HomeNet: A New Home Network

By Matthew Doll

Master Project

School of Architecture and Community Design
University of South Florida
Spring 2011

Chair: Professor Mark Weston

with assistance from Dr. Christensen

DEDICATION

To my parents who continue to encourage me to follow my dreams

ACKNOWLEDGEMENTS

To all the teachers and professors who have helped and inspired me over the years, especially

Ms. Beth, Mr. Oldakowski, Mr. Bartholomew and Professor Mark Weston

Table of Contents

ABSTRACT

Today, computers can be found in items we use everyday like our cars, coffee maker and toys but we have yet to truly integrate them into the buildings we create. Currently, the systems we use in the home are all independent, a mess of wires running through walls. I propose that by using modular/prefab construction, a microcontroller, a cheap single chip computer, can be integrated into every piece of building and joined together during assembly on site to create a distributed computer network that wraps around the building. This network will consolidate existing control systems like security systems and thermostat controls into a single unified system that creates a foundation for a whole new generation of integrated devices. This way, inputs like light switches and security sensors are programmed rather than hard wired to outputs like lights, shades, locks and other actuators, creating a very flexible environment.

HomeNet is a prototype of the future building network and demonstrates how this technology will change how we construct houses. It takes an open source approach to creating a smart house, using all of the tools freely available today. I will look at how our interaction with our built environment changes and what happens when our buildings can interact with us with the same tools we use everyday like the internet, email, Facebook and Twitter. We have the potential to gain finite control over our surroundings and reduce our impact on the environment.

INTRODUCTION

HomeNet is a new way of thinking about home automation and "Smart Houses": DIY, open, free, and collaborative. It's ideas that can be adapted to run on a variety of hardware platforms from simple Arduino microcontrollers to PCs and it all can be connected to the internet. The goal is to create a open platform for creating the next generation of smart house.

This isn't your typical architecture Master Project. I am very interested in how people live and interact with technology. The technology in our homes is increasing every year and I think as architects, we should have a say about how that happens. I don't want to be limited to one brand or manufactures products; I want things in my house to be open and accessible.

The Problem

We are surrounded by smart devices, yet we fail to incorporate them directly into the construction of the Home

The technology inside the American home is sadly lagging behind other industries. Today the cars we drive are infinitely smarter than our homes. They monitor their energy usage, remind you when they need maintenance and automatically turn on headlights and lock the doors. The current home automation systems we use are an afterthought, a mess of wires running through walls back to a central computer. We need to think about architecture in a new way, "not about how building looks but by what a building can do"[1].

I started out my Master Project looking at existing home automation/ home networking technologies but I got frustrated that all the coolest stuff was too expensive and that not much was open source and easy to customize and reprogram.

The Solution

The technology for something better already exists in the form of small low cost microcontrollers. These mini computers on a chip are what the toy and automotive industries have been using to make their products smart. A system of wired and wireless microcontroller device nodes can easily be integrated throughout the home, creating a building network that connects all the devices of the home. The home then becomes a smart entity that can change and react to its occupants. HomeNet simplifies the construction process by using the small nodes to connect devices instead of expensive wiring. Devices like a light switch are connected to these smart nodes and are programmed to send commands, like sending an email, to the light that it should turn on or off. HomeNet aims to assist the creation of a smart house by providing a series of code libraries to help automate the design and programming of these nodes and devices.

[1] Fox, Michael, and Miles Kemp. 2009. Interactive architecture. New York: Princeton Architectural Press., 20

HomeNet allows you create a very flexible environment you can control from anywhere using your laptop or cell phone. You can get home status updates through email, Facebook, and Twitter; turn lights on and off remotely or set it up to happen automatically; get energy usage reports by room or even by power outlet, helping you discover energy vampires; or get status updates about A/C performance so you know when you need a tune up or just need to change the filter. Almost anything can be connected to a device node for monitoring and remote control. This knowledge will break the home free of its ambiguity and silence and reveal it as a living, energy consuming machine and give use new way to control it with smart energy saving mechanisms like automatically turning off lights and power outlets when we leave and provide us a snapshot of how we are living and how we can change our behavior to become more environmentally conscious

Existing Networks

Networks in the home can be separated into two categories: Media Networks and Simple Networks.

Media networks carry data like voice, video, photos, and websites. This network is what your computer uses to connect to the internet, your cell phone uses to surf the web and your TV uses to get channels from the cable company. These devices use high bandwidth communication wiring like Cat5/6 (8 wires), Coaxial and Fiber Optics or wireless networks like 3g/4g Wifi/WiMax.

Simple Networks are one that that communicate things as simple as on/off, open/close, lock/unlock. These networks are found throughout the home.

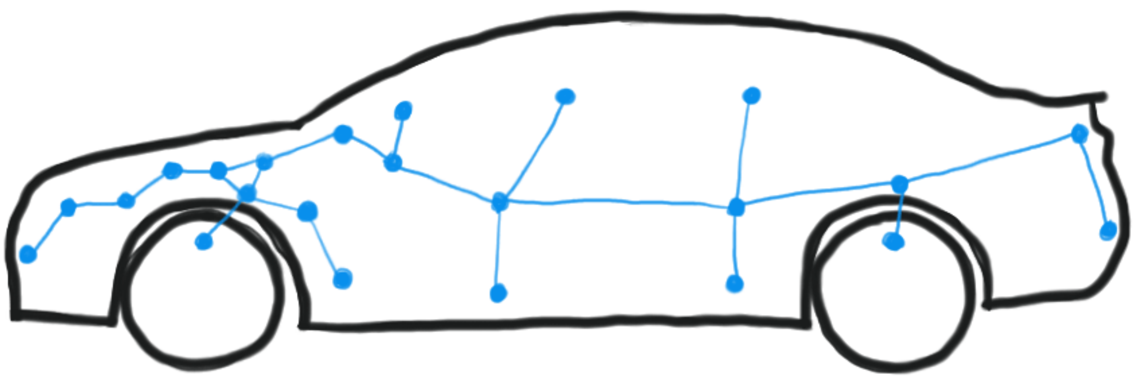
For example:

- Door bell: the push button connects to the chimes
- Climate Control: the thermostat connects to the Air Handler
- A/V system: the remote control connects to the television
- Light networks: a light switch connects to light bulbs
- Security system: sensors connect to a monitoring station which connects to a monitoring service
- Irrigation system: the control timer connect to solenoid valves.
- Access control: a key connects to the lock in the door
- Electricity: the power meter connects to circuit breakers which connect to electrical outlets

Currently all of these simple networks are independent of each other. What if these networks could talk to each other with a common language? What if we connect them to a media network? This opens a whole new world of opportunities to improve the future home. We do things like program the irrigation system timer from our cell phone. Why spend money on expensive controllers with proprietary interfaces when we already have devices with great interfaces that we can use. We can change from complex and expensive hardware to simple and cheap hardware that we can program from a complex device. Many of our modern appliances already contain smart computer chips capable of interacting with other devices, now we need a common language for devices so we can unite them in the home.

PRECEDENT STUDIES

The Modern Automobile

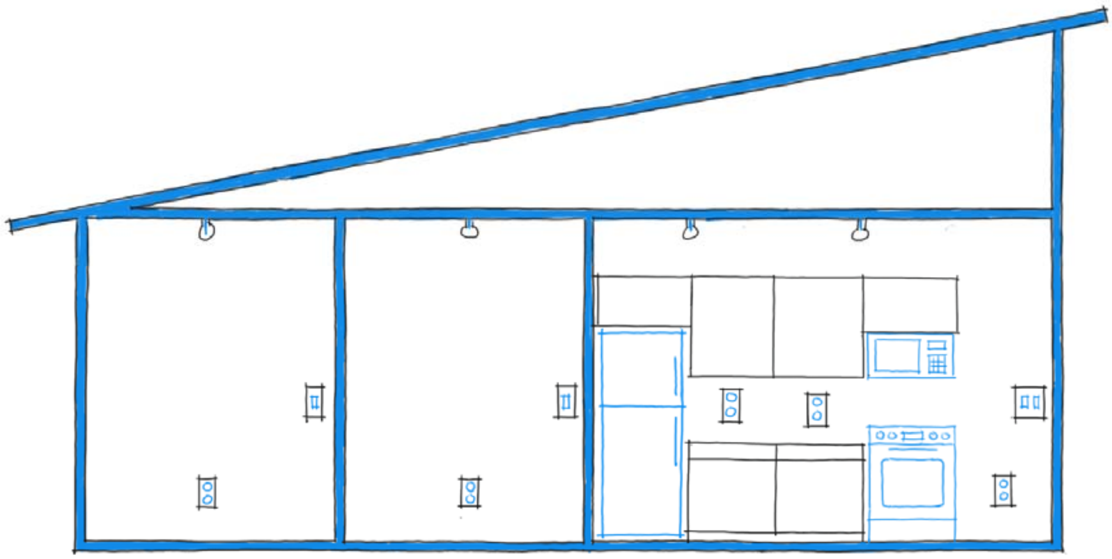


Today our vehicles are far smarter than our homes. Engineers have integrated a complex sensor network into the core of today's cars, allowing the car's computer to monitor all aspects of the combustion process, and enabling the computer to adjust the combustion process to maximize performance and fuel economy. Over the last decades, "nearly 80% of all innovations within automobiles are derivatives of the electronic systems"[2] The cars computer can monitor the drivers input via the steering wheel, accelerator and brake pedals and make adjustments to prevent losing control of the vehicle in an emergency. The best part is that all of this complex technology is hidden from the driver and presented as a series of indicator lights on the dashboard. If the computer detects engine malfunction or inefficiency it turns on the check engine light so you know that you have a mechanic take a look at your car before the problem causes expensive damage.

---

[2] Fox, Michael, and Miles Kemp. 2009. Interactive architecture. New York: Princeton Architectural Press., 21

MIT Media House Project



The Media House Project was a joint project between UPC. Barcelona Tech and the MIT Media Lab during 2001-2002. MIT focused on developing new technology for the home and Barcelona Tech did a series of studies on how to integrate MIT's work into a new architecture.

The House is the Computer, The Structure is the Network

This statement sets the framework for the project. Instead of the computer being separate from the house, it is integrated into the structure of the building. They developed an agenda for the Information Home[3] and built a series of projects that demonstrated what the future home might look like.

I was most interested in points one, five, and six. Internet-0 is a language they developed for the home. The open dwelling focused on how to rewire the home to make it more flexible. The house's website was demo of new ways to interact with the home.

A lot of new hardware and software has been developed since this project that has made building a prototype smart house a lot easier and simpler, so that someone like myself can quickly experiment with new ideas.

