## References

* IoT Hub device SDK for C intro.

<https://azure.microsoft.com/zh-tw/documentation/articles/iot-hub-device-sdk-c-intro/>

* How to process IoT Hub device-to-cloud messages using .Net

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-csharp-csharp-process-d2c>

## Requirements

* Finished the part 2 of HOL
* PuTTY
* Device Explorer tool
* For the simulated .Net Windows Wind Turbine
  + Microsoft Visual Studio 2015/2017 with Azure SDK
  + Microsoft.Azure.Devices.Client (Azure IoT Device SDK) Package from NuGet Management

## Goals

* Build the simulated wind turbines for Linux and Windows devices
  + Fixed the device name
  + Generate the message ID, wind speed, depreciation and power capacity with the random value by timestamp.
  + Send the telemetry data to cloud every 5 seconds.

## Step 1: Modify simulated wind power device code of Linux (LinuxTurbine)

* Download and modify the provided sample code(**simplesample\_amqp\_windpower.c**) from this workshop
  + Open **simplesample\_amqp\_windpower.c**
  + Update the **connection string** of device (LinuxTurbine) in the sample code



Replace the **simplesample\_amqp\_windpower.c** to sample code

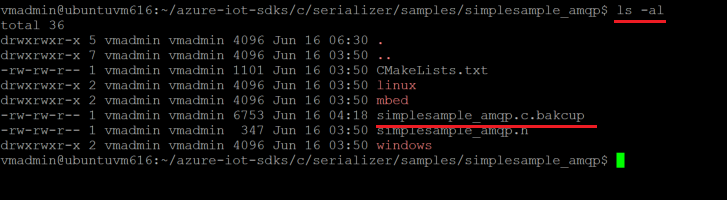
* + SSH connect to VM
  + Use mv command to backup the **simplesample\_amqp.c** file

