

# IoT using OpenHAB and MQTT

Kirk Carlson

[Kirk.Carlson@att.net](mailto:Kirk.Carlson@att.net)

<https://github.com/kirkcarlson>

# Personal Objective

- Integrate information from various sources on single page
- Have platform for storage and retrieval of data
- Reuse existing sensors
- Learn about One-Wire devices and MQTT
- Enforce the “Castle Doctrine”
- *“Control” is obviously missing*

# Desired Data

- Local weather conditions
- Local weather forecast
- 'Esoteric' weather conditions and water level data
- Sunrise/sunset times
- Moon phase and rise time
- Door sensors, especially garage door
- Motion sensors

# IoT Integrator Alternatives

- NodeRED
- IFTTT – If this, then that
- FHEM
- Cayenne
- OpenHAB
- Maybe 20 others not considered

# What is OpenHAB?

- Home Automation Bus / Information integrator
  - Abstracts data to number, string, datetime, contact (open/closed), switch (on, off), dimmer (%), rollerblind(%)
  - Technology agnostic
  - Open, non-proprietary
- Interfaces to many (most?) Internet of Things things
- Data can be accessed with RESTful interface
- Actively developed
- Built on Eclipse Smarthome framework with Java

# OpenHAB Works With Many IoT Things



# Version

- This talk is about version 1.8.3
- Run under Ubuntu
- Dependent on Java 1.8 (the proprietary Oracle version)
- Version 2.0 was released late January
- OpenHABian is easy way to install Version 2 on a Raspberry Pi

# OpenHAB Basics

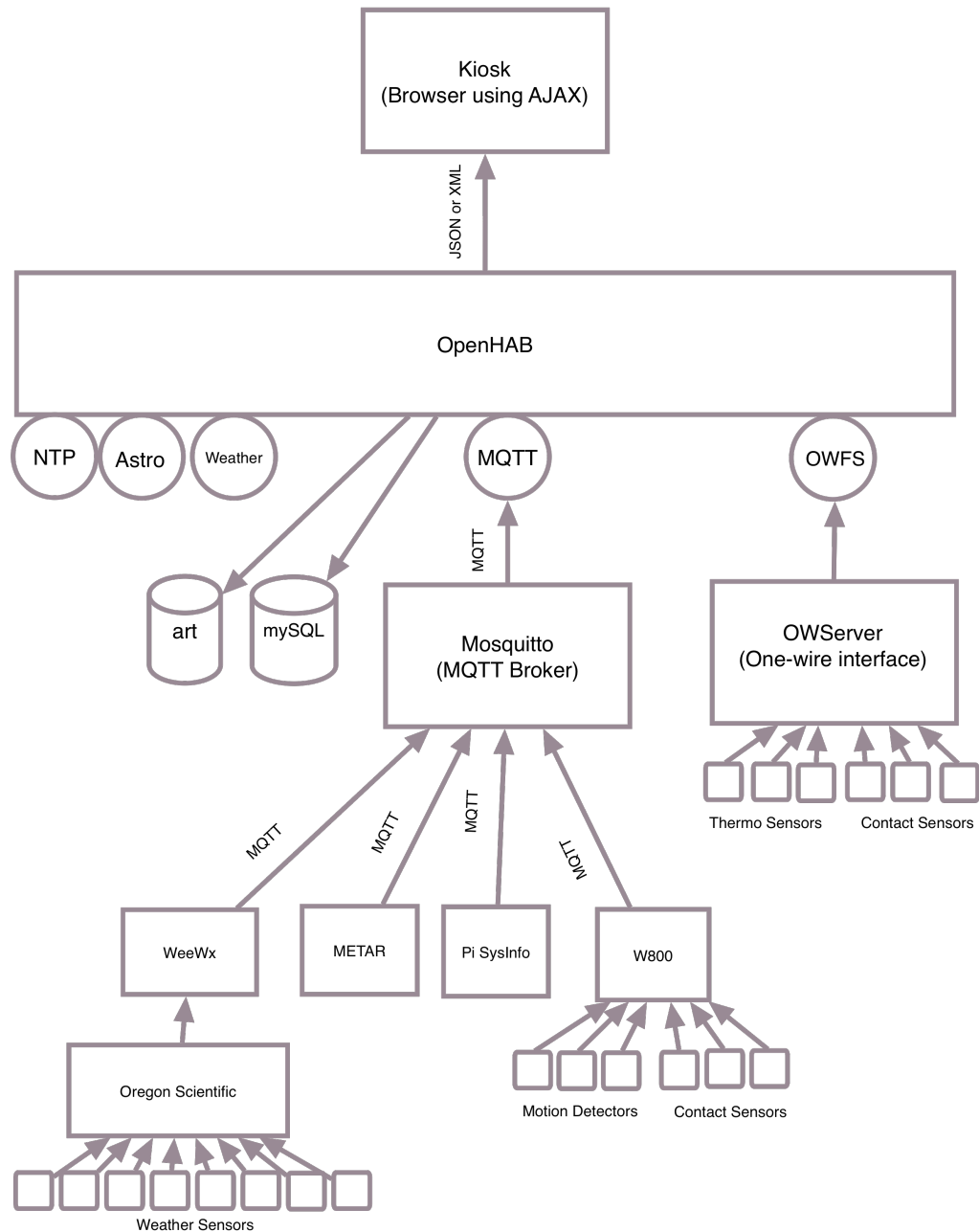
- Driven by configuration files and add-ons
- Main configuration file (for add-ons)
- Items
- Sitemaps
- Persistence
- Rules
- Apps



# Add-ons

- Interface to services and devices
- Nearly 100 of these
- This talk used:
  - Astro – astronomical information
  - Weather – conditions and forecasts from several providers
  - MQTT – simple information messaging technique
  - OWFS – one-wire devices

# My Experience with OpenHAB



# Main configuration file

- Configure the add-ons
- Examples...