Agenda for the Informational Home

1. Internet 0: the house a great brain
2. Internet 2: large format video.
3. Informational structures (building the network into structure of the house)
4. Manufacturing dwellings
5. Open dwelling: flexibility of distribution
6. The house's website
7. The neighborhoods website: more social
8. Media spaces: break free of the screen
9. Extended space: virtual links
10. The interface-space
11. The knowledge house
12. The bio house - a technoloagricultural environment
13. The architecture of the home

[3] Massachusetts Institute of Technology Media Laboratory. *Media House Project*. Barcelona: IAAC, 2004.

DEFINITIONS

Open Source
is to literally make the source of things freely available. It's a movement of open sharing of knowledge. It originated in the software industry as a way to share and collaborate on complex software programs. The philosophy can be applied to all sort of creative endeavors including the creation of a smart home.

Internet Of Things
is an idea that originated from MIT to connect everyday items to the Internet so that they can be inventoried, tracked and interacted with remotely. Every human is surrounded by 1000 to 5000 objects[4], so this is a formidable task.

Intelligent Environments
defined as "spaces in which computation is seamlessly used to enhance ordinary activity." [5] They enable users to interact with technology in a new way that is intuitive and natural like gesture, context and speech.

Physical Computing
is hardware and software that can sense and respond to the analog world. It's reshaping what we think of as a typical computer. It is about creating a conversation between the physical world and the virtual world of the computer.

Embedded Computation
computer technology integrated into devices to make them smart. In architecture it refers to "a system that is literally embedded into the building that has the ability to gather information, process it, and use it to control the behavior of the actual physical architecture"[6]

Home Automation
is the automation of work in the home. For example, programming a light to be turned on and off on a schedule.

Smart Grid
is a better power grid that allows consumers and producers of electricity to communicate their supply and demands in real time to help make the grid more efficient by reducing the need to run expensive generators to supply peak demand. It allows buildings with solar panels to choose to store the energy in batteries or sell the power they generate based on market demands. It also enables the power company to inform the house when it can defer tasks to off peak times in exchange for cheaper electricity

[4] *Introduction to Internet of Things* http://cpschina.org/IOTS/ 5/3/2011
[5] Fox, Michael, and Miles Kemp. 2009. Interactive architecture. New York: Princeton Architectural Press.
[6] Fox, Michael, and Miles Kemp. 2009. Interactive architecture. New York: Princeton Architectural Press.

Home Area Network

is simplified building network for linking together smart devices in the home. HAN for short.

Cloud Computing

allows you to run software on demand on a remote cluster computers that easily scales and can be accessed from anywhere. Online email services like Gmail and Hotmail are very popular cloud programs

OBJECTIVES

There has to be a reason to spend money on making the home smarter. These can be broken down into 4 categories:

Energy

First and foremost any smart house system has to be energy conscious. The system needs to be able to save more energy than it consumes to run and the amount of energy saved needs to easily offset the cost of installing the system.

Homes account for a substantial amount of energy in the US. In a 2009 study by the Department of Energy, showed that residential users consume 22 percent of the energy the US produces [7]. Inside the home over 50% of the energy is use to heat and cool the home[8]. We have a lot to gain from conditioning our homes more efficiently. Many homes are empty during the work day, wasting electricity during peak demand. A smart house could very effectively reduce energy use while you are away and still have it ready when you arrive home.

A lot of energy can be saved by following LEED design guidelines like increasing the use of natural light but the smart house can take that one step further and help automate the blinds so natural light is being used as the designer intended. It can also provide validation that the design is working and show areas that could be further improved.

Loss prevention

A smart house system has to alleviate worries rather than cause them. It has to provide security and watch over the home, providing reassurance while you are away. It needs to monitor the status of appliances and notify the owner before they break so they can be repaired before one becomes costly replacement. In a disaster like a fire or tornado, not only can the house call the authorities, the home can also notify first responders about how many people are home and which rooms they were last in

Convenience

A smart house needs to provide a convenience and not be a hassle to use. The modern house is getting more and more complicated and technology can remove some of that burden. No longer is it a pain to program the sprinklers, you can do from your tablet while lounging on the couch.

Personalization

The home can be reprogrammed to meet your needs and change based on different occupants. Spaces can be dynamic and you can change a space on the fly. This will open a new possibilities of architectural design and allow spaces to be used for multiple purposes like a family room becoming a home theater with a push of the a button.

---

[7] http://www.eia.doe.gov/energyexplained/index.cfm?page=us_energy_use 4/4/2011
[8] http://buildingsdatabook.eren.doe.gov/docs/xls_pdf/2.1.5.pdf 4/4/2011

PROCESS

Gather Data

The first step of a smart home system is to *gather data* about the home environment. This can be from sensors that measure temperature and humidity to monitoring a light switch to know if it is turned on or off. The data needs to be tagged with the current date/time and where the data came from so the system can analyze it later.

Interpret the Data

The next step is to *interpret the data* and figure out what it represents. What do we need to save to a database to use later? What can be ignored so the system doesn't get overwhelmed? Some data like from environmental sensors we will want to save on a schedule but other data from devices like a light switch, only needs to be saved when it changes, like from off to on.

Present the Data

The collected data needs to be presented the user. This way you can check on your home in real time or see all of the past historical information. In HomeNet, this is done with a web based interface, HomeNet.me that can accessed from a computer or cell phone anywhere.

Remote Control

The fourth stage is *remote control*, where we can interact with the smart house indirectly using the web interface. This way you can turn devices on and off from anywhere.

Automate Control

The next step is *automated control* where the computer can be given control to make decisions based on predefined sensor input like turning off lights when house detects no one is home.

Intelligent Control

Finally we come to *intelligent control* where the smart house can recommend changes and react dynamically based on user behavior.

Building the Home Area Network

Hardware

Microcontroller
is a simple self contained computer on small microchip that is designed to be embedded into devices

Sensors
measure the physical world and convert it into digital signals that a computer can understand.

Actuators
are things that produce a physical action, like a turning on a light bulb, motor, or unlocking a door.

Tools

Arduino
an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. [9]
http://arduino.cc

JeeLabs/JeeNode
Produces an Arduino derivative called a JeeNode that has a modular "plug" interface and a integrates a low cost wireless chip. JeeLabs also produces a range of useful "plugs" that easily expands the JeeNode capability with sensors, buttons, and more.
http://jeelabs.org/

Processing
an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. [10] http://Processing.org

[9] Arduino.cc used under Creative Commons License
[10] Processing.org used under Creative Commons License

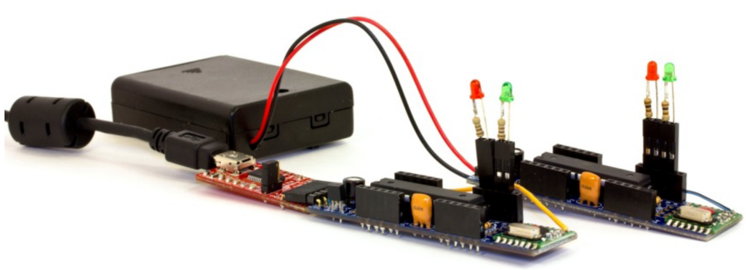
EARLY EXPERIMENTS

I started this project with a series of experiments to gain an understanding of how to work with the Arduino microcontrollers, sensors, and transmit the data they collect to the Internet. I started with a single microcontroller and expanded to build a wireless network of them.

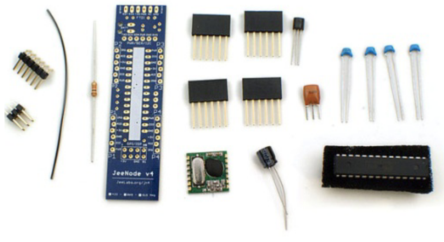
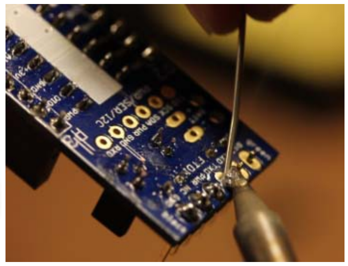
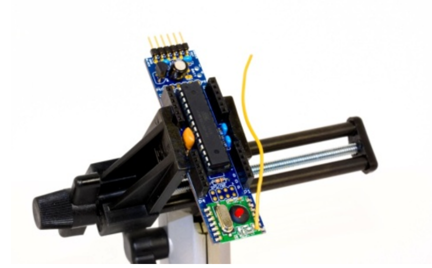
***Experiment 1*:** Send Data to the Internet

The first experiment was to learn how to use an Arduino to gather data, transfer it to a PC and store it on the Cloud. I chose an interesting temperature/humidity Sensor (SHT21) that already had open source code library written for using it with an Arduino. Reading the sensor and printing the values to the serial console was pretty easy. The challenge came when trying to get a Processing sketch to read the serial values and upload them to Pachube. The Processing sketch had to interpret the data the Arduino sent and turn it into numbers it could send to Pachube. Luckily there are some great examples on the web that guided me.

Experiment 2: Build a Wireless Arduino Called a JeeNode

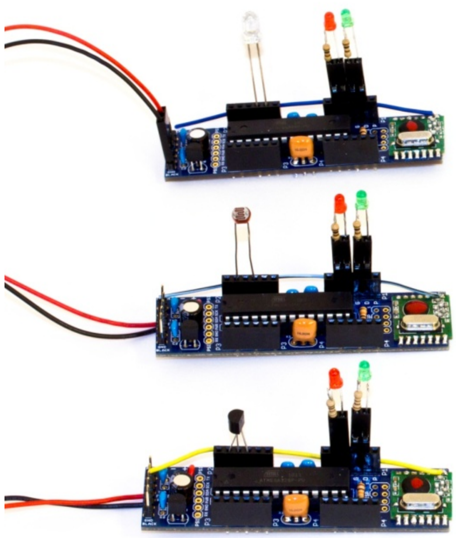
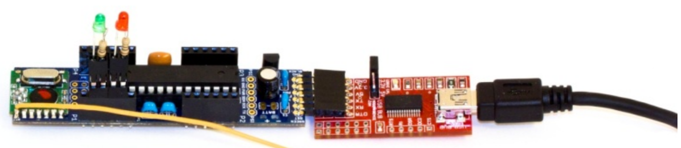


The second experiment was to expand the range of the sensor by making it wireless. I looked at using some of the common ways of do this like using a xBee shield for the Arduino. This was too expensive to deploy on the scale of a whole home so searched for a cheaper alternative. I found an Arduino derivative called a JeeNode (by JeeLabs) that used the same hardware and software as an Arduino but in a smaller cheaper form factor that also included a low cost wireless transceiver

The JeeNode comes as a kit with “some assembly required”. I learned how to solder all of the parts together to assemble the kit. Soldering can be quite a challenge but over time I a developed a new skill It was fairly easy to get the data from one node to be sent to the other using the code libraries from JeeLabs.