cd ~/azure-iot-sdk-c/serializer/samples/simplesample\_amqp

mv simplesample\_amqp.c simplesample\_amqp.c.bakcup

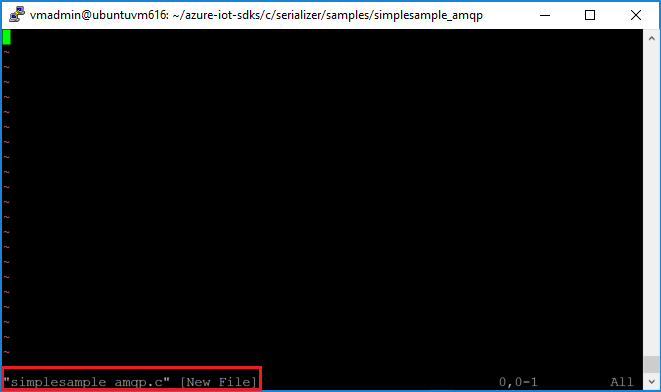
ls -al

* + Check the output

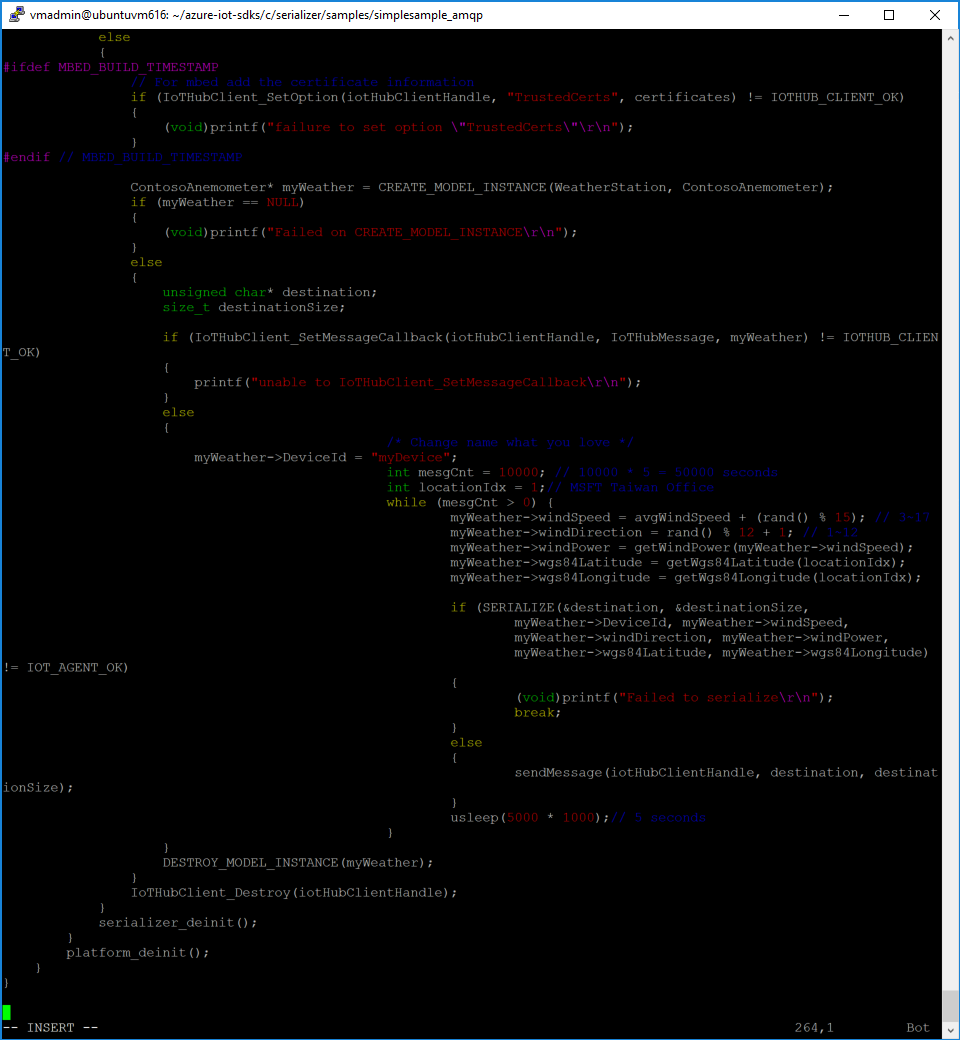


* + Create a new file and named it as **simplesample\_amqp.c**

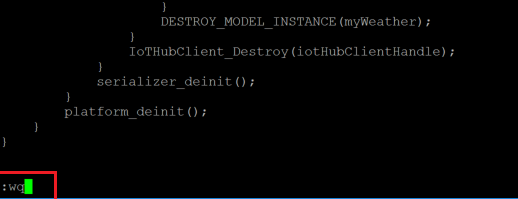
vi simplesample\_amqp.c



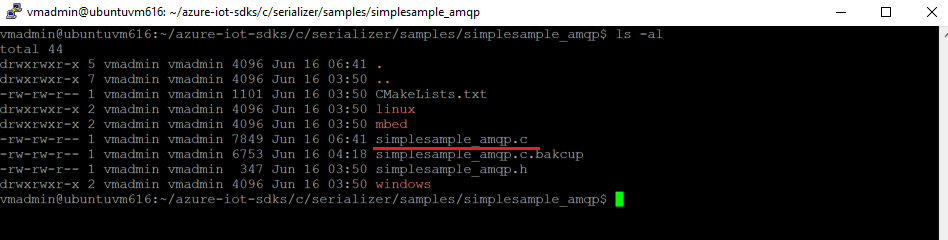
* + Type “a” to switch the insert mode, and paste all of sample code.



* + Press Esc key, then type :wq. The editor will save the file and quit.



* + Confirm whether the file is existing.

