



Integrate IR Trans with digitalSTROM Developer Days Hackaton (15.10.2015)

Team:

Goran Petrovikj

Tomasz Hasinski

Jurica Mihačić

Krzysztof Klimek

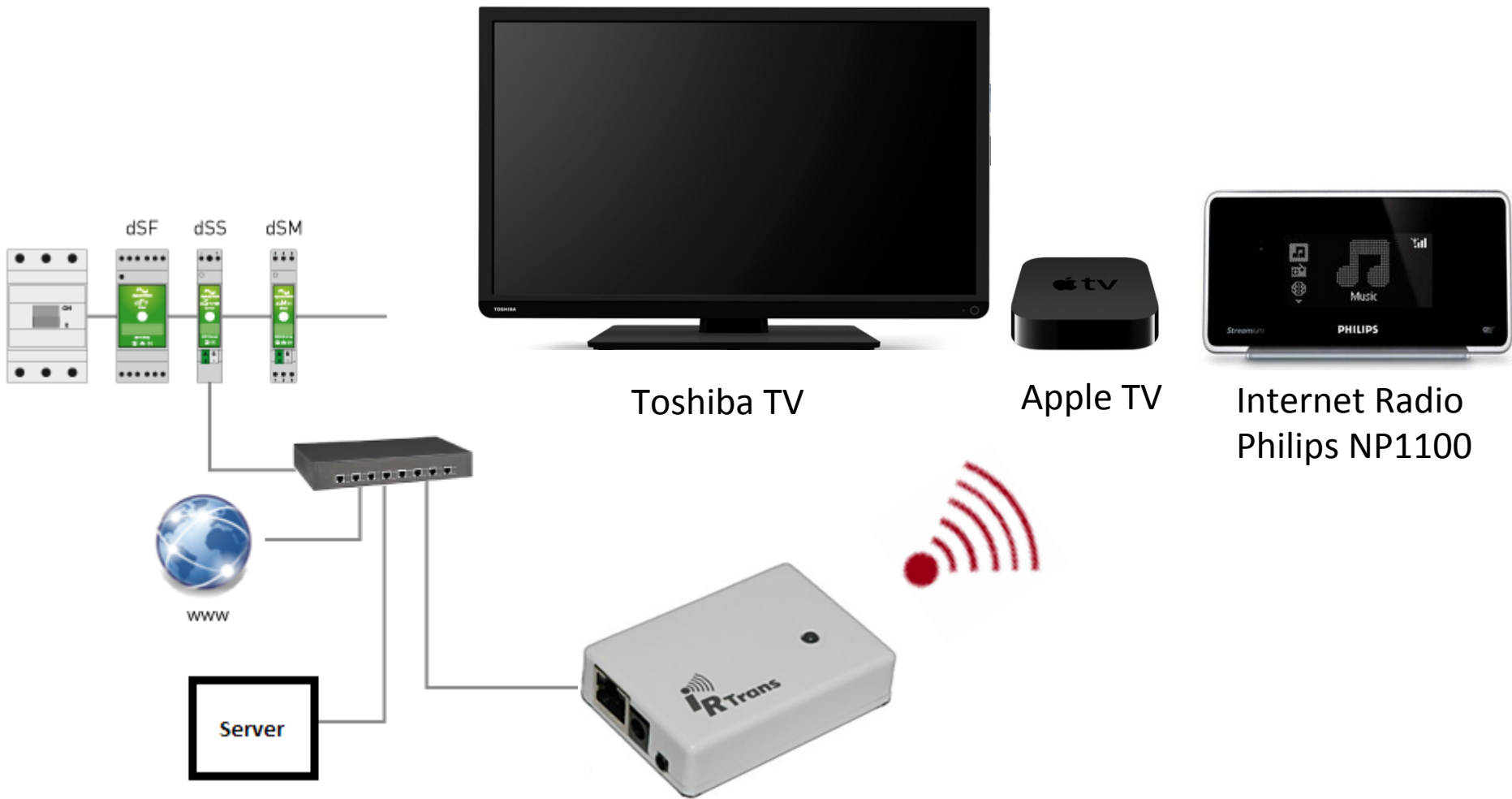
The Goal

- Integrate IR controlled devices such as TV, Radio, Streaming media player with digitalSTROM system
- Play some TV channel / radio station / movie trailer on a scene call in a digitalSTROM equipped apartment

Hardware needed

- digitalSTROM Developer kit
- IR Trans device with Ethernet capability
- Radio controlled devices, in our case
 - Toshiba TV
 - Apple TV
 - Philips Internet Radio
- Small Web Server (which might be later integrated in dSS)
- A router

