Developing IoT solutions with Windows 10 and Raspberry Pi

Sebastian Brandes
Tech Evangelist
Session Objectives And Takeaways

1. Understand how IoT solutions work and how the Microsoft platform is central to IoT development.

2. Get an overview of the IoT landscape and what offerings are available to makers in the marketplace right. Also: Understand how the different IoT components all work with Microsoft's platform (largely, Azure).

3. Know the tools needed for IoT development and know where to start to build my own IoT solutions.
Agenda

Brief Introduction to IoT
1. IoT on Windows 10
2. Devices in the Maker Space
3. Services for IoT in Microsoft Azure

Demo: Windows Watering System
on Arduino UNO and Arduino YÚN
on Raspberry Pi 2

Conclusion
What is IoT?
“At its core, IoT is simple: it’s about connecting devices over the internet, letting them talk to us, applications, and each other.”

The Guardian
Brief Introduction to IoT

Windows 10 IoT Core

Universal Windows Platform (UWP)

Microsoft Azure
IoT – Why Now?

Devices are getting cheaper and more plentiful.

Developing applications for embedded systems does not require a PhD in electrical engineering any longer.

Cloud computing is becoming more accessible.

Microsoft supports all three trends.
IoT on Windows 10
“Windows 10 represents the culmination of our platform convergence journey with Windows now running on a single, unified Windows core.”

Kevin Gallo
Dir., Windows Dev Platform
A converged core means we can leverage our Windows dev skills for IoT solutions
IoT Platform Convergence

Windows Embedded Standard
Windows Embedded Handheld
Windows Embedded Compact
Windows 10

Converged OS kernel
Converged app model
Porting Tools

One core
Multiple SKUs

Windows on Devices
Windows 10 and the IoT SKUs

- Windows 10 comes with three IoT SKUs
  - Windows 10 IoT Core
  - Windows 10 IoT for mobile devices
  - Windows 10 IoT for industry devices
- Windows 10 IoT Core is available for preview on windowsondevices.com – when released, it’ll be free
- IoT Core currently supports Raspberry Pi 2 and MinnowBoard Max
- Available for preview to the general public
Windows 10 IoT Editions

Windows 10 IoT for industry devices
Desktop Shell, Win32 apps, Universal apps and drivers
Minimum: 1 GB RAM, 16 GB storage
X86/x64

Windows 10 IoT for mobile devices
Modern Shell, Mobile apps, Universal apps and drivers
Minimum: 512 MB RAM, 4 GB storage
ARM

Windows 10 IoT Core
Universal Apps and Drivers
No shell or MS apps
Minimum: 256MB RAM, 2GB storage
X86/x64 or ARM

+ 

Windows Updates
Visual Studio & UWP
New User Interfaces
Security & Identity
AllJoyn
Integrated Device Connectivity
Microsoft Azure IoT
Windows for Industry Devices
Windows for Industry Devices

Windows 10 IoT for industry devices
Desktop Shell, Win32 apps, Universal Windows Apps and Drivers
1 GB RAM, 16 GB Storage
X86

Windows 10 IoT for mobile devices
Modern Shell, Universal Windows Apps and Drivers
512 MB RAM, 4 GB storage
ARM

Windows 10 IoT Core
No Shell, Universal Windows Apps and Drivers
256MB RAM, 2GB storage
X86 or ARM

Device Capabilities
Installing Windows 10 IoT Core

Get Started
Learn how to set up the Raspberry Pi 2 and connect it to your computer.

1. Select your Device
2. Set up your PC
3. Set up your Device
4. Develop

What you need
1. A PC running Windows 10 Insider Preview (Prepare in the previous step)
2. Raspberry Pi 2
3. 5V micro USB power supply - with at least 1.1A current
   (If you plan on using several power-hungry USB peripherals, use a higher current power supply instead (~2.5A))
4. 8GB micro SD card - class 10 or better. (We suggest this one or this one)
5. HDMI cable and monitor
6. Ethernet cable
7. Micro SD card reader

Put the Windows 10 IoT Core Insider Preview image on your SD card
1. Insert a micro SD card into your SD card reader.
2. Use IoTCoreImageHelper.exe to flash the SD card. Search for “WindowsIoT” from start menu and select the shortcut “WindowsIoTImageHelper”
Tool 1: Using Windows 10 and UWP to develop apps for IoT devices
Devices in the Maker Space
How many in here are makers?
...and many more!
Arduino UNO

Processor: ATmega328P
Speed: 16 MHz (8-bit AVR)
Storage: 32 KB
Memory: 2 KB
OS: Wiring
Voltage: 5V DC
GPIO: 20 (14 digital, 6 with PWM, 6 analog input pins)
Release Date: September 24, 2010 (approx. $25)
Arduino YÚN

