Introduction to OpenHAB

Kirk Carlson

Kirk.Carlson@att.net
https://github.com/kirkcarlson

What is Internet of Things (IoT)

- Sensors
  - Temperature, motion, closures, flow, power, etc.
- Actuators
  - Thermostats, lights, valves, sprinkler, locks
- Communication
  - Sensor to hub
  - Hub to actuators
  - Hub to other users... web page, kiosk, etc.

Motivation

- Envisioned a kiosk with diverse information
  - Current local weather conditions
  - Weather forecast
  - Esoteric weather (marine, buoy, light house)
  - Water level/wave/wake data
  - Door and window closures
  - Motion detectors
  - Sunrise and sunset times
- High enough SAF to continue

Personal Requirements

- Integrate information from various sources
- Re-use existing sensors
- Scrape specialized information from websites
- Enforce the “Castle Doctrine”
- *Learn about One-Wire devices and MQTT*
- *“Control” comes later – Not first rodeo*

IoT Integrator Alternatives

- NodeRED
- IFTTT – If This, Then That
- FHEM
- Cayenne
- Maybe 20 others not considered
- OpenHAB

What is OpenHAB?

- Home Automation Bus / Information integrator
  - Abstracts data to number, string, datetime, contact (open/closed), switch (on, off), dimmer (%), rollerblind(%), etc.
  - Technology agnostic
  - Open, (mostly) non-proprietary
- Interfaces to many (most?) Internet of Things things
- Uses a open-source Java-based server

OpenHAB Works With Many IoT Things

AVM
Qualcomm AllPlay
amazon dash
amazon echo
ANEL Elektronik AG
wemo
chromecast
DSC A Tyco International Company
DENON
ecobee
enocean
EPSON EXCEED YOUR VISION
Garadget
Global Caché
31
HTTP
heatmiser
HomeMatic
PowerView
IFTTT
InfluxDB
INSTEON
KNX
KEBA Automation by innovation.
KODI
KOSTAL
LCN Local Control Network
LG
LIFX
Logitech Squeezebox
LUTRON
MAX!
MQTT
Miele
MINECRAFT

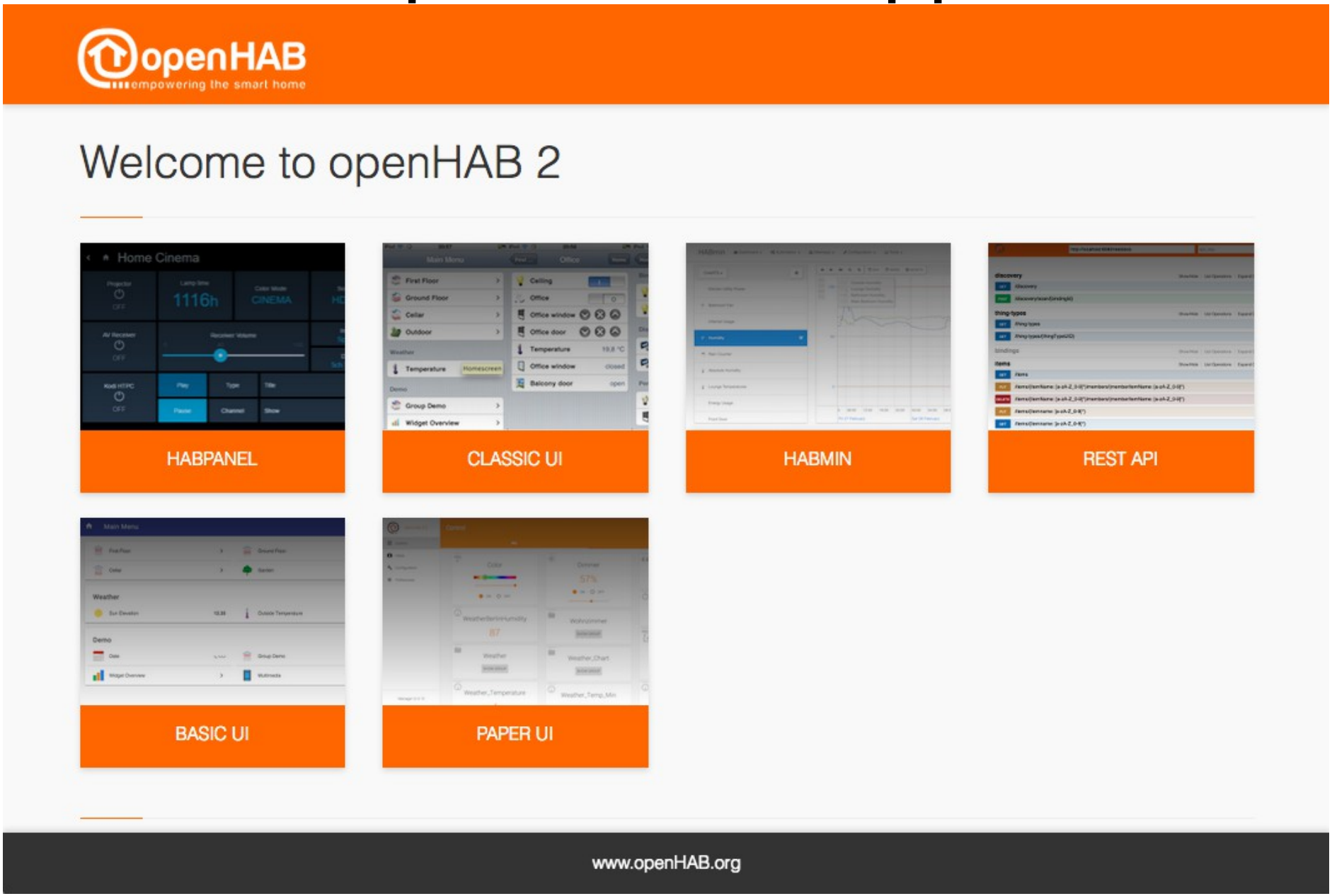
Modbus
nest
netatmo
Oceanic
ONKYO
OpenSprinkler
ORVIBO Smart Home Smart life
hue PERSONAL WIRELESS LIGHTING
Pioneer
PLEX
PulseAudio
RFXCOM
RUSSOUND
SMA
SAMSUNG
SilverCrest
somfy
SONOS
TellStick
TESLA
VIESSMANN
XMPP
YAHOO!
YAMAHA
Z-WAVE
digitalSTROM

Software

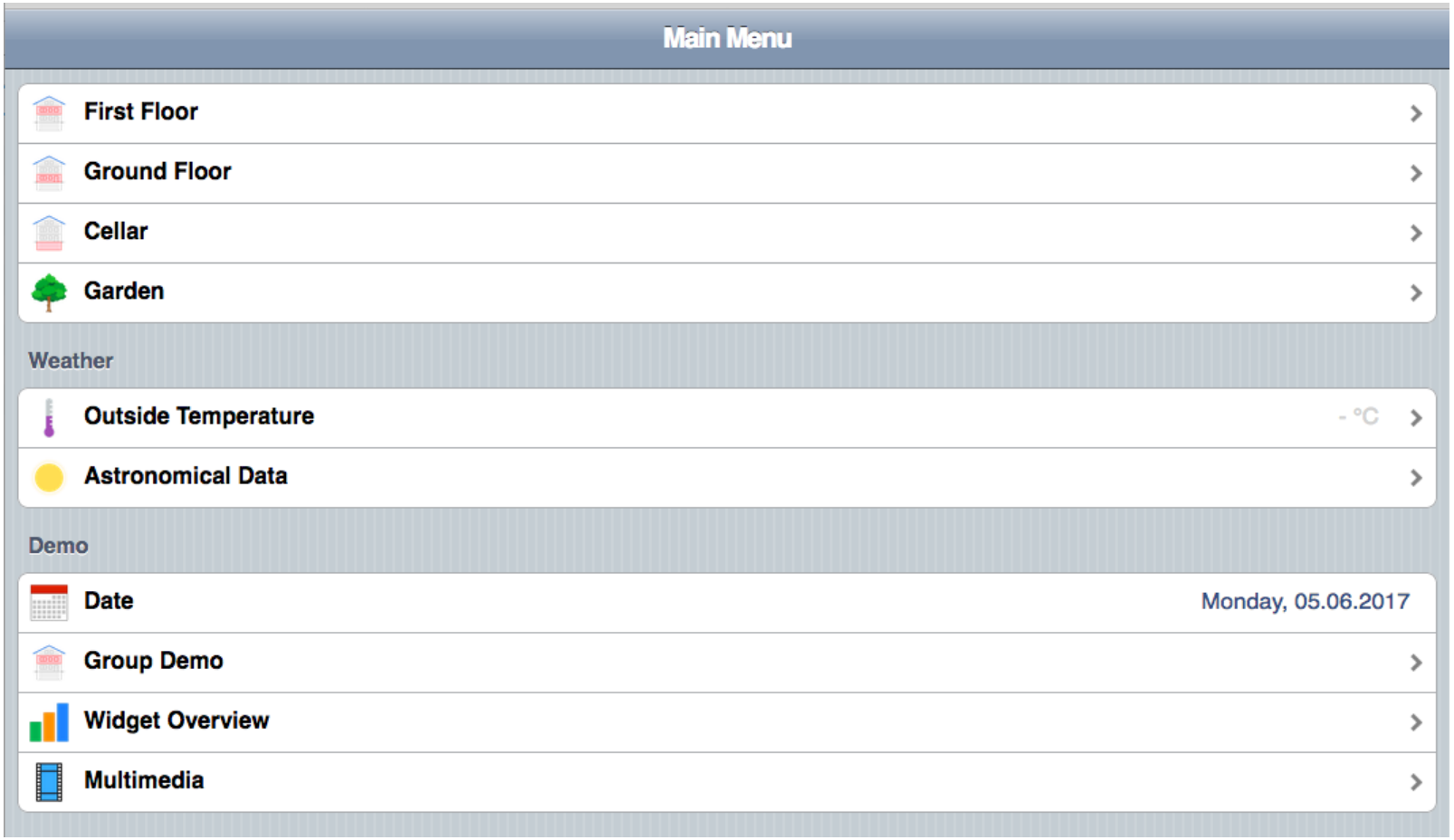
- Uses Java 1.8 (the proprietary Oracle version)
- Built on Eclipse SmartHome