# OpenHAB General Configuration

```
#####
#####              General configurations              #####
#####

# Configuration folders (must exist as a subdirectory of "configurations"; the value
# tells the number of seconds for the next scan of the directory for changes. A
# value of -1 deactivates the scan).
# A comma separated list can follow after the refresh value. This list defines a filter
# for valid file extensions for the models.
folder:items=10,items
folder:sitemaps=10,sitemap
folder:rules=10,rules
folder:scripts=10,script
folder:persistence=10,persist

# configures the security options. The following values are valid:
# ON = security is switched on generally
# OFF = security is switched off generally
# EXTERNAL = security is switched on for external requests
#             (e.g. originating from the Internet) only
# (optional, defaults to 'OFF')
#security:option=

# the Netmask to define a range of internal IP-Addresses which doesn't require
# authorization (optional, defaults to '192.168.1.0/24')
#security:netmask=

# The name of the default persistence service to use
persistence:default=mysql

# The refresh interval for the main configuration file. A value of '-1'
# deactivates the scan (optional, defaults to '-1' hence scanning is deactivated)
#mainconfig:refresh=

# Bind service discovery to specific hostname or IP address
#servicediscovery:bind_address=127.0.0.1
```

# Astro Binding Configuration

---

```
##### Astro Binding #####  
#  
# The latitude  
#astro:latitude=  
#columbia SC  
astro:latitude=34.1313  
  
# The longitude  
#astro:longitude=  
#columbia SC  
astro:longitude=-80.8775  
  
# Refresh interval for azimuth and elevation calculation in seconds  
# (optional, defaults to disabled)  
astro:interval=300
```

# Weather Add-on Configuration

```
##### Weather Binding #####  
#  
# The apikey values for the different weather providers  
# Note: Hamweather requires two apikeys: client_id=apikey, client_secret=apikey2  
weather:apikey.ForecastIo=  
#weather:apikey.OpenWeatherMap=  
#weather:apikey.WorldWeatherOnline=  
weather:apikey.Wunderground=  
#weather:apikey.Hamweather=  
#weather:apikey2.Hamweather=  
  
# location configuration, you can specify multiple locations  
# Note: latitude and longitude are NOT required for Yahoo  
#       woeid is ONLY required for Yahoo  
weather:location.cola.name=Sand Hills/Columbia, SC  
weather:location.cola.latitude=34.1313  
weather:location.cola.longitude=-80.8775  
#weather:location.cola.woeid=  
weather:location.cola.provider=ForecastIo  
weather:location.cola.language=en  
weather:location.cola.updateInterval=15
```

# Items

- Type (number, string, datetime, switch, contact, etc.)
- Unique name
- Text to display when item is used
- Value associated with text
- Icon used with text
- Binding of item to input
- Group(s) that item belongs to
- Each item is essentially a “variable”
- Use freely: °F, °C, “x °F (y °C)”





# 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" }	








# Astro Control Page


## Sun Rise Times

	<b>Astronomical Dawn Start</b>	5:52 AM
	<b>Nautical Dawn Start</b>	6:21 AM
	<b>Civil Dawn Start</b>	6:51 AM
	<b>Civil Dawn End</b>	7:17 AM
	<b>Sunrise</b>	7:17 AM

## Sun Set Times

	<b>Sunset</b>	6:01 PM
	<b>Civil Dusk Start</b>	6:01 PM
	<b>Nautical Dusk Start</b>	6:27 PM
	<b>Astronomical Dusk Start</b>	6:57 PM
	<b>Astronomical Dusk End</b>	7:26 PM




## Moon Rise and Set Times

	<b>Moonrise</b>	2:48 PM
	<b>Moonset</b>	4:09 AM

## Moon Phase

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

## Orbit

	<b>Moon distance</b>	369,446 km
	<b>Moon distance</b>	229,563 miles
	<b>Moon distance from</b>	07 February, 2017 3:05 PM








# Weather Items

Group	Weather_Chart (Weather)			
Number	Weather_Cola_Temperature	"Columbia Temperature [%.1f °C]"	<temperature>	{ weather="locationId=cola, type=temperature, property=current" }
Number	Weather_Cola_FTemperature	"Columbia Temperature [%.1f °F]"	<temperature>	{ weather="locationId=cola, type=temperature, property=current, unit=fahren
heit" }			(OutdoorTemps)	
Number	Weather_Cola_Humidity	"Columbia Humidity [%.1f %]"	<temperature>	{ weather="locationId=cola, type=atmosphere, property=humidity" }
Number	Weather_Cola_Humidex	"Columbia Humidex [SCALE(humidex.scale):%s]"	(Weather)	
Number	Weather_Cola_Temp_Max	"Todays Maximum [%.1f °C]"	<temperature>	
Number	Weather_Cola_Temp_Min	"Todays Minimum [%.1f °C]"	(Weather)	
Number	Weather_Cola_FTemp_Max	"Todays Maximum [%.1f °F]"	<temperature>	
Number	Weather_Cola_FTemp_Min	"Todays Minimum [%.1f °F]"	(Weather)	
Number	Weather_Cola_Chart_Period	"Chart Period"	<temperature>	
DateTime	Weather_Cola_LastUpdate	"Last Update [%1\$ta %1\$tR]"	<clock>	