Wiring



What is IR Trans

- It's a little programmable device that can simulate IR remote control - can send IR commands
- It can learn the IR commands that are usually sent by the remote control, and store them in its internal database
- Features an Ethernet Interface for Network connectivity
- Offers an HTTP API that allows sending commands in format

`http://irtrans.local/send.cgi?`

`remote={remoteName}&command={command}`

Learning the commands

- Use the web interface offered by IR Trans
- For each remote control
 - Give the remote control a name
- For each key:
 - Give the key a name
 - Press the learn button
 - Point the remote control to IR Trans and press the key
 - The code received is shown on the web interface and stored in the internal database

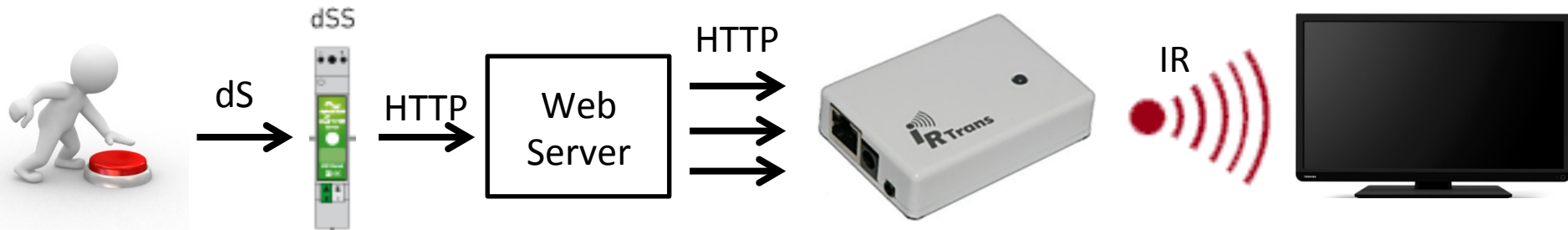


Demo Scenarios

- Light Preset 1 in Room 1
 - Switch TV to channel 3
- Light Preset 2 in Room 1
 - Switch TV to PC input
- Light Preset 3 in Room 1
 - Switch TV to Apple TV input and play the trailer of the top movie of the week
- Light Presets 1, 2, 3 in Room 2
 - Play Internet Radio Station 1, 2, 3 correspondingly

How to Integrate the IR Trans to digitalSTROM

- The easy way is creating a User Defined Action that would call an URL on the IR Trans device
 - Limitation – in order to achieve some action, we usually need to execute a sequence of key presses with some short delay in between
- The flexible way is creating a server app that would handle sending the sequence of commands to IR Trans, and would provide single URL that can be called by a User Defined Action, e.g. <http://myServer.local/tvPreset1>



Server Implementation (Using python web.py)

```
import web
import requests
import time

baseUrl = 'http://172.16.0.7' #IR Trans URL

#Remote names
toshibaTV = 'Toshiba TV'
philipsRadio = 'Philips Internet Radio'
appleTV = 'Apple TV'

def sendCommand(remote, command):
    url = baseUrl + '/send.cgi?remote=' + str(remote) + '&command=' + str(command)
    requests.get(url)
    time.sleep(0.3)

class radioPreset1:
    def GET(self):
        sendCommand(philipsRadio, "Stop")
        sendCommand(philipsRadio, "Home")

        sendCommand(philipsRadio, "Internet Radio")
        sendCommand(philipsRadio, "OK/Play")
        sendCommand(philipsRadio, "OK/Play")
        return 'ok'

...
```

Switch TV to channel 3

- URL: <http://myServer.local/tvPreset1>
- Assumptions:
 - the TV is already on
- Using TV Remote:
 - Press button 3
 - Press OK //channel 3 selected

Switch TV to input PC

- URL: <http://myServer.local/tvPreset2>
- Assumptions:
 - the TV is already on
- Using TV remote:
 - Press button 1
 - Press OK //Starting with a clean state, channel 1 selected
 - Press Source select
 - Press Up //Navigate to PC input source
 - Press OK

Switch TV to input source Apple TV and play trailer of the movie of the week

- URL: <http://myServer.local/tvPreset3>
- Assumption: the TV is already on
- Using TV Remote:
 - Press button 1
 - Press OK //Starting with a clean state, channel 1 selected
 - Press Source select
 - Navigate Down x 5 //Navigate to HDMI 1
 - Press OK
- Using Apple TV Remote:
 - Press Menu x 2
 - Press Down x 2
 - Press Menu //Selection is in second row
 - Press Left //Select movies
 - Press Up //Select movie of the week
 - Press Center //Open movie of the week
 - Press Center //Play trailer

