WaveForum 2012
SOLUTIONS · EXPERTS · ANSWERS

XBee Internet Gateway

The easy path for the Internet of Things

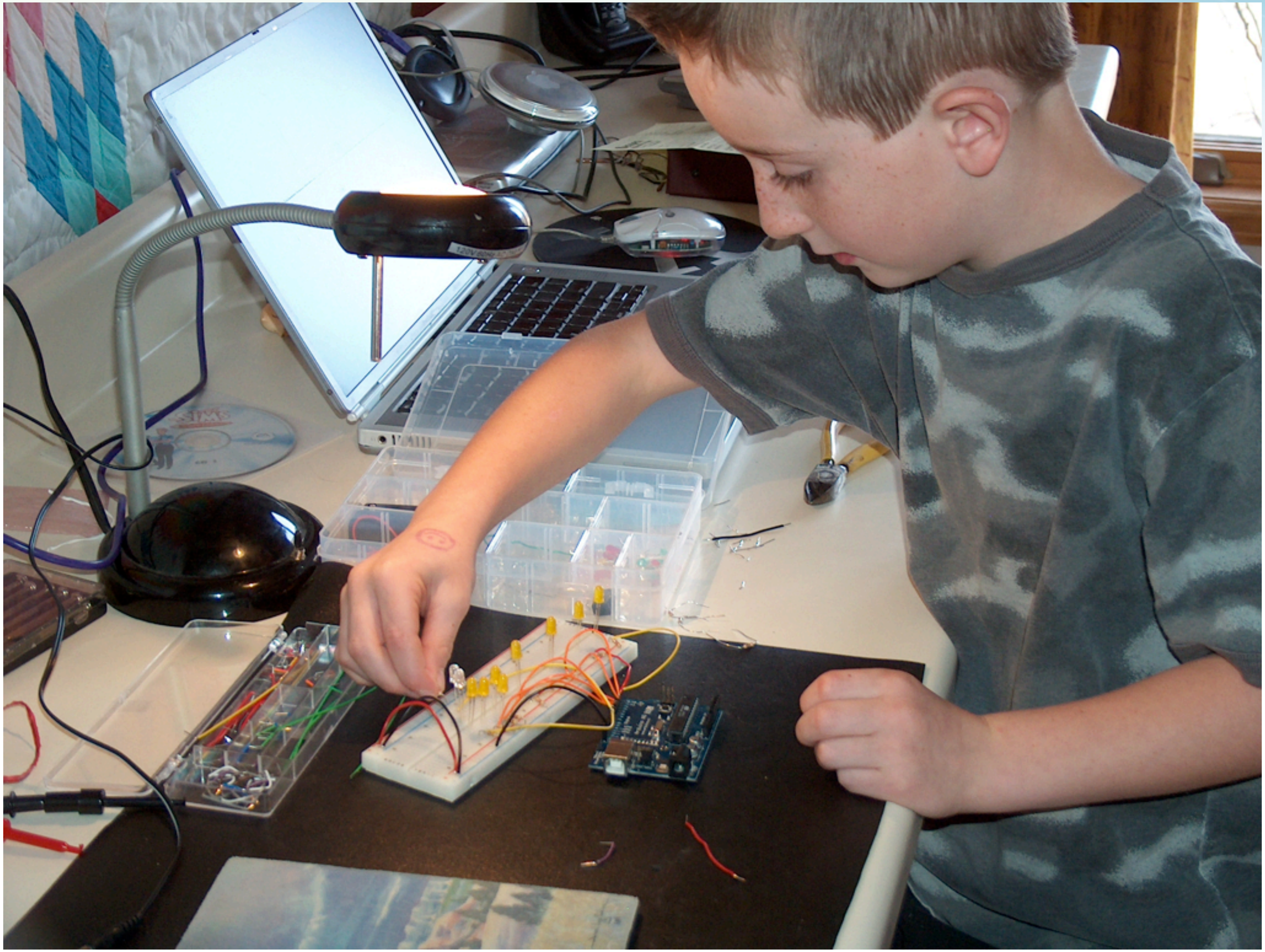
Introductions

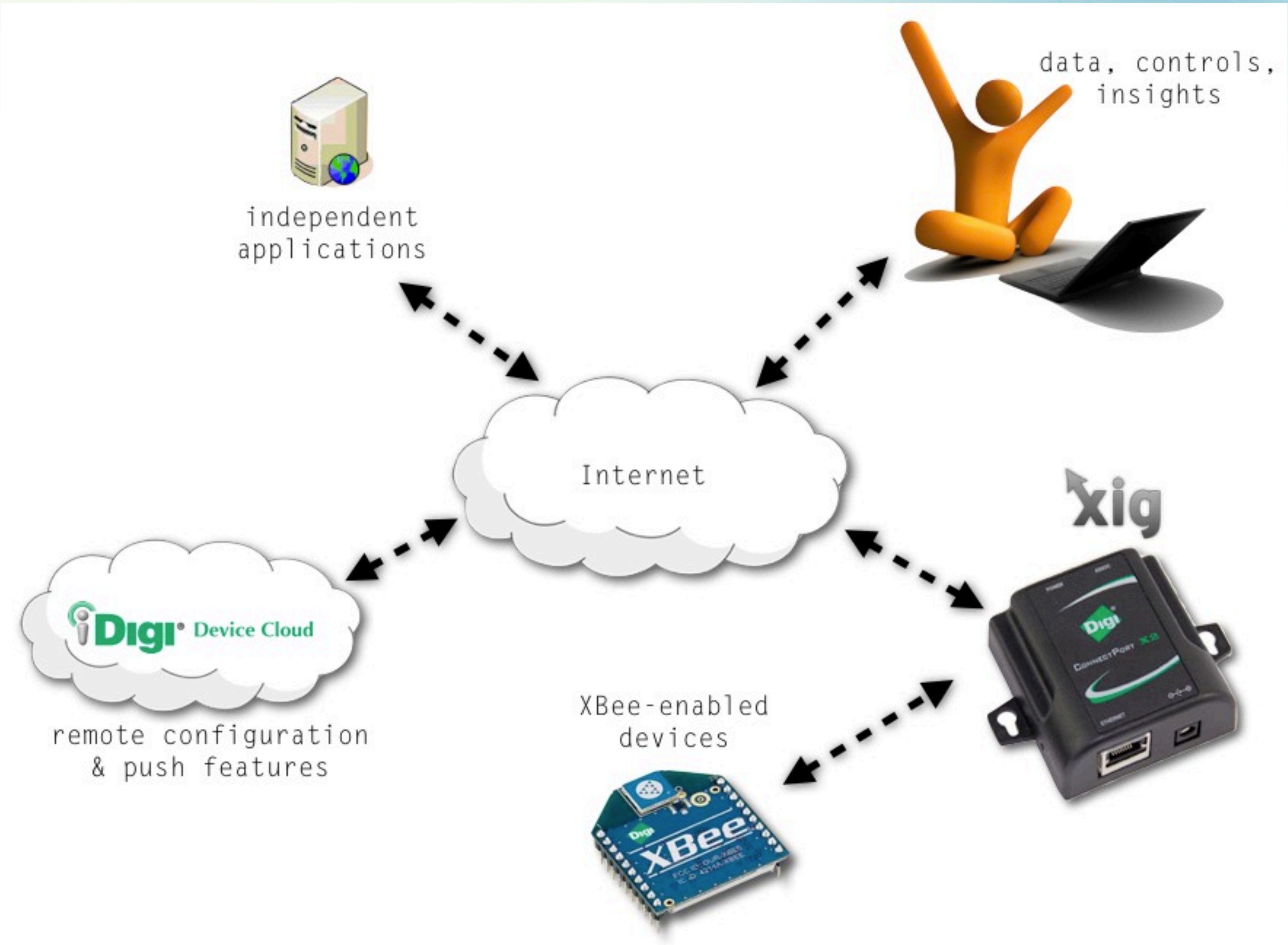
- Rob Faludi
- Collaborative Strategy Leader, R&D
- Author: *Building Wireless Sensor Networks*
- NYU ITP Professor
- SVA MFA Professor, interaction design
- Botanicalls

What’s the XIG?

- Easy serial Internet communication for devices
- Web browser for objects
- Initially developed for transparent wireless connection to Internet for prototypers
- Uses Digi ConnectPort hardware, XBee radios

1607-0
MaxStream
XBee
Series 2
www.maxstream.net





independent
applications

data, controls,
insights

Internet

remote configuration
& push features

XBee-enabled
devices

Before XIG

Send:

```
GET /path/file.html HTTP/1.1
Host: www.host1.com:80
User-Agent: HTTPTool/1.0
<CR><LF>
```

Receive:

```
HTTP/1.0 200 OK
Date: Tue, 8 Jun 2010 12:01:35 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
<h1>Happy Birthday Tim!</h1>
<p>Thanks for inventing all this stuff...
</body>
</html>
```

After XIG

Send:
http://twansform.appspot.com/usweekly/text/1

Receive:
“*Lindsay Lohan cried as she arrived to jail, but the inmates cheered her on*”

Reasons for XIG

- simple
- wireless
- shared connection
- single radio: inexpensive
- low power
- endlessly flexible

Short History

- Me
- ITP
- Ted Hayes
- Building Wireless Sensor Networks
- Jordan
- Digi

Why open-source?