Experiment 3: Gather Information from Multiple Nodes

Expanding from two to four nodes proved much more challenging. Suddenly it became very time consuming to sort out which device or sensor was sending data. Each node had to be programmed individually with custom code. I discovered I needed to standardize the code and automate the communication process.   I needed a common protocol to organize the data that described where the packet is coming from, where it needed to go, what type of data it was carrying and the data payload it’s self.

Developing a Language for the House

There are many existing device protocols like LonWorks, BACet and x10 that were developed to control lighting and building automation but are too specialized to interface with the wide variety of smart devices that will be found in the future smart house. Many of them lack modern security practices like encryption and rolling codes and are very susceptible to being hacked. I also looked at some proposed protocols like Internet-0 and ZigBee but decided that they would be too difficult for me to implement using Arduino hardware with its limited amount of memory and my limited time. Future hardware would not have this issue and a more complex protocol could be implemented.

Internet-0 is a very interesting protocol developed by MIT and used in their Media House project. They focused on implementing a compact version of the *Internet Protocol version 4* (IPv4) networking protocol that connects computers on the Internet today. I found it had too much overhead in comparison to the size of the payload it carries. While in theory it supported both wired and wireless systems, the Internet 0 designers focused on a wired system where packet overhead was not a major concern. However, in a wireless system, every byte of data has a cost to battery life. If it's possible to reduce the number of bytes that are sent by half, it reduces how long the transmitter is on and could almost double battery life. While using the IPv4 stack allows devices to have a direct presence on the internet, I think that for security reasons, we don't really want our homes exposed like that and we will have our Home Area Network's behind a firewall and router for protection. If we eliminate that requirement, the protocol can be simplified to reduce its overhead and lower the cost of the hardware. For future devices, I see the technology improving where adding additional bytes of data wouldn't be a concern and could easily implement the next generation IPv6 protocol.

The current front runner for a standard language for the home are the standards that the ZigBee Alliance is developing but I think they have several drawbacks. I think their standards are too closely tied to ZigBee's own hardware. In twenty years, the hardware will have changed and improved and they need to look at how their standards can work on hardware that hasn't been invented yet. Recently, the National Institute of Standards and Technology (NIST) agreed and recommended that ZigBee change their Smart Grid standard to support the next generation IPv6 by using a protocol called 6LoWPAN.
For me, another drawback is its license. ZigBee advertises their standards as being "open", but it's only free and open for non commercial use. The license terms prevent it from ever being included in anything that is licensed under GPL, like Arduino and Linux.[11] If ZigBee was a true open source standard, it would be much more appealing to projects like mine.

I see the future of ZigBee becoming what Microsoft is to PC's, the large consumer oriented HAN. I think ZigBee needs competition. My project was not really meant to replace ZigBee but build the roots of an open source competitor, a Linux like alternative. Mine might not be successful but it could lead to a better derivative. I look at the Internet Explorer versus Firefox battle and I'm impressed at how an open source project can challenge the dominate brand and force both to innovate and the web is better for it.

[11] http://freaklabs.org/index.php/Blog/ZigBee/ZigBee-Linux-and-the-GPL.html 4/4/2011

HomeNet Protocol, an Open Source Language for the Home

I developed my own language for a smart home called the HomeNet Protocol. It is a simple messaging system for sharing information over a variety of transport methods. The HomeNet Protocol uses light weight packets with a 8 byte header and a 2 byte checksum which is just a fraction the length of other protocols like TCP/IPv4. This balances features with overhead and still supports the ability for different versions, packet types, priorities and encryption.

It was inspired by the packets of Internet-0. I broke away from having direct compatibility with IPv4 and that simplified things and allowed for an even smaller packet. I shrunk the node address length from 32 bits to 12bits to save memory but it reduces the number of nodes in a network from billions to a maximum of 4096 nodes but this still more than enough nodes for the this first generation. Most nodes will be behind some sort of gateway/firewall and do not need direct access to the Internet. *Public Nodes* will have dual address: a local HomeNet address and a remote Internet Address and can translate data between protocols.

HomeNet Packet Format

Packet Length
Sending the packet length first allows the serial port or wireless radio to know how many bytes of data to look for.

Packet Version
This allows future changes to the protocol without breaking backwards compatibility

Packet Type
Currently there are three supported types:

00: TCP*- Reliability is required. Nodes send a reply or an ACK packet back to the sender to acknowledge that it was received. This way lost packets can be resent or an error is generated and logged.

01: UDP- Reliability is not required. This is for sensors that regularly report data and the occasional lost data isn't a concern.

10: Broadcast* - A packet is sent to all nodes until it's *time to live* (TTL) becomes 0 . The *To Devices* field serves as TTL and the *To Node* field serves as an optional Group ID .

ACK flag*
Marks that a packet is a reply to a previous packet. The packet contains an extra byte after the ID containing the ID of the previous packet.

Encrypted Flag*
Encryption hasn't been implemented yet but AES encryption is planned.

Packet Format

8 bits	Packet Length
2 bits	Version
2 bits	Packet Type
1 bit	Reply/ACK
1 bit	Encrypted
1 bit	0:Automatic/1:Manual
1 bit	Priority
12 bits	From Node
4 bits	From Device
12 bits	To Node/TTL
4 bits	To Device
8 bits	Packet ID
8 bits	*Reply/ACK ID (only if it has flag)*
8 bits	Encrypted Payload Length (only with Encrypt Flag)
8 bits	Command/Payload Type
up to 48 bytes	Payload
16 bits	CRC16 Checksum

When this flag is set, it means that the rest of the packet is encrypted starting with the "FromNode" field. AES Encryption is done in 128bit (16 byte) blocks so smaller payloads will require padding. to account for this, there is an addition byte added after the ID to indicate the new payload size and any excess space inside the block is filled with random data.

Automatic/Manual flag
This flag marks when a packet is a manual user override versus one that is sent automatically by a device. This way if someone manually turns a light on, the system has to check if it has permission to override a manual settings like turning that light off after 30 minutes of no activity.

*Priority flag**

This flag marks when a packet should take priority over other packets

Some actions like a light switch turning on a LED light need to happen immediately where as others like sensors reporting scheduled data can be deferred a few seconds without harm.

To/From Node

The node address 0-4095 (0x0 to 0xFFF in hexadecimal) Some typical addresses:

0xFF (255) is usually the primary *Public Node*
0xFFF (4095) is usually the primary *Master Node* (like HomeNet.me)

To/From Device

Nodes will rarely have a just a single device attached. These field provides a standard way of accessing different devices attached to a node

Device 0 is always the node it's self for accessing battery level and other node properties

Packet ID

This allows the node to tell the difference between two identical packets received in close proximity. it also ensures that even identical encrypted packets will look different.

ACK/Reply ID

This contains the id of the previous packet that the node is replying to so the sender node can confirm that the packet it sent arrived intact and it knows that this packets contains the response.

Command

The command informs the node how to process the data that the payload contains.

See Appendix A for the full list of commands

Payload

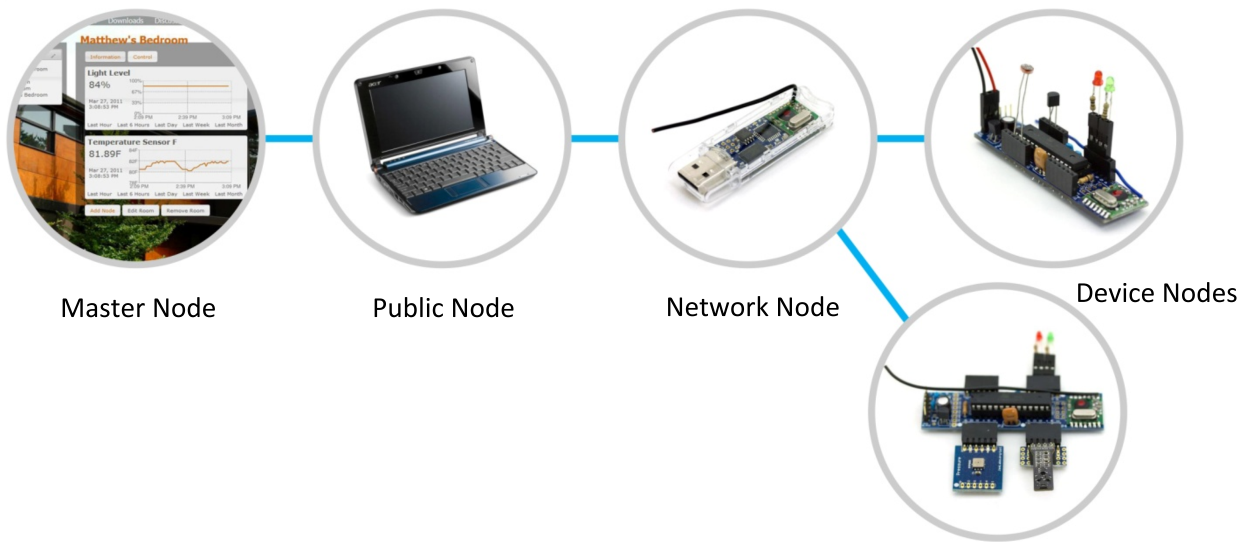
This part of the packet contains the data being sent and can be up to 48 bytes long.

Checksum

Provides a fast way to validate that the packet arrived intact and without errors.

*planned but not implemented yet.

Hardware of HomeNet



Nodes

Nodes provide the brains for the devices and link them into HomeNet. They translate HomeNet commands into actions for the devices and read data from devices and send it to other nodes and devices. In my project, these nodes are based on JeeNode boards, a cheaper Arduino derivative that also incorporates a low cost wireless transmitter. Future nodes could be based on ARM, PIC or other microcontrollers. Nodes can be broken out into 4 types based on their roles: *Device Nodes*, *Network Nodes* and *Public Nodes*, and *Master Nodes*. Depending on a nodes configuration, it might have multiple roles.

Device Nodes

Device Nodes are remote nodes around the home that gather data from sensors or control actuators connected to them. They connect wired or wirelessly to a *Master Node* that tracks and collects the nodes' data. Future versions of HomeNet could implement a mesh network to build a more robust network.

Network Nodes

Network Nodes translate packets between different transport methods. For example, from a wireless network to a wired and back. These nodes may also have the capability to store packets in the case of a connection failure. In my HomeNet prototype, a network node links your HomeNet network to your PC via USB.

Public Nodes

Public Nodes are nodes directly connected to the Internet. These could be in the form of an App running on your PC or it could be a node with a built in Ethernet card. It provides a link for HomeNet *Device Nodes* to send data to the cloud to be stored and archived. The current *Public Node* was written in Processing (Java) and running on a netbook.