Processor: Atmega32U4 and Atheros AR9331
Speed: 16 MHz / 400 MHz
Storage: 32 KB / 64 MB
Memory: 2 KB / 16 MB
OS: Wiring / Linino
Voltage: 5V DC
GPIO: 20 (14 digital, 6 with PWM, 12 analog input pins)
Release Date: September 10, 2013 (approx. $75)
MinnowBoard MAX

Processor: 64-bit Intel® Atom™ E38xx Series SoC

Speed: 1.33-1.75 GHz (Dual Core)

Storage: SD Card

Memory: 1 GB ($99) or 2 GB ($139)

OS: Windows 10 and others

Voltage: 5V DC

GPIO: 8 (2 with PWM) + SATA2, 2*USB, SPI, I2C, more

Release Date: June 2014 ($99-$139)
Intel Galileo

Processor: Intel Quark X1000
Speed: 400 MHz
Storage: SD Card
Memory: 256 MB
OS: Windows 8.1 and others
Voltage: 5V DC
GPIO: 20 (14 digital, 6 with PWM, 6 analog input pins)
Release Date: October 17, 2013 (approx. $70)
Raspberry Pi 2
Processor: Broadcom BCM2836
Speed: 900 MHz quad-core ARM Cortex-A7
Storage: SD Card
Memory: 1 GB
OS: Windows 10 and others
Voltage: 5V DC
GPIO: 17 (only digital)
Release Date: February 2015 ($35)
So far the Raspberry Pi 2 appears to be the most interesting of the bunch.
Today I brought...

- Arduino UNO
- Arduino YÚN
- Intel Galileo (running Windows 8.1)
- Raspberry Pi 1 (running Raspbian)
- Raspberry Pi 2 (running Windows 10)

- MinnowBoard MAX is in the WoD program but I do not own one (donations are welcome)
Tool 2: Using maker boards supported by Windows 10 to run UWP apps
IoT Services in Azure
## IoT Services in Microsoft Azure, July 2015

<table>
<thead>
<tr>
<th>Device Connectivity</th>
<th>Storage</th>
<th>Analytics</th>
<th>Presentation &amp; Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Hubs</td>
<td>SQL Database</td>
<td>Machine Learning</td>
<td>App Service</td>
</tr>
<tr>
<td>Service Bus</td>
<td>Table/Blob Storage</td>
<td>Stream Analytics</td>
<td>Power BI</td>
</tr>
<tr>
<td>External Data Sources</td>
<td>DocumentDB</td>
<td>HDInsight</td>
<td>Notification Hubs</td>
</tr>
<tr>
<td></td>
<td>External Data Sources</td>
<td>Data Factory</td>
<td>Mobile Apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BizTalk Services</td>
</tr>
</tbody>
</table>
Watering System Architecture
Services for Device Connectivity

Azure Service Bus Topics
- Sending messages from a mobile client
- Consuming messages via a subscription on the IoT device
- Using the AMQP 1.0 protocol

Azure Event Hubs
- Great for telemetry
- Possible to ingress millions of events per second (>1 GB/s)
- Examples of events include sensor readings (e.g., a humidity sensor)
Azure Service Bus Topics
Azure Service Bus Topics

wateringcommands

**General**
- DEFAULT MESSAGE TIME TO LIVE: 1 minutes
- DUPLICATE DETECTION HISTORY: 10 minutes
- PUBLISHING: Filter Message Before Publishing
- TOPIC STATE: Enabled

**Shared Access Policies**

<table>
<thead>
<tr>
<th>NAME</th>
<th>PERMISSIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagePolicy</td>
<td>Manage, Send, Listen</td>
</tr>
<tr>
<td>SendPolicy</td>
<td>Send</td>
</tr>
<tr>
<td>ListenPolicy</td>
<td>Listen</td>
</tr>
</tbody>
</table>

**Shared Access Key Generator**
- POLICY NAME: ManagePolicy
AMQP 1.0 and HTTPS

- Advanced Message Queuing Protocol 1.0
  - AMQP is an open standard application layer protocol for message-oriented middleware.
  - More advanced than MQTT.
  - In Azure, it can be used for sending and receiving messages.

- HTTPS is supported too
  - HTTPS is supported for sending messages.
  - Some devices might not have client libraries for AMQP and, thus, HTTPS can be used instead to ingress events.

