Integrating Device Connectivity in IoT & Embedded devices

Tom Zamir
IoT Solutions Specialist
tom@iot-experts.net
About me

- **Tom Zamir** – IoT Solutions Specialist
  - Independent Consultant
  - Helping Companies design and implement IoT strategies from Device to Cloud to Customer
  - Background in IT/OT and DevOps Engineering
  - Experience with Agro-tech startups and device manufactures
  - Working experience with various IoT Platforms, hardware and software configurations

- Why am I here
  - Partnership with **EASTRONICS** to offer complete end-to-end IoT Solutions on Azure
  - Expertise in IoT Cloud Infrastructure and Azure
  - Passion for emerging IoT Technologies and the **Azure IoT Offerings**
Agenda

- IoT Architecture and Device Lifecycle
- The Azure IoT Hub
- Adding Connectivity to your devices with Azure IoT Device SDK
- Managing your devices with Azure IoT Services SDK
- Operating on the edge with IoT Edge Gateway SDK
- Orchestrating your IoT solution with the Azure IoT Suite
- Demo - Cloud to Cloud capabilities
Azure IoT Reference Architecture

- Devices and Data Sources
  - IP capable devices
  - Existing IoT devices
  - Low power devices

- Data Transport

- Device and Event Processing
  - Provisioning API
  - Solution Portal
  - Device Registry Store
  - Device State Store
  - Stream Event Processor
  - Storage

- Presentation
  - Data Visualization & Presentation
Sensor to User Application Solutions
**IoT Device Lifecycle**

- **Retire**: Replace or decommission devices after failure, upgrade cycle or service lifetime.
- **Plan**: Group devices and control access according to your organization's needs.
- **Monitor**: Monitor device inventory, health & security while providing proactive remediation of issues.
- **Provision**: Securely authenticate devices, on-board for management and provision for service.
- **Configure**: Provide updates, configuration & applications to assign the purpose of each device.
Azure IoT Hub

- Establish bi-directional communication with billions of IoT devices
- Authenticate per device for security-enhanced IoT solutions
- Register devices at scale with IoT Hub Device Provisioning Service
- Manage your IoT devices at scale with device management
- Extend the power of the cloud to your edge device
Azure IoT Hub

- Device Twins
- Device authentication and secure connectivity
- Device-to-cloud messages
- Rule-base data routing
- Monitoring device connectivity and Operations
- IoT protocols and extensibility
- Scale and High Availability
- Open-source SDKs
Azure IoT Hub – Manage your Devices
Add, remove, view, interact with your devices
Azure IoT Hub – Monitor your Hub/Devices
Quickly visualize your network, identify bottlenecks, alerts, and overall health
Azure IoT Hub – Query Explorer

Query traffic on your IoT Hub, from incoming payloads, to job execution
Azure IoT SDKs (open source)
Azure IoT SDKs

- **Device SDK** - enable you to build apps that run on your IoT devices. These apps send telemetry to your IoT hub, and optionally receive messages from your IoT hub.

- **Service SDK** - enable you to manage your IoT hub, and optionally send messages to your IoT devices.

- **Azure IoT Edge SDK** - enables you to build gateways for devices that don't use one of the supported protocols. Gateways can also process messages on the edge.
Azure IoT Device SDK
Azure IoT **Device SDK**

- Device Authentication
- Send and receive messages to the Azure IoT Hub.
- Communicate with the service via AMQP, MQTT or HTTP.
- Synchronize an Azure IoT Hub device Twin with Azure IoT Hub from a device
- Implement Azure IoT Hub Direct Device Methods on devices
- Implement Azure IoT Device Management features on devices
Azure IoT Device SDK – Authentication

- Symmetric Key
- X.509 Certificate
- TLS 1.2 Encryption
- Trusted Platform Module (TPM)
Azure IoT **Device SDK** – Device Twins

- Device Metadata
- Device tags
- Device configurations
- Device long-running operations (firmware updates)
Azure IoT Device SDK – cont.

More features
- Direct methods
- Upload to blob storage
- Send/Receive Messages
- Retry polices
Azure IoT Device SDK - Example

Add IoT functionality to Digital Door Viewer

**Previous model functionality:**
- When motion sensor active, show camera feed on digital screen
- Click to take picture and save to SD card

**New model**
- When motion sensor active, capture image and upload to cloud
- Send Alerts to Mobile Phone
- Analyze with FACE API
- Optional: unlock door if face recognized using BLE door-lock, remote-unlock door
Azure IoT Services SDK
Azure IoT Services SDKs

- Send and Receive messages from Azure IoT Hub.
- Communicate with the service via AMQP, MQTT or HTTP.
- Update Azure IoT Hub device Twins
- Set Jobs to run operations on multiple devices
- Implement Azure IoT Hub Direct Device Methods on devices
- Implement Azure IoT Device Management features on devices

```javascript
'use strict';

var Client = require('azure-iotHub').Client;

var connectionString = '<Hub Connection String>'; var targetDevice = '<Device Id>'; var methodParams = {
  methodName: '<Method Name>',
  payload: '[Method Payload]',
  responseTimeoutInSeconds: 15 // set response timeout as 15 seconds
};

var client = Client.fromConnectionString(connectionString);

client.invokeDeviceMethod(targetDevice, methodParams, function (err, result) {
  if (err) {
    console.error('Failed to invoke method `' + methodParams.methodName + '`');
  } else {
    console.log(methodParams.methodName + ' on ' + targetDevice + ':');
    console.log(JSON.stringify(result, null, 2));
  }
});
```
Azure IoT Services SDKs