Master Node

The *Master Node* tracks and monitors the health of nodes and stores the collected sensor data and schedules for devices that have been programmed. In my prototype, this is running on the HomeNet.me website, but in the future, the task of the master node could be running on hardware inside of the home.

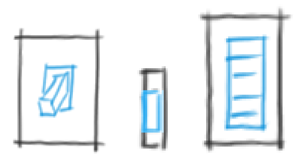
Devices

HomeNet can interface with practically any device in the home. Devices are any smart objects, sensors and actuators that you wish to integrate into HomeNet. They can be inputs like temperature and humidly sensors or outputs like lights, window blinds, and your air conditioner.

Control Devices

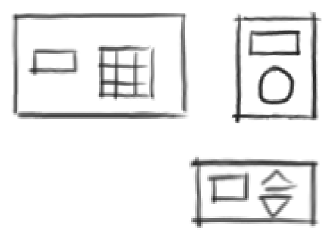
Control devices vary from being very simple switch to your cell phone running an app that can control and interact with every device in the home.

Light Switch, Door bell, Button panels



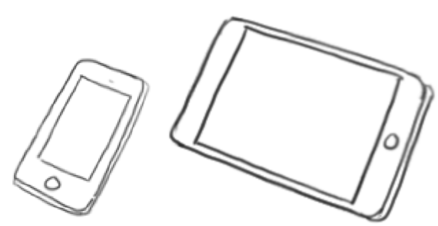
These simple controls are no longer hardwired to devices, rather they are programmed to meet your needs and can be reprogrammed if your needs change. You can control one device or one hundred devices and could change based on the time of day or what is going on the home

Multifunction Panels: Thermostat, Security Panel, Sprinkler System Controller, Pool Timer



These typically have been self contained devices that had their own user interfaces. In a smart house these devices break free of their housing and join a larger network so you can program these devices from more user friendly devices like your laptop or cell phone.

Portable: Cell phone, Tablet



These devices can interact with any devices in the home and can act as a universal remote to the home.

These devices connect directly to the master node to access all of the devices in the network.

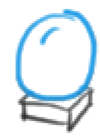
Total Control: PC, Program Hardware, Visit Website

A PC is the ultimate controller in the system. Not only can you view and interact with the whole system, you can reprogram the hardware and change how the system works.

Alerts
Quickly understand what is going on the home in new ways.

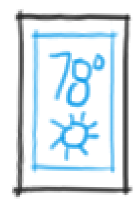
Check House Light
Inspired by the check engine light in your car, it serves as a visual reminder to check on your house and can alert you to for maintenance, bad air quality and broken devices

Ambient Information
Originating from a project at MIT, it provide information through color that can be casually observed. For example, it could change from yellow to blue to indicate the current weather or glow green to red depending on the homes energy use. A trial in Canada of a similar device that showed energy use, lead to an average 6.5% energy savings.[12]

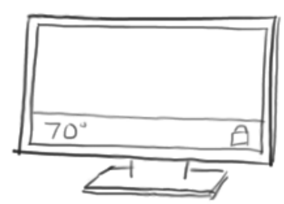


Audio Alerts: Buzzer, Alarm
In case of an emergency, like a fire, alarms can sound through the house.

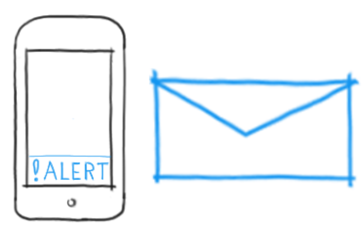
Small Screens: Digital Photo frames, Weather displays
Small digital displays are becoming increasing common in the home. A digital photo frame could also display real time information about the home.



Television
Information about the home could be overlaid on the screen, for example showing video from the front door when the door bell rings.



Digital Alerts: Cell Phone, Email, Social Network
If your away from home, you can be quickly be alerted via text message, email, or post to social media sites like Facebook and Twitter.



[12] Eiden, Joshua M. *Investigation Into The Effects Of Real-Time, In-Home Feedback To Conserve Energy In Residential Applications* http://www.currentcost.net/University%20of%20Nebraska.pdf 5/1/2011, 31

Room Sensor

A room can be made smart by installing just a single device that has collection of different sensors. It adds additional functionality to a smoke detector that would already need to be installed.

Occupancy Sensor

Detect when people are actually in a room. When the room is empty lights can be turned off and the HVAC system adjusted.

Light Sensor

Lighting in the room can be adjusted based on natural light. Based on other sensors, blinds could be opened or additional lights turned on.

Sound Sensor

Adjust the volume of media in different rooms, or listen for voice commands

Temperature Sensor

Serves as a thermostat for HVAC system so that room is only heated and cooled as much as needed.

Flame/Smoke Sensor, Carbon Monoxide

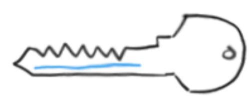
Can trigger alerts and alarms throughout the house if something dangerous is detected

Access Control

In a smart house it becomes economical to implement robust access control that used to be only available to businesses and institutions.

Digital Keys

Keys can be individually coded so that lost keys are merely an inconvenience and not a security concern

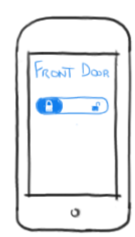


Wireless FOB

Inspired by remote unlocking for your car, it brings the same convenience to your front door

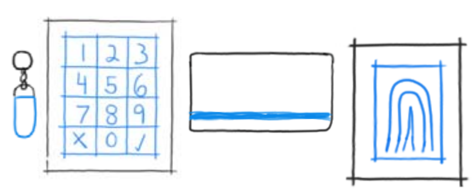
Cell phone

Your cell phone can act like a key, letting you unlock the door for yourself or friends who are locked out. It is inspired is by the new apps that automobile manufacturers are building.

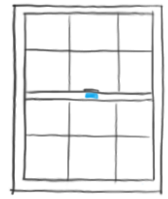


Others: RFID, Keypad, Swipe Card, Biometric Access

The possibilities are limitless. Many choices are available based on convenience or amount of security desired.

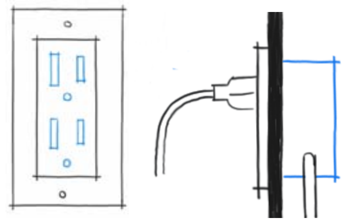


Window



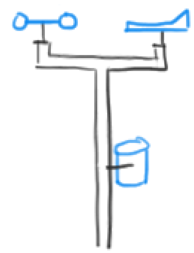
It could contain sensors to let you know if it is closed and locked or trigger an alarm if it is shattered. Sensors are integrated directly into the frame of the window, simplifying the installation process and making it harder for security measures to be sabotaged. In combination with other devices, it could alert you if window is open when AC should be on or let you know that the window is open and it's about to rain. Window blinds and louvers can be controlled maximizing the use of natural light.

Smart Power Outlet



Enables the Remote control and energy monitoring of power outlets. The system can turn them off if no one is home and has a built in power monitor that can report exactly how much energy you are using.

Weather Station



Get a precise weather forecast. It has a range of sensors like temperature, humidity, barometric pressure, anemometer (wind speed), wind vane (wind direction), a rain gauge and more. It can help determine whether to water the lawn or wait a day and see if it rains. It can monitor outdoor air quality and pollen counts and recommend when it's a good day to open the windows. All of the meteorological data it generates can be shared on the cloud to help build better weather forecasts the whole community.

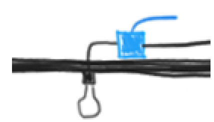
Home Power Generation

Monitor your home's power production from solar panels. Track your power production from your cell phone. Quickly detect when a panel is underperforming and needs maintenance.

Smart Appliances

Monitor their performance and health to prevent untimely break downs. Appliances can get real-time power rates and can choose to defer energy use until off-peak times in exchange for a reduced power rate. For example, if you drive an electric car and it takes 6 hours to charge, you will hook it to charge when you get home but it could defer charging several hours until off peak and still be ready by morning.

Lighting



Lighting can be tied to sensors in the room so only as much light as needed is used. Smart home network also enables cool new lighting opportunities like using RGB LED's that contain a red, green and blue LEDS to reproduce almost any color that you can set from your cell phone.

```
class HomeNet{

public:
    HomeNet(uint16_t id);
    void init(HomeNetPort* [], uint8_t, HomeNetDevice* [],uint8_t);
    void receive();
    HomeNetPacket *getNewPacket();
    HomeNetPacket* clonePacket(HomeNetPacket*);
    boolean process();
    boolean processPacket(HomeNetPacket*);
    boolean processCommand(HomeNetPacket*);

    HomeNetPayload packetToPayload(HomeNetPacket*);

    HomeNetPacket* addTcpPacket(const uint8_t, const uint16_t, const uint8_t, const
    HomeNetPacket* addUdpPacket(const uint8_t, const uint16_t, const uint8_t, const
    HomeNetPacket* addBroadcastPacket(const uint8_t, const uint8_t, const HomeNetPay
    HomeNetPacket* addCrc(HomeNetPacket*);
    void debugPacket(HomeNetPacket*);
    int availableMemory();
    void debugPayload(const HomeNetPayload&);

    inline void registerStatusLights(HomeNetDeviceStatusLights& l) { statusLights =

    void registerSchedule(HomeNetSchedule[], uint8_t );
    void registerInterrupts(HomeNetInterrupt[], uint8_t );
    void deviceUpdate();
    void deviceSchedule();
    void deviceInterrupt(boolean = false);
    void loop();
    void sleep(uint8_t = 0);
    uint16_t getDeviceId(uint8_t);
    inline uint8_t getVersion(){ return 0x01; };

    uint16_t _nodeId;
    HomeNetDeviceStatusLights * statusLights;
    volatile boolean runInterupt;

private:
    HomeNetPort** _ports;
    uint8_t _portCount;
    HomeNetDevice** _devices;
    uint8_t _deviceCount;
    uint8_t _uniqueId;
    HomeNetPacket _packets[PACKET_BUFFER];
    HomeNetSchedule * _deviceSchedule;
    uint8_t _deviceScheduleCount;
    HomeNetInterrupt * _deviceInterrupts;
    uint8_t _deviceInterruptCount;
    uint8_t _scheduleCount;
    long _deviceTimer;
    uint8_t _getId();
};
```
Example of the code in HomeNet code library.