- More brains to create better code
- More people involved means more ideas
- Open source gets widest adoption
- Security, transparency, community

Ted Hayes

Jordan Husney

Thanks

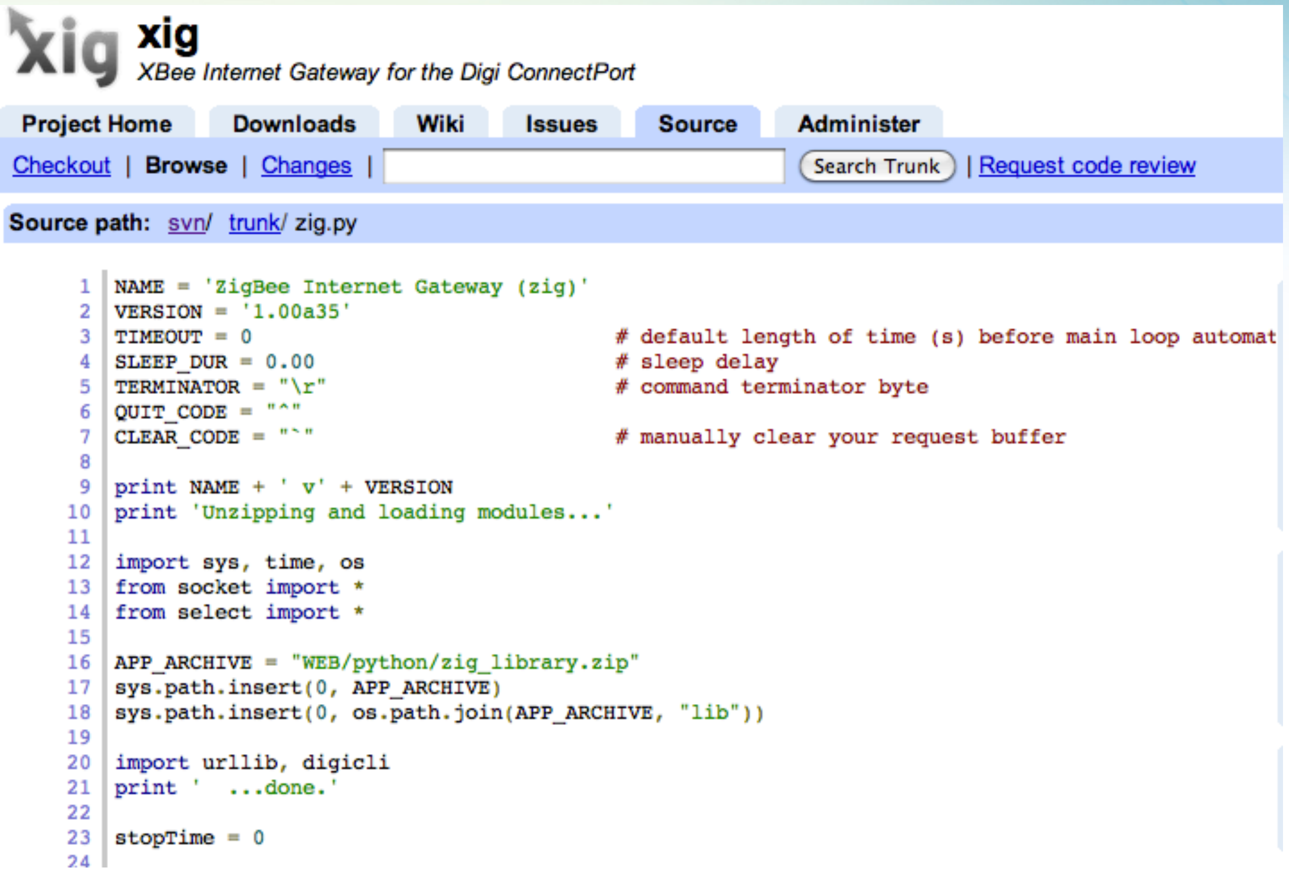
Basics

HOW TO XIG

The Setup

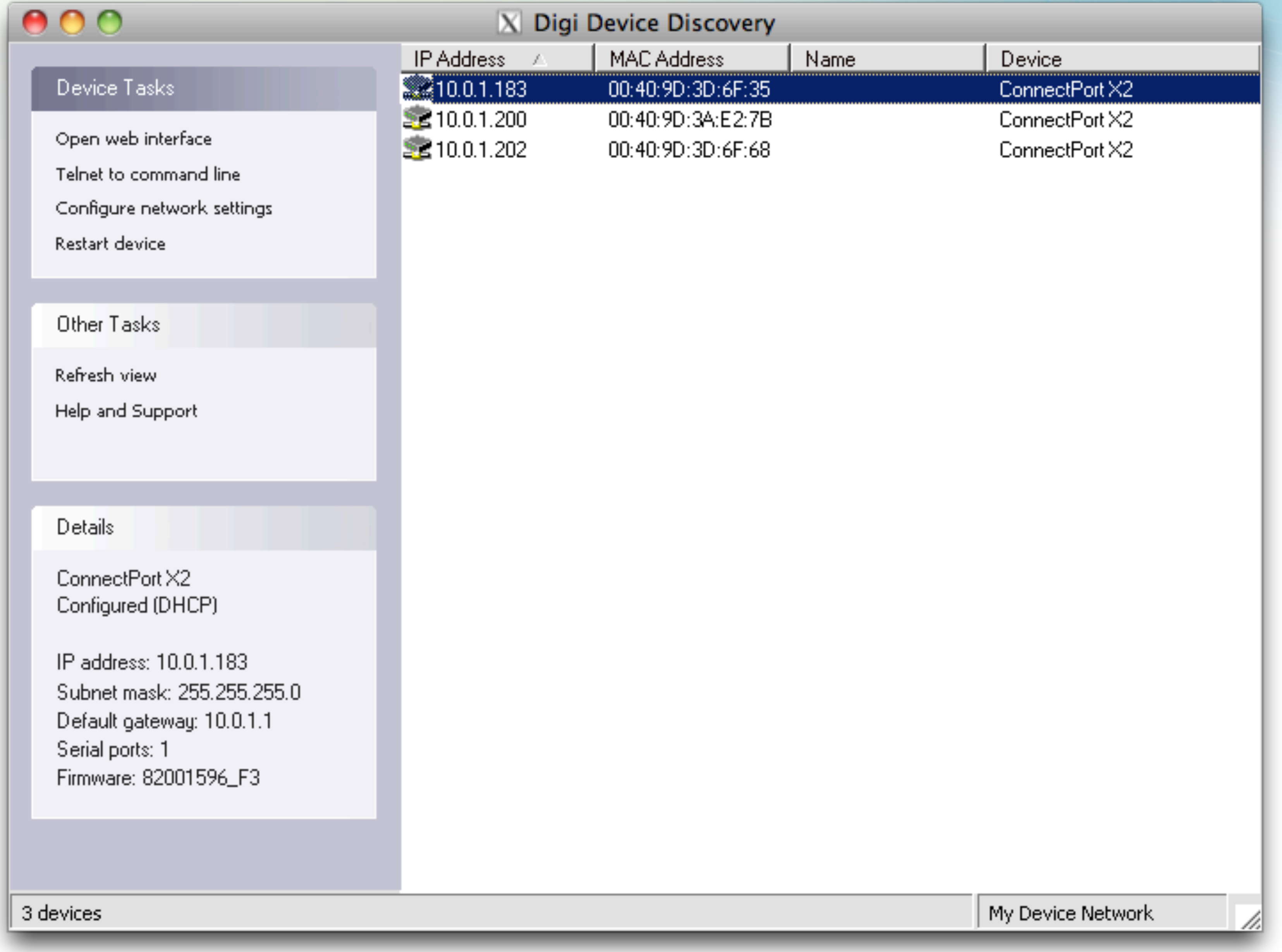
ConnectPort WAN
Digi
SIGNAL STRENGTH
RESET
STATUS

CABLE 4-PAIR 24AWG E132226-A (UL) CMG 75
CONNECTIVITY SYSTEM ETL VERIFIED TIA/EIA-568
CATEGORY 6 UTP PATCH CABLE 4-PAIR


xig
xig
XBee Internet Gateway for the Digi ConnectPort
Project Home
Downloads
Wiki
Issues
Source
Administer
Checkout | Browse | Changes |
Search Trunk
| Request code review
Source path: svn/ trunk/ zig.py

```
NAME = 'ZigBee Internet Gateway (zig)'
VERSION = '1.00a35'
TIMEOUT = 0                             # default length of time (s) before main loop automat
SLEEP_DUR = 0.00                        # sleep delay
TERMINATOR = "\r"                       # command terminator byte
QUIT_CODE = "^"
CLEAR_CODE = "`"                        # manually clear your request buffer

print NAME + ' v' + VERSION
print 'Unzipping and loading modules...'

import sys, time, os
from socket import *
from select import *

APP_ARCHIVE = "WEB/python/zig_library.zip"
sys.path.insert(0, APP_ARCHIVE)
sys.path.insert(0, os.path.join(APP_ARCHIVE, "lib"))

import urllib, digicli
print '  ...done.'

stopTime = 0

```

http://code.google.com/p/xig/



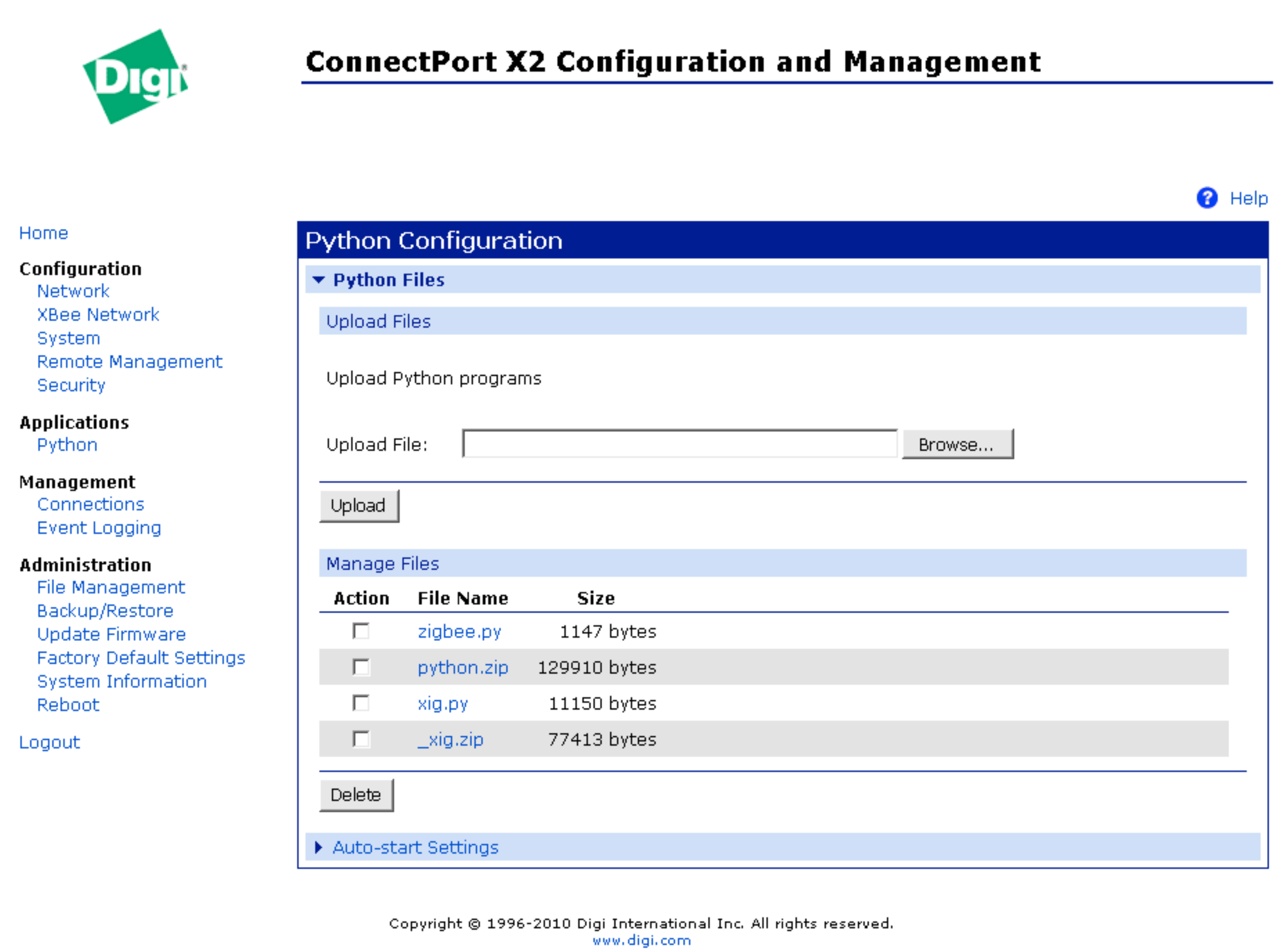
IP Address	MAC Address	Name	Device
10.0.1.183	00:40:9D:3D:6F:35		ConnectPort X2
10.0.1.200	00:40:9D:3A:E2:7B		ConnectPort X2
10.0.1.202	00:40:9D:3D:6F:68		ConnectPort X2

