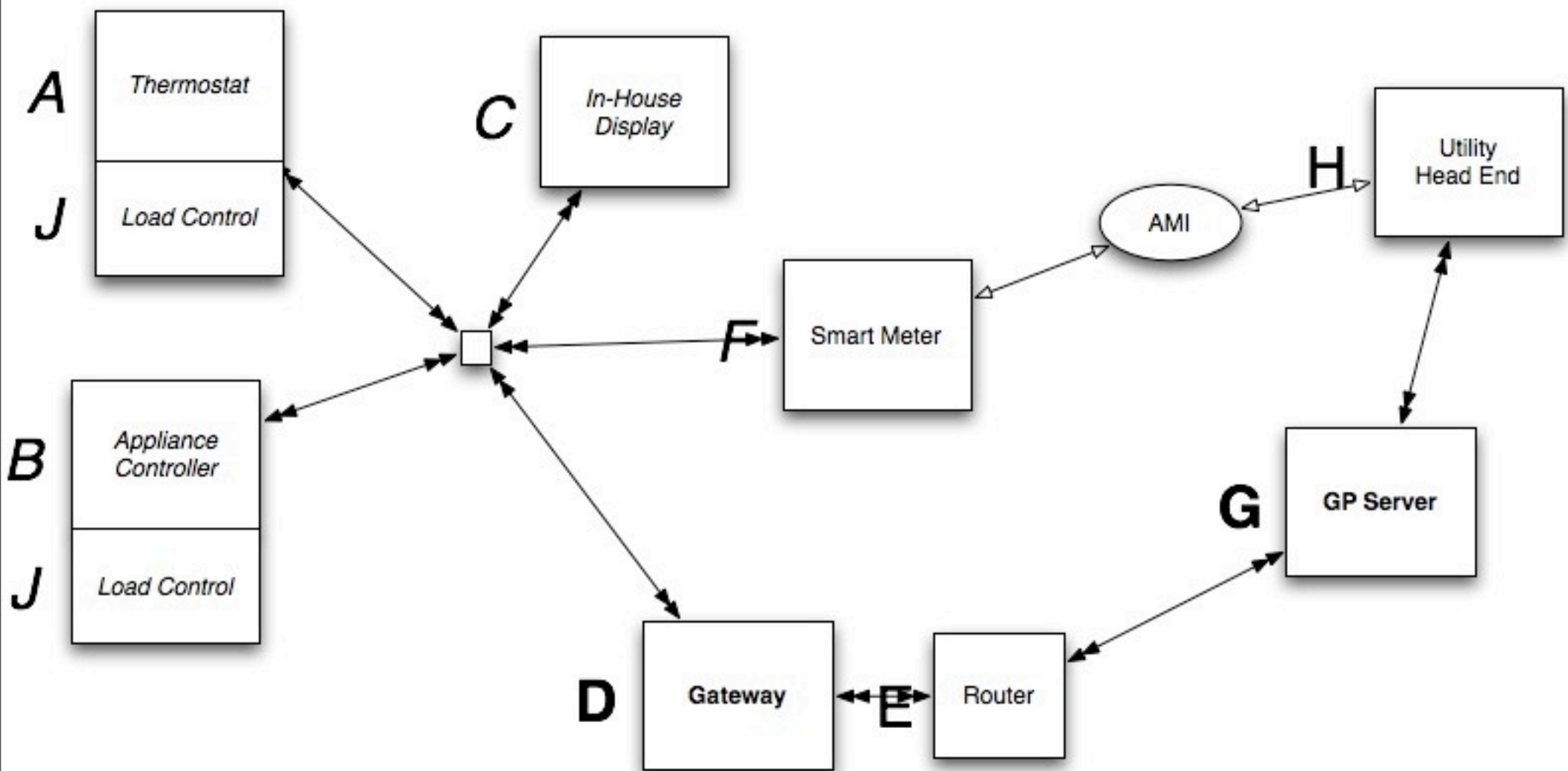
- more stable,

- sometimes...