INSTALLATION

New Construction

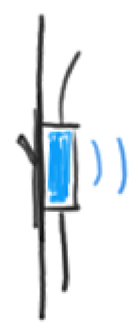
Starting from scratch allows a smart home system to be used to its full potential. With a HomeNet System, the wiring in the home is simplified. Devices no longer have to be hard wired together, lights no longer have to be wired directly to a wall switch for control and it reduces the amount of expensive copper wiring required. Sensors can be embedded deep into the structure so that the system can monitor for things like structural instability, moisture intrusion, insect infestation, and leaking pipes.
In new construction, it is preferred to use a simple wired network rather than go wireless so you don't have to worry about replacing batteries , interference with neighbors and provides better security against outside intrusions and snooping.

Any additional installation and setup costs can be minimized by using prefabricated construction. Sensors and actuators can be integrated directly into components like doors and windows in the factory rather than during construction on the job site. Nodes and sensors can be installed and setup inside the factory and are ready to start working as soon as the building is assembled.

Designers should keep in mind that the system will require maintenance and some component may need to replaced over time. Nodes and devices should be installed where they can be accessed in the future if needed, especially until the technology is proven. Motorized actuators are likely to wear out over time and sensors might break and fail, especially those exposed to the elements.   Designers should be conscious of the systems needs and place them where they won't need any additional access covers like installing them behind planned electrical outlets and light switches.

Retrofit

A smart house doesn't need to be built from scratch. Smart components can easily be retrofitted into existing homes. In a retrofit, using wireless nodes is worth the extra expense to avoid running new wires. Nodes may be plugged in to wall outlets or could use batteries.   In the future, it could even be possible to gather energy from their surrounding environment.



Nodes and devices should be clustered together to reduce installation costs. Some devices like the *Room Sensor* replaces an old smoke detector and can easily be installed by the homeowner. Other devices, like installing smart light switches, will require an electrician.

The Virtual House, The Interface to the Home

Everything these days is being virtualized so that we can interact with them through our electronic devices. We have virtualized the art of letter writing into email and opening an encyclopedia into searching Wikipedia. We have virtualized ourselves on Facebook, creating a digital representation of ourselves on the Internet. We post what we like and dislike about our lives. We write our attributes, what we do, where we live and where we go to school. It's a database of our friends and a history of our lives. In video games we create 3d Avatars that represent ourselves as we interact in the virtual world. It's only logical to think that one day architecture is going to be virtualized into an electronic form. What will this look like? How do we interact with our home virtually? Who is going to design it?

In the Future, who will design the virtual home?

These are questions I've been trying to answer over the last semester as I start exploring how to build the user interface for HomeNet. I built a website, http://HomeNet.me that allows you to interact with your smart house from anywhere. I was inspired by Facebook and wanted to build a website that allows you to create a virtual profile of your home. It allows you to create a detailed profile of your home listing all of its rooms and properties and allow you to view and control all of the smart HomeNet devices you have. You can personalize each page with photos of your house creating a unique personal link to your home.

TOOLS

Many open source products were used to quickly build and prototype the HomeNet web interface

Backend

These are tools running on the cloud that generate the webpage you see in your web browser.

Apache

an open source web server used by 62% of the websites on the web[13]

http://httpd.apache.org/

PHP

an open source programming language for creating dynamic websites

http://php.net

mySQL

an open source database (used by Facebook, Twitter)

http://mysql.com/

Zend Framework

an open source MVC framework for PHP that provides a collection of reusable code components.

http://zendframework.com

Front End

These are tools that enhanced the user interface with dynamic elements and simplify the programming required.

Google Chart API

Provides a easy interface for creating graphs from raw data.

http://code.google.com/apis/chart/

jQuery

an open source JavaScript toolkit for manipulating the code on a webpage.

http://jquery.com

jQueryUI

an extension of jQuery that focuses on reusable User Interface widgets.

http://jqueryui.com

JQuery Mobile

an extension of jQuery that provides tools for creating mobile web apps.

http://jquerymobile.com

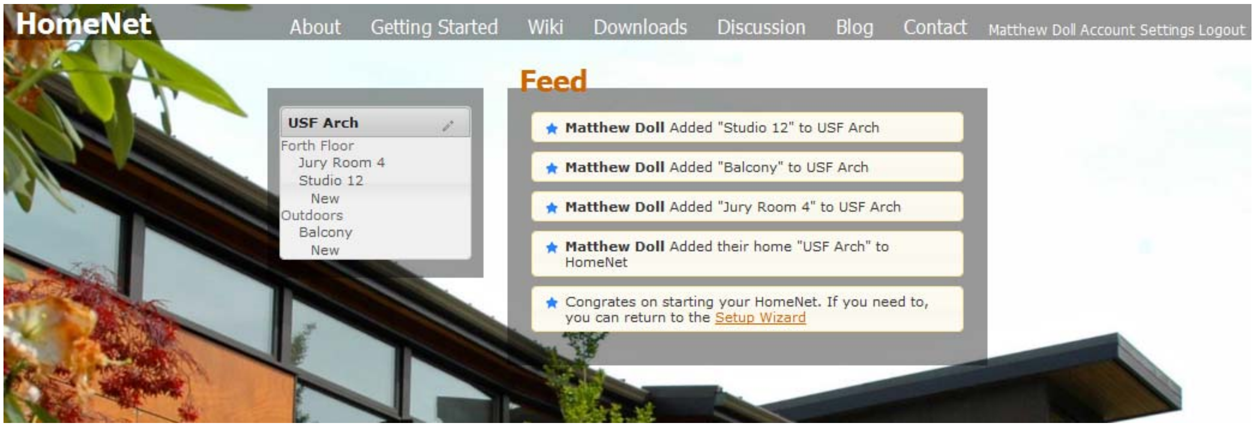
[13] http://news.netcraft.com/archives/2011/05/02/may-2011-web-server-survey.html (5/3/2011)

User Interfaces

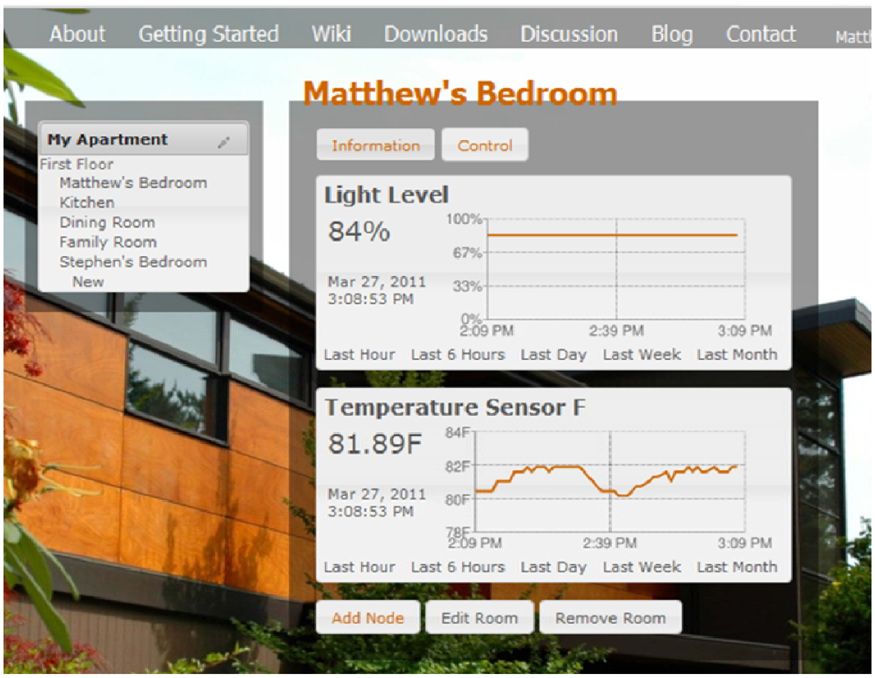
Website

http://HomeNet.me

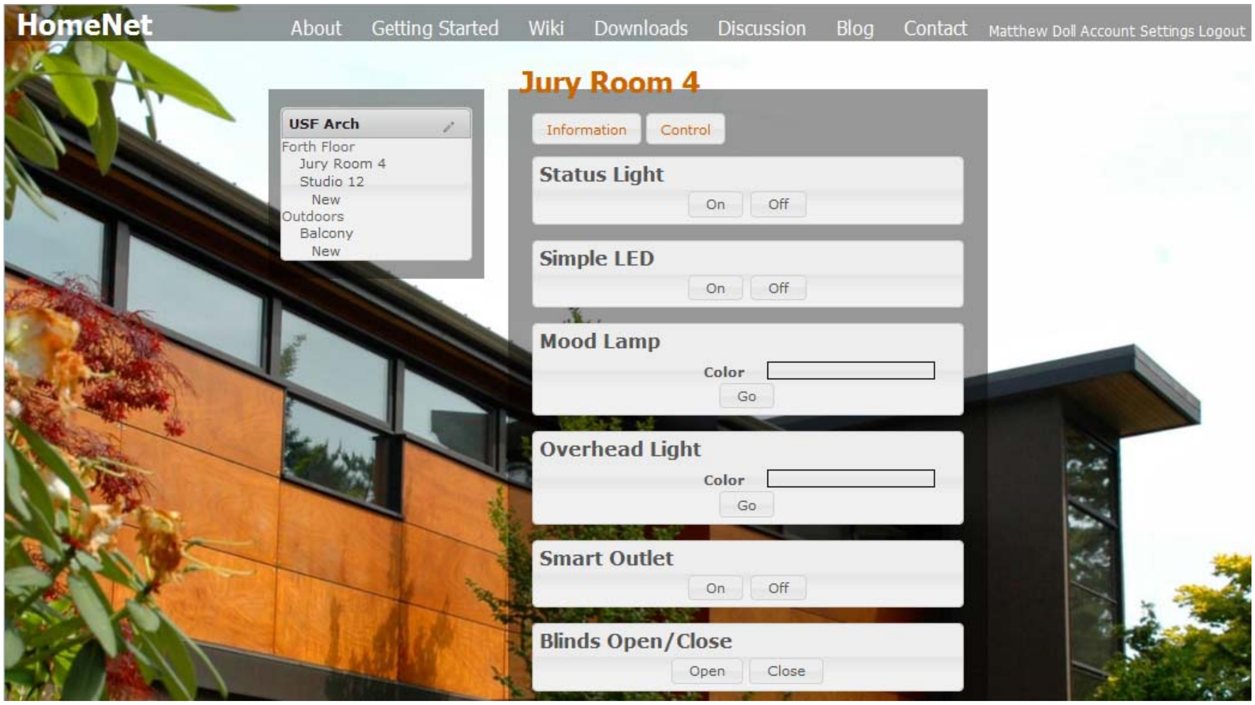
It is your gateway to your home. It provides an easy interface to setup and configure your HomeNet. It serves as the *Master Node*, maintaining the data collected by your home and generates detailed reports and graphs so you can understand how your house is performing. The website also helps generate the code needed to ease the setup process for inexperienced users.