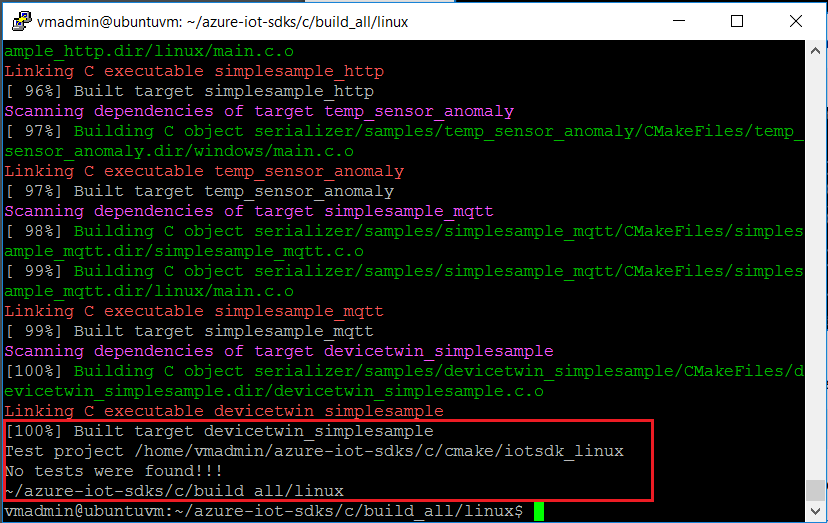


* Build the code again

cd ~/azure-iot-sdk-c/build\_all/linux

sudo ./build.sh

* + Confirm the output of build

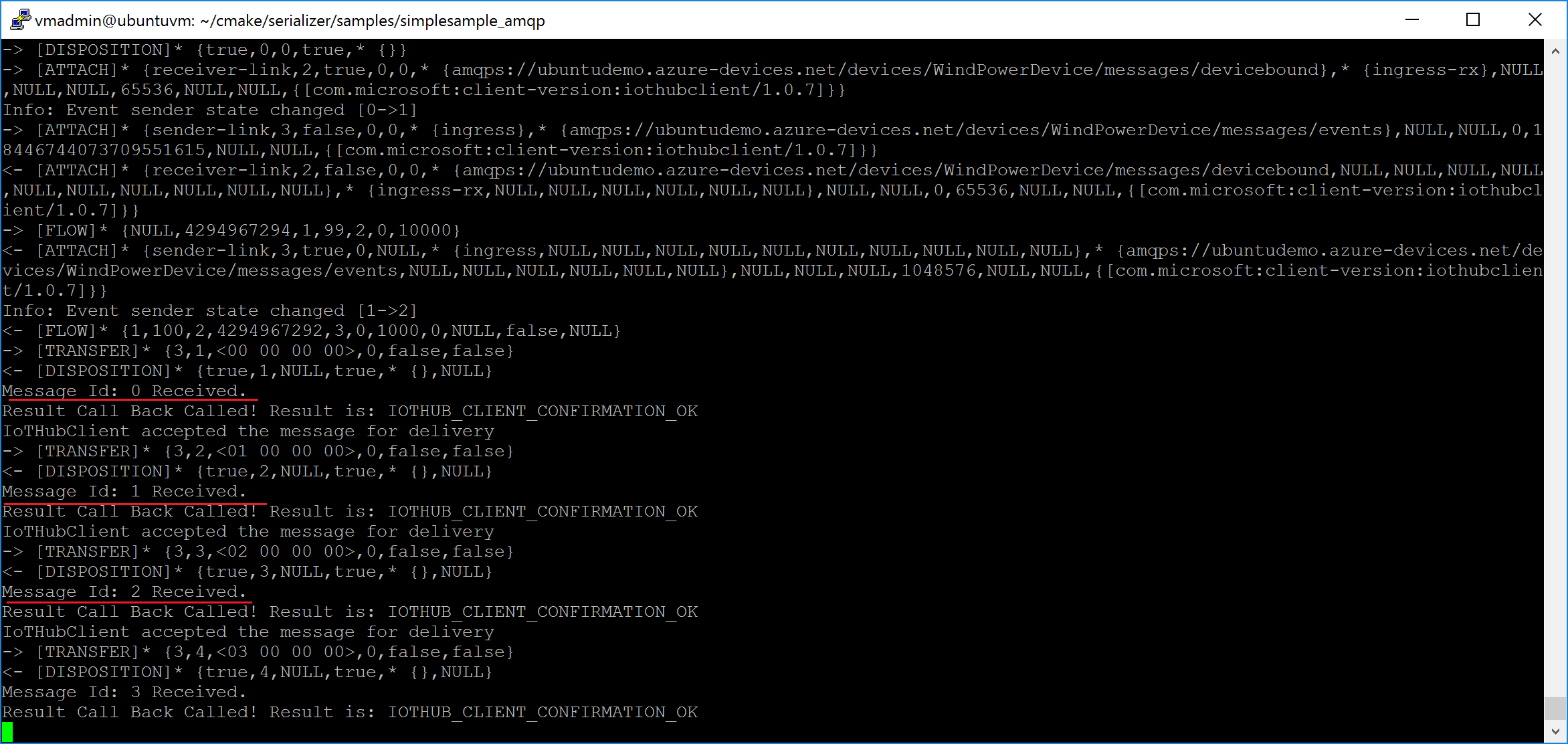


## Step 2: Run the simulated wind power device (LinuxTurbine)

* Run AMQP Simple Sample in C SDK

cd ~/azure-iot-sdk-c/cmake/iotsdk\_linux/serializer/samples/simplesample\_amqp/