# One-Wire Items

Group	OutdoorTemps			
Group	IndoorTemps			
Group	DoorContacts			
Number	FrontTemp	"Front Temperature [%1f °C]"	<temperature>	{onewire="deviceId=22.92C04B000000;propertyName=temperature12;"}
Number	GarageCarDoorTemp	"Garage Car Door Temperature [%1f °C]"	<temperature>	{onewire="deviceId=22.19E44B000000;propertyName=temperature12;"}
Number	GarageDoorTemp	"Garage Door Temperature [%1f °C]"	<temperature>	{onewire="deviceId=28.77D881000000;propertyName=temperature12;"}
Number	DeckTemp	"Deck Outside Temperature [%1f °C]"	<temperature>	{onewire="deviceId=28.6D0883000000;propertyName=temperature;"}
Number	ManCaveClosetDoorTemp	"Man Cave Closet Temperature [%1f °C]"	<temperature>	{onewire="deviceId=28.63B480000000;propertyName=temperature12;"}
Number	AtticDoorTemp	"Attic Temperature [%1f °C]"	<temperature>	{onewire="deviceId=28.591F82000000;propertyName=temperature;"}
Number	DoorTemp	"Temperature [%1f °C]"	<temperature>	{onewire="deviceId=28.C6F580000000;propertyName=temperature;"}
String	FrontTempD	"Front Temperature [%s]"	<temperature>	
String	GarageCarDoorTempD	"Garage Car Door Temperature [%s]"	<temperature>	
String	GarageDoorTempD	"Garage Door Temperature [%s]"	<temperature>	
String	DeckTempD	"Deck Outside Temperature [%s]"	<temperature>	
String	ManCaveClosetDoorTempD	"Man Cave Closet Temperature [%s]"	<temperature>	
String	AtticDoorTempD	"Attic Temperature [%s]"	<temperature>	
String	DoorTempD	"Temperature [%s]"	<temperature>	
Number	FrontFTemp	"Front Temperature [%1f °F]"	<temperature>	(OutdoorTemps)
Number	GarageCarDoorFTemp	"Garage Car Door Temperature [%1f °F]"	<temperature>	(IndoorTemps)
Number	GarageDoorFTemp	"Garage Door Temperature [%1f °F]"	<temperature>	(IndoorTemps)
Number	DeckFTemp	"Deck Outside Temperature [%1f °F]"	<temperature>	(OutdoorTemps)
Number	ManCaveClosetDoorFTemp	"Man Cave Closet Temperature [%1f °F]"	<temperature>	(IndoorTemps)
Number	AtticDoorFTemp	"Attic Temperature [%1f °F]"	<temperature>	(IndoorTemps)
Number	DoorFTemp	"Temperature [%1f °F]"	<temperature>	(IndoorTemps)
DateTime	FrontTempTime	"Front Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	DeckTempTime	"Deck Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	GarageCarDoorTempTime	"Garage Car Door Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	GarageDoorTempTime	"Garage Door Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	ManCaveClosetDoorTempTime	"Man Cave Closet Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	AtticDoorTempTime	"Attic Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	DoorTempTime	"Temperature Time [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
