Internet of Things Communication Framework

Rev: 0.1

Date: March 20th, 2014

By: Samuel Phung
Windows Embedded MVP
Twitter: @Samuelp101
Blog: http://Embedded101.com/Samuelp101

Contents

INTRODUCTION ---------- 3

IOTCF APPLICATION SCENARIO ---------- 4

ioTCF Framework General Design Guideline ----------5

ioTCF Application Scenario----------6

Introduction

The Internet of Things Communication Framework (ioTCF) project is setup to host the development of an open-source application framework to help simplify development tasks, as part of the process in developing application for Internet of Things (IoT) devices.

While there are other application frameworks that have been around for a while, most of these existing frameworks are designed to support general purpose development. The ioTCF framework is designed specifically to support IoT device application.

This document is put together, as a place holder, to collect and organize information relate to the design and development of the ioTCF framework.

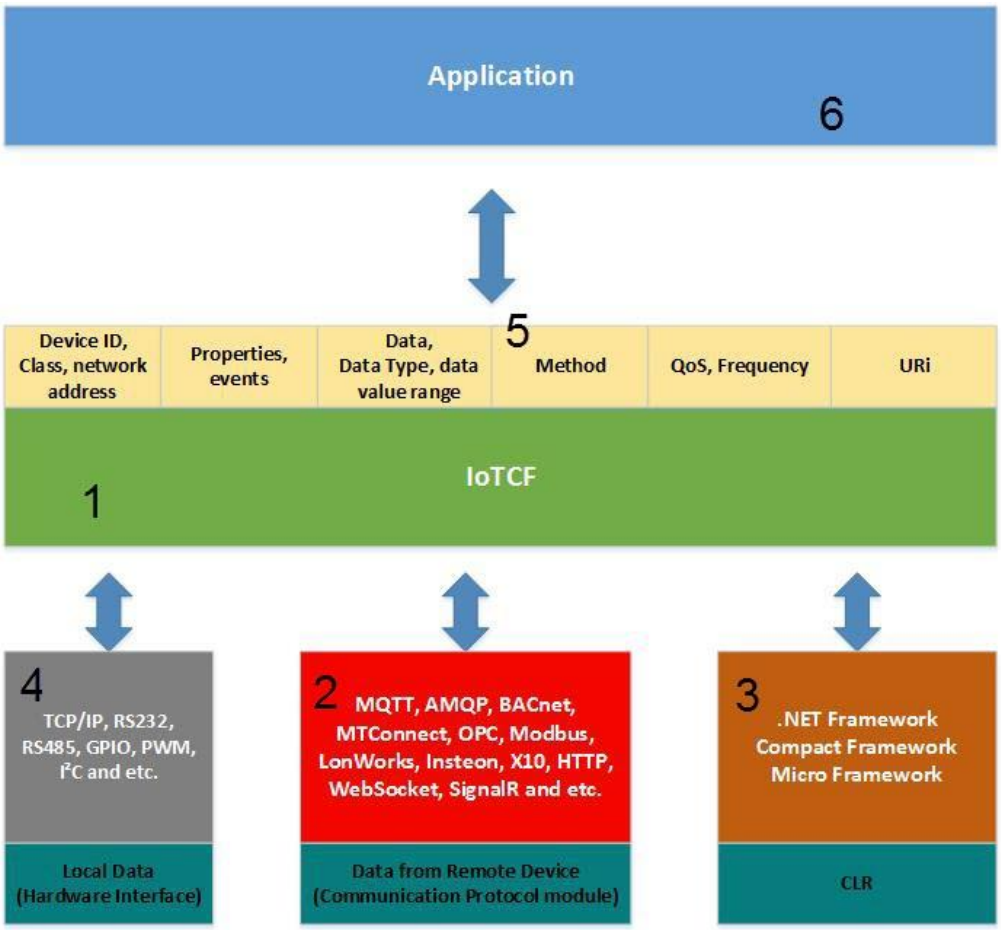
The ioTCF project is initiated by the Embedded101 community. Software components created by the ioTCF project will release as open-source.

An advisory board will be formed to oversee the functions and features for the ioTCF project.

ioTCF Application Scenario

The following diagram shows a high level view to the ioTCF framework.

Note: The information in this document is part of the early planning and is expected to change.



Let's review each of the function blocks in the above diagram and associate them in an application.

ioTCF Framework General Design Guideline

The main IoTCF module is derived from the underline .NET Framework, Compact Framework and Micro-Framework, to provide functionality unique to IoT and automation application.

If we look at a typical automation-control application, it generally has to deal with the following:

- Detect whether a device or appliance is activated or deactivated.
- Detect whether a switch, door or window is open or close.
- Turning on a device or appliance.
- Capture sensor data from a temperature, humidity, pressure or light sensor.
- Control a servo's movement and position.
- Control a motor's speed.
- Control lighting intensity.
- Etc.

Regardless of the industry and market segment, all automation-control application typically needs to work with the following:

- Digital input: Used to detect the On/Off or Open/Close status of a device.
- Digital output: Used to control the On/Off state of a device.
- Analog input: Used to detect sensor data.
- Analog output: Used to control servo, motor, lighting intensity and etc.

  Note: Some application uses analog output to an user interface, where the analog data stream is used to provide graphical presentation, such as waveform on a oscilloscope.

The ioTCF framework is designed to encapsulate these common I/O scenarios and provide a simple and structured application development environment, which greatly simplify the tasks to develop application.

Following are proposed application interfaces for the ioTCF framework:

- **Device identification**, which include device ID, class of device and location ID.
- **Properties** to configure and identify the device's capabilities.
- **Event** is triggered when new data is available or device status is changed.
- **Data** from the device
- **Data type & Range of Value** to identify the type of data
- **Method** or command that can be used to control the device
- **QoS** set the interaction frequency with the device.
- **Uri** is used to enable developer to interact with the device without the need to know the underline connectivity mechanism.

To promote interoperability and simplify future ioTCF development to support additional resources, a flexible interface to the data source is needed to support the following:

- Interface to device connected via an industrial standard communication protocol (Function Block #2, communication protocol), such as:
  - MQTT
  - AMQP
  - BACnet
  - MODBUS
  - MTConnect
  - LonWorks
  - WebSocket
  - SignalR
  - And more...
- Interface to directly attached I/O peripherals (Function Block #4, hardware interface), such as:
  - Serial port
  - USB port
  - GPIO
  - Ethernet
  - Built-in peripherals

ioTCF Application Scenario

When developing an application using the ioTCF framework, the application is expected to interact with multiple devices. In addition, Internet connectivity is expected to occasionally disconnect and reconnect. An IoT application must address the connectivity issue in order to maintain positive user experience.

ioTCF framework must be designed with necessary mechanism to support the multi devices application scenario and address occasionally disconnect/reconnect issues, such as:

- Incorporate array of devices
- Establish a local cache to keep the application running while disconnected from the Internet.

ioTCF Design Consideration

Based on preliminary observation, it's in this document's author's opinion the ioTCF framework should be designed to support event driven application, and incorporate the following design patterns for the framework:

- Publiser
- Subscriber
- Observer
- Interface

ioTCF Project Participation

If you are interested to participate and contribute to this project, please send us an email with "ioTCF Contributor" in the subject to the following alias:

- info@iotcf.com

Please include the following information in the email:

- Your contact information, including your full name, country/region you are from.
- Tell us about your interest and how you like to contribute to the project.
- If you have a website or blog, please let us know the URL.