POW-R

Power Outlet Wireless Reporter

Charles Hathaway
Grace De Geus
Niloc Quimby
Nate Pickett

December 7^{th}, 2012

Project Overview

- What is POW-R?
- Satellite
- Server
- Display

Functional Requirements

Satellite Requirements

- Standard NEMA 5-15 mains electrical outlets
- Two outlet sockets on the opposite side of the plugs
- Less than 5% error on readings
- Power draw less than 1W per Satellite
- Broadcasts information up to 500M
- Transmits every 1.0 seconds
- Zigbee compatible
- Encrypted communication

Server Requirements

- Physical Server inside the monitored building
- Runs on less than 10 W
- Hosts web server for Display
- Button to connect Satellites
- Ability to connect to the user's network
- Zigbee compatible
- Encrypted communication

Display Requirements

- User Management
- Group Management
- Device Management
- Satellite Management

Display Requirements Cont.

- View Data
  - View Device Power Consumption
  - View Power Consumption Over Time
  - View Device Cost Over Time

Display Requirements Cont.

- Power Bill Guesstimator
  - View cost to run a specified device over a period of time in the future
  - View cost to run multiple devices over a period of time
  - View expected power bill if a device is added
  - Allow the user to specify cost of power for
    - Specific time ranges
    - Specific power-usage ranges
    - E.g. Power cost the user $0.08 per kW/hr for the first 500 kW/hrs, and $0.10 after that

Non-functional Requirements

Satellite Requirements

- Able to work in groups of up to 255 for any one Server within a 500 meter radius
- Conveniently sized and shaped
  - Unobtrusive
  - As small as possible given the hardware
- A small LED near each socked to indicate power
- A button for turning On or Off
- A button to initiate connection to the Server
- Will be an unobtrusive color other than white
- Data sent from Satellite to Server shall be encrypted

Server Requirements

- Enough hard drive space to hold 5 years of data
- Supports up to 255 Satellites
- Receives a reading from any Satellite at any time
  - Losing readings is considered an error
- Hardware specifications:
  - 1.2 GHz CPU, i386 architecture
  - 512 MB Ram
  - 40G SSD Drive

Server Requirements Cont.

- Small and unobtrusive case
- Color shall not be white
- User can add Satellites at any time
- Encryption used on all communication to Satellites
- User cannot log in to the Server and get a shell

Display Requirements

- Interface is easy to learn
- Maximum of 500 milliseconds to load a page
- Data is not lost when a device is moved to a new Satellite
- Not vulnerable to SQL injection
- Not vulnerable to XSS
- No unauthorized access
- Data transmitted to the Display is encrypted

Display Requirements Cont.

- User has a modern web browser:
  - Internet Explorer 9
  - Chrome 22
  - Firefox 15
- User's PC can render a page in their web browser that contains:
  - Javascript
  - Images
  - Extensive mark up

What we're working on - Group

- Project Concept
- Requirements
- Architectural Design
- Low Level Design
- Prototyping
- Development

What I'm working on - Charles

- Overall software architecture
- The server software/hardware
- The front-end
  A. All the templates
- The back-end
  A. The REST API
  B. Verifying the database (all modules)

What I'm working on - Grace

- Display
  A. Front-end
- Database
  A. Back-end Satellite module
- Graphs

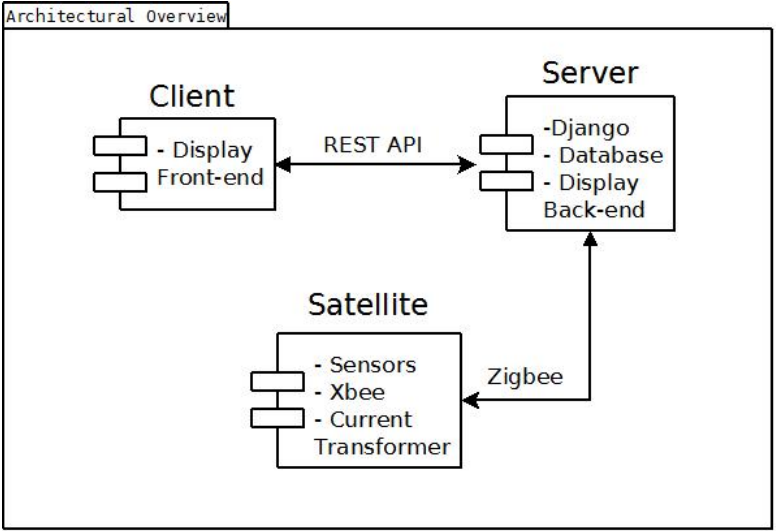
What I'm working on - Nate

- Prototyping Hardware
- Satellite

What I'm working on - Niloc

- Prototyping hardware
- ZigBee specification

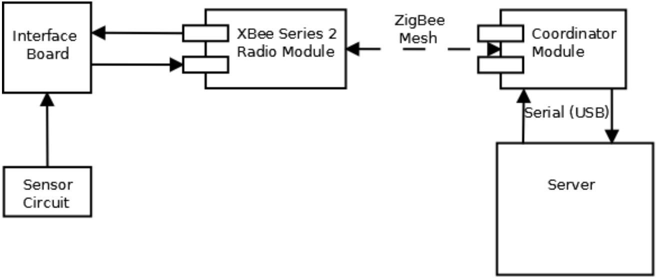
Overview


Architectural Overview
Client
- Display
Front-end
REST API
Server
-Django
- Database
- Display
Back-end
Satellite
- Sensors
- Xbee
- Current
Transformer
Zigbee