Contact	GarageCarDoor	"Garage Car Door [MAP(en.map):%s]"	(DoorContacts)	
Contact	GarageDoor	"Garage Door [MAP(en.map):%s]"	(DoorContacts)	
Contact	ManCaveClosetDoor	"Man Cave Closet Door [MAP(en.map):%s]"	(DoorContacts)	
Contact	AtticDoor	"Attic Door [MAP(en.map):%s]"	(DoorContacts)	
Contact	Door	"Door [MAP(en.map):%s]"	(DoorContacts)	
DateTime	GarageCarDoorOpenTime	"Garage Car Door Last Opened [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	GarageDoorOpenTime	"Garage Door Last Opened [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	ManCaveClosetDoorOpenTime	"Man Cave Closet Door Last Opened [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	AtticTempDoorOpenTime	"Attic Door Last Opened [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	DoorOpenTime	"Door Last Opened [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	GarageCarDoorClosedTime	"Garage Car Door Last Closed [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	GarageDoorClosedTime	"Garage Door Last Closed [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	ManCaveClosetDoorClosedTime	"Man Cave Closet Door Last Closed [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	AtticTempDoorClosedTime	"Attic Door Last Closed [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	
DateTime	DoorClosedTime	"Door Last Closed [%1\$ta %1\$tb %1\$td, %1\$ty %1\$tl:%1\$tm %1\$tp]"	<clock>	

# One-Wire Devices Raw Temperatures

One-wire devices		
Raw Temperatures		
	Front Temperature	5.9 °C
	Garage Car Door Temperature	12.9 °C
	Garage Door Temperature	16.1 °C
	Deck Outside Temperature	3.9 °C
	Man Cave Closet Temperature	- °C
	Attic Temperature	8.3 °C
	Temperature	18.1 °C

# Temperatures as Strings

Combined Temperatures		
	Front Temperature	42.8°F (6.0°C)
	Garage Car Door Temperature	55.3°F (12.9°C)
	Garage Door Temperature	61.1°F (16.2°C)
	Deck Outside Temperature	
	Man Cave Closet Temperature	-
	Attic Temperature	47.0°F (8.3°C)
	Temperature	64.6°F (18.1°C)

# Sitemaps

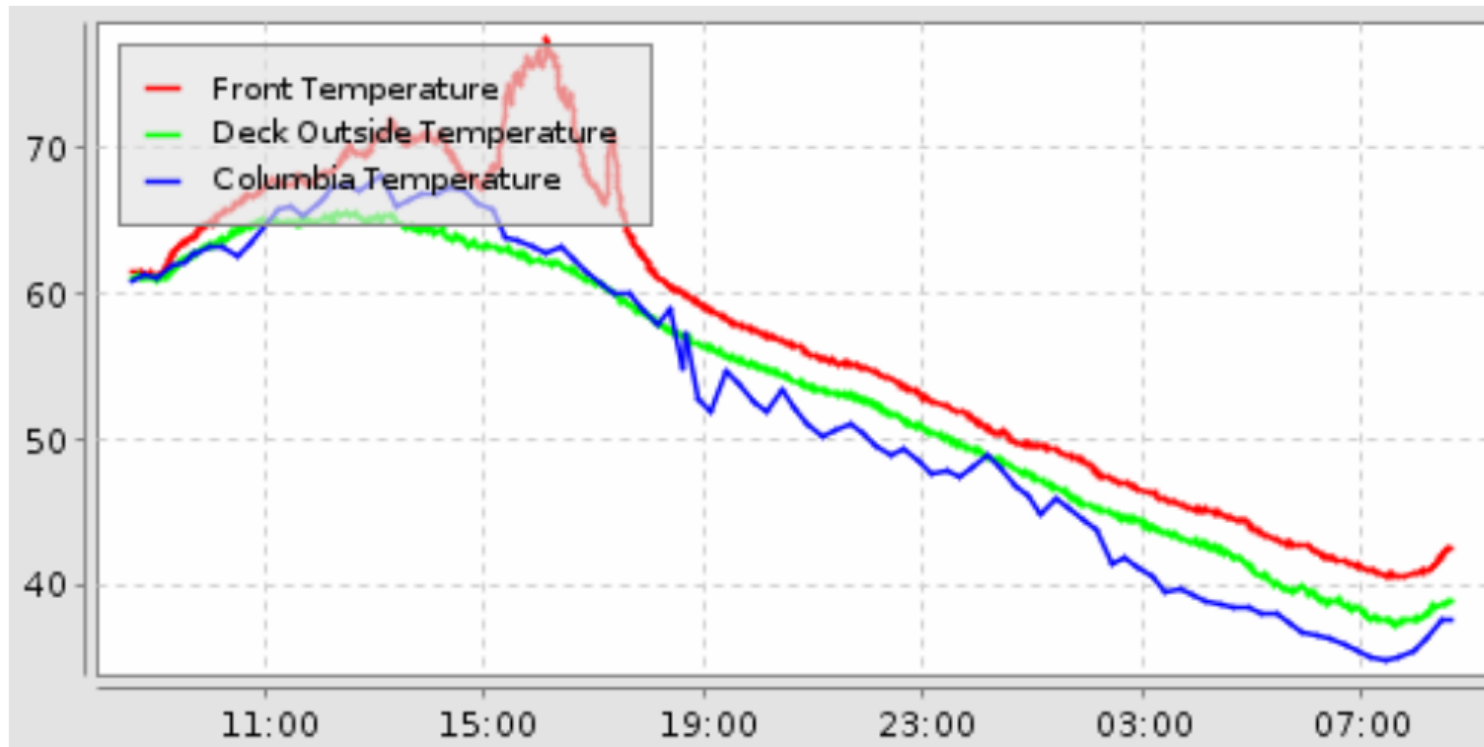
- Defines which items are displayed and where
- Can be hierarchical
- Used by default page viewer, smart phone apps, and others
- There can be more than one per system
  - Debugging sitemap for ALL items
  - Specialty maps with only certain items

