
Sunkara Vinodh Kumar
IIT Kharagpur
svinodhkumar.1031@gmail.com

Vlado Handziski
TU Berlin
handziski@tkn.tu-berlin.de

Abstract

The TWIST Actu platform is aimed at extending the capabilities of the TWIST testbed at TU Berlin with support for remote experiments involving building automation sensors and actuators, as part of a wider effort to migrate the testbed from a pure sensor network testbed to one that can also effectively host more challenging cyber-physical system experimental scenarios. In this demo paper we summarize the main features of the hardware and software architecture of TWIST Actu, focusing on the design of the RESTful remote experimentation API that supports a “Testbed as a Service” model of use of the testbed resources. We also present a prototype implementation of the platform and scenario for demonstrating its capabilities.

1 Introduction

The TKN Wireless Indoor Sensor Network Testbed (TWIST) [4] is a flexible testbed architecture implemented on top of open hardware and software. The TWIST instance at TU Berlin is one of the largest indoor sensor network testbeds with public remote access. It integrates 204 sensor nodes, distributed over three floors of the TKN building, offering more than 1500 m² of instrumented office space.

The TWIST Actu platform extends the hardware capabilities of the TWIST testbed with typical building automation sensors and actuators like heating plate valve controllers, light switches/dimmers, window tilt and blinds controllers. Wrapped behind an expressive and flexible remote experimentation API, the TWIST Actu extension enables new test scenarios including:

- evaluation of different Internet/HVAC interfacing approaches, middleware, programming abstractions;
- evaluation of interference effects on the correct operation of different sensing/actuation control loops;
- evaluation of the impact of coexistence problems between different wireless technologies on the operation of building automation systems, etc.

In the following we provide a brief overview of the hardware and software architecture of the platform. More detailed description can be found in [6].

2 Hardware Platform

The TWIST Actu extension has been realized using products from the HomeMatic line of sensors and actuators from the ELV/eQ-3 group [5]. The HomeMatic system is one of the most popular “retrofitting” building automation systems in Germany. It features a very large product palette and its products are available through major distributors of electronic equipment. The HomeMatic nodes use a Texas Instruments CC1100 radio in the 868 MHz band, with a custom communication protocol named BidCos. The protocol uses a form of XOR obfuscation and initial ASK-protected handshake, which have already been reverse-engineered. This opens the opportunity to freely communicate with the HomeMatic equipment using any sensor node platform that has a radio compatible with the CC1100, enabling low-level integration with the rest of the TWIST sensor networks infrastructure. In the current configuration, the TWIST testbed has been extended with seven different types of HomeMatic nodes which have been installed in the CPS Lab room on the second floor of the TKN building.

3 RESTful Remote Experimentation API

The user-facing service API for remote experimentation with building automation actuators has been designed as an extension of the COTEF Testbed Abstraction API (TAA) [1]. Following the Representational State Transfer (REST) architectural style [3], the API is specified as a set of resources which are manipulated through a uniform method set, comprised of the standard HTTP methods like GET, PUT, DELETE, POST, etc. The role and the semantics of the methods are independent from the resources that are being manipulated, specified by a Universal Resource Locator (URL). The abstraction of the access to the building automation sensors and actuators in TWIST Actu has been achieved through two new resources that extend the resource model of the TAA: Channel and Parameter.

The API supports two types of channel resources: sensors and actuators. The sensor channels perform a supervisory function. They sense a particular parameter in the...
environment and generate events when necessary. Actuator channels perform control functions and influence the state of physical objects in the environment. Both sensor and actuator channels are characterized by state values, data points and attributes which are exposed through the parameter resource concept.

Figure 1 illustrates the core resources in the service API of TWIST Actu, their attributes and types. The representation of the resources is realized using JSON serialization that simplifies the development of JavaScript client implementations and increases the human-readability and encoding efficiency in comparison to XML-based solutions. For example, issuing a HTTP GET command to https://www.twist.tu-berlin.de:8001/nodes/IEQ0042249/channels/1/parameters/2 provides the API client with a JSON representation of the parameter representing the state of the window-tilting actuator which is part of the HomeMatic node with unique ID of IEQ0042249:

```
"id": "2",
"media_type": "application/json",
"name": "LEVEL",
"type": "FLOAT",
"value": 0,
"unit": "%",
"min": 0,
"max": 1
```

Following the general RESTful design, the Twist Actu API resources can be modified by issuing a HTTP POST or PUT methods. To optimize the modification of small subsets of the resource, for example just changing the "value" of a given actuator, the service API also supports the HTTP PATCH command.

In addition to this “forward” interface, TWIST Actu also supports efficient, low-latency notification channel in the “reverse” server-to-client direction, leveraging the HTML5 WebSockets concept [2].

4 Demo Description

This demo makes use of a prototypical implementation of the TWIST Actu API which has been developed and deployed for testing. Figure 2 illustrates the system architecture of the current prototype. It shows how the XMLRPC-based interfaces offered by the HomeMatic Central Control Unit (CCU) have been integrated and abstracted via an extended CONET Testbed Adaptation Server implemented in Django [7].

Figure 2. TWIST Actu prototype

The demo illustrates the use of the RESTful TWIST Actu API for experimentation with building automation sensors and actuators. Using the developer tools panel of the Chrome web browser, the users will be able to follow in real-time the interaction between the web client and the server, including the HTTP requests in the client-to-server direction, as well as the websockets-based dispatching of notifications in the server-to-client direction. The correct execution of the actuation commands will be validated using a live video stream from the CPS Lab at the TKN building hosting the TWIST Actu deployment.

5 Acknowledgments

The work has been performed during the first author’s internship at the TKN group at TU Berlin. It has been partially supported by EIT ICT Labs, as part of the activity 12149, “From WSN Testbeds to CPS Testbeds”.

6 References