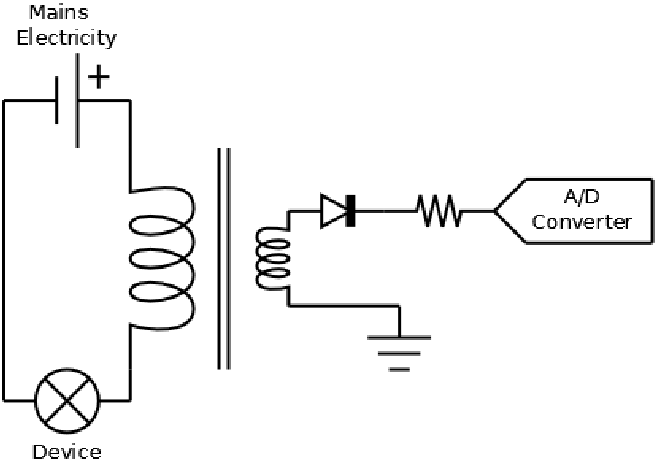
What are the Satellites?

- Satellites are the devices in the outlets
    - Do all of the data measuring
    - Talk to each other

Satellite Overview

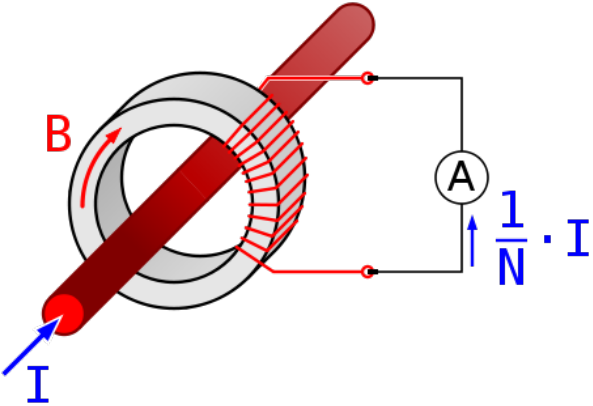


Measurement Hardware Overview



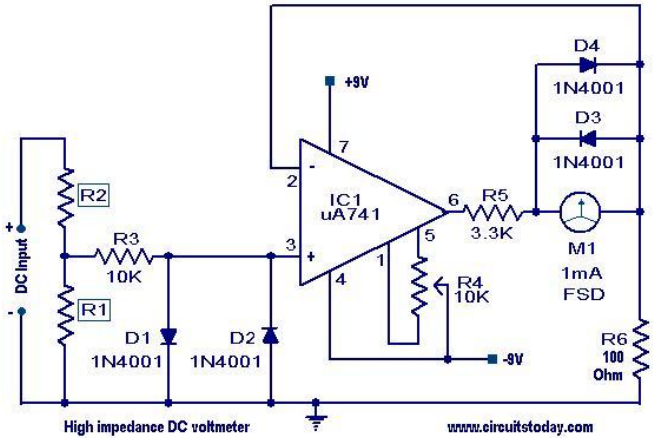
Measurement Hardware

- Current Transformer



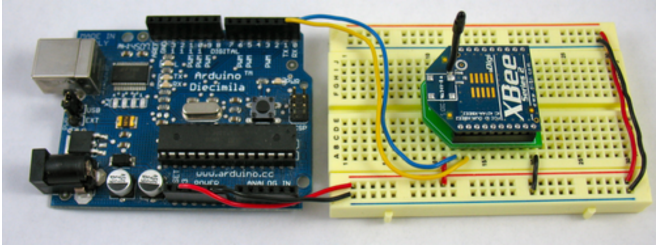
Measurement Hardware

- Voltmeter



Interface Board

- Arduino



Communications Hardware

- XBee Radio

ZigBee Overview

- All inter-Satellite (”Mesh”) communication
- Different than XBee:
  - XBee is a small digital radio (Hardware)
  - ZigBee is the specification it talks over (’Protocol’)

What is ZigBee?