# Persistence

- Allows values to be recalled
  - Graphing
  - High or low temperatures
- Defaults to a round robin database
- Can use other databases like MySQL
- Stores data as time-value pairs

# Graph of Outside Temperatures

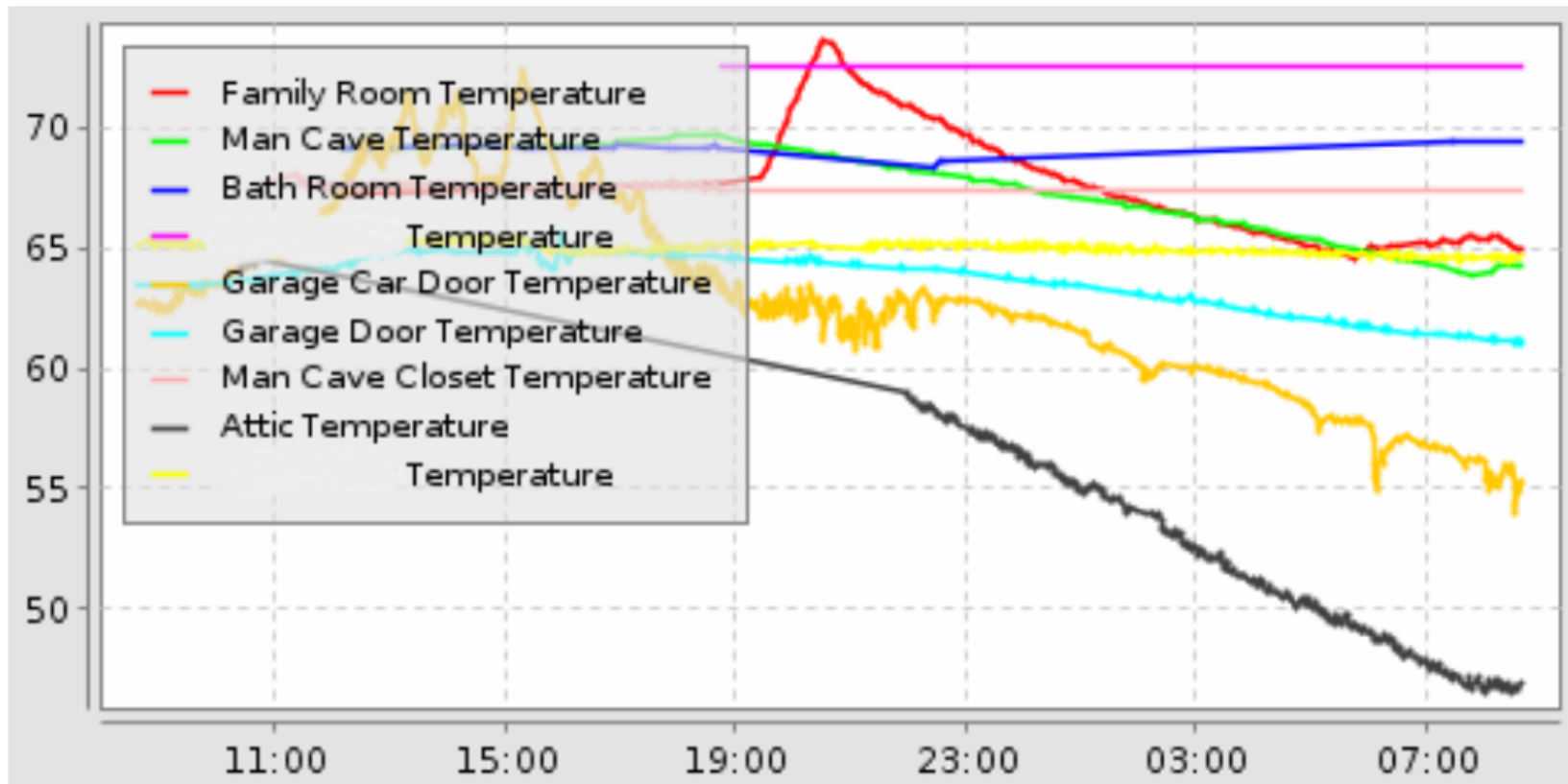
Outdoor Temperatures



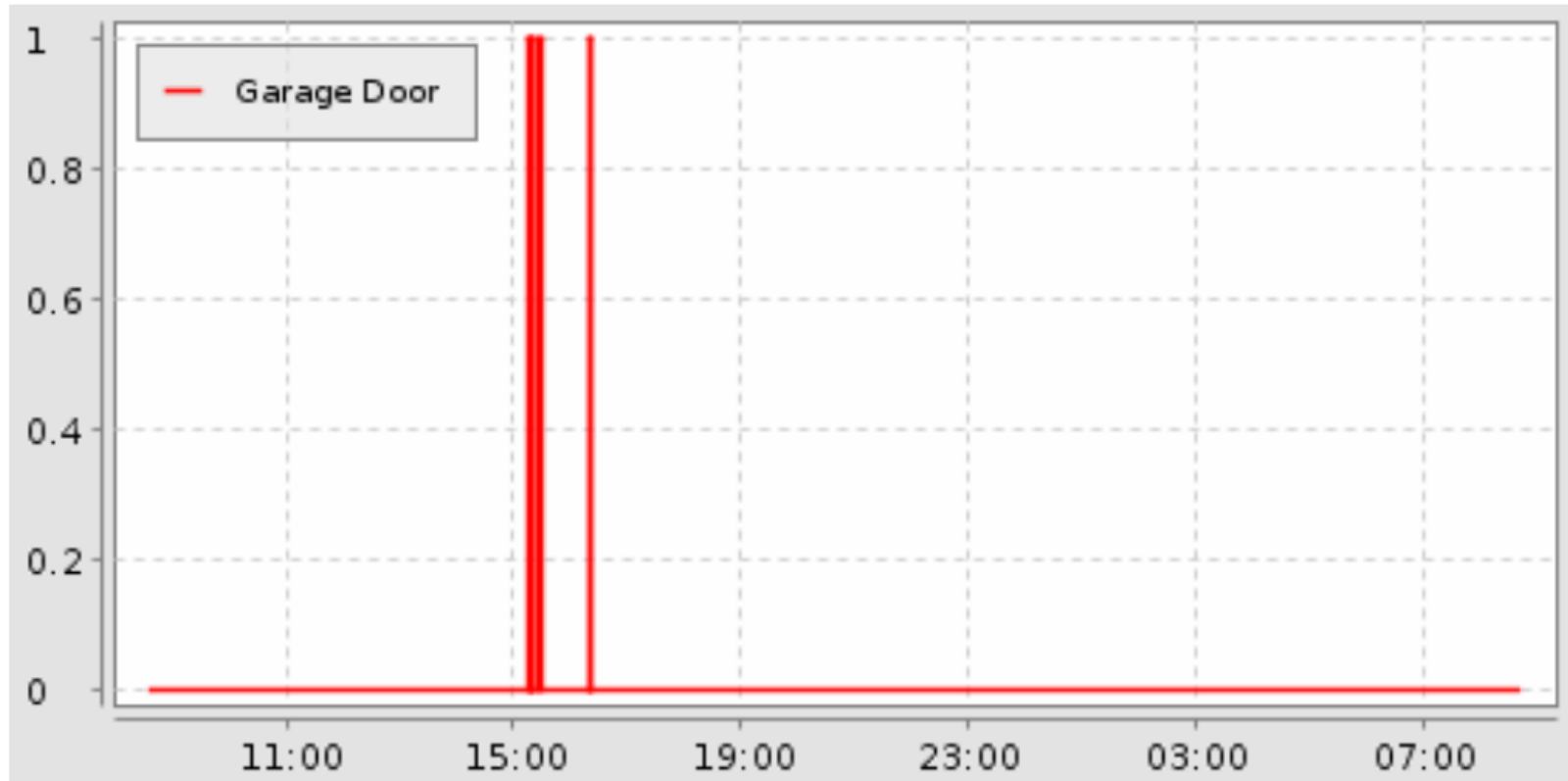


# Graph of Inside Temperatures

Indoor Temperatures



# Graph of Garage Door Contact



# Rules

- Sort of an If This Then That type of interface
- Trigger, the when part
  - update of a value
  - change of value
  - specific change in value (off to on)
  - periodically (like crontab)
  - timer expiry
  - start up
  - others...
- Actions, the then part
  - basically a snippet of Java code
  - can convert units of updated values (e.g, °C to °F)
  - can start a timer for a reminder (you left the garage door open)

# 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

- Generate a web page (default)
- Generate a weather page
- Access RESTful data
- Access graph .png

# 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%

# MQTT

- Simple messaging protocol
- Simple interface:
  - Topic string (“major/minor”)
  - Payload (value)
  - QOS (at most once, at least once, exactly once)
  - Retention
- Need an MQTT broker
  - Publish information 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