Hacked

Manufactured

Gateway Examples

= phase 2

Light Sensor

Soil Moisture Sensor

Display

Arduino

Xbee

10 plants
10 soil sensors
10 photocells
10 arduinos
10 xbees

Xbee Mesh

Humidity Sensor
Temperature Sensor
Light Sensor
CO2/Oxygen Sensor

Ambient Sensors
Arduino, Xbee

Xport

PHP

Database

General
- Time
- Temperature
- Humidity
- Light
- Gas Readings

Per Plant
- Plant Bio
- Current State
- History of State
- Threshhold Levels

Asterisk

Output to plasma screen, web

Message to email

Asterisk calls personal phones of people paired with plants

People can call plants for current status

Light and Moisture data sent to Arduino
data smoothing, thresholds per plant

Arduino triggers display on container

Arduino sends data to Xbee

Xbee sends into Xbee mesh

Xbee mesh sends to Xport which sends to PHP

Data stored in database

PHP compares data, generates signals

PHP sends to Flash which outputs on web and plasma

Asterisk checks for signal

Asterisk calls lounge phone

BlueWay System Diagram

A — Thermostat

J — Load Control

B — Appliance Controller

J — Load Control

C — In-House Display

F — Smart Meter

AMI

H — Utility Head End

G — GP Server

D — Gateway

E — Router

XBees inside Kill-a-Watt
transmit power usage

Kitchen
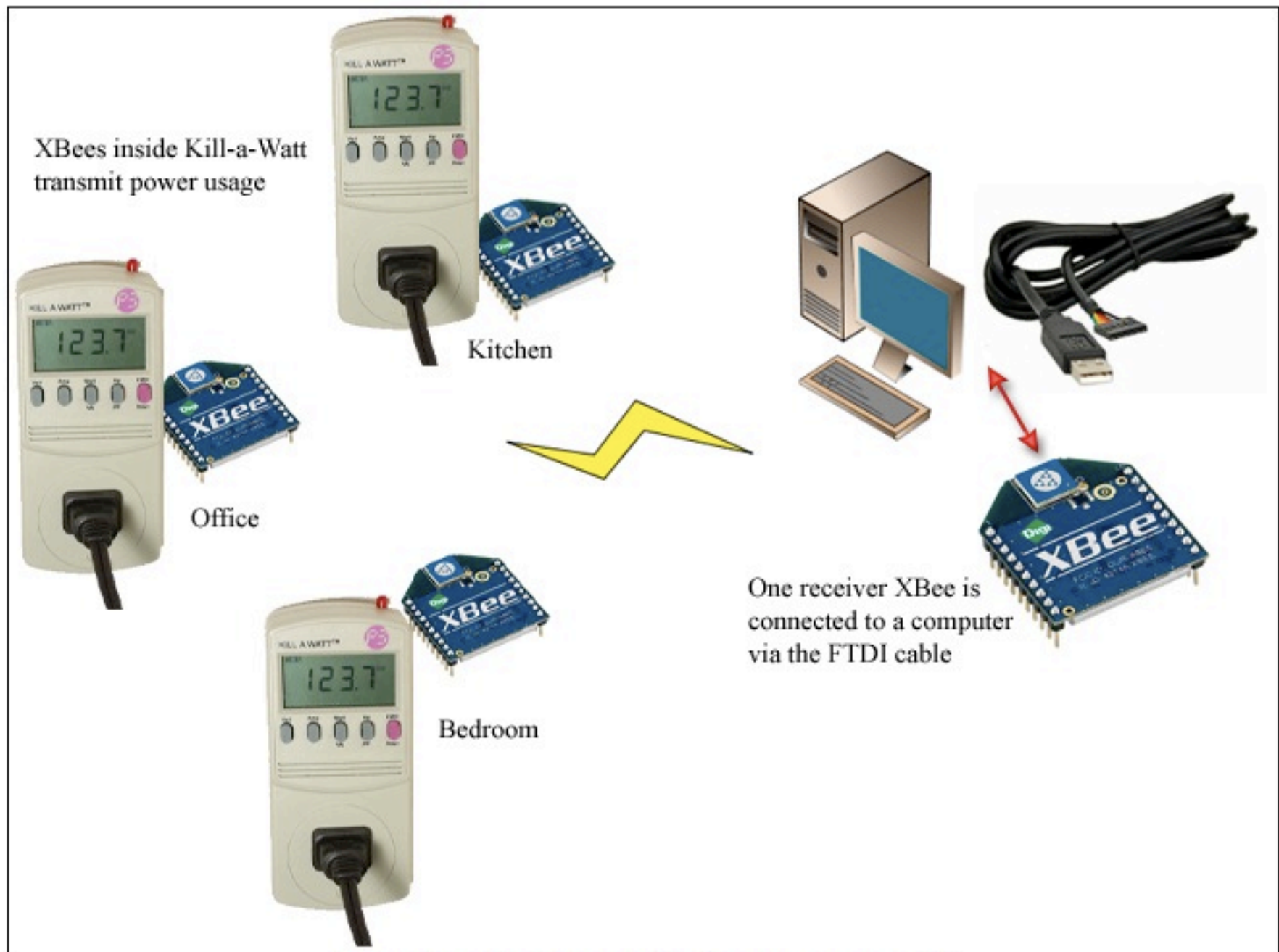
Office

Bedroom

One receiver XBee is
connected to a computer
via the FTDI cable

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

Home

Getting Started

Tutorial   Not sure what 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:	00000000-00000000-00409DFF-FF380571

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

Network Configuration

▼ **Ethernet IP Settings**

○ Obtain an IP address automatically using DHCP *

◉ Use the following IP address:

* IP Address: `10.0.1.100`

* Subnet Mask: `255.255.255.0`