ConnectPort X2
Configured (DHCP)

IP address: 10.0.1.183
Subnet mask: 255.255.255.0
Default gateway: 10.0.1.1
Serial ports: 1
Firmware: 82001596_F3

3 devices

My Device Network



ConnectPort X2 Configuration and Management

Python Configuration

▾ Python Files

Upload Files

Upload Python programs

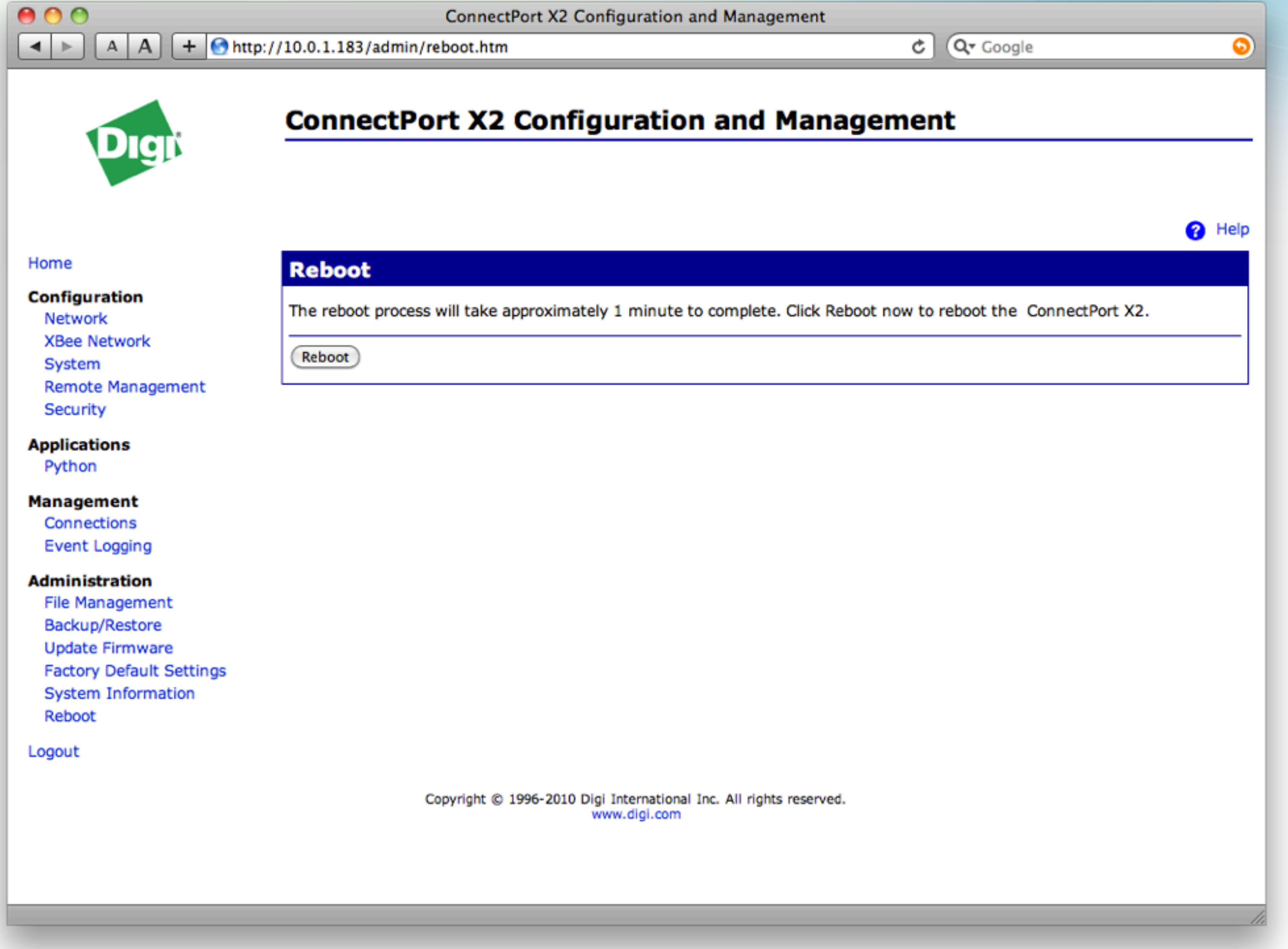
Upload File:

Manage Files

Action	File Name	Size
□	zigbee.py	1147 bytes
□	python.zip	129910 bytes
□	xig.py	11150 bytes
□	_xig.zip	77413 bytes

▸ Auto-start Settings

Copyright © 1996-2010 Digi International Inc. All rights reserved.
www.digi.com


ConnectPort X2 Configuration and Management
http://10.0.1.183/admin/reboot.htm
Google
Digi