- Version 1.8.3 is mature, but now dated
- Version 2.0 was released late January
  - Improved architecture for multiple “things”
  - OpenHABian is a minimal install for Raspberry Pi
  - Recommended... this talk is about 2.0

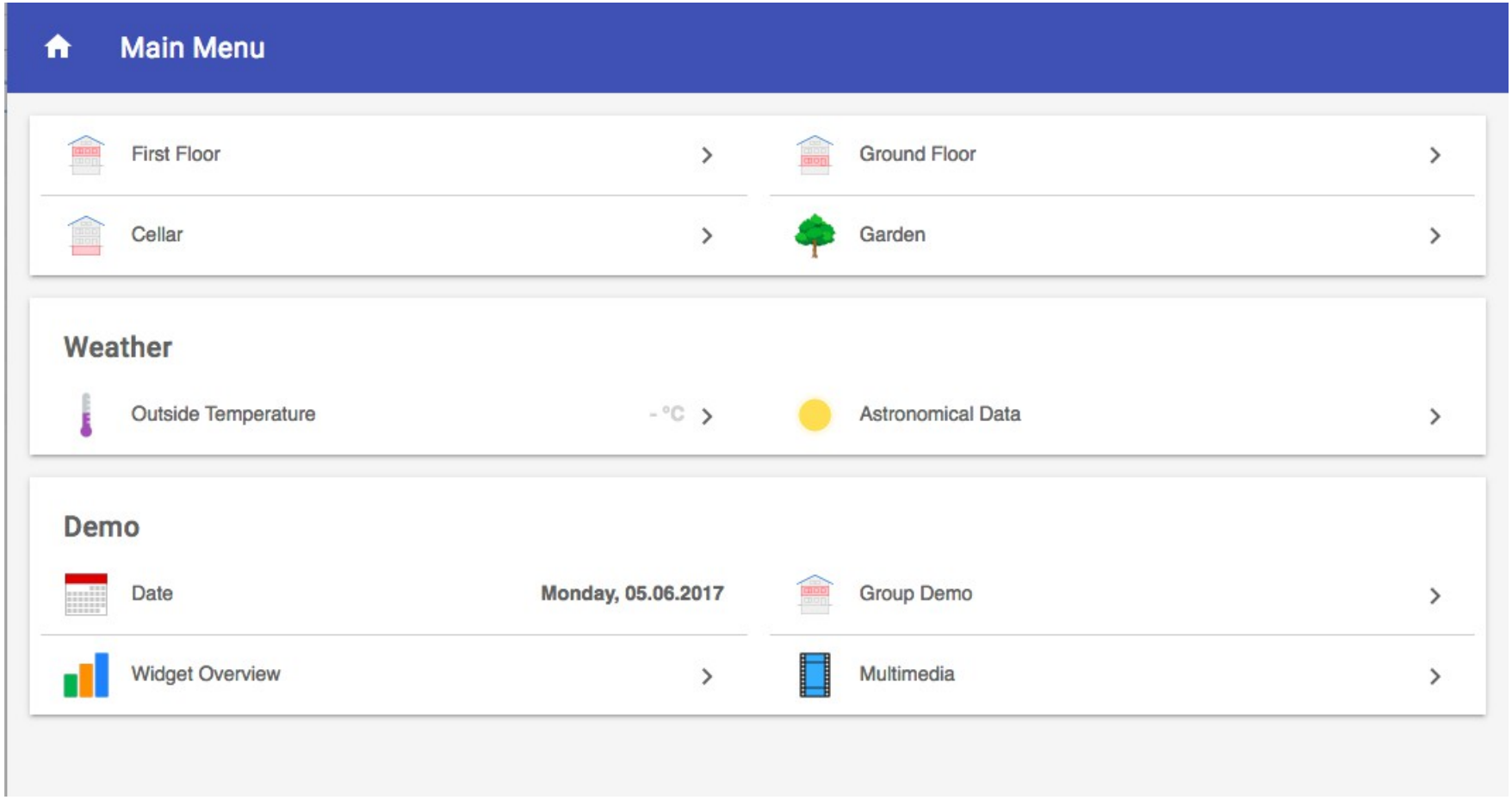
OpenHabian Apps



