Building IoT systems with openHAB

Matt Porter
Konsulko
mporter@konsulko.com
Overview

+ Timeline
+ Frameworks and Protocols
+ openHAB architecture
+ openHAB features
+ openHAB configuration
+ openHAB examples
+ Demo
Timeline

- ARPANET online in 1969 with “things” talking Network Control Program (NCP)
- Internet born in 1983: ARPANET “things” start talking TCP/IP
- Many Internet connected appliances created from 1990 to present
- Kevin Ashton (Auto-ID) coins “IoT” in 1999
- Media goes into a frenzy about IoT that just won’t quit.
- openHAB started in 2010
Frameworks

+ AllJoyn - framework for distributed applications
  + [https://allseenalliance.org/developers/learn/architecture](https://allseenalliance.org/developers/learn/architecture)
+ IOTivity - framework for Machine to Machine (M2M) communication
  + [https://www.iotivity.org/](https://www.iotivity.org/)
+ Kura - OSGi-based framework for M2M applications
  + [https://eclipse.org/kura/](https://eclipse.org/kura/)
+ Mihini - Lua-based M2M framework
  + [https://eclipse.org/mihini/](https://eclipse.org/mihini/)
+ openHAB - Home Automation and IoT gateway framework
  + [http://openhab.org](http://openhab.org)
+ ...
Protocols

+ **CoAP (Constrained Application Protocol)**
  + request/response, low overhead, translates to HTTP
+ **MQTT**
  + pub/sub, low overhead
+ **RESTful HTTP**
  + request/response, one way from devices to service
+ **XMPP (Extensible Messaging and Presence Protocol)**
  + pub/sub, built in authentication
+ ...

...
MQTT

- OASIS standard: MQTT v3.1.1
- Publish/Subscribe and hub/spoke model
  - MQTT brokers provide the communication hub
- Mosquitto 1.3.4 broker supports MQTT v3.1.1
- Fixed header required, variable header and payload optional

Fixed header just 2 bytes

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte 1</td>
<td>MQTT Control Packet type</td>
<td>Flags specific to each MQTT Control Packet type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte 2...</td>
<td>Remaining Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
openHAB basics

+ FOSS automation software
+ [http://www.openhab.org](http://www.openhab.org)
  + Eclipse Public License 1.0
+ Written mostly in Java
+ Central automation hub
+ Hardware/protocol agnostic
  + Many bindings [http://www.openhab.org/features-tech.html](http://www.openhab.org/features-tech.html)
+ Component-based architecture
  + Each component is an OSGi bundle
openHAB architecture

+ **Event bus**
  - Asynchronous communication bus
+ **Repository**
  - Persistent storage
+ **Items**
  - Objects that can be read or written
  - Have a type and state
+ **Bindings**
  - Translate event bus events to/from external system
+ **Actions**
  - Programmatic methods available for use by scripts/rules
openHAB features

+ Plugin framework
+ Rules engine
+ Logging mechanism
+ UI abstraction
  + Sitemap - Tree structure of UI widgets
  + Item UI Providers - Dynamic UI configuration
+ UI implementations
  + Web
  + Android
  + iOS
+ Designer tool - graphic configuration of runtime
openHAB add-ons

**Actions**
- HTTP - access URL on event
- Mail - ancient notification technology
- Pushover/Prowl - notifications
- Twitter - Tweet that your toilet flushed

**Bindings**
- Bluetooth - device proximity events
- GPIO - Linux sysfs-based GPIO events
- KNX - home automation events
- MQTT - raw protocol support
- OneWire - various sensor events
- Serial - RS-232 will never die
- ZWave - home automation events
Running openHAB

+ Runs well on any x86 or ARM board with 512MB+ RAM
+ OpenJDK or Oracle JREs are supported
  + Some bindings may not work on OpenJDK on ARM
+ Packaged on some distros
  + Debian Cloudbees repository has the core and all bindings packaged
  + openhab-runtime
  + openhab-addon-*
  + $ cat /etc/apt/sources.list.d/openhab.list
deb http://repository-openhab.forge.cloudbees.com/release/1.6.1/apt-repo/
openHAB configuration

$(openhab)/configurations/

openhab.cfg
items/*.items
persistence/*.persist
rules/*.rules
scripts/*.script
sitemaps/*.sitemap
transform/*.map
openhab.cfg

#------------------------------------------------------------------------------
# Mail Action configuration
#------------------------------------------------------------------------------
# The SMTP server hostname, e.g. "smtp.gmail.com"
mail:hostname=smtp.gmail.com