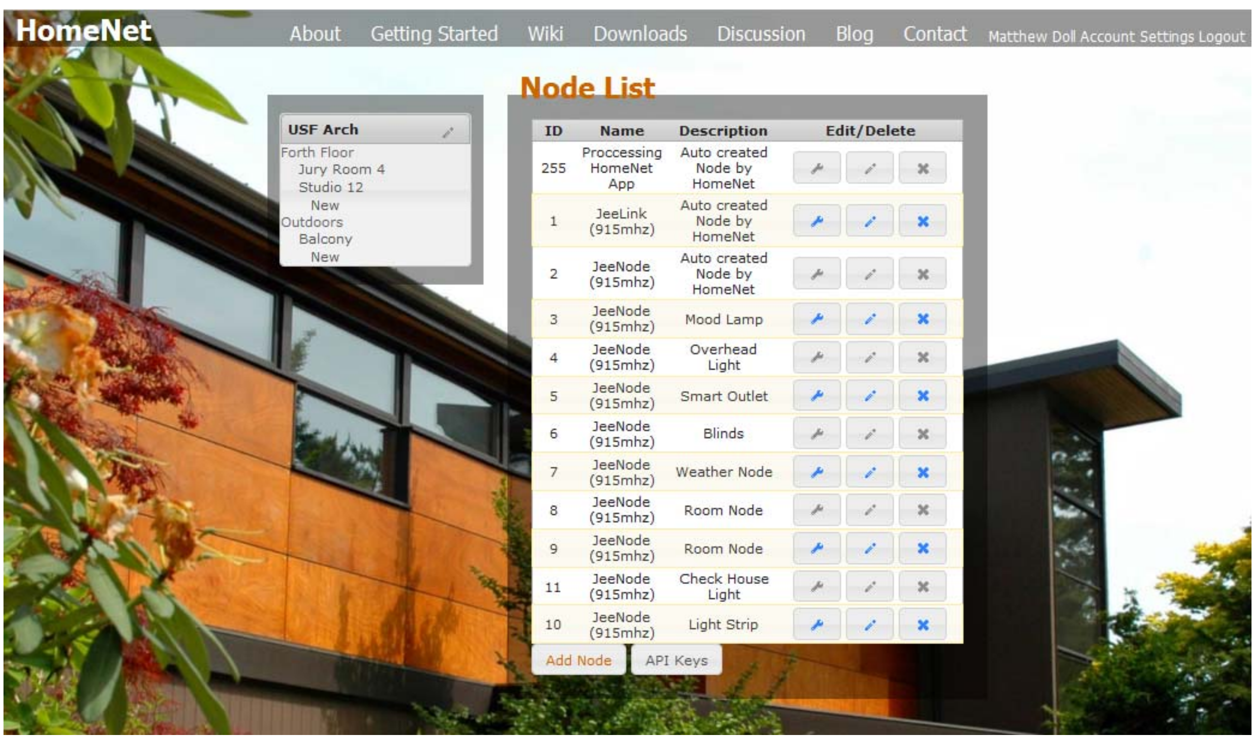
Home front page, with the latest news alerts



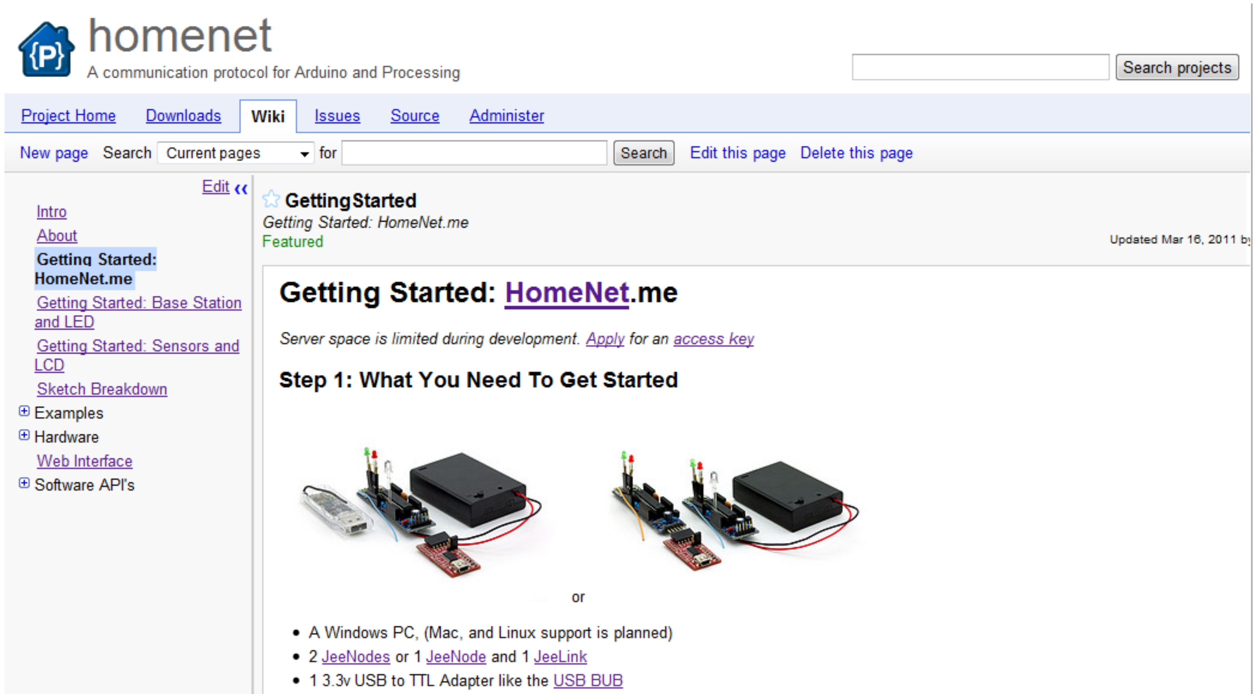
Room information graphs



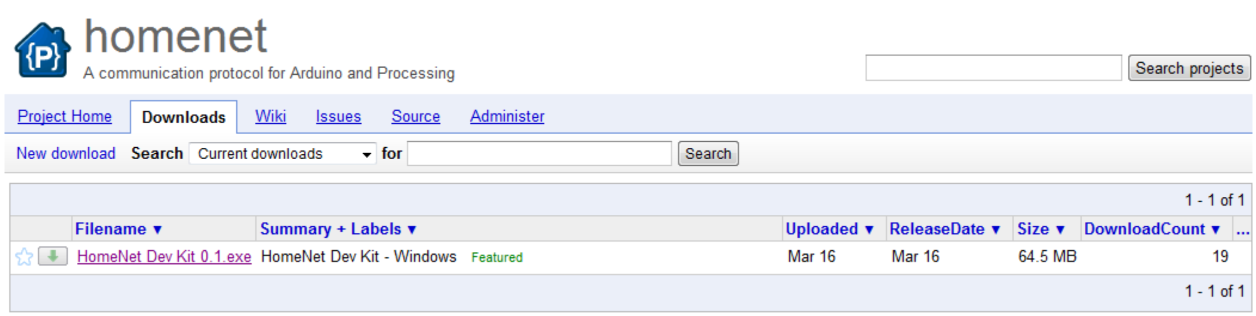
Control devices in a room



Management of nodes



Documentation is online in wiki format at http://homenet.googlecode.com



All of the source code can be downloaded at http://homenet.googlecode.com

Cell Phone

http://m.HomeNet.me

The cell phone interface serves as a universal remote for the home. Quickly get information about your home from anywhere. Emergency alerts can be sent to your phone instantly so you stay informed. You can check on your house anytime, anywhere giving you a new piece of mind.

Social Media

There are a lot of interesting opportunities to integrate social media into the virtual home. You could get status updates from your house in your "newsfeed". Energy use could be shared online and you can compete against other family members or friends to see who is the greenest.[14]

[14] http://www.readwriteweb.com/archives/6_ways_to_live_better_inside_an_internet_of_things_homes.php

The Future, Home 2.0

Only time will tell what the future holds but one thing is for sure, The future home is going to be full of technology. How will this affect us? For some, technology is seen as a beneficial advancement but others might see it as stressful nuisance. Integrating technology into the home is going to happen whether we like it or not. It is the architect's responsibility to make sure it is done properly. Smart home technology allows us to create a dialog with the user in a way we haven't been able to do before and create homes that can truly adapt to the user's needs. This new architecture will help make people more aware of how they are living and help them to live better, more environmentally friendly lives.

The future architect will need to design how one interacts with all the electronics and technology that the future home will contain. I think that the future architect needs to look beyond just the physical aspects of a smart house and also look at the virtual interface to the house. The virtual world is currently dominated by engineers and graphic designers. I think that architects need to get involved so that people have the same quality experience in the virtual home as they do in the physical home. I'm not sure if it's the architects role to build the virtual home but I think that the architect needs to represent the owners interests in its construction the same way an architect works with a contractor to build a physical building.

It's exciting to look at the potential future of HomeNet. It shows the future smart house is just around the corner. We need to carefully plan and make sure the smart houses we design will meet our needs for the next 50 years. They need to be secure, robust and upgradeable. It is easy to ignore security but doing so could have devastating consequences. Security needs to be designed in to the system from the ground up. Criminals are increasingly tech savvy and if a smart house is on the internet, they could hack in to it and analyze the owners patterns to find the perfect time to unlock the front door and rob the house. Another concern is protecting the owner's privacy. A smart house will collect a lot data about its occupants. Who has a right to access it? Is it protected by the owner or can the data be subpoenaed in a criminal investigation? I think new laws will be needed to define privacy boundaries in this new age. Lastly, we need to be concerned how this technology will change our physical interaction with architecture. It needs to add to the architecture of the home instead of replacing it. The future designer will challenged to find the right balance.

Works Cited

4dsocial: Interactive Design Environments, edited by Lucy Bullivant Wiley, 2007.

4dspace : Interactive Architecture. Architectural Design (London, England : 1971) ; v. 75, no. 1., edited by Lucy Bullivant. Chichester: Wiley-Academy, 2005.

"An Electronic Paradise." House Beautiful 138, no. 11 (Nov., 1996): 204-205.

Ander, Gregg. "Connectivity for Smart Buildings." Architectural Lighting 21, no. 2 (2007): 25.

Berl, Andreas, Roman Weidlich, Michael Schrank, Helmut Hlavacs, and Hermann de Meer. "Network Virtualization in Future Home Environments." In Integrated Management of Systems, Services, Processes and People in IT, edited by Claudio Bartolini and Luciano Gaspary. Vol. 5841, 177-190: Springer Berlin / Heidelberg, 2009.

Bettinelli, Eugenio. "Living with Home Automation." Abitare no. 388 (1999): 208.

