EdgeX DevKits

Tech Talks – Session 12
Today’s Agenda

• What are EdgeX DevKits?
• EdgeX reminder/overview
• The first EdgeX Community DevKit
  • Raspberry Pi and Grove Sensors
  • Grove Device Service
  • How to use it
  • Example scenario
• Demo
• Q & A
James Butcher

• Senior Solutions Architect at IOTech
• 15 years working in middleware and IoT
• Focused on usability, adoption and roll-out of the technology
• Email - james@iotechsys.com
What are EdgeX DevKits?

• Aimed at helping users get started quickly with EdgeX

• Pre-configured software to work against specific hardware and sensor combinations

• Good for learning EdgeX – get real sensor data flowing quickly

• Can easily sketch out and prototype Edge IoT Proof of Concepts

• Verify real use cases with the sensors and then transition to real/industrial devices

• Both Community and Commercial DevKit options available
Quick EdgeX Reminder
Introducing EdgeX Foundry

• An open source, vendor neutral project (and ecosystem)
• A micro service, loosely coupled software framework for IoT edge computing
• Hardware and OS agnostic, optional use of containers
• Linux Foundation, Apache 2 project
• Goal: enable and encourage growth in IoT solutions
  • The community builds and maintains common building blocks and APIs
  • Plenty of room for adding value and getting a return on investment
  • Collaborative approach, allowing best-of-breed solutions
It's 102° outdoor. Stop the Machine.
EdgeX Micro Service Layers

- Contextually, EdgeX micro services are divided into 4 layers
- Crudely speaking, the layers of EdgeX provide a dual transformation engine
  - 1x - Translating information coming from sensors and devices via hundreds of protocols and thousands of formats into EdgeX
  - 2x - Delivering data to applications, enterprises and cloud systems over TCP/IP based protocols in formats and structures of customer choice
- The DevKits are going to use and demonstrate all of the layers!
Grove Sensors
What are Grove Sensors?

- Seeed Studios Grove Kit consist of the base shield and numerous available sensors
- Supported on Raspberry Pi and other dev boards
- Communicates via I2C interface
- Sensors use GPIO/AIO pins
- Light, Sound, Rotary Angle, Relay, LEDs, LCD, Buzzer, etc
- Easy to use, low cost and widely available
An EdgeX Grove Device Service

• Needed to write a southbound Device Service that can interface with the Grove Devices

• Implemented with C Device Service SDK

• Based on open source libmraa https://github.com/intel-iot-devkit/mraa/

• Requires Device Profile (see next page)

• Typically ran as Docker container with default port 49992
Grove Device Profile

- Maps each sensor’s input/output to EdgeX variables within an EdgeX Device (see table)
- Initially supports sensors from GrovePi+ starter kit
- Two options:
  1. **Single Grove Device** which lists Resources & Commands all of the sensors
  2. **Multiple Grove Devices** with Resources & Commands for each sensor
- Default profile automatically creates a single device named “GroveDevice” – i.e. option 1
- `/res/configuration.toml` and `/res/Grove_Device.yaml`
- Individual devices modelled as yaml files under `/profiles`
Grove Device Profile – Blue LED example

name: "Grove_Device"
manufacturer: "Seeed"
labels:
- "Grove LED, Grove Button, Grove Buzzer, Grove LCD, Grove LightSensor, Grove Relay, Grove RotarySensor, Grove SoundSensor"
description: "Grove Device to Read/Write sensors connected to GrovePi."

deviceResources:
- name: Blue-LED
description: "Turn the Blue LED to On/Off"
attributes:
- Pin_Num: "D2", Interface: "GPIO", Type: "OUT"
properties:
value:
- type: "Bool", readWrite: "RW", size: "1", minimum: "0", maximum: "1", defaultValue: "0"
units:
- type: "String", readWrite: "R", defaultValue: "Enabled/Disabled"

resources:
- name: Set_BlueLed
  set:
  - operation: "set", object: "Blue-LED", property: "value", parameter: "Blue-LED"