'use strict';

var Client = require('azure-iotlua').Client;

var connectionString = '<Hub Connection String>.CV
var targetDevice = '<Device Id>'.
var methodParams = {
    methodName: '<Method Name>',
    payload: '[Method Payload]',
    responseTimeoutInSeconds: 15 // set response timeout as 15 seconds
};

var client = Client.fromConnectionString(connectionString);

client.invokeDeviceMethod(targetDevice, methodParams, function (err, result) {
    if (err) {
        console.error('Failed to invoke method \'' + methodParams.methodName + '\': ' + err.message);
    } else {
        console.log(methodParams.methodName + ' on ' + targetDevice + ':');
        console.log(JSON.stringify(result, null, 2));
    }
});
# Azure IoT Services SDK

<table>
<thead>
<tr>
<th>Features</th>
<th>Support</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity registry (CRUD)</td>
<td>✓✓</td>
<td>Use your backend app to perform CRUD operation for individual device or in bulk.</td>
</tr>
<tr>
<td>Cloud-to-device messaging</td>
<td>✓✓</td>
<td>Use your backend app to send cloud-to-device messages in AMQP and AMQP-WS, and set up cloud-to-device message receivers.</td>
</tr>
<tr>
<td>Direct Methods operations</td>
<td>✓✓</td>
<td>Use your backend app to invoke direct method on device.</td>
</tr>
<tr>
<td>Device Twins operations</td>
<td>✓✓*</td>
<td>Use your backend app to perform device twin operations. *Twin reported property update callback and replace twin are in progress.</td>
</tr>
<tr>
<td>Query</td>
<td>✓✓</td>
<td>Use your backend app to perform query for information.</td>
</tr>
<tr>
<td>Jobs</td>
<td>✓✓</td>
<td>Use your backend app to perform job operation.</td>
</tr>
<tr>
<td>File Upload</td>
<td>✓✓</td>
<td>Set up your backend app to send file upload notification receiver.</td>
</tr>
</tbody>
</table>
Azure IoT Services SDKs - Example

When inventory is added to ERP
- Provision new device in IoT Hub – set Device twin {
  - Status = Inventory
  - Location = Warehouse 1
}

When item is ordered, where Order to Cash process is: Inventory -> Test -> Ship
- Set Device twin { status = testing}

When device is shipped to Customer
- Set device twin { status = production; customer = John Brown, Counter=Canada}
Azure IoT Edge Gateway SDK
Azure IoT Edge **Gateway** SDK

- BYOH – Bring your own Hardware
- BYOC - Bring your own code
- Artificial Intelligence and Machine Learning on the edge
- IoT Edge cloud interface
- Multiple Protocols
Azure IoT Edge Gateway SDK

- Deploy Cloud Logic
- Modular approach
- Filter data
- Machine Learning on the Edge
- Azure Functions on the Edge
- Manageable at Scale
- Runs on Windows/Linux
Azure IoT Edge Gateway SDK - Modules
Your Industry 4.0 Customer wants to connect a new machine to the Gateway and to see the data in his Connected Factory Solution.

- After receiving the specifications, your engineers develop a new module to support it.
- Deploy the new module to the IoT Edge Gateway at Customer location
- Provision device to Customer Solution
- Assign new rules to device
- View data
Azure IoT Suite & Time Series Insights
Azure IoT Suite

- Remote Monitoring / Predictive Maintenance / Connected Factory
- Start in minutes
- Customize to your own needs
- Connect your devices and systems
- Discover new insights
- Enhance security
Azure IoT Suite

Remote Monitoring

- View and interact with Alerts
- Monitor devices
- Filter by device type
- Add new devices
Azure IoT Suite

Connected Factory

- Connect to OPC UA server
- Deploy gateway to connect
- Setup Alerts and KPIs
- Filter by device type
- Acknowledge alerts
Time Series Insights

- Storing time series data in a scalable way.
- Near real-time data exploration.
- Root-cause analysis and anomaly detection.
- It can ingress from 1 million to 100 million events per day, with a default retention span of 31 days.
Azure IoT Cloud to Cloud Demo
Azure IoT – Cloud to Cloud Capabilities

- Use REST API to interact with Azure IoT Hub
- Send real-time data to Azure Event Hub
- Use Stream Analytics to route the data everywhere
- Use PowerBI to Visualize and Analyze

Demo
Cloud To Cloud - DEMO

Monitoring Flow - Sensor to Cloud
Light Sensor -> U-Control Wireless Node -> U-Control Gateway -> Telit DeviceWise
Telit DeviceWise -> Azure Event Hub -> Stream Analytics -> Database & PowerBI

Control Flow Cloud to Light Relay
Telit DeviceWise <--> U-control Gateway <--> U-Control Control Unit -> Light
Thank You!

COME VISIT ME AT THE BOOTH

Tom Zamir
tom@iot-experts.net