- Specification, much like Bluetooth
- Low power communication over digital radios
- Based on 802.15.4 (PAN standard)
- 'Slightly' open-source

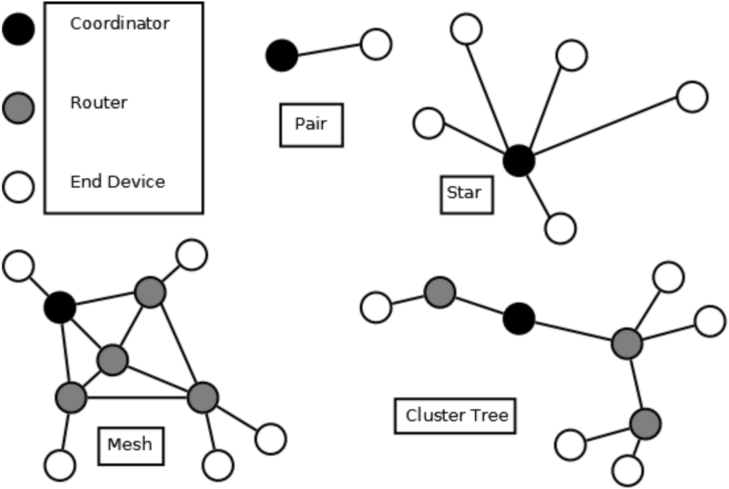
Why ZigBee?

- Ad-hoc 'Mesh' networking
- Capable of adding nodes on-the-fly
- Localized and yet expandable network
- Has it's own encryption

How ZigBee Works

- Three 'roles' in ZigBee networks:
  - Coordinator
  - Routers
  - End-devices
- Addressing scheme supports
- Supports many network topologies

Network Topology Examples



Software Overview

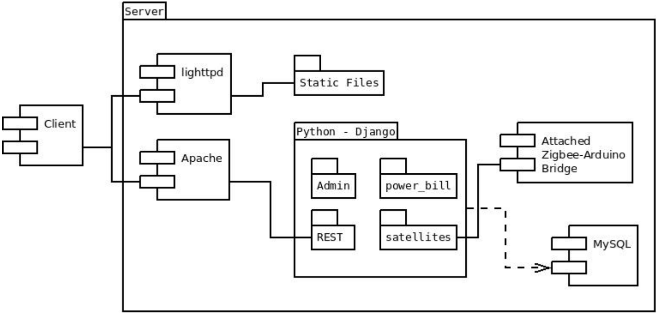
Software Overview

- Python on the backend
  A. Django for the REST, HTTP stuff
  B. Custom python to interact with Arduino interface
- Heavy Javascript on the frontend
  A. jqplot for creating graphs (jQuery included)
  B. RequireJS for module dependency and loading
  C. BackboneJS for MVC architecture

Backend Overview

- Django will be used to handle database
- Tastypie will be used to prototype the REST API
- Django admin will be used to prototype the admin interface
- Each functional area of the project will be a Django module
  - A. Power Bill Guestimator
  - B. Satellite
  - C. REST API

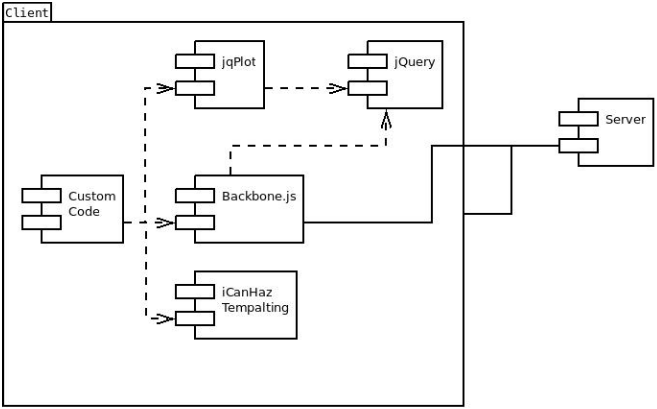
Backend Overview



Frontend Overview

- RequireJS - Loads all the required modules for each page
  - A. Helps keep the scope clean
  - B. Enhances modular development
- Backbone.js - MVC Architecture on the client-side
- iCanHaz - Provides the 'V' part of MVC
- jqplot - Renders charts and graphs
- jQuery - Deals with all the behind-the-scenes AJAX stuff

Frontend Overview



Interesting Problems

Power Bill Guestimator

- Power companies have different ways of charging
  A. Power consumptions tiers
  B. Prime-time vs Off-time
- Cost of power can fluctuate (over long periods of time)
- Need to keep a history
- Database-consolidation can't mix power "tiers"

Power Bill Guestimator Solutions

- Support all methods of charging power
- Store key in data table indicating what power "tier" applies
- Calculate current power tier at runtime
  - A. Scheduling problem
  - B. Calculate during save for efficiency

Limited Resources

- Server must be low powered
  A. Limited processing power
  B. Limit memory capacity
- Server must store data for years
- Server must be able to serve many clients
- Server must be online 24/7 with some reliability
- Server must be able to process data from all Satellites

Limited Resources Solution

- Bare-bones operating system
- Compacting database
- Offload templating to clients via Javascript
  A. REST API
  B. JSON
- Use well-known and tested components
- Use a database which supports simultaneous read/writes

Limited Time

- We have 2 software engineers, not full time
- We have 3 computer engineers, not full time
  A. Most focus on the hardware
- Lots of implied requirements
  A. Access control
  B. Extensible API
  C. History tables

Limited Time - Solutions

- Modular - Prototype components with the expectation of complete replacements
- Open Source - Utilize free libraries
- One stone for many birds
  1. Use this project for other classes
  2. Use architecture and design that we're familiar with
  3. Reusable components

Questions?

Made with LaTeX