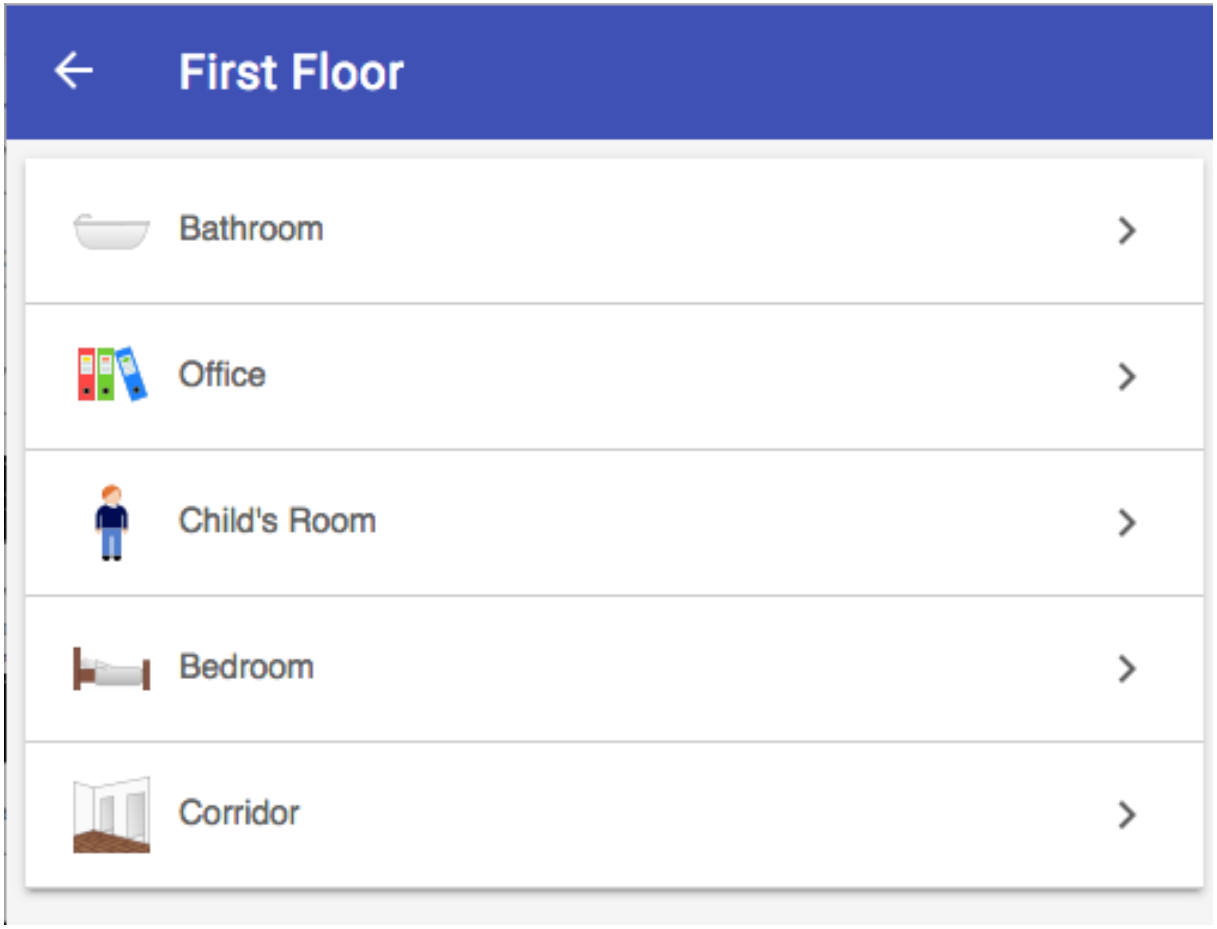
Classic User Interface



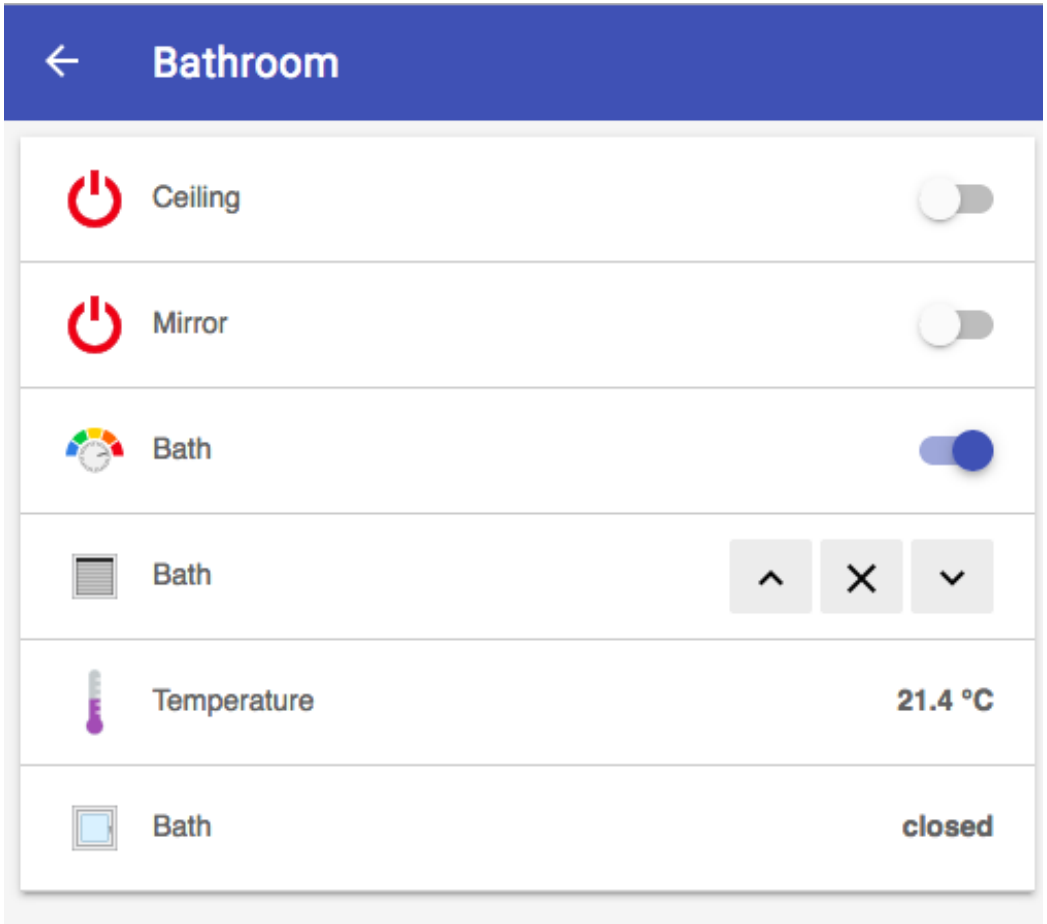
Basic User Interface



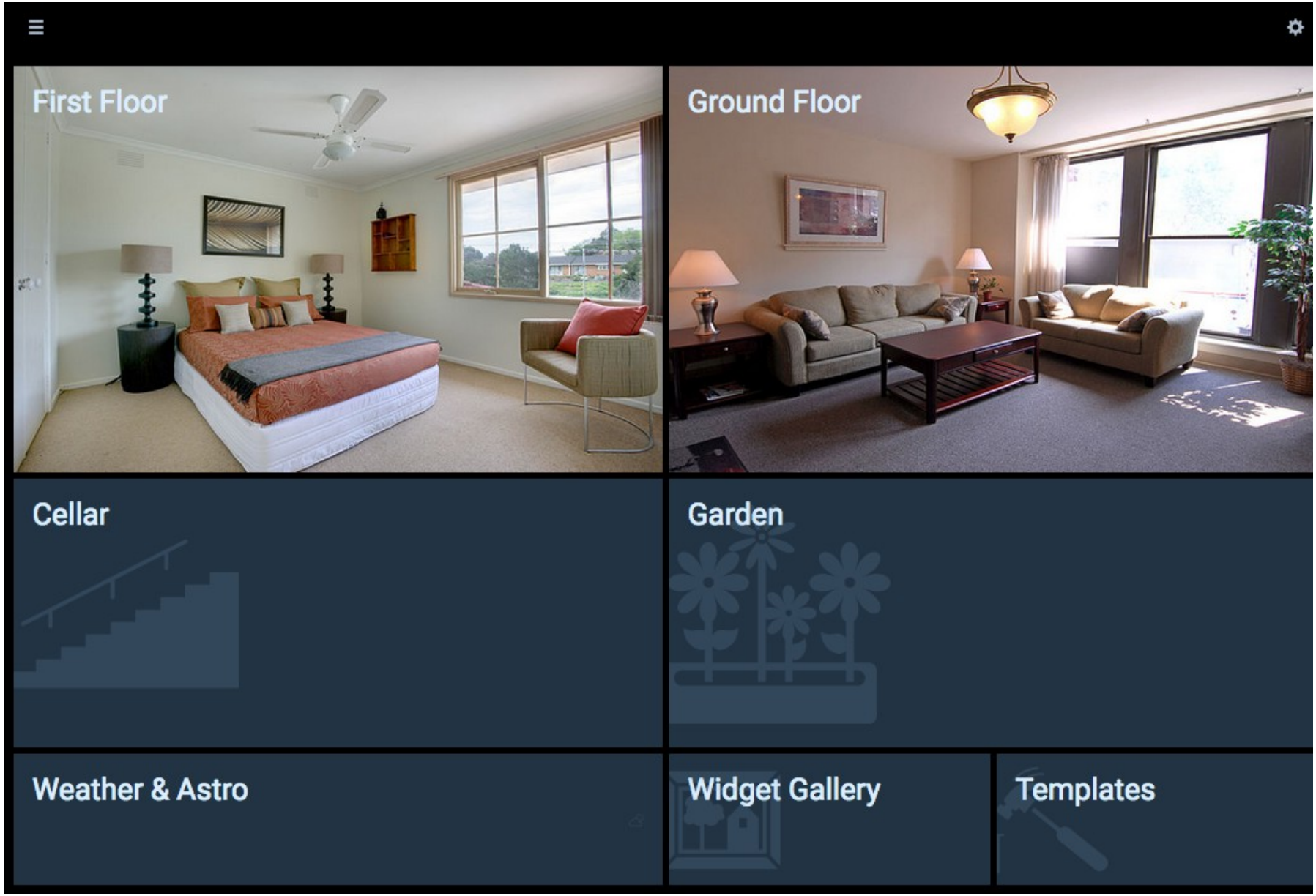
Basic User Interface



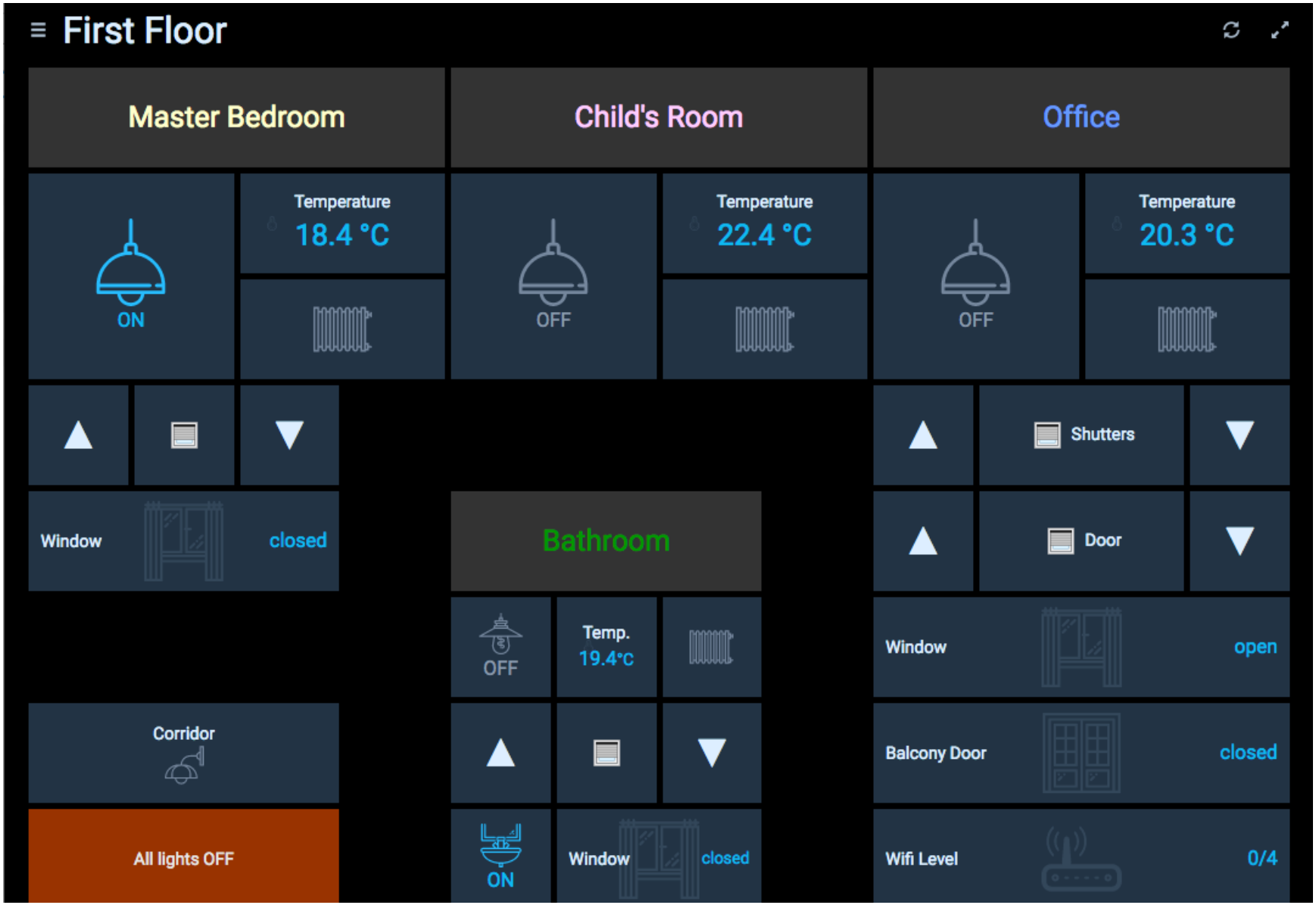
Basic User Interface



HABPanel User Experience



HABPanel (continued)