ConnectPort X2 Configuration and Management

Help

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

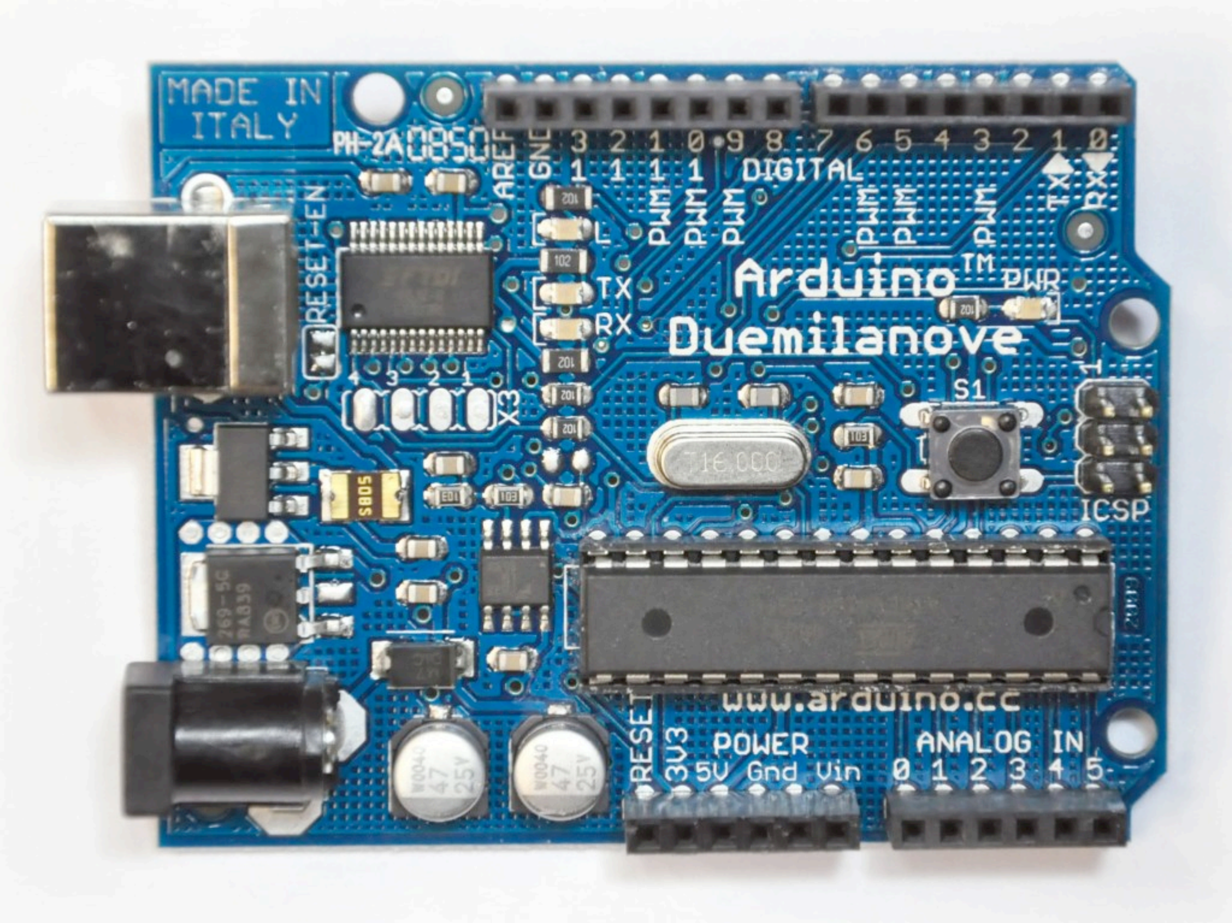
Logout

Reboot

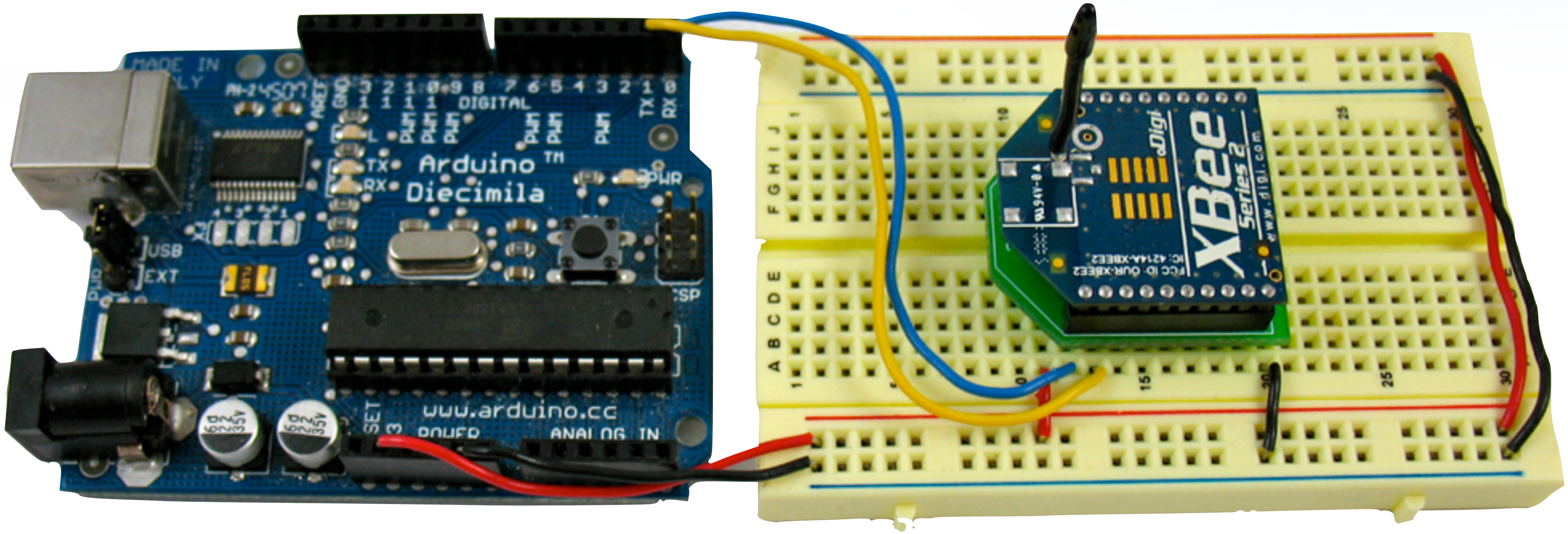
The reboot process will take approximately 1 minute to complete. Click Reboot now to reboot the ConnectPort X2.

Reboot

Using XIG


MADE IN ITALY
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
DIGITAL
PWM
TX
RX
L
RESET-EN
Arduino
Duemilanove
PWR
S1
ICSP
X3
www.arduino.cc
RESET
3V3
5V
Gnd
Vin
POWER
ANALOG IN
0
1
2
3
4
5

1607-01
MaxStream
XBee
Series 2
www.maxstream.net


Arduino
Diecimila
www.arduino.cc
DIGITAL
POWER
ANALOG IN
Digi
XBee
Series 2
www.digi.com

Send a request

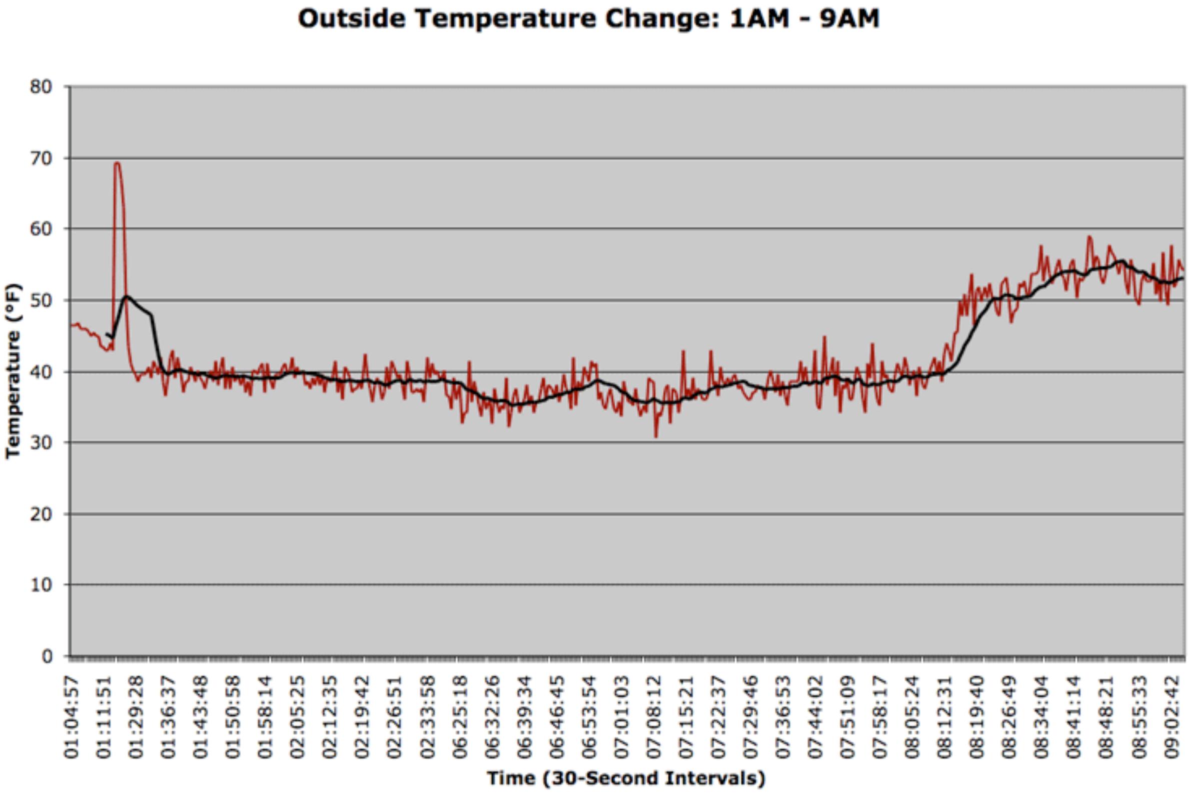
```
Serial.println(“http://faludi.com/test.html”);
```

Read a response

```
if (Serial.available()) {
  char inChar = Serial.read();
  debug.print ( inChar );
}
```

Send a value

```
Serial.println(“http://faludi.com/testpage.php?value=137”);
```



Jordan: XIG Getting Started

xig

Getting Started

http://code.google.com/p/xig/

XIG

COMMANDS AND SERVICES

Commands

```
help or xig://help:   displays this 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 a URL 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)
```

HTTP: Prototyping with URLs

- Shared access – good for groups
- Fast development
- Simple for web developers
- Powerful, flexible, extensible approach

Examples

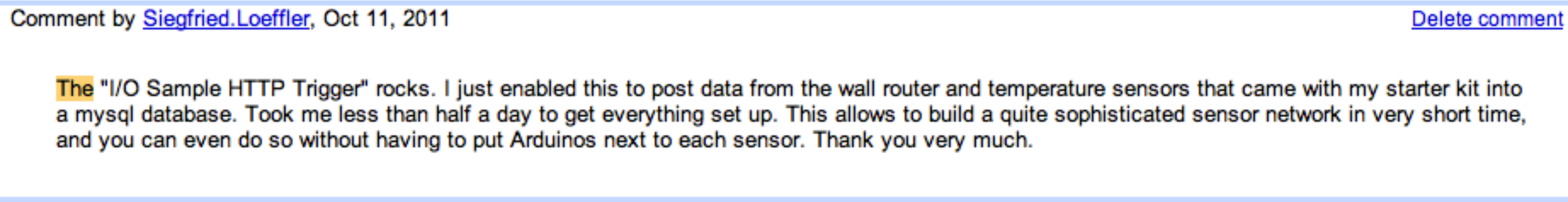
- http://www.whattimeisit.com
- http://yourwebapplication.appspot.com/?name=sensor1&temp=72

I/O Sample HTTP Trigger

- Sends data to any server in a standard format

- Config:
  io_sample_destination_url = http://xbee-data.appspot.com/io_sample

- Returned to server:
  http://xbee-data.appspot.com/io_sample?addr="00:13:a2:00:40:3a:8b:90"&DIO2=1&DIO3=1&DIO0=1&DIO1=1



Comment by Siegfried.Loeffler, Oct 11, 2011

The "I/O Sample HTTP Trigger" rocks. I just enabled this to post data from the wall router and temperature sensors that came with my starter kit into a mysql database. Took me less than half a day to get everything set up. This allows to build a quite sophisticated sensor network in very short time, and you can even do so without having to put Arduinos next to each sensor. Thank you very much.