- Client libraries exist for many languages
  - I am using Azure SB Lite, which is an open source project on CodePlex.
public async Task SendToTopicAsync(string topic, string message)
{
    var builder = new ServiceBusConnectionStringBuilder(TOPIC_CONNECTION_STRING);
    builder.TransportType = TransportType.Amqp;

    var factory = MessagingFactory.CreateFromConnectionString(TOPIC_CONNECTION_STRING);

    TopicClient client = factory.CreateTopicClient(topic);

    MemoryStream stream = new MemoryStream(Encoding.UTF8.GetBytes(message));
    BrokeredMessage brokeredMessage = new BrokeredMessage(stream);
    brokeredMessage.Properties["time"] = DateTime.UtcNow;

    await Task.Run(() => client.Send(brokeredMessage));

    client.Close();
    factory.Close();
}
public async void ReceiveFromTopicSubscriptionAsync(string topic, string subscription) {
    var builder = new ServiceBusConnectionStringBuilder(TOPIC_CONNECTION_STRING);
    builder.TransportType = TransportType.Amqp;
    var factory = MessagingFactory.CreateFromConnectionString(TOPIC_CONNECTION_STRING);
    SubscriptionClient client = factory.CreateSubscriptionClient(topic, subscription);
    while (true) {
        try {
            BrokeredMessage request = await Task.Run(() => client.Receive());
            request.Complete();
            BrokeredMessageReceived(this, new BrokeredMessageReceivedEventArgs(request));
        }
        catch (Exception ex) {
            // TODO: Handle bad message from WateringCommands topic
        }
    }
}
Azure Event Hubs
Azure Event Hubs

- Based on a Partitioned Consumer model
  - In contrast to the Competing Consumer model employed by Queues and Topics.
  - The Competing Consumer model results in complexity and scale limits for stream processing applications.
  - Partitioned Consumer model enables for horizontal scaling

- Up to 1,024 partitions in a single hub
  - Default number of partitions is 16.
  - This number cannot be changed after the creation of the hub.
  - Publications to a partition (a single event or a batch) can be max. 256 KB.

- Partitions can be read from an offset
  - Allowing for correct replay of events in a partition.
public async Task SendToPartitionAsync(string message, string partitionId)
{
    var builder = new ServiceBusConnectionStringBuilder(EVENT_HUB_CONN_STRING);
    builder.TransportType = TransportType.Amqp;

    var factory = MessagingFactory.CreateFromConnectionString(EVENT_HUB_CONN_STRING);
    EventHubClient client = factory.CreateEventHubClient(EVENT_HUB_NAME);
    EventHubSender sender = client.CreatePartitionedSender(partitionId);

    EventData data = new EventData(Encoding.UTF8.GetBytes(message));
    data.Properties["time"] = DateTime.UtcNow;

    await Task.Run(() => sender.Send(data));

    sender.Close();
    client.Close();
    factory.Close();
}
public async void ReceiveFromPartitionAsync(string partitionId, string eventHubEntity) {
    var builder = new ServiceBusConnectionStringBuilder(EVENT_HUB_CONN_STRING);
    builder.TransportType = TransportType.Amqp;
    var factory = MessagingFactory.CreateFromConnectionString(EVENT_HUB_CONN_STRING);
    EventHubClient client = factory.CreateEventHubClient(eventHubEntity);
    EventHubConsumerGroup group = client.GetDefaultConsumerGroup();
    EventHubReceiver receiver = group.CreateReceiver(partitionId);

    while (true) {
        EventData data = await Task.Run(() => receiver.Receive());

        if (data == null)
            continue;

        EventHubMessageReceived(this, new EventHubMessageReceivedEventArgs(data));
    }
}
Azure Stream Analytics

- Perform real-time analytics for your Internet of Things solutions
- Stream millions of events per second
- Mission critical reliability, performance and predictable results
- Create real time dashboards and alerts over data from devices and applications
- Correlate across multiple streams of data
- Rapid development with familiar SQL-based language
Tool 3: Using Microsoft Azure to communicate with devices over the Internet
Demo: Windows Watering System

My career as a gardener
Abandoned garden projects = unhappy girlfriend 😞
Shopping List

- 1 * Pantry Pump (12 V, 2 A)
- 1 * Battery (12 V, 4.5 Ah)
- 1 * Transparent Rubber Tube (10 meters)
- 1 * Bucket (5 liters)
- 1 * Battery Charger (12 V, 4 A)
- Some cables
- A new soldering iron
  - There was a great one on sale – yay! 😊
Grill café
Vi har tændt op for grillen og det gode vejr
Grillet polse m. brød: 15 kr.
Frankfurter: 10 kr.
Franck hønse: 25 kr.
Alpha version using Arduino (video)
Electrical Circuit Design

- GPIO1 is controlled by one of the maker boards
- The pump was a little too strong so I reduced it with 2 * 1.6Ω resistors (~50% reduced power)
- Ohm’s Law: 2A * (2*1.6Ω) = 6.4V
Let’s Code
Arduino
Intel Galileo
Raspberry Pi 2
In Review: Session Objectives And Takeaways

1. Understand how IoT solutions work and how the Microsoft platform is central to IoT development.

2. Get an overview of the IoT landscape and what offerings are available to makers in the marketplace right. Also: Understand how the different IoT components all work with Microsoft's platform (largely, Azure).

3. Know the tools needed for IoT development and know where to start to build my own IoT solutions.
Q&A

If you have questions please proceed to the Q&A MICROPHONE located in your session room.