IoT Connect the Dots - Quick Start (Outline)

# Prerequisites:

1. Microsoft Azure subscription ([free trial subscription](http://azure.microsoft.com/en-us/pricing/free-trial/) is sufficient)
2. Access to the [Azure Streaming Analytics Preview](https://account.windowsazure.com/PreviewFeatures)
3. Visual Studio 2013 – Community Edition

## Hardware:

1. Raspberry PI B/B+ with Internet access
2. Arduino Uno R3
3. [SparkFun Weather Shield](https://www.sparkfun.com/products/12081) for Arduino (make sure you also have the required headers etc. as specified on the SparkFun site)

Note: Only the models have been tested. The Weathershield for example is known to not work reliably on Arduino above Due or Arduino Uno R2.

# Intro

Diagram, conceptual overview etc.

# Step-by-step

1. Download and Build “Connect the Dots”:
   1. Download or clone the “Connect the Dots” depot from <https://github.com/MSOpenTech/connectthedots-pr>
   2. Open ConnectTheDots.sln in Visual Studio and build the solution
   3. If the AMQP.Net Lite package cannot be found, you may need to build it yourself:
      1. Download or clone from <http://amqpnetlite.codeplex.com>
      2. Open amqp.sln in Visual Studio and build the solution
         1. To avoid errors for platforms for which you don’t have tooling, comment out or remove all <file> elements except for “bin\Release\Amqp.Net\\*.\*” in Amqp.Net.nuspec
      3. Add a private NuGet location:
         1. Open Tools, Options, NuGet Package Manager, Package Sources and add “<yourpath>\amqpnetlite\Build\Packages” as a new package source.
2. Create Azure resources for Event Hub:
   1. Download publishsetting file
      1. Go to <https://manage.windowsazure.com/publishsettings/index?client=powershell>) and save to local disk <publishsettingsfile> (contains keys to manage all resources in your subscriptions, so handle with care).
      2. If you have access to multiple subscriptions, make sure the file only contains the subscription that you want to use, otherwise, remove the other XML elements for the other subscriptions.
   2. Run ConnectTheDotsCloudDeploy from a command prompt, passing a name to be used for all cloud resources, and the publishsettings file
      1. Chose a <name> that has only letters and number
      2. ConnectTheDotsCloudDeploy\bin\debug\ ConnectTheDotsCloudDeploy.exe –n <name> -ps <publishsettingsfile>
   3. Note the device connection strings displayed by the tool, as you will need them to provision the devices later, i.e.

Device AMQP address strings (for Raspberry PI/devices):

amqps://D1:xxxxxxxx@yyyyyyyy.servicebus.windows.net

amqps://D2:xxxxxxxx@yyyyyyyy.servicebus.windows.net

amqps://D3:xxxxxxxx@yyyyyyyy.servicebus.windows.net

amqps://D4:xxxxxxxx@yyyyyyyy.servicebus.windows.net

1. Create Azure Stream Analytics jobs:
   1. Make sure you have access to the preview: if not sign up at <https://account.windowsazure.com/PreviewFeatures>
   2. Create a new job “**Aggregates**”:
      1. “+” in lower left corner -> Data Services -> Stream Analytics -> Quick Create -> Job name “Aggregates”.
      2. Add Input
         1. Inputs tab -> Add an Input -> Data Stream, Event Hub
         2. Input Alias: “DevicesInput”
         3. Choose the namespace <name>-ns,
         4. Event Hub “ehdevices”
         5. Policy Name: “StreamingAnalytics”
         6. Serialization: JSON, UTF8
      3. Query:
         1. Query tab
         2. Copy/paste contents “Queries\Aggregates.sql”
         3. Save
      4. Create output
         1. Output tab -> Add an Output, Event Hub,
         2. Choose the namespace <name>-ns,
         3. Event Hub “ehalerts”
         4. Policy name “StreamingAnalytics”
         5. Serialization “JSON”, UTF8
      5. Start the Job
         1. Dashboard, Start
   3. Create a new job “**Alerts**”: as above, but use “alerts.sql” contents for the query (in step iii/2.)
2. Publish the Azure Websites
   1. In VS: Right-click on WebClient, Publish.
   2. Select Azure Web Sites, create new one.
   3. Publish
   4. Enable WebSockets for the new Azure Web site
      1. Browse to <https://manage.azure.com> and select your Azure Web Site.
      2. Click on the Configure tag. Then set WebSockets to On
   5. Open the site in a browser to verify it has deployed correctly.
3. Provision the Raspberry PI:
   1. Install mono and required SSL root certs:
      1. Run the following from a shell (i.e. via SSH):
         1. sudo apt-get update
         2. sudo apt-get upgrade
         3. sudo apt-get install mono-complete
         4. mozroots --import --ask-remove
   2. Update Devices\RaspberryPiGateway\scripts\autorun.sh with one of the amqp address strings returned by ConnectTheDotsCloudDeploy.exe, i.e. amqps://D1:xxxxxxxx@yyyyyyyy.servicebus.windows.net
   3. Copy required files from Devices\RaspberryPiGateway\bin\Debug to Raspberry /usr/pi/RaspberryPiGateway/
      1. Files: RaspberryPiGateway.exe, Amqp.Net.dll, Newtonsoft.Json.dll, scripts/autorun.sh
      2. Optionally use Devices\RaspberryPiGateway\scripts\scprpi.cmd to copy via SSH
   4. Modify /etc/rc.local by adding one line:
      1. /home/pi/RaspberryPiGateway/autorun.sh &
   5. Make /usr/pi/CloudPI/autorun.sh executable:
      1. chmod 755 /usr/pi/RaspberryPiGateway/autorun.sh
4. Prepare Arduino Uno R3
   1. Flash with Arduino\Weathershield.ino in this project (modified from the Funspark sample to send data in JSON format) (from the Arduino IDE)
   2. Download and install Weathershield libraries
5. Run Raspberry PI + Arduino
   1. Plug Arduino’s USB cable into one of the Raspberry PI USB ports.
   2. (Re-)start Raspberry PI

You should now see average temperature measurements showing up in the web browser every minute.

If you select “All”, you should see raw readings from the device coming in every second.

If the temperature exceeds 75 degrees (F), you should see an Alert showing in the Alerts table, once per minute while the temperature on any of the devices exceeds 75 degrees (F).

To add more devices: …

RPI tips & tricks / detailed steps

Arduino tips & tricks / detailed steps