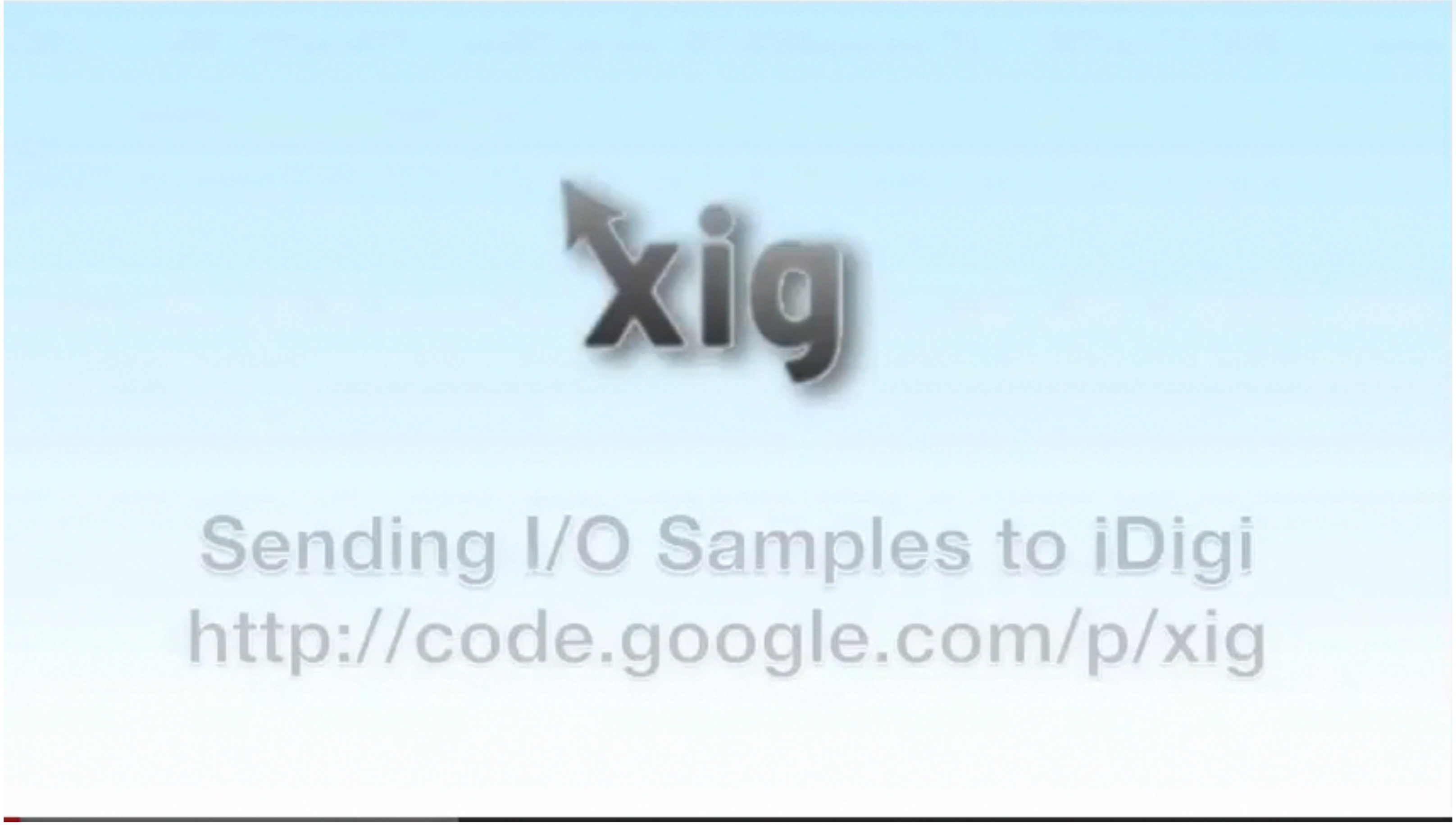
Sending Sample Data to iDigi

- By default, I/O sample frames sent to XIG will be uploaded to iDigi and available as a file and via the iDigi Dia interface
- Calls to this URL: http://(my|developer).idigi.com/ws/DiaChannelDataHistoryFull
- Return the following XML:

Sending Sample Data to iDigi

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<result>
  <resultTotalRows>8</resultTotalRows>
  <requestedStartRow>0</requestedStartRow>
  <resultSize>8</resultSize>
  <requestedSize>1000</requestedSize>
  <remainingSize>0</remainingSize>
  <DiaChannelDataFull>
    <id>
      <devConnectwareId>00000000-00000000-00409DFF-FF43FA07</devConnectwareId>
      <ddInstanceName>XBee_40485A23</ddInstanceName>
      <dcChannelName>AD3</dcChannelName>
    </id>
    <cstId>93</cstId>
    <xpExtAddr>00:13:A2:00:40:3B:CD:B8</xpExtAddr>
    <dcDataType>0</dcDataType>
    <dcdUpdateTime>2012-01-21T13:35:33.607Z</dcdUpdateTime>
    <dcdStringValue>520</dcdStringValue>
    <dcdIntegerValue>520</dcdIntegerValue>
  </DiaChannelDataFull>
  <!-- ... more records ... -->
</result>
```

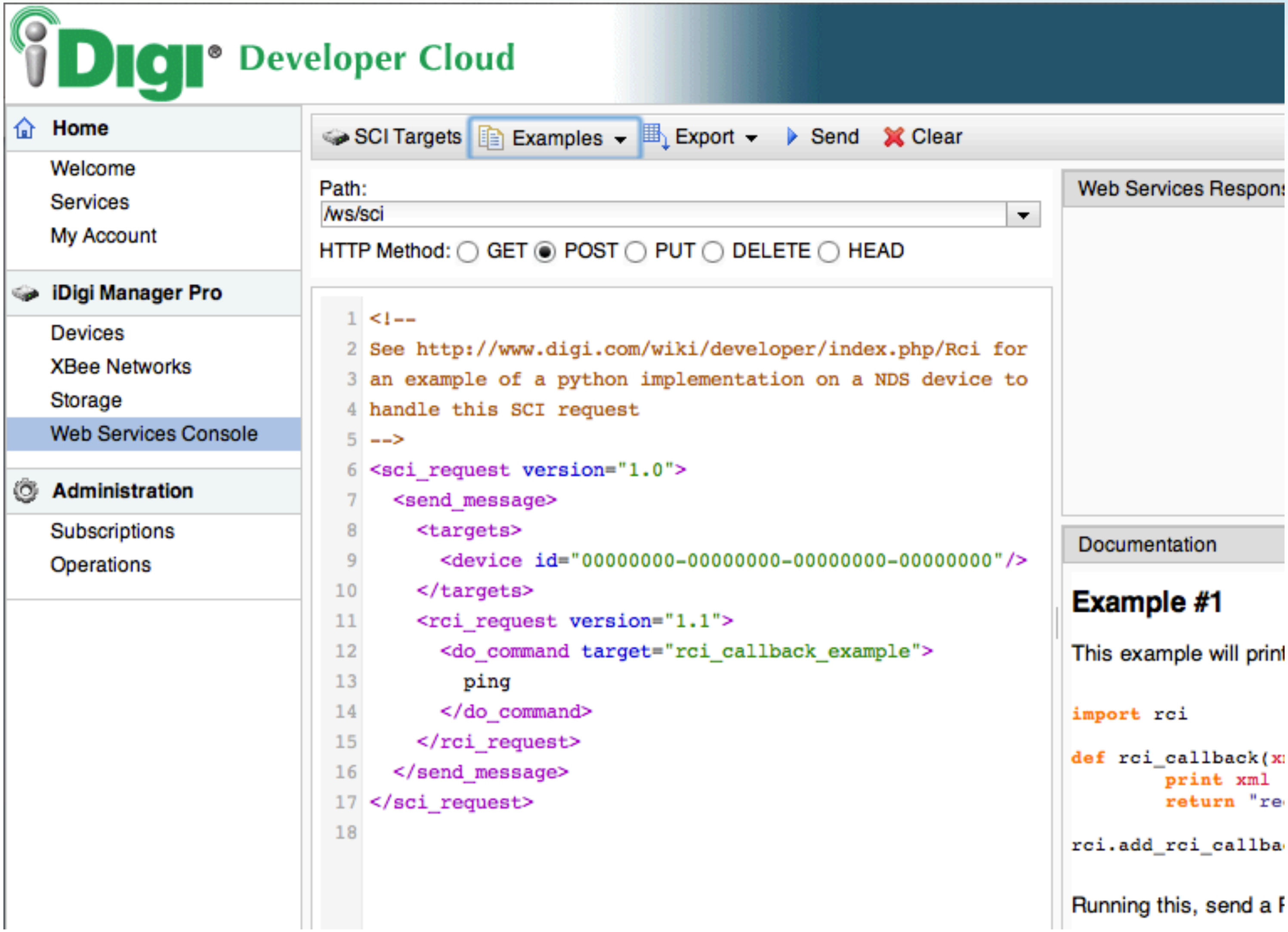
Jordan: XIG I/O Samples



xig

Sending I/O Samples to iDigi
http://code.google.com/p/xig

Sending Messages *from* the Internet to an XBee Using iDigi RCI



Path: /ws/sci

HTTP Method: ○ GET ● POST ○ PUT ○ DELETE ○ HEAD

```
<!--
See http://www.digi.com/wiki/developer/index.php/Rci for
an example of a python implementation on a NDS device to
handle this SCI request
-->
<sci_request version="1.0">
  <send_message>
    <targets>
      <device id="00000000-00000000-00000000-00000000"/>
    </targets>
    <rci_request version="1.1">
      <do_command target="rci_callback_example">
        ping
      </do_command>
    </rci_request>
  </send_message>
</sci_request>
```

Example #1

This example will prin

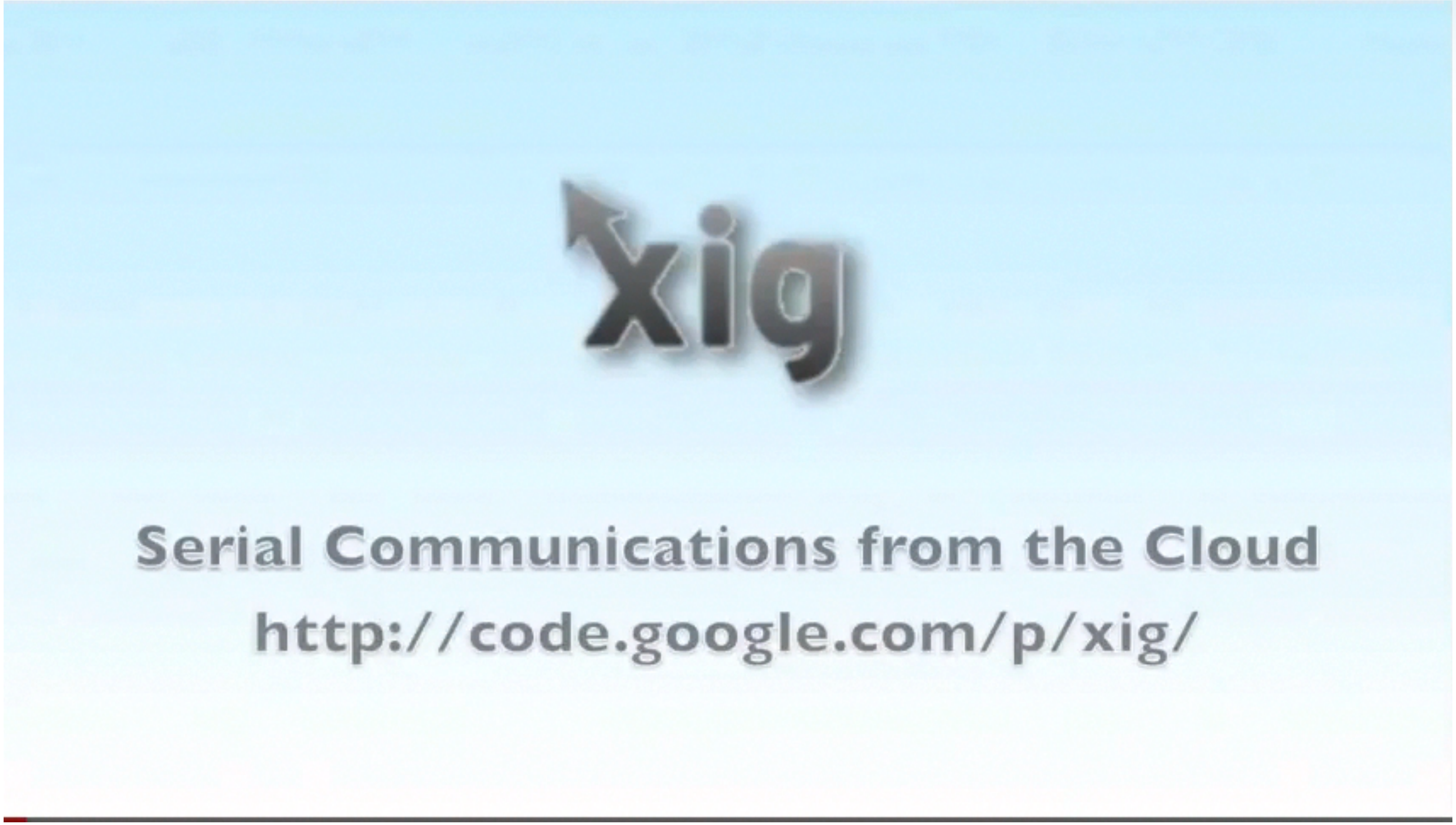
```
import rci

def rci_callback(x
        print xml
        return "re

rci.add_rci_callba
```

Running this, send a F

Sending Messages *from* the Internet to an XBee Using iDigi RCI