Commands:
- name: Set_BlueLed
  put:
  path: "/api/v1/device/\{deviceId\}/Set_BlueLed"
  parameterNames: ["Blue-LED"]
  responses:
  - code: "204"
    description: "valid and accepted"
    expectedValues: []
  - code: "400"
    description: "bad request"
    expectedValues: []
  - code: "503"
    description: "service unavailable"
    expectedValues: []

To turn LED on:
curl '{"BlueLED": "true"}' http://localhost:49992/api/v1/device/all/Set_BlueLed

To turn LED off:
curl '{"BlueLED" : "false"}' http://localhost:49992/api/v1/device/all/Set_BlueLed
Using a DevKit
Where to get an EdgeX DevKit?

- Community and Commercial DevKit options referenced from EdgeX website
- First EdgeX Community version just released:
  - Raspberry Pi 3 64 bit with GrovePi+ Sensors
  - See [https://www.edgexfoundry.org/devkits/](https://www.edgexfoundry.org/devkits/)
  - Links to buy a Pi and GrovePi+ Sensor Kit
  - Instructions to install custom 64 bit Ubuntu OS image
  - Instructions to download and run the EdgeX software
Aim to Validate Real Use Cases

- Simulate real world scenarios with easy to use sensors
- Can span multiple industries and use cases
- Collect, visualize and make decisions with real data at the edge
- Prove there is value in the project
- Verify that EdgeX technology helps with this
- Evolve to use more industrial devices
Today’s DevKit Demo – Smart Factory

• Production / Output Quality Monitoring
  • Validate that the factory output meets required QA levels
  • e.g. bottles, cartons, packages, etc
  • Monitor and analyze size and weight of each item
  • Actuate control devices

• Production Line / Machine Monitoring
  • Validate machinery is operating correctly
  • Safety of human operators
  • Spot performance degradation ahead of breakage
  • Perform timely maintenance routines
  • Visualize on dashboards
  • Send to Cloud for storage and analysis
Today’s DevKit Demo – Smart Factory

What does the demo do?

• Collect sensor data from some of the Grove Sensors with Grove Device Service
• Ingest into EdgeX through Core Services
• Translate data to northbound side, e.g. to MQTT
• Display data and make decisions at the Edge, e.g. with Node-RED:
  • Easy dashboarding
  • Flow programming capability
  • MQTT subscriptions to get data
  • REST calls to actuate

For convenience, Mosquito broker and Node-RED both come pre-installed on supplied Ubuntu Image
Today’s DevKit Demo – EdgeX Architecture

Node-RED

Package Monitoring

Production Line Monitoring

Grove

Today's DevKit Demo – EdgeX Architecture

Grove Production Line Monitoring

Package Monitoring
Today’s DevKit Demo – Physical Architecture

Edge Decision Making and Dashboard

MQTT Broker

EdgeX Core Services

Grove Device Service

- Rotary Angle Sensor
- Light Sensor
- LEDs
- Sound Sensor
- Button
- Buzzer
- LED / LCD

Package Monitoring:
Weight and Size

Line Monitoring:
Vibrations and Emergency Buttons

Web Browser

Laptop
Demo Time!!
Following On

- More DevKits - different environments and sensors
- Sample use cases for different industries
- Blogs/Videos/Tutorials
Key Project Links

Access the code:  
https://github.com/edgexfoundry

Access the technical documentation:  
https://docs.edgexfoundry.org/

Access technical video tutorials:  
https://wiki.edgexfoundry.org/display/FA/EdgeX+Tech+Talks

EdgeX Blog:  
https://www.edgexfoundry.org/news/blog/

Join an email distribution:  
https://lists.edgexfoundry.org/mailman/listinfo

Join the Slack Channels:  
https://edgexfoundry.slack.com

Become a project member:  
https://www.edgexfoundry.org/about/members/join/

LinkedIn:  
https://www.linkedin.com/company/edgexfoundry/

Twitter:  
https://twitter.com/EdgeXFoundry

Youtube:  
https://www.youtube.com/edgexfoundry

EdgeX DevKits:  
https://www.edgexfoundry.org/devkits/community-devkit/
Question and Answer Time

Email: james@iotechsys.com