OpenHAB Basics

- Everything is configured
  - Most things with text editor
  - Some things can use administrative tools
- Functionality is extended with add-ons
  - On the order of 200 add-ons

OpenHAB Configuration

- Services – inclusion of add-ons
- Items – e.g. temperature
- Sitemaps – list of items
- Persistence – what to save, where and how
- Rules – IFTTT functionality
- Apps – to access data
- Things – provide sensor or actuator items
- Channel – selector within a thing

Add-ons

- Bindings abstract sensors
  - Astro – astronomical information for a lat/long
  - Weather – conditions and forecast for a lat/long from a specific provider
  - MQTT – simple information messaging technique
  - OWFS – one-wire devices
- Actions abstract actuators
  - Mail
  - pushover

Service configuration

```
# A comma-separated list of bindings to install (e.g. "sonos,knx,zwave")
binding = ntp,weather1,astro,mqtt1,owserver1

# A comma-separated list of UIs to install (e.g. "basic,paper")
ui = paper,classic,basic,habpanel,habmin,restapi

# A comma-separated list of persistence services to install (e.g. "rrd4j,jpa")
persistence = rrd4j

# A comma-separated list of actions to install (e.g. "mail,pushover")
#action =

# A comma-separated list of transformation services to install (e.g. "map,jsonpath
")
#transformation =
"services/addons.cfg" [Modified] 52 lines --69%--                36,0-1        77%
```

Weather service configuration

```
# The apikey for the different weather providers, at least one must be specified
# Note: Hamweather requires two apikeys: client_id=apikey, client_secret=apikey2
#apikey.ForecastIo=
#apikey.OpenWeatherMap=
#apikey.WorldWeatherOnline=
#apikey.Wunderground=
#apikey.Hamweather=
#apikey2.Hamweather=
#apikey.Meteoblue=

# location configuration, you can specify multiple locations
#location.<locationId1>.name=
#location.<locationId1>.latitude=   (not required for Yahoo provider)
#location.<locationId1>.longitude=  (not required for Yahoo provider)
#location.<locationId1>.woeid=      (required for Yahoo provider)
#location.<locationId1>.provider=
#location.<locationId1>.language=
#location.<locationId1>.updateInterval=
                                                          1,1           Top
```

Thing configuration

```
yahooweather:weather:berlin [ location=638242 ]
astro:sun:home   [ geolocation="52.5200066,13.4049540", interval=60 ]
astro:moon:home  [ geolocation="52.5200066,13.4049540", interval=60 ]
ntp:ntp:demo [ hostname="nl.pool.ntp.org", refreshInterval=60, refreshNtp=30 ]

// vim: syntax=Xtend
~
~
~
~
                                                                    5,0-1         All
```

Thing configuration

```
yahooweather:weather:berlin [ location=638242 ]
astro:sun:home   [ geolocation="52.5200066,13.4049540", interval=60 ]
astro:moon:home  [ geolocation="52.5200066,13.4049540", interval=60 ]
ntp:ntp:demo [ hostname="nl.pool.ntp.org", refreshInterval=60, refreshNtp=30 ]

// vim: syntax=Xtend
~
~
~
~
                                                                    5,0-1         All
```

Items

- Holds one sensor value or control element
- May be for alternate units (°F (°C))
- May be for derivative information
  - Accumulation
  - Time of last activity
  - Battery Failure (no activity in a certain time)

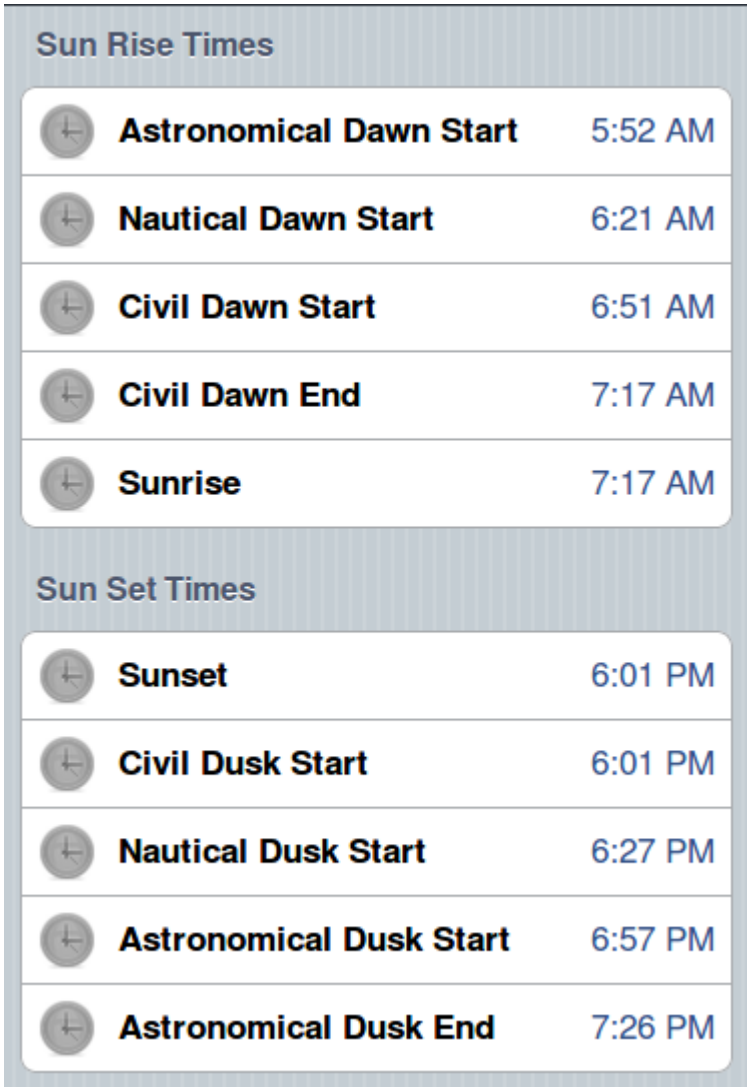
Item Configuration

- Type (number, string, datetime, contact, etc.)
- Unique name
- Text to display when item is used
- Formatted value associated with text
- Icon used with text
- Binding of item to input thing
- Group(s) that item belongs to