```
<sci_request version="1.0">
  <send_message>
    <targets>
      <device id="00000000-00000000-00409DFF-FF43FA07"/>
    </targets>
    <rci_request version="1.1">
      <do_command target="xig">
        <send_data hw_address="00:13:a2:00:40:3a:8b:90!">Hello World!\r\n</send_data>
      </do_command>
    </rci_request>
  </send_message>
</sci_request>
```

Jordan: XIG Serial from Cloud



Serial Communications from the Cloud

http://code.google.com/p/xig/

Setting or Getting Remote XBee AT Settings via iDigi RCI

- Change remote radio’s configuration via POSTing XML to iDigi:

```
<sci_request version="1.0">
  <send_message>
    <targets>
      <device id="00000000-00000000-00409DFF-FF43FA07"/>
    </targets>
    <rci_request version="1.1">
      <do_command target="xig">
        <at hw_address="00:13:a2:00:40:48:5a:23!" command="D0" value="1" />
        <at hw_address="00:13:a2:00:40:48:5a:23!" command="IC" value="0x000C" />
        <at hw_address="00:13:a2:00:40:48:5a:23!" command="WR" apply="True" />
      </do_command>
    </rci_request>
  </send_message>
</sci_request>
```

Jordan: XIG Remote I/O

Remote XBee I/O via iDigi

http://code.google.com/p/xig/

UDP

- To stream UDP data from an XBee node:
  udp://servername:port

- To end the streaming
  xig://abort

Open Sound Control

- Contributed by Axel Roest
- Open Sound Control session allows for XBees to multicast Open Sound Control events to remote Open Sound Control severs
- Configuration of Open Sound Control servers is specified within the XIG configuration file's "osc_targets" section

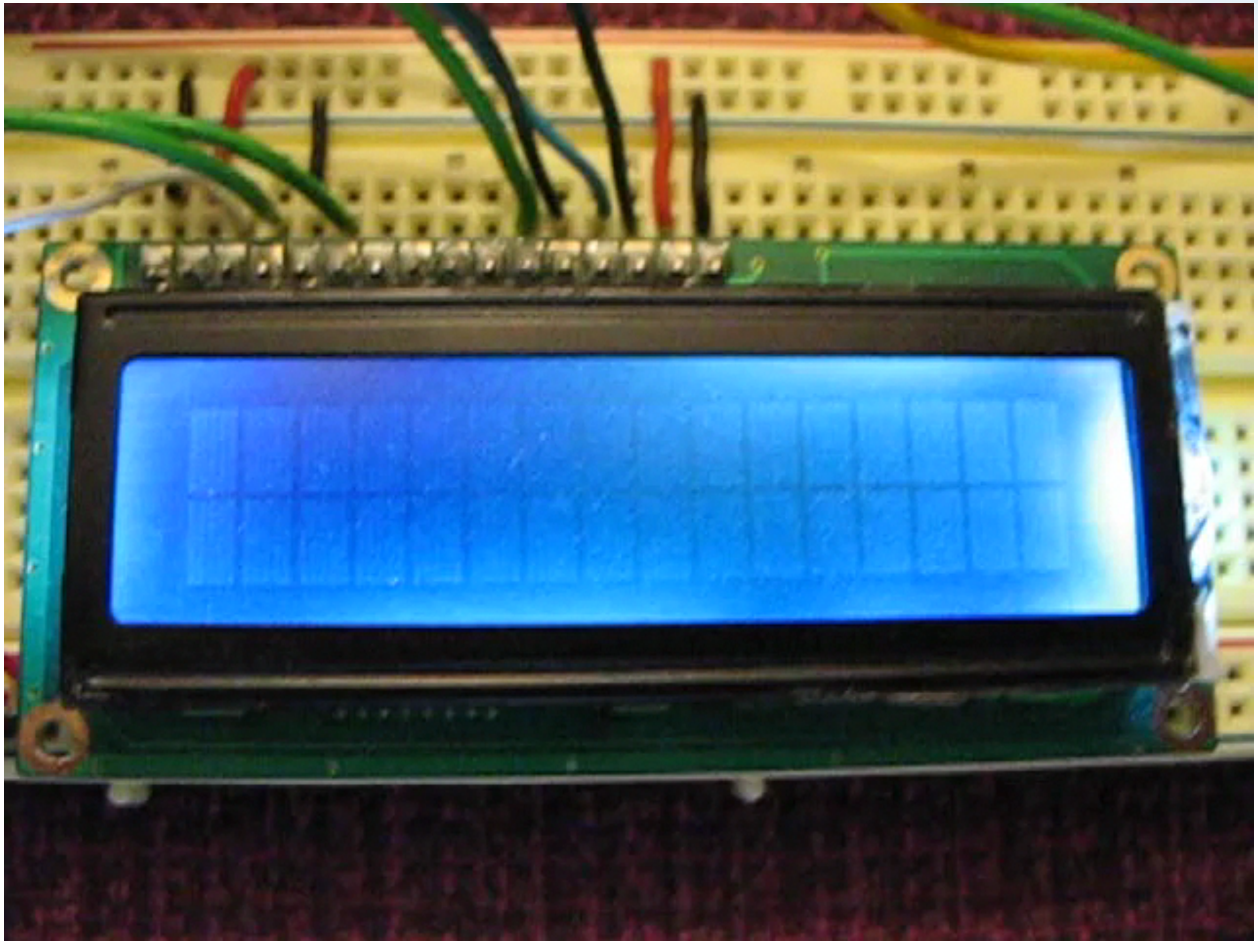
More Features

- Time
- Help
- Abort
- Quit

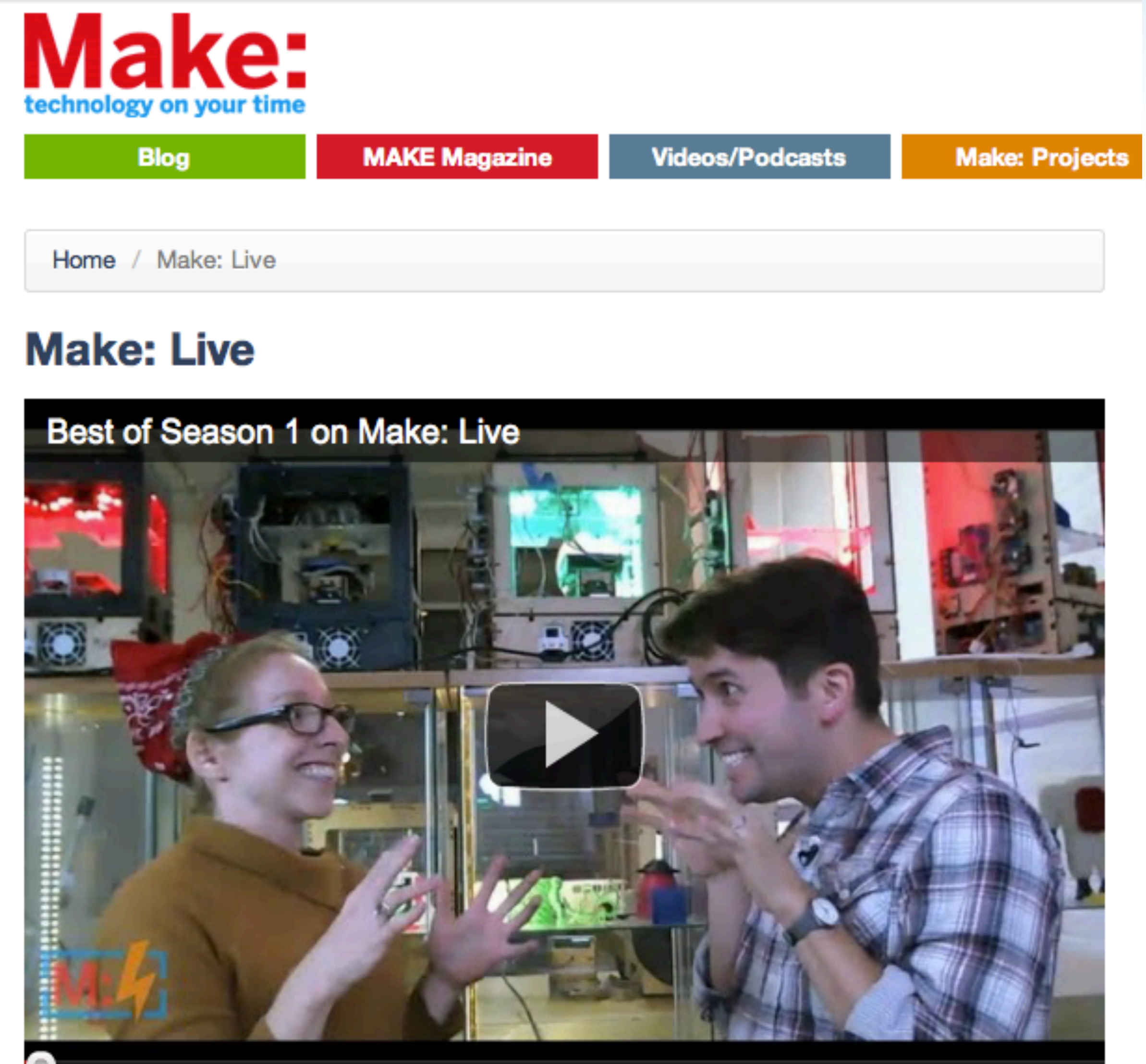
XIG

PROJECTS

Twitter Reader BWSN book project



Matt Richardson



Make: Live

Best of Season 1 on Make: Live

Matt Richardson: On Air

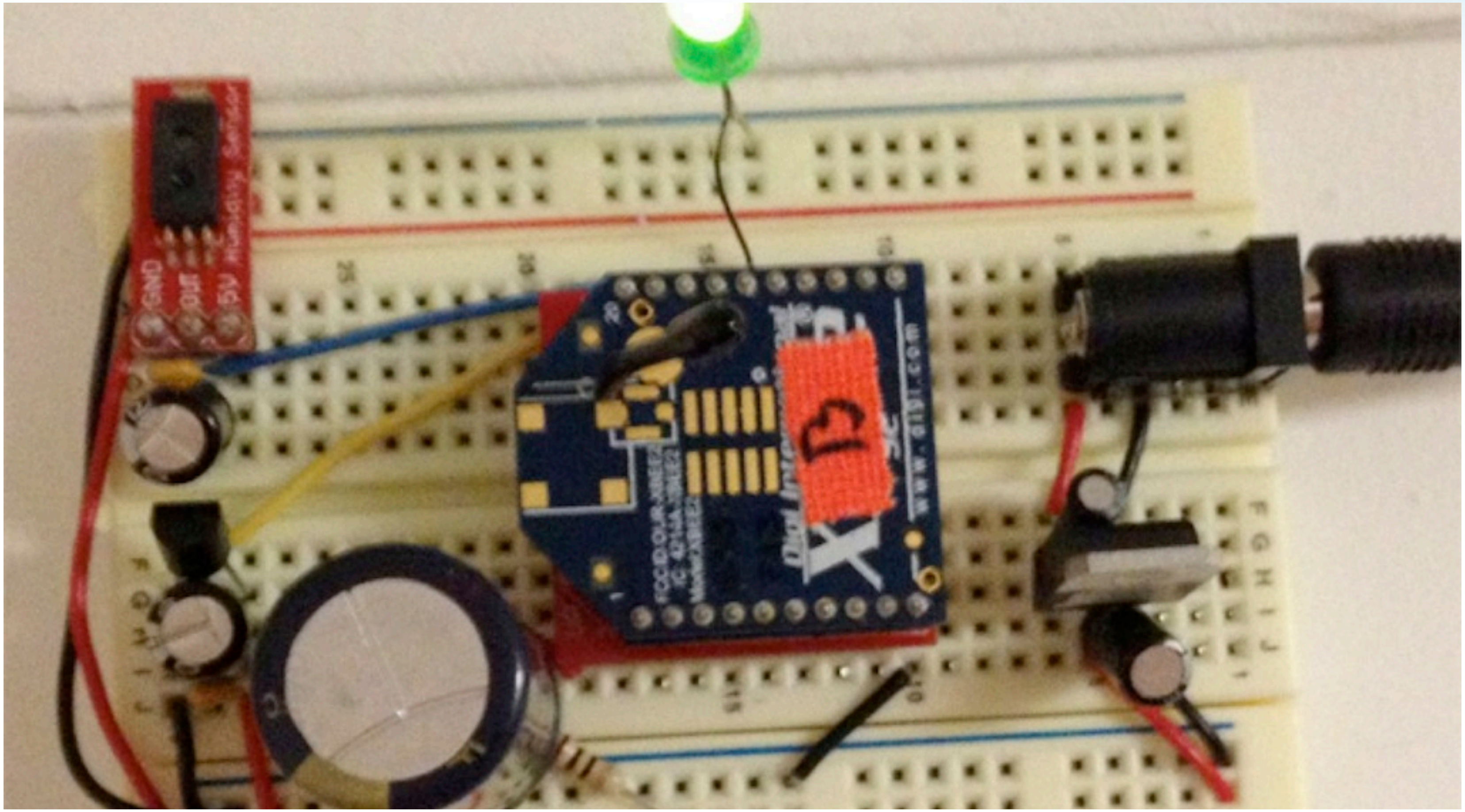
Matt Richardson: Ego Ticker 1

Matt Richardson: Ego Ticker 2

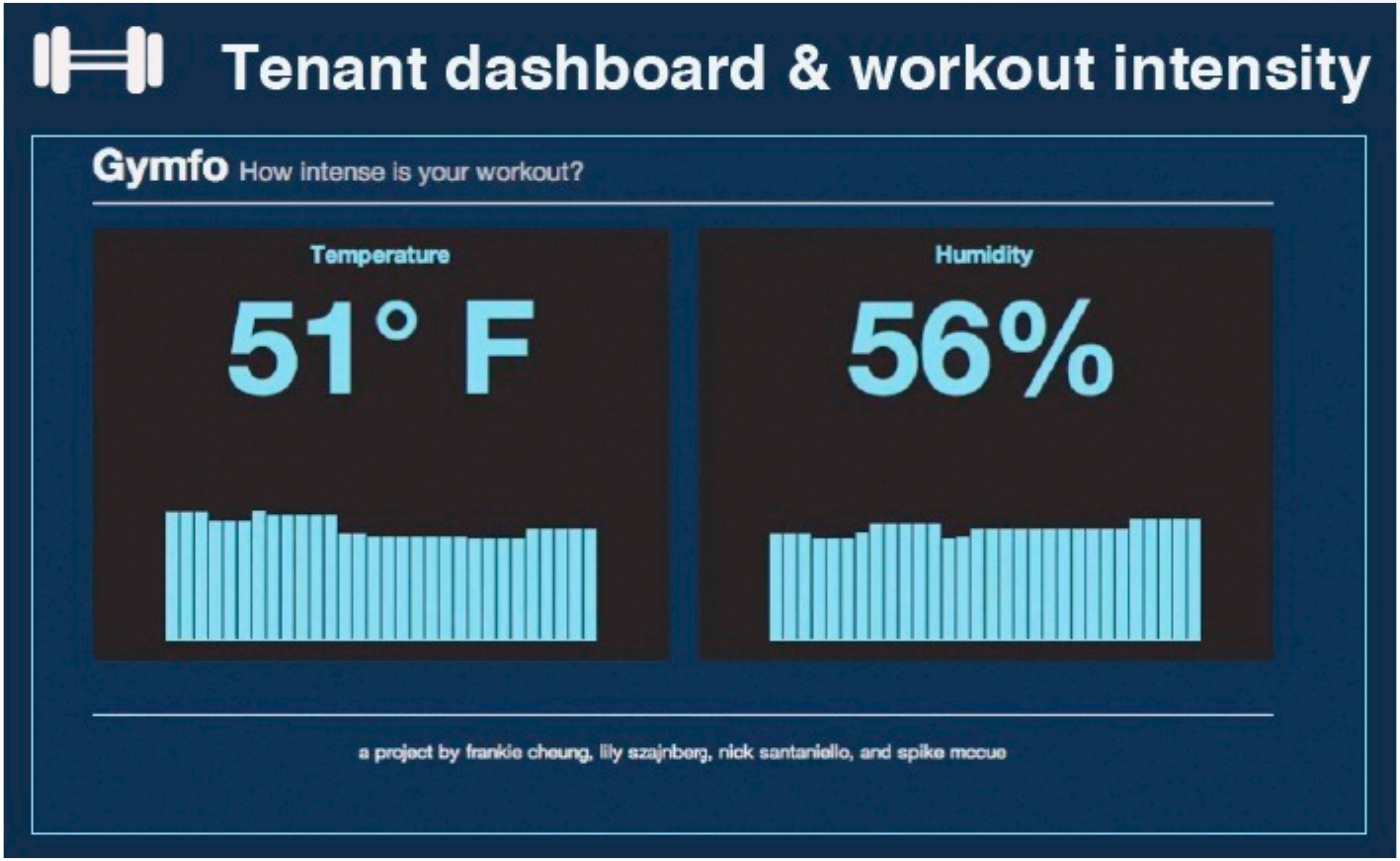
Click here for the video introducing the Ego Ticker