Bowen, Ted. "As Prefabrication Sheds its Off-the-Rack Image, Automation Via 3D Printing Threatens to Transform Conventional Construction." Architectural Record 195, no. 4 (2007): 166.

Bullivant, Lucy. Responsive Environments : Architecture, Art and Design. V & A Contemporary. London; New York: V & A Publications; Distributed in North America by Harry N. Abrams, 2006.

Burke, Anthony and Therese Tierney. Network Practices : New Strategies in Architecture and Design. 1st ed. New York: Princeton Architectural Press, 2007.

Capone, A., M. Barros, H. Hrasnica, and S. Tompros. "A New Architecture for Reduction of Energy Consumption of Home Appliances." Proceedings of the European Conference Towards eEnvironment (March, 2009).

Cox, Christopher Emilio Emiliucci. Living Chassis : Learning from the Automotive Industry; Site Specific, Prefabricated, Systems Architecture [Electronic Resource]. Tampa, Fla: University of South Florida, 2008.

De Botton, Alain. The Architecture of Happiness. 1 American ed. New York: Pantheon Books, 2006.

de Milleville, Hugues. "Smart Houses." The Canadian Architect 41, (1996): 33.

Flade, Antje. "No Place Like Home." Scientific American Mind (February/March 2007, 2007): 70.

Fox, Michael 1967 Aug and Miles Kemp. Interactive Architecture. New York: Princeton Architectural Press, 2009.

Gershenfeld, Neil, Stephen Samouhos, and Bruce Nordman. "Intelligent Infrastructure for Energy Efficiency." Science, 327, (2009): 1086-1088.

Haeusler, M. Hank. Media Facades : History, Technology, Content. Ludwigsburg: avedition, 2009.

He, Mike M., Evan M. Reutzel, Xiaofan Jiang, Randy H. Katz, Seth R. Sanders, David E. Culler, and Ken Lutz. "An Architecture for Local Energy Generation, Distribution, and Sharing." IEEE Energy2030 (November, 2008).

Iwamoto, Lisa. Digital Fabrications: Architectural and Material Techniques. Architecture Briefs. New York: Princeton Architectural Press, 2009.

Joch, Alan. "Digital Architect: Digital Tools for Age-Smart Housing." Architectural Record 190, no. 7 (July, 2002): 161-162.

Klooster, Thorsten, Niels Boeing, Simon Davis, and Almut Seeger. Smart Surfaces : And their Application in Architecture and Design. Basel ; Boston: Birkhäuser, 2009.

Kolarevic, Branko and Kevin Klinger. Manufacturing Material Effects: Rethinking Design and Making in Architecture. First Edition ed. Routledge, 2008.

Kolarevic, Branko. Architecture in the Digital Age : Design and Manufacturing. New York, NY: Spon Press, 2003.

Kronenburg, Robert. Flexible : Architecture that Responds to Change. London: Laurence King, 2007.

Kurokawa, Noriaki. Taisei Overseas System: (TOS): Reference: Modular Components for Construction of Living Units.

Leach, Neil, Neil Leach, and RIBA Future Studies. Designing for a Digital World. Chichester: Wiley-Academic, 2002.

Lee, Hui Lin and Yudong Liu. Defining Digital Architecture : 2001 Far East International Digital Architectural Design Award. Basel ; Boston: Birkhäuser, 2002.

LeMay, Michael, Jason J. Haas, and Carl A. Gunter. "Collaborative Recommender Systems for Building Automation." Proceedings of the 42nd Annual IEEE Hawaii International Conference on System Sciences (HICSS '09) (January, 2009).

LeMay, Michael, Rajesh Nelli, George Gross, and Carl A. Gunter. An Integrated Architecture for Demand Response Communications and Control. Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences. IEEE Computer Society, 2008.

Merz, Hermann, Thomas Hansemann, and Christof Hübner. Building Automation: Communication Systems with EIB/KNX, LON and BACnet. Translated by James Backer, Viktoriya Moser and Leena Greefe. First ed. Springer, 2009.

Norman, Bruce. "Networks in Buildings: Which Path Forward?" Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings (August, 2008).

Novitski, B. J. "Wired for Living, Houses Go High-Tech." Architectural Record 188, no. 4 (2000): 173.

O'Sullivan, Dan, Tom Igoe, and Inc NetLibrary. Physical Computing [Electronic Resource] : Sensing and Controlling the Physical World with Computers. Boston: Thomson, 2004.

Pfammatter, Ulrich. Building the Future : Building Technology and Cultural History from the Industrial Revolution Until Today. Munich ; New York: Prestel, 2008.

Piette, Mary Ann, Sila Kiliccote, and Girish Ghatikar. Linking Continuous Energy Management and Open Automated Demand Response, 2008.

Stadler, Ingo. "Power Grid Balancing of Energy Systems with High Renewable Energy Penetration by Demand Response." Utilities Policy 16, no. 2 (June, 2008): 90-98.

Staub, Robert. "DigitalSTROM: Building Automation using High-Voltage Technology." Detail Green no. 2 (Nov., 2009): 56-57.

Sullivan, C. C. "Robo Buildings: Pursuing the Interactive Envelope." Architectural Record 194, no. 4 (2006): 149.

Wacks, Kenneth. "Home Area Networks for Electricity Demand Management," IHomes and Buildings 5, no. 2 (Summer, 2008): 15-17.

Appendix A: HomeNet Commands

id	Name	Type	Reply Type	
0x00	ERROR	STRING	ACK	Error message
0x01	VERSION	FLOAT	ACK	Get current version of firmware on the node
0x03	BATTERY LEVEL	INT	ACK	Get current battery level
0x04	FREE MEMORY	INT	ACK	Get free amount of memory
0x33	PING	STRING	PONG	Send Test Ping, node should reply back pong
0x3e	PONG	STRING	ACK	Response to a Ping Command
0x11	ACK	BYTE	ACK	letting the sender of a packet know that data arrived safely
0x21	GET NODE ID	INT	ACK	Get Node's current ID (Used with node 0 broadcast)
0x22	SET NODE ID	INT	ACK	Changes Node ID, used for initial setup
0x23	GET DEVICE	BYTE	ACK	Get device id code
0x24	SET DEVICE	BYTE	ACK	Set device code (future use)
Auto Commands				
0xB1	AUTO SEND START	BYTE	ACK	Get device id code
0xB2	AUTO SEND STOP	BYTE	ACK	Set device code (future use)
Simple Commands				
0xC0	ON/OPEN	BYTE	ACK	Turn Device on
0xC1	OFF/CLOSE	BYTE	ACK	Turn Device off
0xC2	LEVEL	BYTE	ACK	Set device light level
0xC3	CLEAR	BYTE	ACK	Clear device
Get Types				
0xD0	GET VALUE	BYTE	REPLY VALUE	Get value in native format from subdevice byte
0xD1	GET BYTE	BYTE	Varies	Get value
0xD2	GET STRING	STRING	Varies	Get value
0xD3	GET INT	INT	Varies	Get value
0xD4	GET FLOAT	FLOAT	Varies	Get value
0xD5	GET LONG	LONG	Varies	Get value
0xD6	GET BINARY	BINARY	Varies	Get value
0xD7	GET BOOLEAN	BOOLEAN	Varies	Get value
Set Types				
0xE0	SET VALUE	BYTE	ACK	Set raw value
0xE1	SET BYTE	BYTE	ACK	Set value
0xE2	SET STRING	STRING	ACK	Set value
0xE3	SET INT	INT	ACK	Set value

0xE4	SET FLOAT	FLOAT	ACK	Set value
0xE5	SET LONG	LONG	ACK	Set value
0xE6	SET BINARY	BINARY	ACK	Set value
0xE7	SET BOOLEAN	BOOLEAN	ACK	Set value

Reply Types

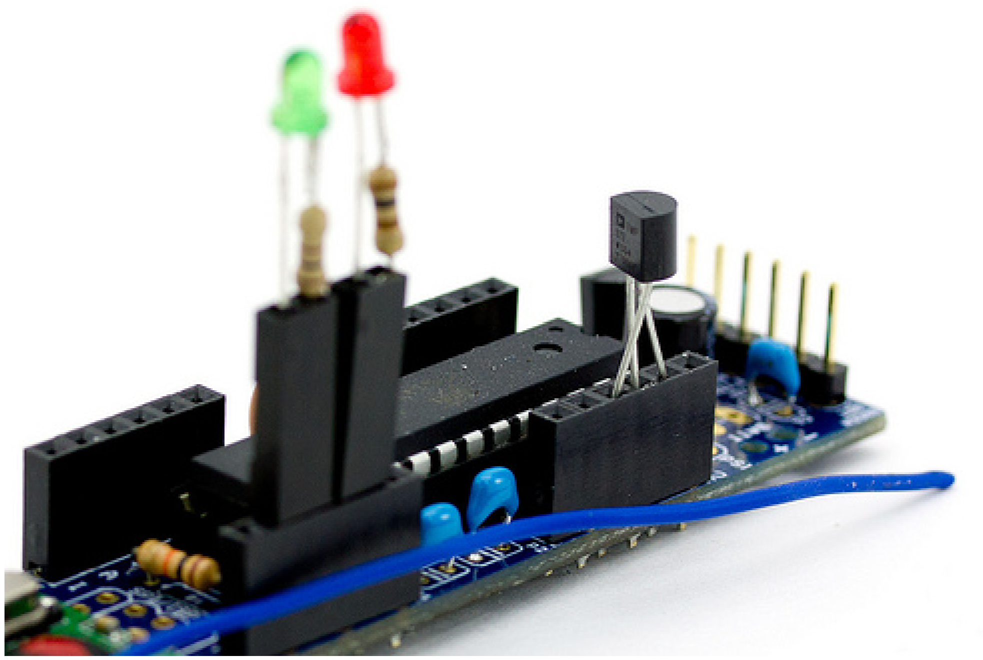
0xF0	REPLY VALUE	BYTE	ACK	A reply value
0xF1	REPLY BYTE	BYTE	ACK	A reply value
0xF2	REPLY STRING	STRING	ACK	A reply value
0xF3	REPLY INT	INT	ACK	A reply value
0xF4	REPLY FLOAT	FLOAT	ACK	A reply value
0xF5	REPLY LONG	LONG	ACK	A reply value
0xF6	REPLY BINARY	BINARY	ACK	A reply value
0xF7	REPLY BOOLEAN	BOOLEAN	ACK	A reply value

Appendix B: Example Node Code

```
// Test of TMP37 and LDR sensors.
// <mdoll@email.usf.edu> http://opensource.org/licenses/mit-license.php

#include <Ports.h>
#include <RF12.h>
#include <HomeNet.h>
#include <HomeNetDevices.h>

//Start HomeNet Packet Stack
HomeNet stack(0x02);//0x01 is RF12 base station //0xFF is PC uplink

//Setup Network Adapters
HomeNetPortRF12   portRF12(stack, SEND_RECEIVE, RF12_915MHZ, 33);

//Setup attached devices
HomeNetDeviceJeeNode jeeNode(stack);
HomeNetDeviceStatusLights statusLights(stack);
HomeNetDeviceTMP37 tmp37(stack);
HomeNetDeviceLDR ldr(stack);

//package the setup info in a nice neat arrays
HomeNetPort * ports[] = {&portRF12};
HomeNetDevice * devices[] = {&jeeNode, &statusLights,  &tmp37, &ldr,};

//delay (sec), frequency (sec), device, sendToNode, sendToDevice, Command, 
Payload
HomeNetSchedule schedule[] = {{5,10,&tmp37,4,2,CMD_GETVALUE,0},
                              {0,10,&ldr,4,2,CMD_GETVALUE,0}};

void setup() {
  //initialize HomeNet with the setup info
  stack.init(ports, sizeof(ports)/2, devices, sizeof(devices)/2);
  stack.registerStatusLights(statusLights); //setup status lights
  stack.registerSchedule(schedule,sizeof(schedule)/sizeof(schedule[0]));
}

void loop() {
  stack.loop();
}
```

Appendix C: HomeNet Devices

Temperature Sensor: TMP37



Features

- Analog
- Range: 5ºC to 100ºC
- Accuracy: ±1 ºC.
- Datasheet

Installation

1. +V goes to +(3.3v)
2. Vout to A
3. GND to G.