Default Gateway: `10.0.1.1`

☑ Enable AutoIP address assignment

* Changes to DHCP, IP address, and Subnet Mask may effect your browser connection.

(Apply)

▶ Network Services Settings

▶ Advanced Network Settings

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 performing refresh

(Refresh)

▶ Firmware Update

Python Configuration

▼ **Python Files**

Upload Files

Upload Python programs

Upload File: [Choose File] no file selected

[Upload]

Manage Files

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
☐	mimetypes.py	17638 bytes
☐	email.zip	155588 bytes
☐	quopri.py	6969 bytes
☐	ftplib.py	26935 bytes

[Delete]

▶ Auto-start Settings

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)*
☑	xig.py
☐	
☐	
☐	

[Apply] [Cancel]

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

▼ Basic Settings

Basic Radio Settings

Extended PAN ID (ID): `0x000000000000aaaa` 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)

Advanced Radio Settings

Transmit 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 ▼`

Serial Interface Settings

Baud Rate (BD): `9600 ▼`

```
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)
...
      GORDIE       router   [d21c]! [00:13:a2:00:40:30:cf:e3]!
   RECEPTION       router   [f43e]! [00:13:a2:00:40:30:cf:dc]!
         ROB       router   [fffe]! [00:13:a2:00:40:31:f9:ee]!
                   router   [51e9]! [00:13:a2:00:40:30:d0:22]!
                   router   [fffe]! [00:13:a2:00:40:31:7c:80]!
       QUIET       router   [7b76]! [00:13:a2:00:40:30:d0:0e]!
ZIG Coordinator  coordinator  [0000]! [00:13:a2:00:40:54:ae:03]!
>>>
>>> █
```

Readings and Assignments

- ~~Readings~~ Watch a Movie!

    - Fast, Cheap and Out of Control

        - in the ER, on iPhone or Netflix

- Assignments

    - Sensor/Actuator Project

    - Final Project Proposals