Sensitive Buildings

Sensitive Buildings: Environmental Network



Sensitive Buildings: Exercise Monitoring



Tenant dashboard & workout intensity

Gymfo How intense is your workout?

Temperature

51° F

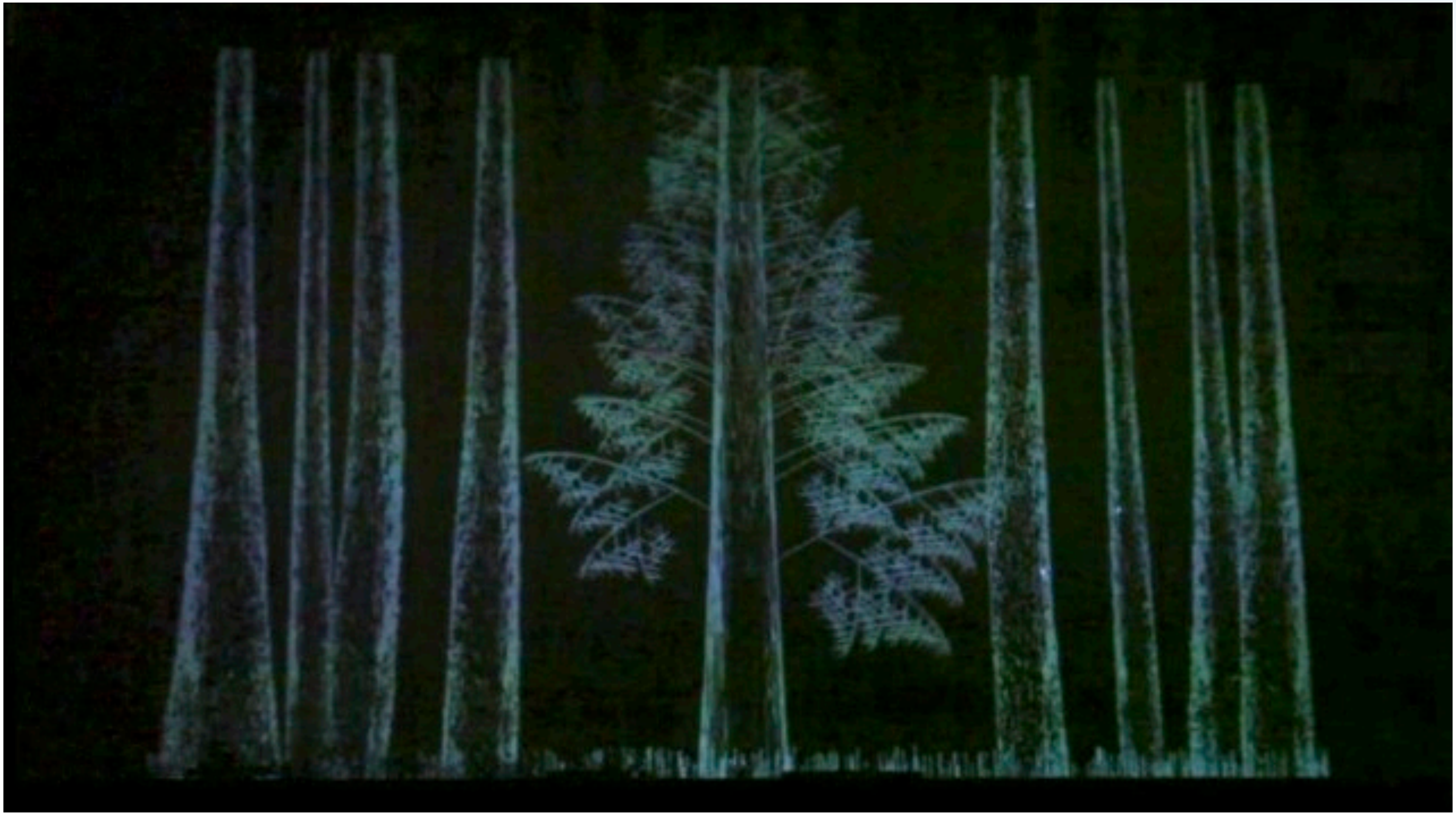
Humidity

56%

a project by frankie cheung, lily szajnberg, nick santaniello, and spike mccue

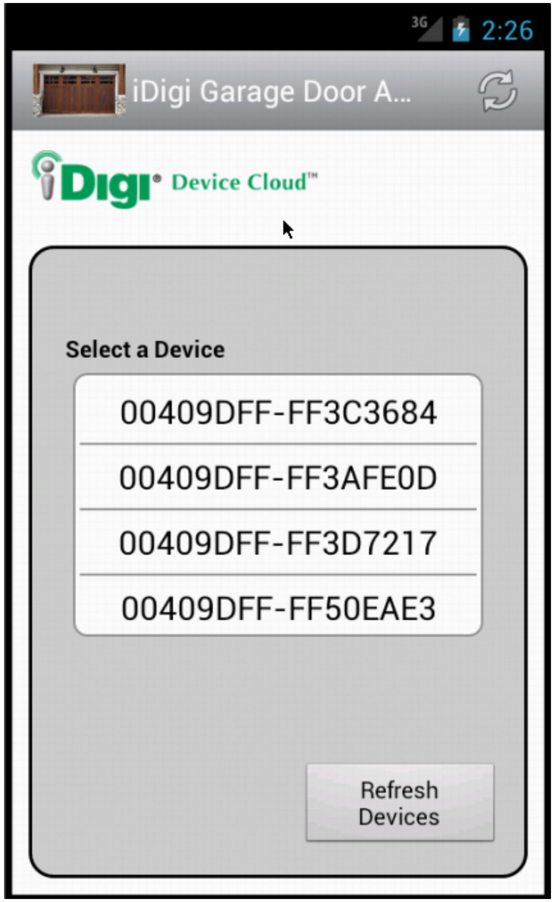
Sensitive Buildings: Postal Mail Chute

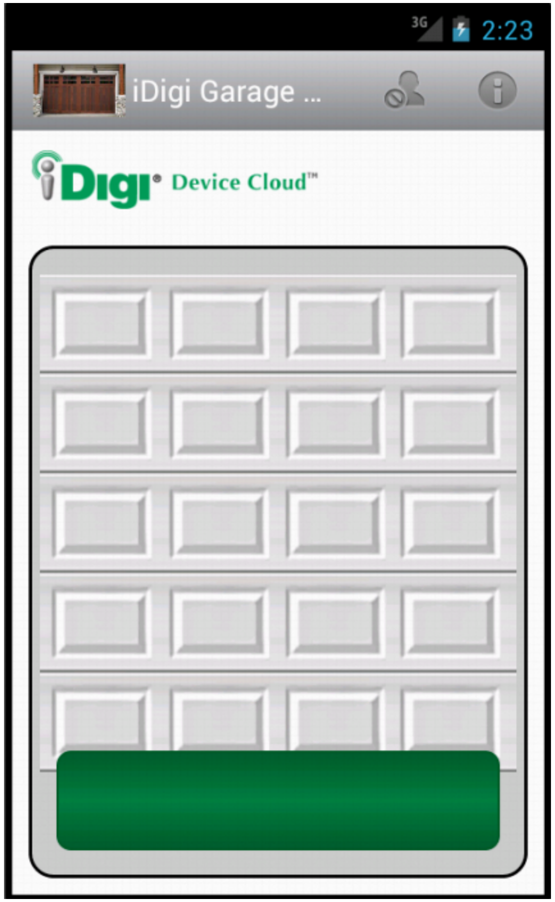
Sensitive Buildings: Elevator Visualization

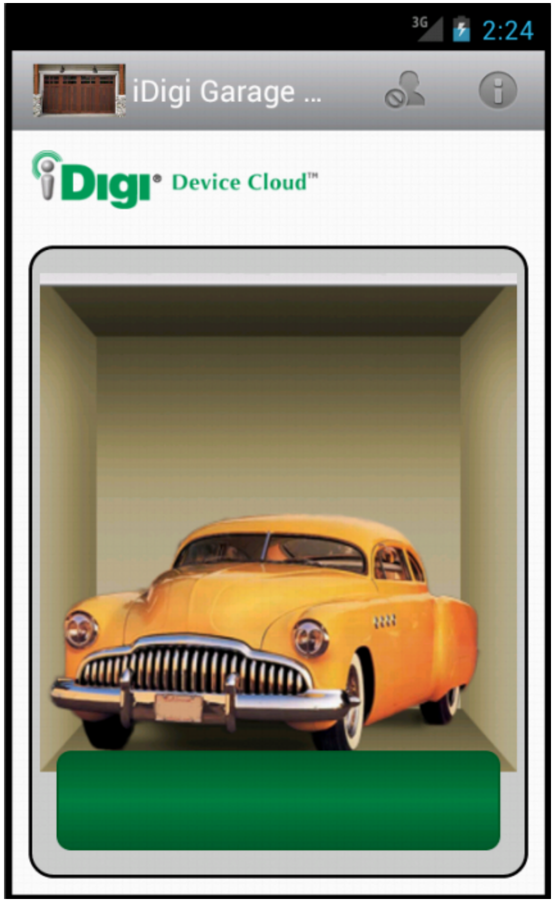


Garage Door for Make

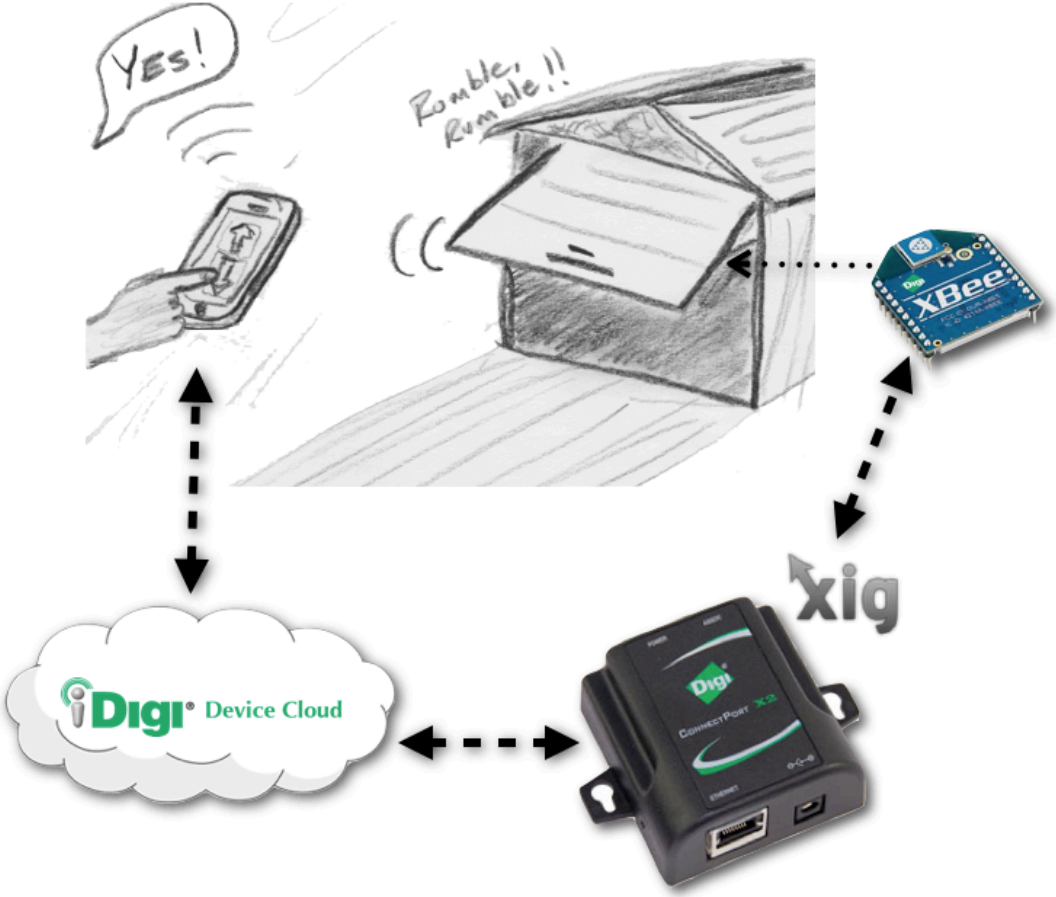
- XBee + ConnectPort + XIG + iDigi





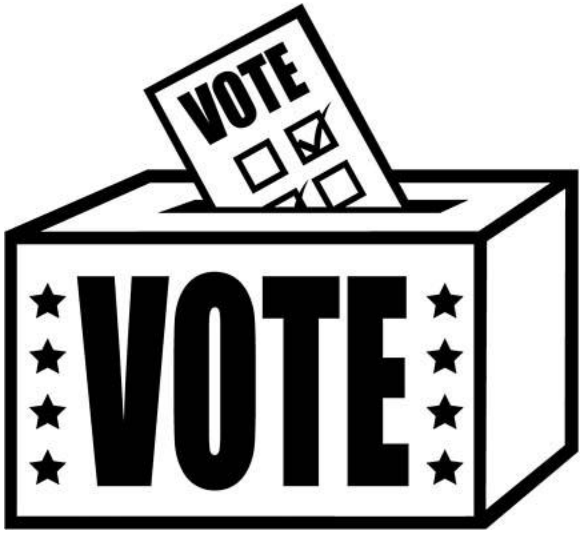


Garage Door for Make



Strata Conference

- Networked wine voting system



Cheerlights

- Tweet a color to change the lights

Cheerlights Video

Imagine...

XIG: the future

- TCP
- HTTP POST
- Other protocols

- Beyond the ConnectPort…

xig

rob.faludi@digi.com

http://faludi.com/xig

THANK YOU!