# the SMTP port to use (optional, defaults to 25 (resp. 587 for TLS))
mail:port=587

# the username and password if the SMTP server requires authentication
mail:username=torvalds
mail:password=linux1991

# The email address to use for sending mails
mail:from=Not Really Linus <torvalds@gmail.com>

# set to "true", if TLS should be used for the connection
# (optional, defaults to false)
mail:tls=true
home.items

Contact FrontDoor "Front Door [MAP(en.map):%s]" {mqtt="<[openhab:/house/frontdoor:state:default]>"}

Contact GarageDoor "Garage Door [MAP(en.map):%s]" {zwave="3:command=sensor_binary"}
openHAB rules

- **Java-like**
  - Imports
  - Variable declarations
  - Rules

```java
var VarType var1

rule “rule name”
  when
    <trigger1> or <trigger2>
  then
    <execute_actions>
end
```
openHAB triggers

+ Item/Event-based

  Item <item> received command [<command>]
  Item <item> changed [from <state>] [to <state>]

+ Time-based

  Time is midnight

+ System-based

  System started
openHAB actions

+ Actions used in rules engine to accomplish a task
+ Core actions

  sendCommand()
  postUpdate()
  logInfo()

+ Binding actions

  sendMail()
  pushNotification()
  sendTweet()
  sendXbmcNotification()
home.rules

rule "Notify Front Door"
when
    Item FrontDoor changed
then
    pushover("Front door is " + FrontDoor.state.toString)
end

rule "Notify Garage Door"
when
    Item GarageDoor changed
then
    pushover("Garage door is " + GarageDoor.state.toString)
end

If the state of the door item changes, send a notification indicating the state of the door.
CLOSED=closed
OPEN=open
-=unknown

A contact item has the state of CLOSED, OPEN, or - (undefined). These are too shouty to print out as is, so transform to lower case before sending to the action.
default.sitemap

sitemap default label="Home"
{
    Frame label="House" {
        Text item=FrontDoor
    }
    Frame label="Garage" {
        Text item=GarageDoor
    }
}
scripts and persistence

- Scripts are another tool useful for reuse of code blocks between rules
  - Java syntax like Xtend language is used
- Persistence allows multiple methods for state to be save
  - Each Item may specify a persistence strategy
- Addons
  - db4o
  - mysql
  - mongodb
  - Logback
ESP8266-based door sensor

- $2-3 WiFi SoC module with Tensilica core and GNU toolchain
  - [http://www.esp8266.com](http://www.esp8266.com)
- NodeMcu - FOSS firmware with lua interpreter for ESP8266
  - [http://www.nodemcu.com](http://www.nodemcu.com)
  - Full I/O library including MQTT v3.1.1 client compatible with openHAB
- Reed switch interfaced to GPIO on ESP8266
- Just 28 lines of lua
  - Configure WiFi
  - Handle GPIO/switch interrupts
  - Publish MQTT “open”/”closed” messages
Door sensor code

-- Door switch contact interrupt callback
function switchcb(level)
    if level == 1 then
        state = "CLOSED"
    else
        state = "OPEN"
    end

    -- Publish a message on each change in state
    m:publish(topic, state, 0, 0, function(conn) print("sent") end)
end

-- Create an MQTT client instance and connect to the broker
m = mqtt.Client(clientid, keepalive, username, password)
m:connect(broker, port, 0, function(conn) print("connected") end)

-- Configure GPIO2 as an interrupt with a pullup
gpio.mode(gpio2, gpio.INT, gpio.PULLUP)
-- Set GPIO2 to call our handler on both edges
gpio.trig(gpio2, "both", switchcb)
ZWave Tilt Sensor

- **Zwave** is a proprietary mesh network
  - Controllers and common sensors have open protocols
  - Fully supported in openHAB
- **ZWave products are easy to purchase in U.S. at any home improvement store**
  - Cheap off the shelf sensors
- **AEON Labs Z-Stick USB controller**
  - [http://aeotec.com/z-wave-usb-stick](http://aeotec.com/z-wave-usb-stick)
  - Push button inclusion of ZWave device to mesh network
  - Works out of the box with openHAB
- **EcoLink garage door tilt sensor**
  - Battery powered tilt sensor suitable for overhead doors
  - Works out of the box with openHAB
Démo
openHAB future

+ More bindings, of course
+ openHAB2 is coming
  + optimize for embedded (hub with 256MB RAM)
  + Minimal runtime
  + Switch to Concierge OSGi?
  + New binding API incorporate concept of “Things”
  + “Things” will be discoverable (IP addresses, etc.)
  + New UI based on material design
Q&A

References

+ https://github.com/konsulko/iot-openhab
+ http://www.openhab.org
+ https://github.com/openhab/openhab/wiki/MQTT-Binding
+ https://github.com/openhab/openhab/wiki/Z-Wave-Binding
+ http://nodemcu.com/
+ http://esp8266.com
+ http://www.openzwave.com/