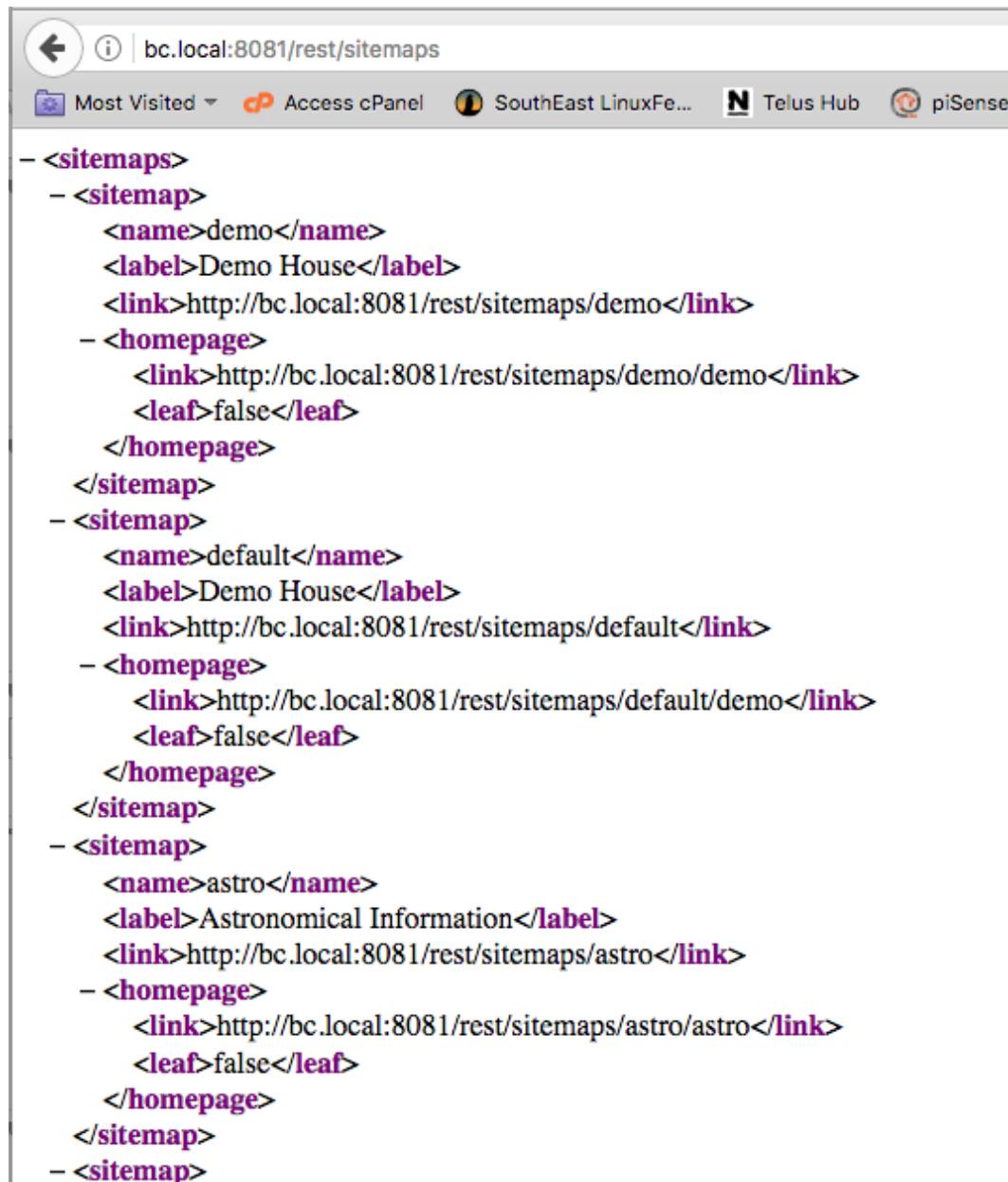
# Example Uses of MQTT

- 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
- (not for non-real-time data like water level data)

# RESTful API

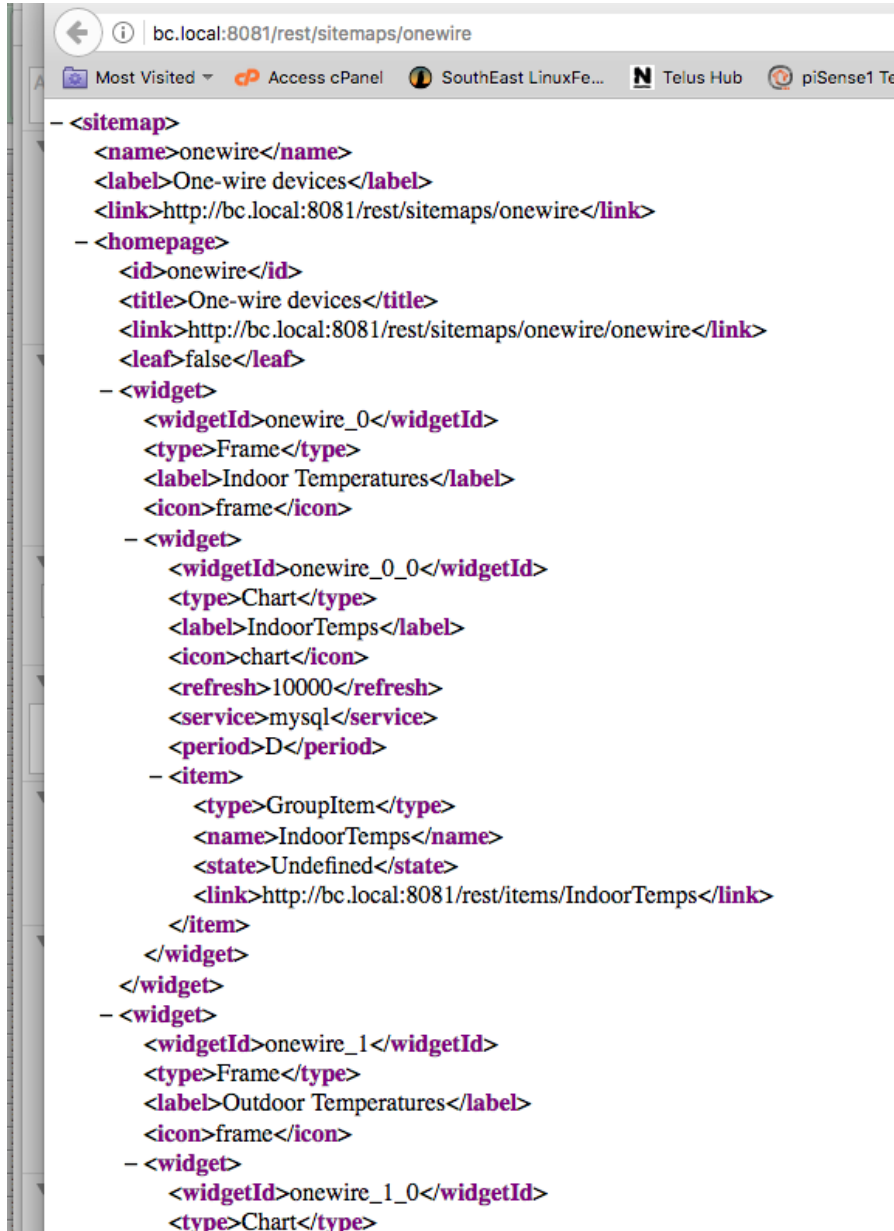
- access to item info and sitemaps
- can speak different protocols: xml, JSON,
- <http://bc.local:8081/rest>
- <http://bc.local:8081/rest/items>
- <http://bc.local:8081/rest/sitemaps>
- [http://bc.local:8081/rest/items/X10\\_Test\\_A1/state](http://bc.local:8081/rest/items/X10_Test_A1/state)
- <http://bc.local:8081/rest/sitemaps/kiosk>
- access historical data with HABmin

# RESTful interface: rest/sitemaps



```
<?xml version="1.0"?>
<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>
</sitemaps>
```

# RESTful interface: rest/sitemaps/onewire



The screenshot shows a web browser window with the address bar displaying `bc.local:8081/rest/sitemaps/onewire`. The browser's address bar also shows a search icon, a magnifying glass icon, and a list of bookmarks including "Most Visited", "Access cPanel", "SouthEast LinuxFe...", "Telus Hub", and "piSense1 Te". The main content area displays a RESTful API response in XML format, which is a sitemap for onewire devices. The XML structure is as follows:

```
<?xml version="1.0"?>
<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>
  </homepage>
  <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>
    </widget>
  </widget>
</sitemap>
```