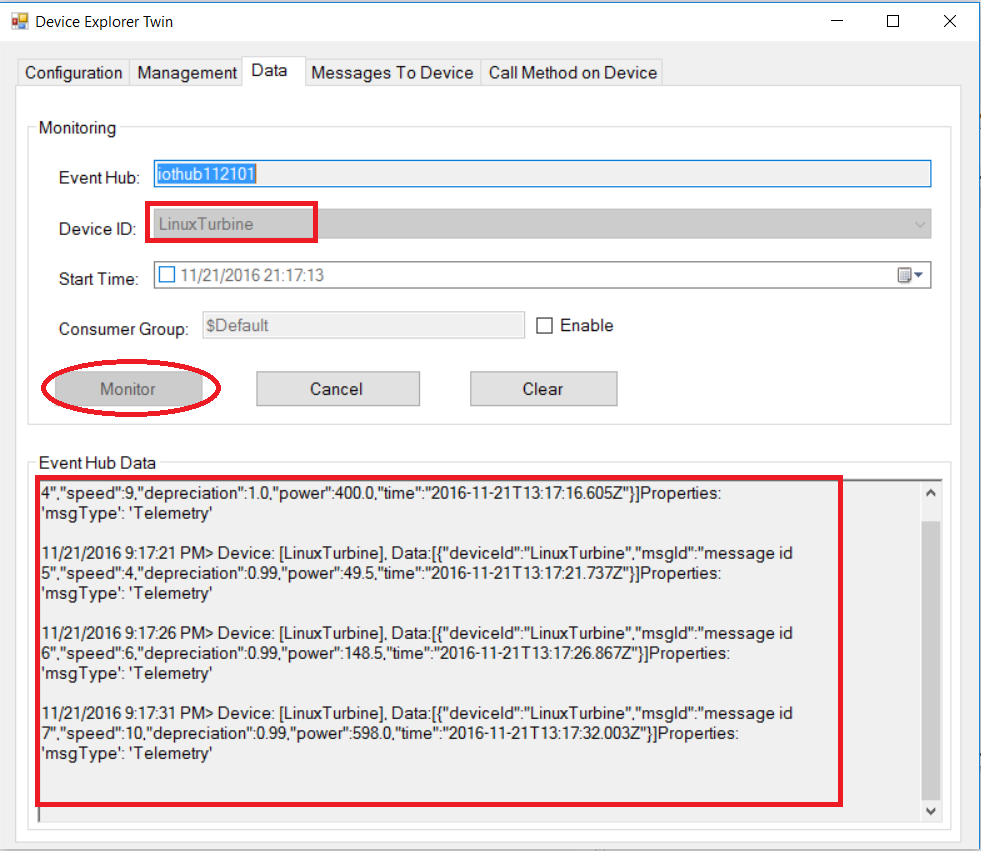
./simplesample\_amqp



**Note: Sample code will send data to IoT Hub every 5 seconds.**

## Step 3: Monitor the simulated wind power device (LinuxTurbine)

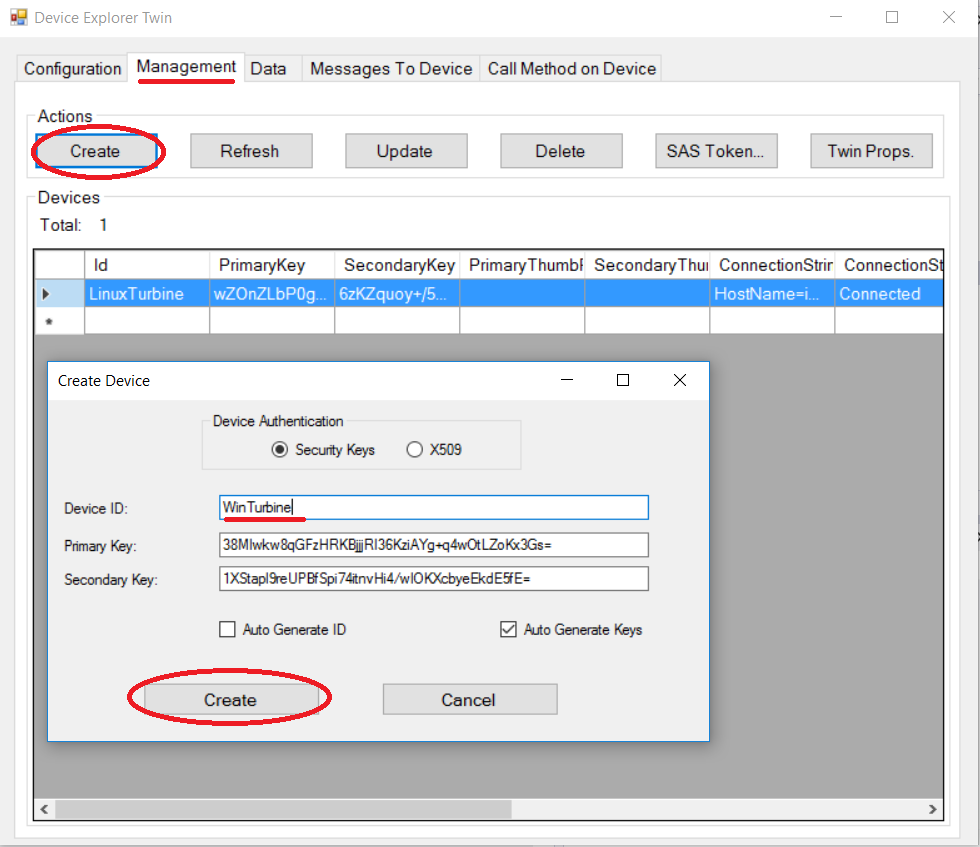
* Open the Device Explorer to monitor data on IoT Hub, it will receive the data every 5 seconds.