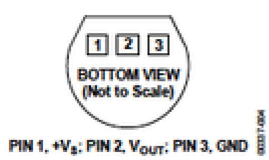


Figure 4. T-3 (TO-92)

Code

```
HomeNetDeviceTMP37(HomeNet);
```

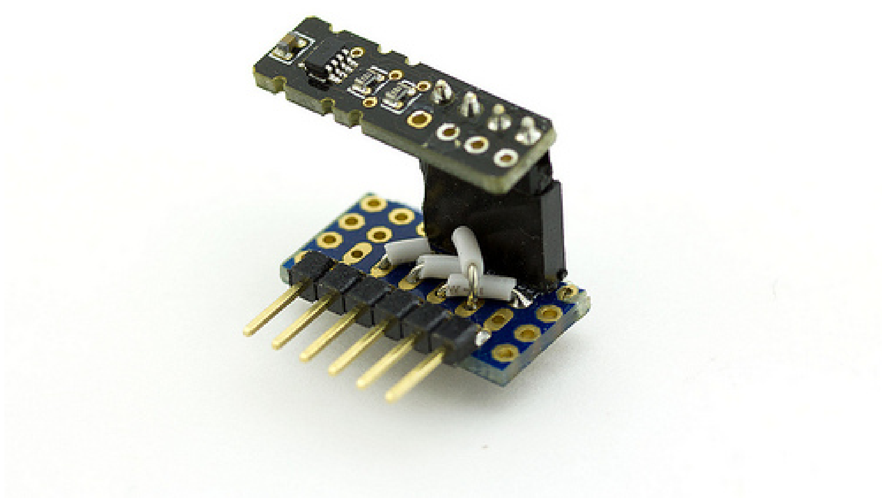
Commands

- GETVALUE
- GETFLOAT

Purchase

http://shop.moderndevice.com/products/temperature-sensor

TEMPERATURE SENSOR: TMP421



Features

- Digital i2c
- Range: –40ºC to +125ºC
- Accuracy: ±1 ºC.
- Datasheet

Installation

1. Ground to G
2. Vin to + (3.3v)
3. SDI to D (Digital)
4. SCL to A (Analog)

Code

```
HomeNetDeviceTMP421(HomeNet);
```

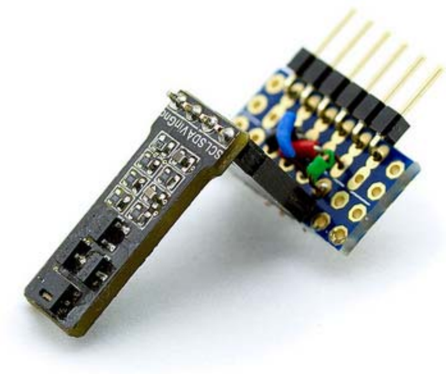
Commands

- GETVALUE
- GETFLOAT

Purchase

http://shop.moderndevice.com/products/tmp421-temperature-sensor

Humidity/Temperature Sensor: SHT21



Features

- Digital i2c
- Relative Humidity Range: 0-100% RH
- Relative Humidity Resolution: 0.04%
- Relative Humidity Accuracy: ±1 2.0%
- Temperature Range: -40°C to +125°C
- Temperature Resolution: 0.01 ºC
- Temperature Accuracy: ±.3 ºC.
- http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT21.pdf

Installation

1. Ground to G
2. Vin to + (3.3v)
3. SDA to D (Digital)
4. SCL to A (Analog)

Code

```
HomeNetDeviceSHT21(HomeNet);
```

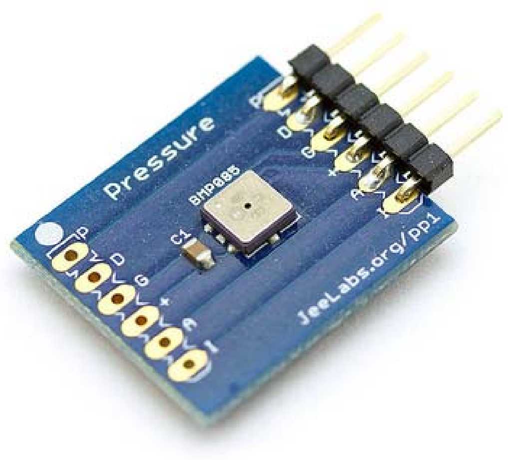
Commands

- GETVALUE
- GETFLOAT returns temp, humidity

Purchase

http://shop.moderndevice.com/products/humidity-and-temperature-sensor

BAROMETRIC PRESSURE SENSOR: BMP085



Features

- Digital i2c
- Range: 300 to 1100hPa
- Accuracy .3hPa
- Datasheet

Installation

- In standard plug form factor

Code

```
HomeNetDeviceBMP085(HomeNet);
```

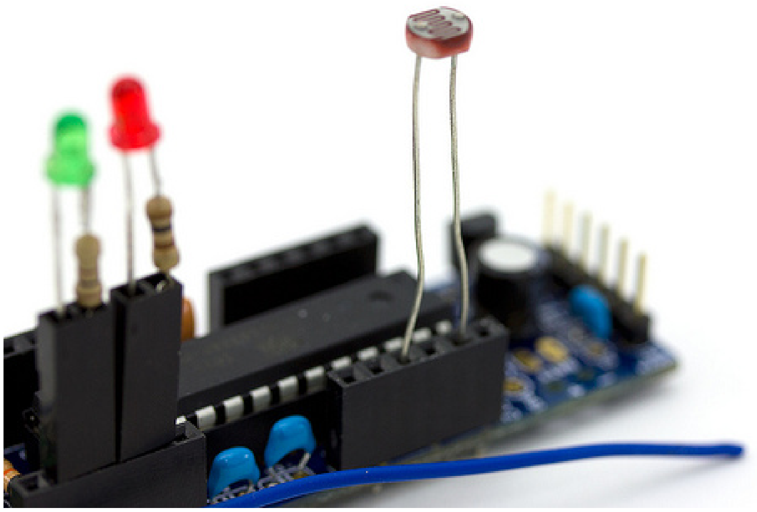
Commands

- GETVALUE
- GETLONG

Purchase

http://shop.moderndevice.com/products/pressure-plug
http://jeelabs.com/products/pressure-plug

Light Sensor: LDR



Installation

- One Leg goes to A
- Other Leg goes to G

Code

```
HomeNetDeviceLDR(HomeNet);
```

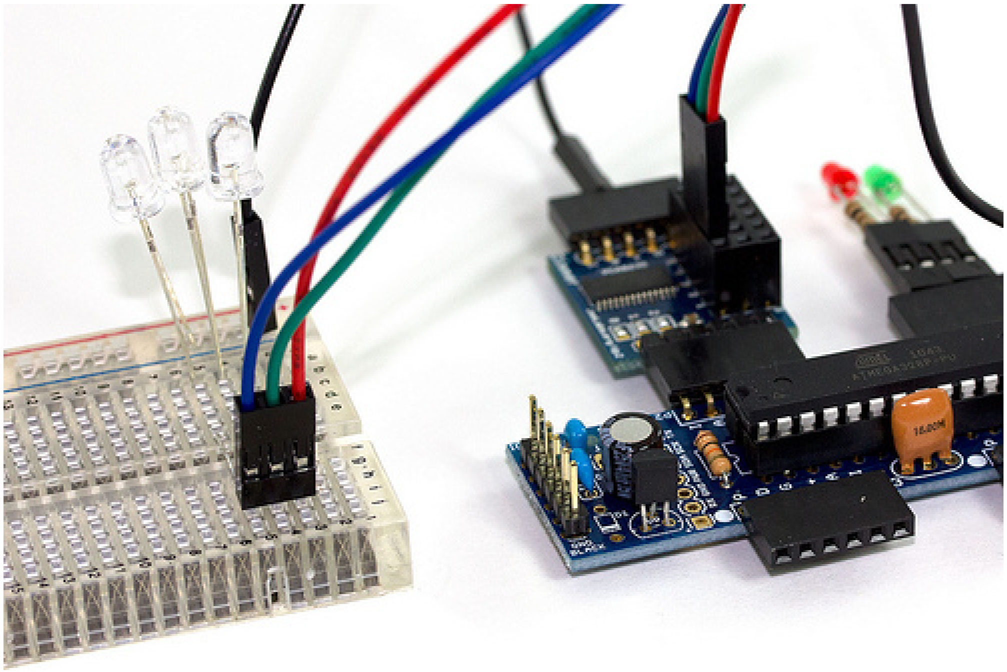
Commands

- GETVALUE
- GETBYTE

Purchase

http://www.sparkfun.com/products/9088

RGB LED



Installation

- Resistors needed to set current.
- Connect Dimmer Plug
- Red LED to 0 on Dimmer Plug
- Green LED to 1 on Dimmer Plug
- Blue LED to 2 on Dimmer Plug
- connect other leg to + or p depending on how bright you want the led

Code

```
HomeNetDeviceRGBLED(HomeNet);
```

Commands

- ON (led 0:red, 1:green, 2:blue)
- OFF (led 0:red, 1:green, 2:blue)
- SETVALUE (red, green, blue)
- SETBYTE (red, green, blue)