Astronomical Items

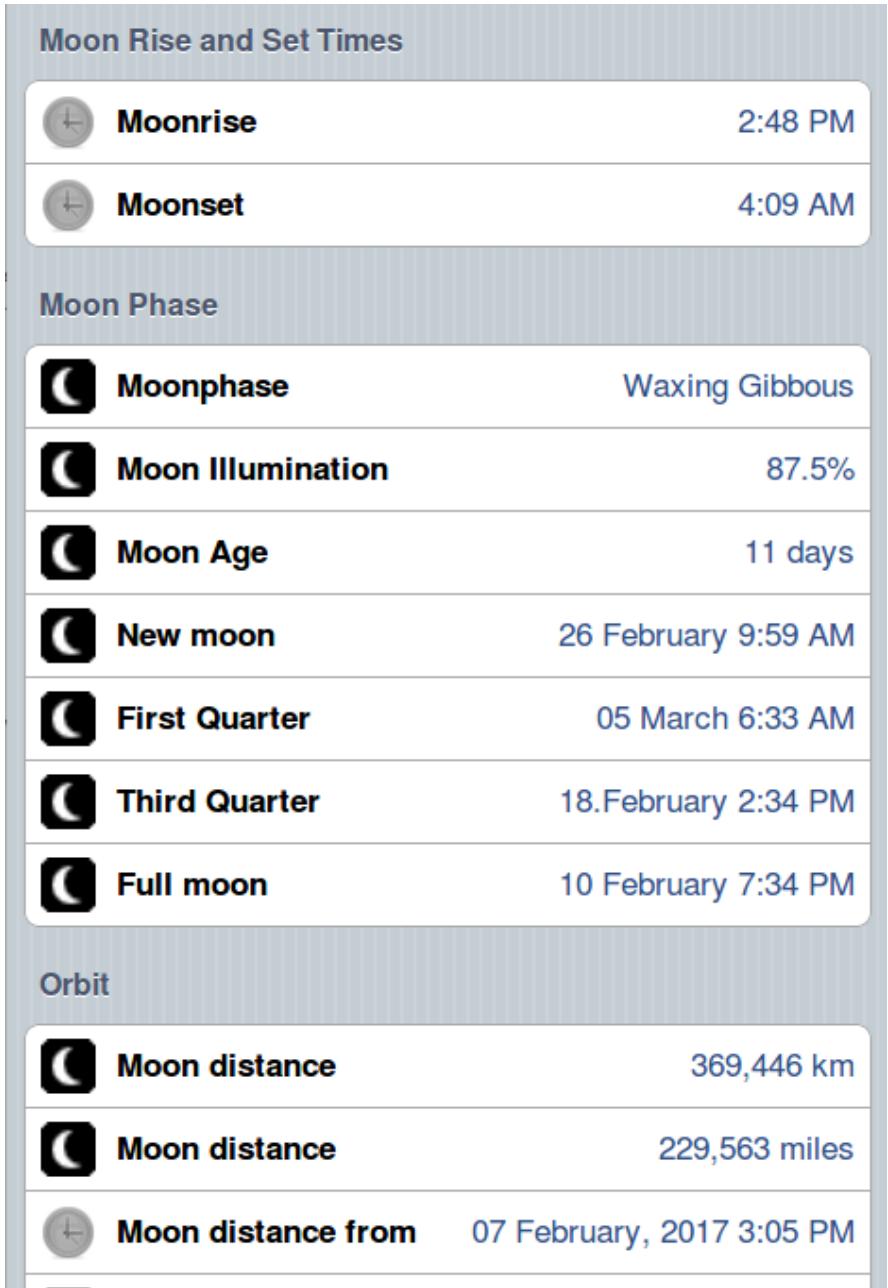
```
Group   aAll
Group   gSun    (aAll)
Group   gMoon   (aAll)
DateTime        Sunrise_Time            "Sunrise [%1$tl:%1$tM %1$Tp]"                                   <clock>        (gSun)  {astro="planet=sun, type=rise, property=start"}
DateTime        Sunset_Time             "Sunset [%1$tl:%1$tM %1$Tp]"                                    <clock>        (gSun)  {astro="planet=sun, type=set, property=end"}
DateTime        Astronomical_Dawn_Start "Astronomical Dawn Start [%1$tl:%1$tM %1$Tp]"                   <clock>        (gSun)  {astro="planet=sun, type=astroDawn, property=start"}
DateTime        Astronomical_Dawn_End   "Astronomical Dawn End [%1$tl:%1$tM %1$Tp]"                     <clock>        (gSun)  {astro="planet=sun, type=astroDawn, property=end"}
DateTime        Nautical_Dawn_Start     "Nautical Dawn Start [%1$tl:%1$tM %1$Tp]"                       <clock>        (gSun)  {astro="planet=sun, type=nauticDawn, property=start"}
DateTime        Nautical_Dawn_End       "Nautical Dawn End [%1$tl:%1$tM %1$Tp]"                         <clock>        (gSun)  {astro="planet=sun, type=nauticDawn, property=end"}
DateTime        Civil_Dawn_Start        "Civil Dawn Start [%1$tl:%1$tM %1$Tp]"                          <clock>        (gSun)  {astro="planet=sun, type=civilDawn, property=start"}
DateTime        Civil_Dawn_End          "Civil Dawn End [%1$tl:%1$tM %1$Tp]"                            <clock>        (gSun)  {astro="planet=sun, type=civilDawn, property=end"}
DateTime        Astronomical_Dusk_Start "Astronomical Dusk Start [%1$tl:%1$tM %1$Tp]"                   <clock>        (gSun)  {astro="planet=sun, type=astroDusk, property=start"}
DateTime        Astronomical_Dusk_End   "Astronomical Dusk End [%1$tl:%1$tM %1$Tp]"                     <clock>        (gSun)  {astro="planet=sun, type=astroDusk, property=end"}
DateTime        Nautical_Dusk_Start     "Nautical Dusk Start [%1$tl:%1$tM %1$Tp]"                       <clock>        (gSun)  {astro="planet=sun, type=nauticDusk, property=start"}
DateTime        Nautical_Dusk_End       "Nautical Dusk End [%1$tl:%1$tM %1$Tp]"                         <clock>        (gSun)  {astro="planet=sun, type=nauticDusk, property=end"}
DateTime        Civil_Dusk_Start        "Civil Dusk Start [%1$tl:%1$tM %1$Tp]"                          <clock>        (gSun)  {astro="planet=sun, type=civilDusk, property=start"}
DateTime        Civil_Dusk_End          "Civil Dusk End [%1$tl:%1$tM %1$Tp]"                            <clock>        (gSun)  {astro="planet=sun, type=civilDusk, property=end"}
Number          Sun_Azimuth             "Sun Azimuth [%.0f°]"                                           <clock>        (gSun)  {astro="planet=sun, type=position, property=azimuth"}
Number          Sun_Elevation           "Sun Elevation [%.0f°]"                                         <clock>        (gSun)  {astro="planet=sun, type=position, property=elevation"}
DateTime        Zodiac_Start            "Zodiac Start [%1$td %1$tB, %1$tY]"                             <calendar>     (gSun)  {astro="planet=sun, type=zodiac, property=start"}
DateTime        Zodiac_End              "Zodiac End [%1$td %1$tB, %1$tY]"                               <calendar>     (gSun)  {astro="planet=sun, type=zodiac, property=end"}
String          Zodiac_Sign             "Current zodiac [%s]"                                           <calendar>     (gSun)  {astro="planet=sun, type=zodiac, property=sign"}
String          Season_Name             "Season [%s]"                                                   <calendar>     (gSun)  {astro="planet=sun, type=season, property=name"}
DateTime        Season_Spring           "Spring [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"                 <calendar>     (gSun)  {astro="planet=sun, type=season, property=spring"}
DateTime        Season_Summer           "Summer  [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"                <calendar>     (gSun)  {astro="planet=sun, type=season, property=summer"}
DateTime        Season_Autumn           "Autumn [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"                 <calendar>     (gSun)  {astro="planet=sun, type=season, property=autumn"}
DateTime        Season_Winter           "Winter [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"                 <calendar>     (gSun)  {astro="planet=sun, type=season, property=winter"}
DateTime        Sun_Eclipse_Total       "Sun total eclipse [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"      <calendar>     (gSun)  {astro="planet=sun, type=eclipse, property=total"}
DateTime        Sun_Eclipse_Partial     "Sun partial eclipse [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"    <calendar>     (gSun)  {astro="planet=sun, type=eclipse, property=partial"}
DateTime        Sun_Eclipse_Ring        "Sun ring eclipse [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"       <calendar>     (gSun)  {astro="planet=sun, type=eclipse, property=ring"}
DateTime        Moonrise_Time           "Moonrise [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"               <clock>        (gSun)  {astro="planet=moon, type=rise, property=start"}
DateTime        Moonset_Time            "Moonset [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"                <clock>        (gSun)  {astro="planet=moon, type=set, property=end"}
DateTime        Moon_First_Quarter      "First Quarter [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"          <moon>         (gSun)  {astro="planet=moon, type=phase, property=firstQuarter"}
DateTime        Moon_Third_Quarter      "Third Quarter [%1$td.%1$tB, %1$tY %1$tl:%1$tM %1$Tp]"          <moon>         (gSun)  {astro="planet=moon, type=phase, property=thirdQuarter"}
DateTime        Moon_Full               "Full moon [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"              <moon>         (gSun)  {astro="planet=moon, type=phase, property=full"}
DateTime        Moon_New                "New moon [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"               <moon>         (gSun)  {astro="planet=moon, type=phase, property=new"}
Number          Moon_Age                "Moon Age [%.0f days]"                                          <moon>         (gMoon) {astro="planet=moon, type=phase, property=age"}
Number          Moon_Illumination       "Moon Illumination [%.1f%%]"                                    <moon>         (gMoon) {astro="planet=moon, type=phase, property=illumination"}
String          Moon_Phase_Name         "Moonphase [%s]"                                                <moon>         (gMoon) {astro="planet=moon, type=phase, property=name"}
Number          Moon_Distance_K         "Moon distance [%,.0f km]"                                      <moon>         (gMoon) {astro="planet=moon, type=distance, property=kilometer"}
Number          Moon_Distance_M         "Moon distance [%,.0f miles]"                                   <moon>         (gMoon) {astro="planet=moon, type=distance, property=miles"}
DateTime        Moon_Distance_Time      "Moon distance from [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"     <clock>        (gMoon) {astro="planet=moon, type=distance, property=date"}
DateTime        Moon_Eclipse_Total      "Moon total eclipse [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"     <moon>         (gMoon) {astro="planet=moon, type=eclipse, property=total"}
DateTime        Moon_Eclipse_Partial    "Moon partial eclipse [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"   <moon>         (gMoon) {astro="planet=moon, type=eclipse, property=partial"}
Number          Moon_Perigee_K          "Moon perigee [%,.0f km]"                                       <moon>         (gMoon) {astro="planet=moon, type=perigee, property=kilometer"}
Number          Moon_Perigee_M          "Moon perigee [%,.0f miles]"                                    <moon>         (gMoon) {astro="planet=moon, type=perigee, property=miles"}
DateTime        Moon_Perigee_Time       "Moon perigee from [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"      <clock>        (gMoon) {astro="planet=moon, type=perigee, property=date"}
Number          Moon_Apogee_K           "Moon apogee [%,.0f km]"                                        <moon>         (gMoon) {astro="planet=moon, type=apogee, property=kilometer"}
Number          Moon_Apogee_M           "Moon apogee [%,.0f miles]"                                     <moon>         (gMoon) {astro="planet=moon, type=apogee, property=miles"}
DateTime        Moon_Apogee_Time        "Moon apogee from [%1$td %1$tB, %1$tY %1$tl:%1$tM %1$Tp]"       <clock>        (gMoon) {astro="planet=moon, type=apogee, property=date"}
String          Moon_Zodiac_Sign        "Moon zodiac [%s]"                                              <clock>        (gMoon) {astro="planet=moon, type=zodiac, property=sign"}
Number          Moon_Azimuth            "Moon azimuth [%.0f°]"                                          <clock>        (gMoon) {astro="planet=moon, type=position, property=azimuth"}
Number          Moon_Elevation          "Moon elevation [%.0f°]"                                        <clock>        (gMoon) {astro="planet=moon, type=position, property=elevation"}
DateTime        Sun_Date                "Date [%1$tA, %1$tB %1$td, %1$tY]"                              <calendar>     { ntp="America/New_York:en_EN" }
```

