OPEN REMOTE
Open Source Home Automation
Eric Bariaux
(Home) Automation State of the Union

Why Open Remote?

Open Remote Details
Home Automation

- Lighting
- Audio
- Video
- HVAC
- Communications
- Energy Management
- Security
- Blinds
- Camera
- Health Care

Monday 6 February 12
Home Automation

Residential

Hotels

Industrial

Offices

Hospitals

...
Functionality

Control

Automation
Platform

Devices, Actuators and Sensors

Logic

User Interaction
Problems

One protocol per vendor

Proprietary protocols

Fragmented standards

Expensive touch panels

Niche and extremely expensive distribution
Our solution

Open source
Our solution

Open source

+ 

Adoption of standard
Our solution

Open source

+ 

Adoption of standard

+ 

Off the shelf hardware

Monday 6 February 12
Hardware
Our solution

- Open source
- Adoption of standard
- Off the shelf hardware
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
The ecosystem
Controller (ORB)

Runtime brain of the system

Talks to the devices

Implemented as a J2SE application

Runs on off the shelf hardware
Panels

Renders a graphical user interface

Controls the ORB

Receives information from ORB

Wall-mountable visible interface to the system
Panels

Android

iOS

Rich Web Client
Online Designer

Cloud based tool for system configuration
Separation between device declaration
and UI definition
Devices

[Image of a KNX system configuration showing a device named "Light 1 on" and an "Edit command" window with details:
- **Name**: Light 1 on
- **Protocol**: KNX
- **KNX attributes**:
  - **Group Address**: 0/0/1
  - **KNX Command**: ON
  - **KNX DataPoint Type (DPT)**: 1.001]

Monday 6 February 12
Online Designer
Online Designer Tech

Java Enterprise application

XML system description for run-time

Uses Beehive as the backend
Beehive

Service and storage repository

Cloud based

Access via designer

and a REST API
Control command

```xml
<commands>
  <command id="46" protocol="knx">
    <property name="command" value="ON" />
    <property name="groupAddress" value="0/0/1" />
    <property name="DPT" value="1.001" />
  </command>
  <command id="47" protocol="knx">
    <property name="command" value="OFF" />
    <property name="groupAddress" value="0/0/1" />
  </command>
</commands>
```

controller.xml

Monday 6 February 12
Control command

controller.xml

Monday 6 February 12
Control command

controller.xml

<commands>
  <command id="46" protocol="knx">
    <property name="command" value="ON" />
    <property name="groupAddress" value="0/0/1" />
    <property name="DPT" value="1.001" />
  </command>
  <command id="47" protocol="knx">
    <property name="command" value="OFF" />
    <property name="groupAddress" value="0/0/1" />
  </command>
</commands>
Control command

controller.xml

```xml
<commands>
  <command id="46" protocol="knx">
    <property name="command" value="ON" />
    <property name="groupAddress" value="0/0/1" />
    <property name="DPT" value="1.001" />
  </command>
  <command id="47" protocol="knx">
    <property name="command" value="OFF" />
    <property name="groupAddress" value="0/0/1" />
  </command>
</commands>
```
Control command

controller.xml
Control command

panel.xml

```xml
<absolute left="30" top="20" width="102" height="102">
  <button id="40" name="" hasControlCommand="true">
    <default>
      <image src="PowerOn1320938966346.png" />
    </default>
  </button>
</absolute>

<absolute left="188" top="20" width="102" height="102">
  <button id="42" name="" hasControlCommand="true">
    <default>
      <image src="PowerOff1320939062377.png" />
    </default>
  </button>
</absolute>
```
Control command

panel.xml
Control command

panel.xml
Control API

controller.xml

Monday 6 February 12
Control API

/rest/control/40/click

controller.xml
Control API

/rest/control/40/click

controller.xml
ORB details
ORB details

```
<<interface>>
ExecutableCommand
send()
```
Read Command Flow

- **Panel read request (immediate response)**
  - HTTP/REST/XML & JSON
  - HTTP response is withheld by server until new state update arrives or request times out (a.k.a. HTTP server push)

- **Panel read request (delayed)**
  - HTTP/REST/XML & JSON
  - HTTP response is withheld by server until new state update arrives or request times out (a.k.a. HTTP server push)

**Diagram Notes:**

1. Throttle concurrent sensor requests to fixed permanent connections (e.g. KNX, Telnet)
2. Return last in-memory state without triggering device read command (poll)
3. Connectionless, non-cached read command (poll) to device (e.g. HTTP)
4. Listeners for active devices (e.g. KNX bus listener instead of GroupValue Read)
Protocol implementation

<<interface>>
CommandBuilder
build()

<<interface>>
ExecutableCommand
send()

<<interface>>
EventListener
setSensor()
stop()
Community

Fantastic platform for hobbyists

Active open source community

Community has many professionals
THANK YOU

www.openremote.org
eric@openremote.org