Play Internet Radio Station

- URL: <http://myServer.local/tvPreset1> (or 2, or 3)
- Assumptions
 - The Radio is on
 - The favorite station list is already defined
- Using Radio Remote:
 - Press Stop
 - Press Home
 - Press Internet Radio
 - Press OK // Select Favorite stations
 - Press Down // x 0, 1, 2 depending on preset
 - Press OK // Play station

User Defined Actions Configuration

The screenshot displays the 'User Defined Actions' configuration page in a web browser. The browser's address bar shows the URL: `https://172.17.0.149/add-ons/system-addon-user-defined-actions/`. The page header includes the 'digitalSTROM' logo and a title 'User Defined Actions'. Below the header, there are several action management buttons: '+ New Action', 'Edit Action', 'Deactivate Action', 'Test Action', and 'Delete Action'. The main content area features a table with the following data:

Active	Name	Initiated Activities	Conditions	Last Edit
<input checked="" type="checkbox"/>	Radio Preset 1	URL: <code>http://172.16.0.8:8080/radioPreset1</code>		15.10.2015 - 14:17:58
<input checked="" type="checkbox"/>	Radio Preset 2	URL: <code>http://172.16.0.8:8080/radioPreset2</code>		15.10.2015 - 14:18:05
<input checked="" type="checkbox"/>	Radio Preset 3	URL: <code>http://172.16.0.8:8080/radioPreset3</code>		15.10.2015 - 14:18:14
<input checked="" type="checkbox"/>	Radio Preset Off	URL: <code>http://172.16.0.8:8080/radioPresetOff</code>		15.10.2015 - 14:18:23
<input checked="" type="checkbox"/>	TV Preset 1	URL: <code>http://172.16.0.8:8080/tvPreset1</code>		15.10.2015 - 14:20:25
<input checked="" type="checkbox"/>	TV Preset 2	URL: <code>http://172.16.0.8:8080/tvPreset2</code>		15.10.2015 - 14:20:45
<input checked="" type="checkbox"/>	TV Preset 3	URL: <code>http://172.16.0.8:8080/tvPreset3</code>		15.10.2015 - 14:20:56
<input checked="" type="checkbox"/>	TV Preset Off	URL: <code>http://172.16.0.8:8080/tvPresetOff</code>		15.10.2015 - 14:29:58

A green-bordered text box highlights the condition used in the actions: `http://myServer.local/ == http://172.16.0.8:8080/`. At the bottom of the interface, there is a 'Ready' status indicator and a 'Refresh View' button.

Event Responder Configuration

The screenshot shows a web browser window titled "Event Responder" with the URL <https://172.17.0.149/add-ons/system-addon-scene-responder/>. The interface features the digitalSTROM logo and a navigation bar with buttons for "New Responder", "Edit Responder", "Disable Responder", "Test Responder", and "Delete Responder".

Active	Name	Trigger	Initiated Activities	Conditions	Delay
<input checked="" type="checkbox"/>	Radio Preset 1	Preset 1 for Light in Radio Room	Action: Radio Preset 1		00:00:00
<input checked="" type="checkbox"/>	Radio Preset 2	Preset 2 for Light in Radio Room	Action: Radio Preset 2		00:00:00
<input checked="" type="checkbox"/>	Radio Preset 3	Preset 3 for Light in Radio Room	Action: Radio Preset 3		00:00:00
<input checked="" type="checkbox"/>	Radio Preset Off	Off for Light in Radio Room	Action: Radio Preset Off		00:00:00
<input checked="" type="checkbox"/>	TV Preset 1 - Channel3	Preset 1 for Light in TV Room	Action: TV Preset 1		00:00:00
<input checked="" type="checkbox"/>	TV Preset 2 - PC	Preset 2 for Light in TV Room	Action: TV Preset 2		00:00:00
<input checked="" type="checkbox"/>	TV Preset 3 - Apple TV	Preset 3 for Light in TV Room	Action: TV Preset 3		00:00:00
<input checked="" type="checkbox"/>	TV Preset Off	Off for Light in TV Room	Action: TV Preset Off		00:00:00

Ready Refresh View

[Help](#)

Problems when automating IR control

- The communication between the Remote Control and the controlled device is usually one way, meaning no feedback is provided to the sender about the success of the command
- For the human beings it's easy to determine the current state of the controlled device, so it's easy to choose the button that should be pressed next
- Automating the sending of IR commands means having no feedback about the current state of the controlled device. This implies that we always have to start with a clean state (e.g. channel 1, home screen) before sending any commands