Astronomy User Interface



Sun Rise Times	
Astronomical Dawn Start	5:52 AM
Nautical Dawn Start	6:21 AM
Civil Dawn Start	6:51 AM
Civil Dawn End	7:17 AM
Sunrise	7:17 AM

Sun Set Times	
Sunset	6:01 PM
Civil Dusk Start	6:01 PM
Nautical Dusk Start	6:27 PM
Astronomical Dusk Start	6:57 PM
Astronomical Dusk End	7:26 PM



Moon Rise and Set Times	
Moonrise	2:48 PM
Moonset	4:09 AM

Moon Phase	
Moonphase	Waxing Gibbous
Moon Illumination	87.5%
Moon Age	11 days
New moon	26 February 9:59 AM
First Quarter	05 March 6:33 AM
Third Quarter	18.February 2:34 PM
Full moon	10 February 7:34 PM

Orbit	
Moon distance	369,446 km
Moon distance	229,563 miles
Moon distance from	07 February, 2017 3:05 PM

Sitemaps

- Defines which items are displayed and where
- Used by user interface programs
- Can make items optional based on value
- Can an color item based on its value
- Define grouping on a page
- Define sub-pages
- As many as you want
  - Debugging sitemap for ALL items
  - Specialty maps with select items

Groups

- Show the status of a group of items
  - How many lights are on:
  - Are all outside doors closed?
  - Are all batteries OK?
- Code defines how to combine item status
- Can control a group of items
  - Turn all lights off
  - Set all thermostats to 40°

Transformations

- Aids for converting values
- Invoked by item text description
- Unit conversion (°F—>°C)
- Range conversion (wind Beaufort scale)
- Language translations

Things

- New concept for 2.0
- Allow multiple instances of an interface
- e.g., allows doing astronomical calculations for two locations

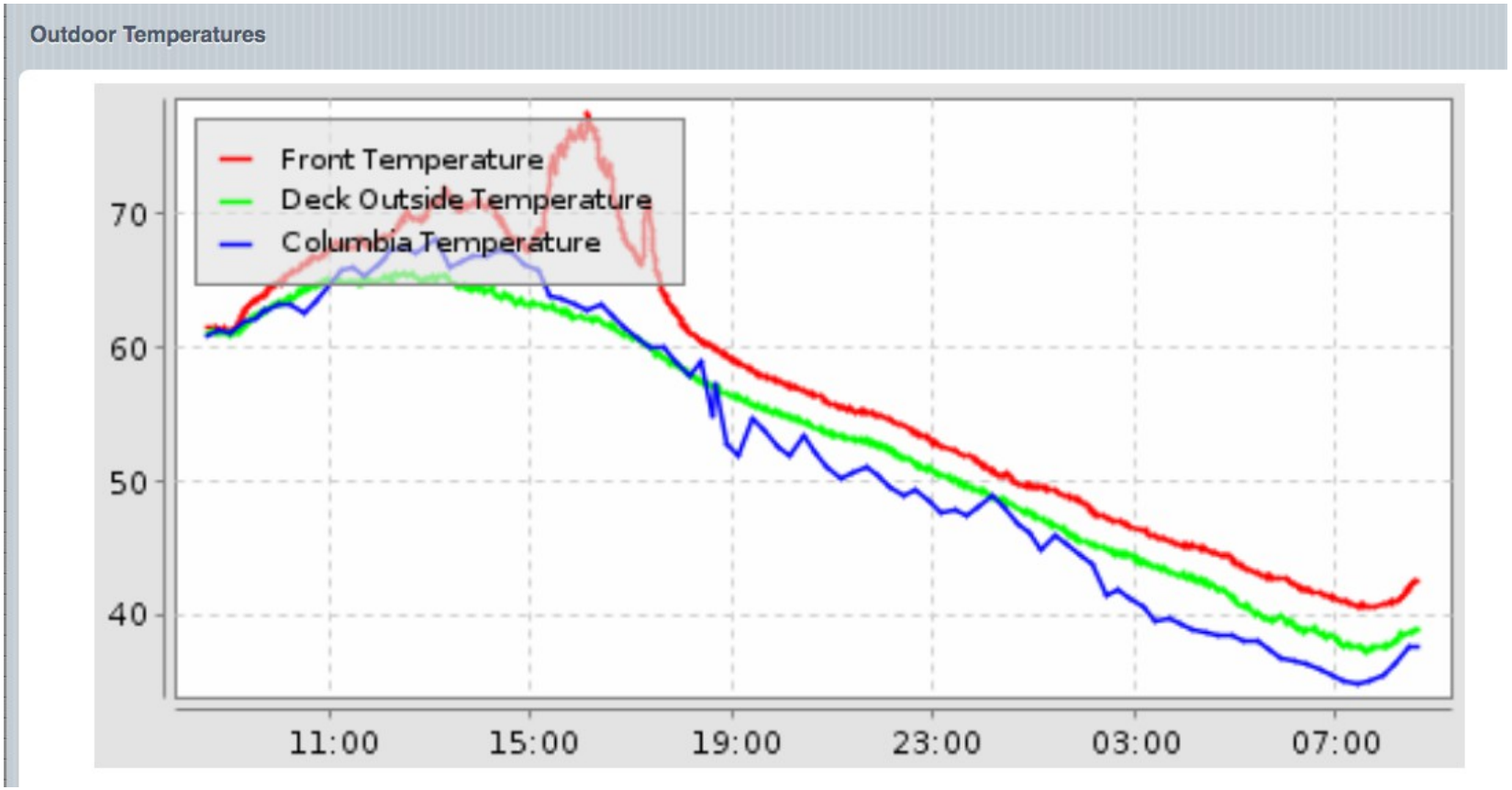
Channels

- Pathway through a thing
- For the weather thing
  - place latitude and longitude
  - provider
- For astronomical
  - Place latitude and longitude

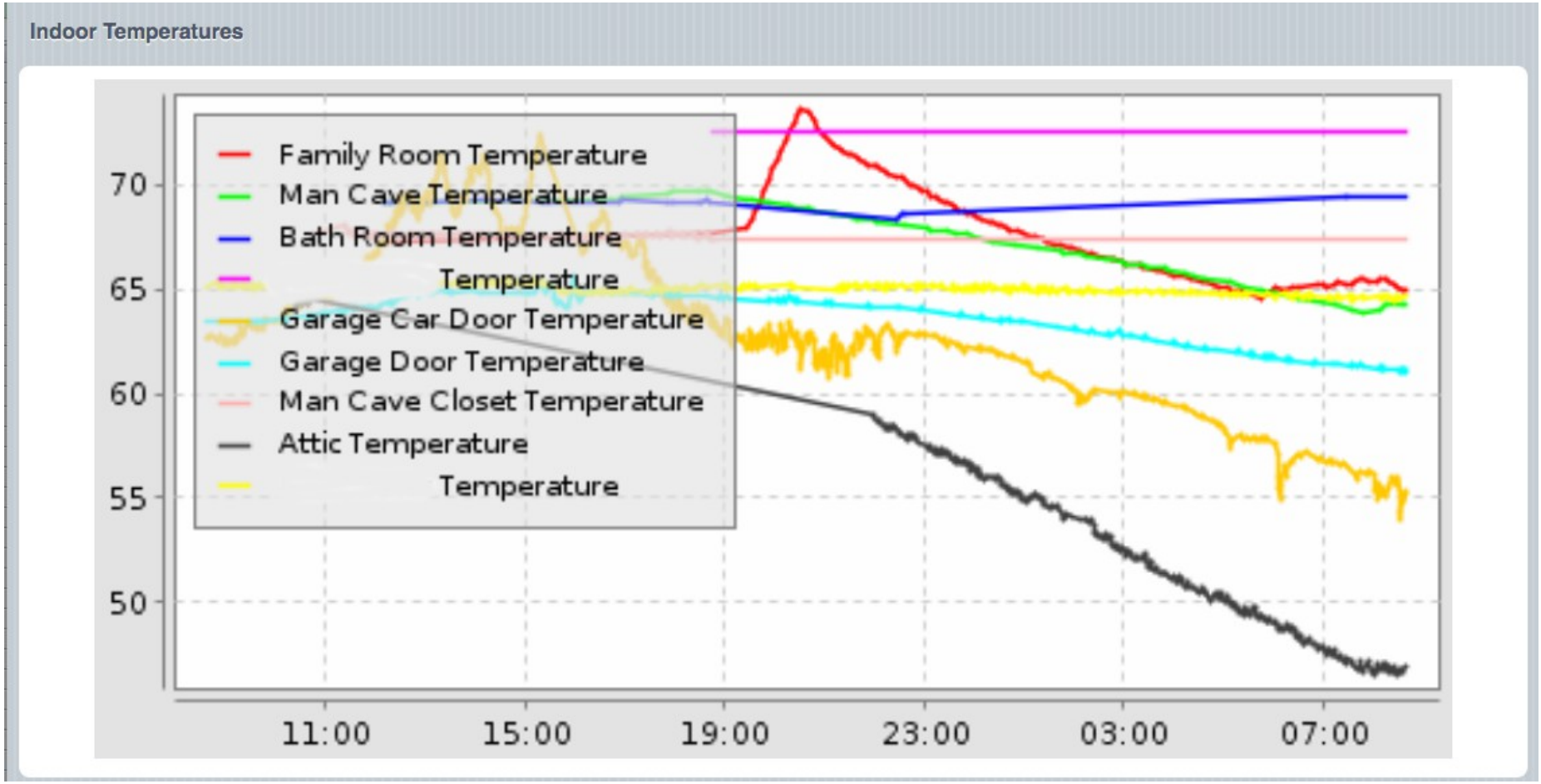
Persistence

- Allows values to be recalled
  - Graphing
  - High-low values over a particular period
  - Yesterday's weather
- Defaults to a round robin database
  - Fixed sized database
  - Consolidates values through averages
- Can use other databases like mySQL
- Stores data as time-value pairs

Graph of Outside Temperatures



Graph of Inside Temperatures



Rules

- Real power of home automation
  - Prevents stupidity of dumb timer
    - Why is he sprinkling when it is raining?
    - Why does that light come on before dark?
- Provides “If This, Then That” functionality

Rule Trigger or “when” clause

- update of a value
- change of value
- specific change to value (off to on, sunset start)
- periodically (like crontab with seconds)
- timer expiry
- start up
- others...