# RESTful Interface: rest/sitemaps/ onewire

```
bc.local:8081/rest/sitemaps/onewire
Most Visited Access cPanel SouthEast LinuxFe... N Telus Hub piSense1

- <widget>
  <widgetId>onewire_3</widgetId>
  <type>Frame</type>
  <label>Farenheit Temperatures</label>
  <icon>frame</icon>
- <widget>
  <widgetId>onewire_3_0</widgetId>
  <type>Text</type>
  <label>Front Temperature [73.5 °F]</label>
  <icon>temperature</icon>
- <item>
  <type>NumberItem</type>
  <name>FrontFTemp</name>
  <state>73.5125</state>
  <link>http://bc.local:8081/rest/items/FrontFTemp</link>
</item>
</widget>
- <widget>
  <widgetId>onewire_3_0_1</widgetId>
  <type>Text</type>
  <label>Garage Car Door Temperature [74.1 °F]</label>
  <icon>temperature</icon>
- <item>
  <type>NumberItem</type>
  <name>GarageCarDoorFTemp</name>
  <state>74.075</state>
  <link>http://bc.local:8081/rest/items/GarageCarDoorFTemp</link>
</item>
</widget>
- <widget>
  <widgetId>onewire_3_0_1_2</widgetId>
  <type>Text</type>
  <label>Garage Door Temperature [67.1 °F]</label>
  <icon>temperature</icon>
- <item>
  <type>NumberItem</type>
  <name>GarageDoorFTemp</name>
  <state>67.1</state>
```

# 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=BasementDoorFTemp
        Text item=GarageCarDoor
        Text item=GarageDoor
        Text item=ManCaveClosetDoor
        Text item=AtticDoor
        Text item=BasementDoor
        Chart item=OutdoorTemps period=D service="mysql" refresh=10000
        Chart item=IndoorTemps period=D service="mysql" refresh=10000
    }
}
```

# 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**

# 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-the
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;

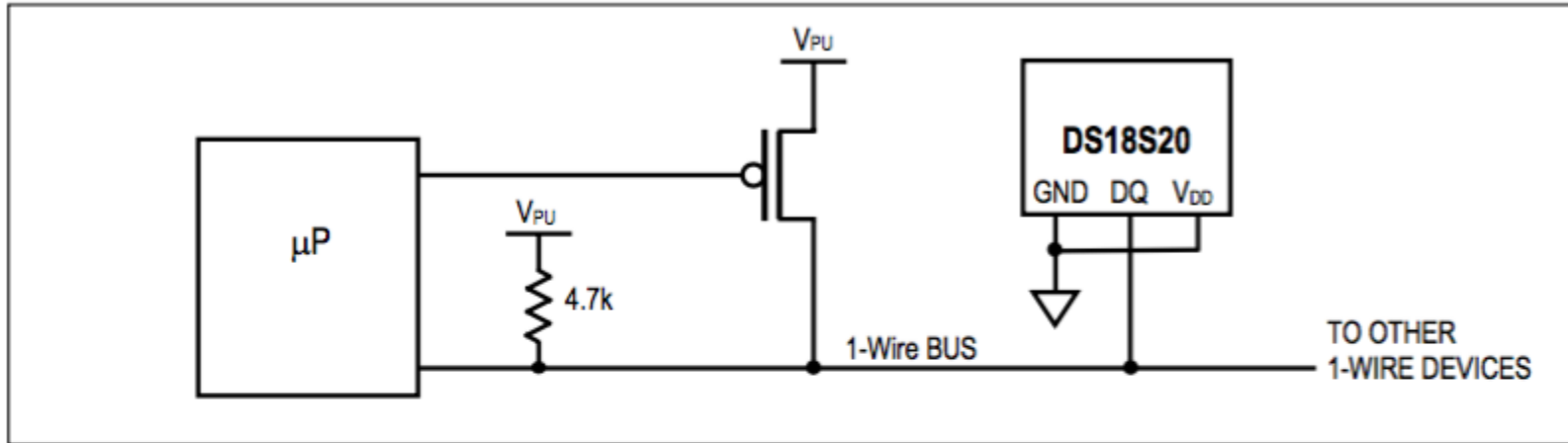
function stopclock (){
    if(timerRunning) {
```

# One-Wire Devices

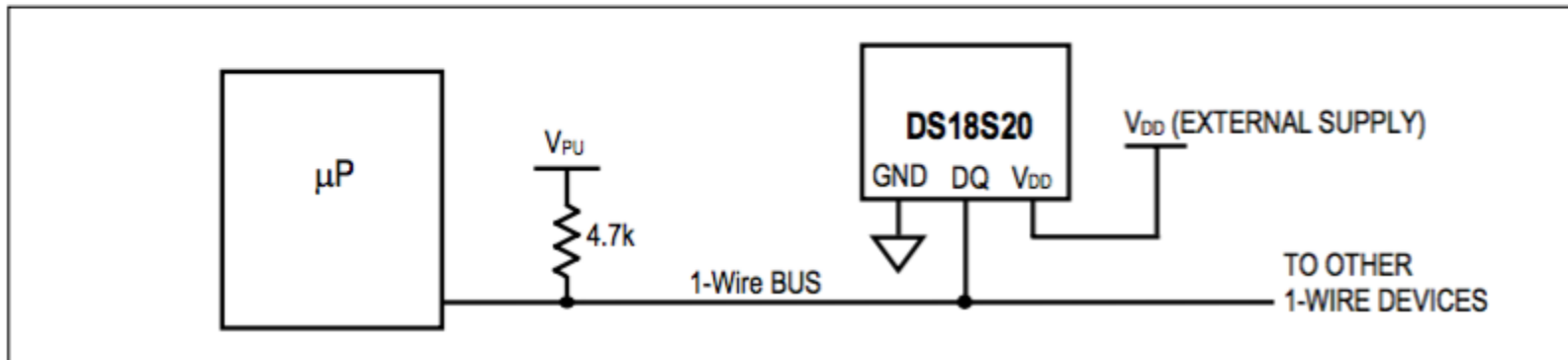
- Mature technology
- Uses one cable: power, ground, signal
- Easiest use to measure temperatures
- An analog to digital chip is available
- Can monitor or control single bits:
  - Door contact
  - Door bell?
  - Garage door opener??
- Can interface with Raspberry Pi boards or USB dongle
- Interfaces managed by OWFS
- OWFS can be integrated into OpenHAB

# Powering One-Wire Devices

**Figure 4. Supplying the Parasite-Powered DS18S20 During Temperature Conversions**



**Figure 5. Powering the DS18S20 with an External Supply**



# OWFS Top Directory Listing

<a href="#">OWFS</a>	<a href="#">Bus listing</a>	<a href="#">OWFS home</a>
<b>directory</b>		
<a href="#">top</a>	highest level	directory
<a href="#">28.C6F580080000</a>	28.C6F580080000	directory
<a href="#">28.591F82080000</a>	28.591F82080000	directory
<a href="#">28.6D0883080000</a>	28.6D0883080000	directory
<a href="#">28.77D881080000</a>	28.77D881080000	directory
<a href="#">22.92C04B000000</a>	22.92C04B000000	directory
<a href="#">22.19E44B000000</a>	22.19E44B000000	directory
<a href="#">bus.0</a>	bus.0	directory
<a href="#">uncached</a>	uncached	directory
<a href="#">settings</a>	settings	directory
<a href="#">system</a>	system	directory
<a href="#">statistics</a>	statistics	directory
<a href="#">structure</a>	structure	directory
<a href="#">simultaneous</a>	simultaneous	directory
<a href="#">alarm</a>	alarm	directory

# OWFS

## Listing of Top Level Busses

<a href="#">OWFS</a>	<a href="#">Bus listing</a>	<a href="#">OWFS hc</a>
<b>bus.0</b>		
<a href="#">top</a>	highest level	directory
<a href="#">interface</a>	interface	directory
<a href="#">28.C6F580080000</a>	28.C6F580080000	directory
<a href="#">28.591F82080000</a>	28.591F82080000	directory
<a href="#">28.6D0883080000</a>	28.6D0883080000	directory
<a href="#">28.77D881080000</a>	28.77D881080000	directory
<a href="#">22.92C04B000000</a>	22.92C04B000000	directory
<a href="#">22.19E44B000000</a>	22.19E44B000000	directory
<a href="#">bus.7</a>	bus.7	directory
<a href="#">bus.6</a>	bus.6	directory
<a href="#">bus.5</a>	bus.5	directory
<a href="#">bus.4</a>	bus.4	directory
<a href="#">bus.3</a>	bus.3	directory
<a href="#">bus.2</a>	bus.2	directory
<a href="#">bus.1</a>	bus.1	directory
<a href="#">bus.0</a>	bus.0	directory
<a href="#">uncached</a>	uncached	directory
<a href="#">settings</a>	settings	directory
<a href="#">system</a>	system	directory
<a href="#">statistics</a>	statistics	directory
<a href="#">structure</a>	structure	directory

# OWFS Subbus 0/0

<a href="#">OWFS</a>	<a href="#">Bus listing</a>	<a href="#">OWFS homepage</a>
<b>bus.0/bus.0</b>		
<a href="#">up</a>	higher level	directory
<a href="#">interface</a>	interface	directory
<a href="#">28.77D881080000</a>	28.77D881080000	directory
<a href="#">22.92C04B000000</a>	22.92C04B000000	directory
<a href="#">22.19E44B000000</a>	22.19E44B000000	directory
<a href="#">simultaneous</a>	simultaneous	directory
<a href="#">alarm</a>	alarm	directory

# OWFS Subbus 0/1

<a href="#">OWFS</a>	<a href="#">Bus listing</a>	<a href="#">OWFS home</a>
<b>bus.0/bus.1</b>		
<a href="#">up</a>	higher level	directory
<a href="#">interface</a>	interface	directory
<a href="#">28.C6F580080000</a>	28.C6F580080000	directory
<a href="#">28.591F82080000</a>	28.591F82080000	directory
<a href="#">28.6D0883080000</a>	28.6D0883080000	directory
<a href="#">simultaneous</a>	simultaneous	directory
<a href="#">alarm</a>	alarm	directory

# OWFS Device Listing for DS18B20

28.77D881080000

[uncached version](#)

<a href="#">up</a>	directory
address	2877D88108000010
alias	<input type="text"/> <a href="#">CHANGE</a>
crc8	10
errata	<a href="#">errata</a>
family	28
fasttemp	15.5
id	77D881080000
locator	FFFFFFFFFFFFFFFF
power	YES (1)
r_address	1000000881D87728
r_id	00000881D877
r_locator	FFFFFFFFFFFFFFFF
scratchpad	F8004B461FFF08106E
temperature	15.5625
temperature10	15.5
temperature11	15.625
temperature12	15.5625
temperature9	15.5
temphigh	<input type="text"/> 75 <a href="#">CHANGE</a>
templow	<input type="text"/> 70 <a href="#">CHANGE</a>
type	DS18B20



# OWFS Device Listing for DS1822

22.19E44B000000

[uncached version](#)

<a href="#">up</a>	directory
address	2219E44B000000E9
alias	<input type="text"/> <input type="button" value="CHANGE"/>
crc8	E9
errata	<a href="#">errata</a>
family	22
fasttemp	16
id	19E44B000000
locator	FFFFFFFFFFFFFFFF
power	YES (1)
r_address	E90000004BE41922
r_id	0000004BE419
r_locator	FFFFFFFFFFFFFFFF
scratchpad	00014B461FFF1010C4
temperature	16
temperature10	16
temperature11	16.125
temperature12	16
temperature9	16
temphigh	75 <input type="text"/> <input type="button" value="CHANGE"/>
templow	70 <input type="text"/> <input type="button" value="CHANGE"/>
type	DS1822

# OWFS Device Listing with Errors

28.6D0883080000

[uncached version](#)

<a href="#">up</a>	directory
address	286D0883080000DD
alias	<input type="text"/> <input type="button" value="CHANGE"/>
crc8	DD
errata	<a href="#">errata</a>
family	28
fasttemp	85
id	6D0883080000
locator	FFFFFFFFFFFFFFFF
power	YES (1)
r_address	DD00000883086D28
r_id	00000883086D
r_locator	FFFFFFFFFFFFFFFF
scratchpad	50054B467FFF0C10
temperature	6.5
temperature10	85
temperature11	85
temperature12	6.5
temperature9	85
temphigh	<input type="text" value="75"/> <input type="button" value="CHANGE"/>
templow	<input type="text" value="70"/> <input type="button" value="CHANGE"/>
type	DS18B20

85 indicates a  
communication  
error

# Next Steps

- Make kiosk and sitemaps more attractive
- Add sensor check logic
  - Are the batteries dead or dying? not heard from unit for a while
  - Is a radio or radio link down? not heard from group of units
  - Concurrence between redundant sensors? vote before acting
- Add reminders
  - Garage door left open too long after dark
  - Open and close window to manage indoor temperature
- Add more sensors (**WARNING: addictive**)
- Clean up persistence
- Play with scraping
- Play with integrating Nagios
- Push notifications
- Explore one-wire more