Rule action or “then” clause

- A snippet of Java Extend code
  - set new values
  - convert units of updated values (e.g, °C to °F)
  - start a timer for some other action
- Should use the Eclipse SmartHome Designer
- Using vi sucks for this
  - *no syntax highlighting/checking*
  - *have to know Java object models (epoch≠epoch, datetime≠datetime, Number is an object, not a type)*

Rule action or “then” clause

- A snippet of Java Extend code
  - set new values
  - convert units of updated values (e.g, °C to °F)
  - start a timer for some other action
- Should use the Eclipse SmartHome Designer
- Using vi sucks for this
  - *no syntax highlighting*
  - *have to know Java object models (epoch≠epoch, datetime≠datetime, Number is an object, not a type)*

Rules for Data Derivatives

- Example: PIR motion sensor has 2 channels
  - Motion: on for motion, off some time after motion
  - Darkness: on, off
  - (programming can limit motion to dark periods)
- Darkness can be feedback for a controlled light
- Darkness can be used to infer battery status

Garage Door Rule

```
rule "Garage car door temperature changed"
  when
    Item GarageCarDoorTemp changed or
    System started
  then
    var Number temp = GarageCarDoorTemp.state
    if (temp == Undefined) {
      /* garage car door is open, temperature is not valid */
      if (GarageCarDoor.state != OPEN) {
        /* change in GarageCarDoor State */
        postUpdate( GarageCarDoor, OPEN)
        postUpdate( GarageCarDoorOpenTime, new DateTimeType())
        logInfo("Door", "Garage car door opened")
      }
    } else {
      /* garage car door is closed */
      if (temp != 85) {
        /* temp is valid */
        if (GarageCarDoor.state != CLOSED) {
          /* change in GarageCarDoor State */
          postUpdate( GarageCarDoor, CLOSED)
          postUpdate( GarageCarDoorClosedTime, new DateTimeType())
          logInfo("Door", "Garage car door closed")
        }
        postUpdate( GarageCarDoorTempD, String::format("%.1f°F (%.1f°C)", (temp.
floatValue() * 9/5 + 32), temp.floatValue()))
        postUpdate( GarageCarDoorFTemp, (temp.floatValue() * 9/5 + 32))
        postUpdate( GarageCarDoorTempTime, new DateTimeType())
      }
    }
end
```

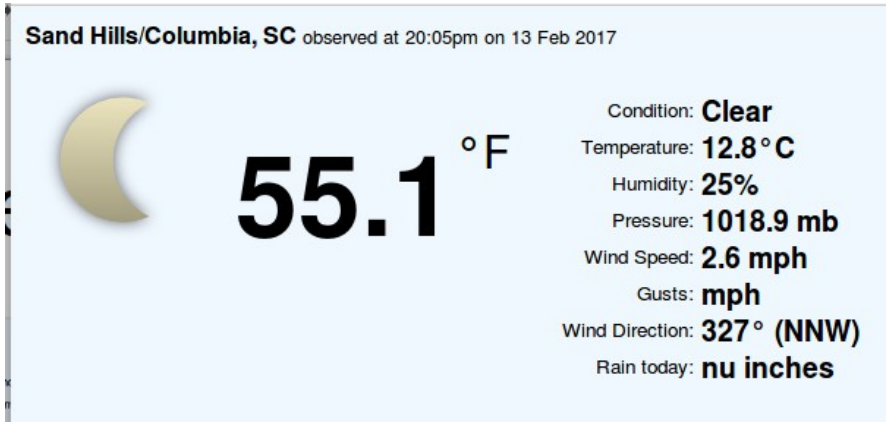
Apps

- Basic UI – generate a web page for mobiles
- Classic UI – generate a basic web page
- HABPanel – tile-based UI for dashboards
- Paper UI – system administration, item access
- HABMin – administration program (Z-wave)

Apps (continued)

- weather – generate image with embedded data
- rest – access RESTful data
- graph – access graph .png images
- Roll your own

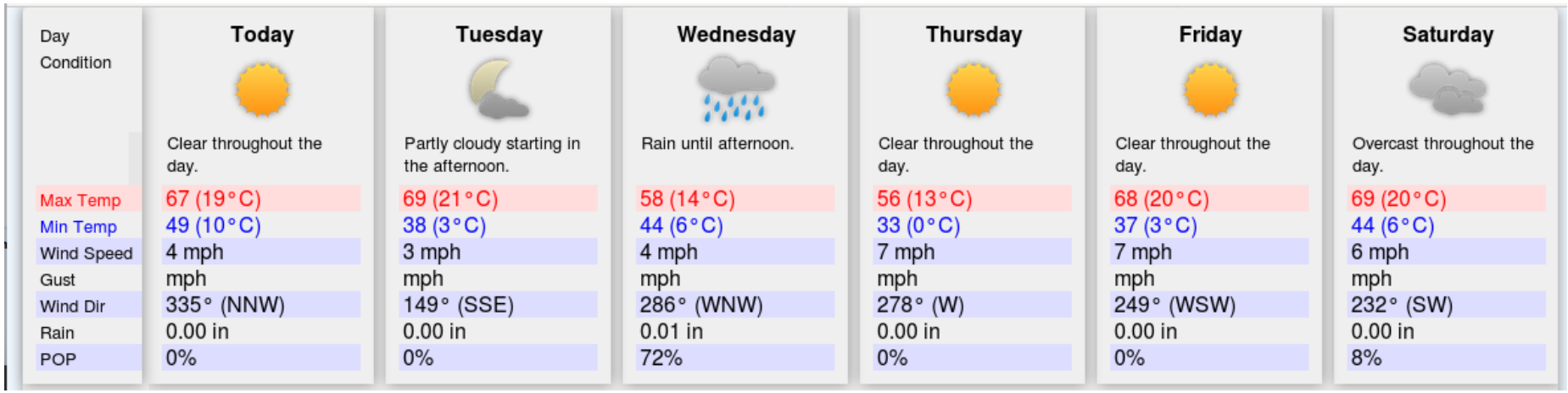
Weather Images



Sand Hills/Columbia, SC observed at 20:05pm on 13 Feb 2017

55.1 °F

Condition: **Clear**
Temperature: **12.8°C**
Humidity: **25%**
Pressure: **1018.9 mb**
Wind Speed: **2.6 mph**
Gusts: **mph**
Wind Direction: **327° (NNW)**
Rain today: **nu inches**



Day	Today	Tuesday	Wednesday	Thursday	Friday	Saturday
Condition	Clear throughout the day.	Partly cloudy starting in the afternoon.	Rain until afternoon.	Clear throughout the day.	Clear throughout the day.	Overcast throughout the day.
Max Temp	67 (19°C)	69 (21°C)	58 (14°C)	56 (13°C)	68 (20°C)	69 (20°C)
Min Temp	49 (10°C)	38 (3°C)	44 (6°C)	33 (0°C)	37 (3°C)	44 (6°C)
Wind Speed	4 mph	3 mph	4 mph	7 mph	7 mph	6 mph
Gust	mph	mph	mph	mph	mph	mph
Wind Dir	335° (NNW)	149° (SSE)	286° (WNW)	278° (W)	249° (WSW)	232° (SW)
Rain	0.00 in	0.00 in	0.01 in	0.00 in	0.00 in	0.00 in
POP	0%	0%	72%	0%	0%	8%

RESTful API

- access to item info and sitemaps
  - Individually or all defined
  - Verging on being too powerful
- uses JSON or xml, so great for AJAX
- Points outs need for securing the HTML port
  - *No authentication out of the box*
  - *Openhabian has a procedure to use NGINX as reverse proxy to provide authentication*

RESTful interface: rest/sitemaps

bc.local:8081/rest/sitemaps

Most Visited | Access cPanel | SouthEast LinuxFe... | Telus Hub | piSense


```
– <sitemaps>
  – <sitemap>
      <name>demo</name>
      <label>Demo House</label>
      <link>http://bc.local:8081/rest/sitemaps/demo</link>
    – <homepage>
        <link>http://bc.local:8081/rest/sitemaps/demo/demo</link>
        <leaf>false</leaf>
      </homepage>
    </sitemap>
  – <sitemap>
      <name>default</name>
      <label>Demo House</label>
      <link>http://bc.local:8081/rest/sitemaps/default</link>
    – <homepage>
        <link>http://bc.local:8081/rest/sitemaps/default/demo</link>
        <leaf>false</leaf>
      </homepage>
    </sitemap>
  – <sitemap>
      <name>astro</name>
      <label>Astronomical Information</label>
      <link>http://bc.local:8081/rest/sitemaps/astro</link>
    – <homepage>
        <link>http://bc.local:8081/rest/sitemaps/astro/astro</link>
        <leaf>false</leaf>
      </homepage>
    </sitemap>
  – <sitemap>
```

RESTful interface: rest/sitemaps/onewire

bc.local:8081/rest/sitemaps/onewire

Most Visited | Access cPanel | SouthEast LinuxFe... | Telus Hub | piSense1 Te


```
– <sitemap>
    <name>onewire</name>
    <label>One-wire devices</label>
    <link>http://bc.local:8081/rest/sitemaps/onewire</link>
  – <homepage>
      <id>onewire</id>
      <title>One-wire devices</title>
      <link>http://bc.local:8081/rest/sitemaps/onewire/onewire</link>
      <leaf>false</leaf>
    – <widget>
        <widgetId>onewire_0</widgetId>
        <type>Frame</type>
        <label>Indoor Temperatures</label>
        <icon>frame</icon>
      – <widget>
          <widgetId>onewire_0_0</widgetId>
          <type>Chart</type>
          <label>IndoorTemps</label>
          <icon>chart</icon>
          <refresh>10000</refresh>
          <service>mysql</service>
          <period>D</period>
        – <item>
            <type>GroupItem</type>
            <name>IndoorTemps</name>
            <state>Undefined</state>
            <link>http://bc.local:8081/rest/items/IndoorTemps</link>
          </item>
        </widget>
      </widget>
    – <widget>
        <widgetId>onewire_1</widgetId>
        <type>Frame</type>
        <label>Outdoor Temperatures</label>
        <icon>frame</icon>
      – <widget>
          <widgetId>onewire_1_0</widgetId>
          <type>Chart</type>
```

HABMin

- Administrative program
- Can access historical data

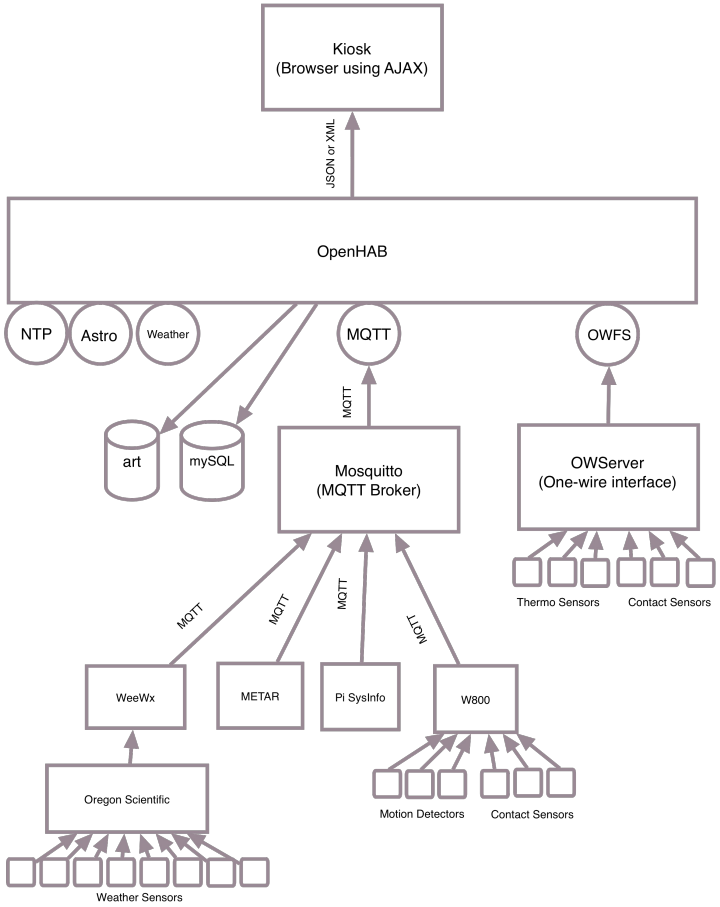
PaperUI

- Intended as a GUI for administering add-ons
- Easy to use
- Lists the add-ons and their status
- Not quite fully functional (so somewhat frustrating)

MyOpenHAB

- Online openHAB server
- For those who want access everywhere
- For those who aren't rationally paranoid
  - Violates “castle doctrine”

My Experience with OpenHAB



MQTT

- Simple messaging protocol
- Simple interface:
  - Topic string (“major/minor”)
  - Payload (value)
  - QOS (at most once, at least once, exactly once)
  - Retention
- Great for ad hoc data

MQTT Broker

- Information is published to the broker
- Information is subscribed from the broker
- Mosquitto is an open source broker that is easy to use

Paho-MQTT Library for Python

- Fairly flexible functions
- Single publish that internally connects, publishes and disconnects
- Multiple publish by connect, publish items, disconnect
- Has callback capabilities for subscribing

Example Uses of MQTT

- Interfaced METAR weather records
- Interfaced W800 to access X-10 RF and “security” devices
- Used MQTT capabilities of weewx to interface Oregon Scientific weather station
- Scraped web for exchange rates
- Scraped web for fire hazard status
- Scraped web for marine forecast

Kiosk

Columbia Temperature **76.4 °F**

Family Room Temperature **68.2°F**

Family Room Humidity **49%**

Sunrise **7:17 AM**

Sunset **6:01 PM**

Moonrise **2:48 PM**

Moon Illumination **87.0%**

Full moon **10 February 7:34 PM**

Tuesday, Feb 7, 2017

1:19:25 PM

Deck Outside Temperature **72.7 °F**

Front Temperature **76.9 °F**

Garage Car Door Temperature **72.8 °F**

Garage Door Temperature **63.6 °F**

Man Cave Closet Temperature

Attic Temperature **74.6 °F**

Cooler Temperature **63.8 °F**

Man Cave Temperature **68.0°F**

Bath Room Temperature **70.2°F**

Cooler Temperature **67.8°F**

Garage Car Door **closed**

Garage Door **closed**

Man Cave Closet Door **open**

Attic Door **closed**

Cooler Door **closed**

X10 Test A1 **on**

X10 Test A7 **on**

Sitemap for Simple Kiosk

```
sitemap kiosk label="OpenHAB Kiosk"
{
        Frame label="Permanent" {
                Text item=Weather_Cola_FTemperature
/* predicted high, low, rain, wind */
                Text item=Watson_Inside_FTemp
                Text item=Sunrise_Time
                Text item=Sunset_Time
                Text item=Moonrise_Time
                Text item=Moon_Illumination
                Text item=Moon_Full
                Webview url="/weather?locationId=ghmd&layout=example2&iconset=colorful" height=7
        }
        Frame label="Rotating" {
                Text item=FrontFTemp
                Text item=GarageCarDoorFTemp
                Text item=GarageDoorFTemp
                Text item=DeckFTemp
                Text item=ManCaveClosetDoorFTemp
                Text item=AtticDoorFTemp
                Text item=[illegible]DoorFTemp
                Text item=GarageCarDoor
                Text item=GarageDoor
                Text item=ManCaveClosetDoor
                Text item=AtticDoor
                Text item=[illegible]Door
                Chart item=OutdoorTemps period=D service="mysql" refresh=10000
                Chart item=IndoorTemps period=D service="mysql" refresh=10000
        }
}
```

Kiosk.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Kirk's OpenHAB Kiosk</title>
  <link type="text/css" rel="stylesheet" href="next.css" />
  <script type="text/javascript" src="next.js"></script>
</head>
<body onload="loadBody();">
  <div id="AddIns">
    <div id="date"></div>
    <div id="time"></div>
  </div>
</body>
</html>
~
~
~
```

Kiosk.js (partial)

```
/* JavaScript for the kiosk */

var oneWire;

function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    console.log ("Got " + this.readyState + " and " + this.status)
    text = ""
    if (this.readyState == 4 && this.status == 200) {
      oneWire = JSON.parse(this.responseText);
      /* convert each frame widget into a list in a div */
      for (i in oneWire.homepage.widget) {
        if (oneWire.homepage.widget[i].type == "Frame") {
          /* convert these widgets into list items within the list */
          console.log (oneWire.homepage.widget[i].widget)
          text = '<ul>'
          for (j in oneWire.homepage.widget[i].widget) {
            if (oneWire.homepage.widget[i].widget[j].type == "Text") {
              text += '<li id="' + oneWire.homepage.widget[i].widget[j].item.name + '">'
              label = oneWire.homepage.widget[i].widget[j].label
              labelText = label.replace (/\s*\[.*$/,'')
              labelValue =  label.replace (/.*\[/, '')
              labelValue =  labelValue.replace (/].*/, '')
              text += "<span class=label>" + labelText + "</span>" + labelValue
              text += '</li>'
            }
          }
          text += '</ul>'
          addElement ("AddIns", "div", oneWire.homepage.widget[i].label, text)
        }
      }
    }
  };
  xhttp.open("GET", "http://bc.local:8081/rest/sitemaps/kiosk?type=json", true);
  //xhttp.open("GET", "http://bc.local:8081/rest/sitemaps/onewire", true);
  //xhttp.setRequestHeader("Content-type", "application/json");
  xhttp.send();
}

/* following function is from https://www.abeautifulsite.net/adding-and-removing-elements-on-th
function addElement(parentId, elementTag, elementId, html) {
    // Adds an element to the document
    var p = document.getElementById(parentId);
    var newElement = document.createElement(elementTag);
    newElement.setAttribute('id', elementId);
    newElement.innerHTML = html;
    p.appendChild(newElement);
}
//The following script and many more are available free online at -->
//The JavaScript Source!! http://www.javascriptsource.com -->

var timerID = null;
var timerRunning = false;

function stopclock (){
  if(timerRunning) {
```