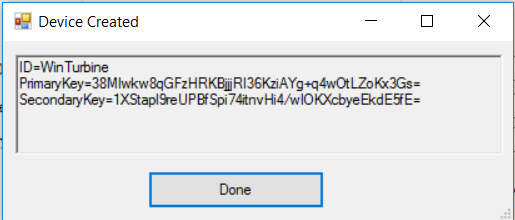


**Note: The data fields include the deviceId, msgId, speed, depreciation, power and time. Also we set a property for msgType as Telemetry for the later data processing.**

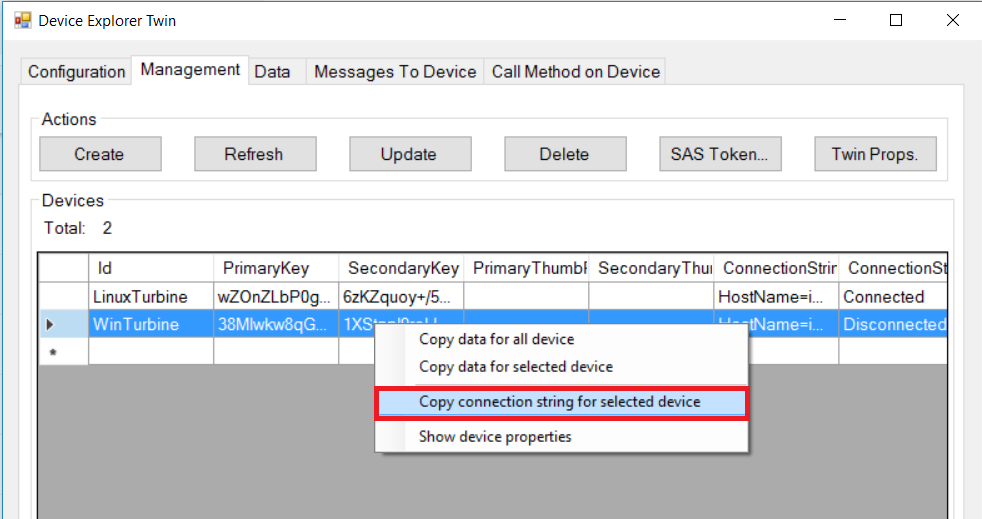
## Step 4: Create a simulated Windows device (WinTurbine) in IoT Hub

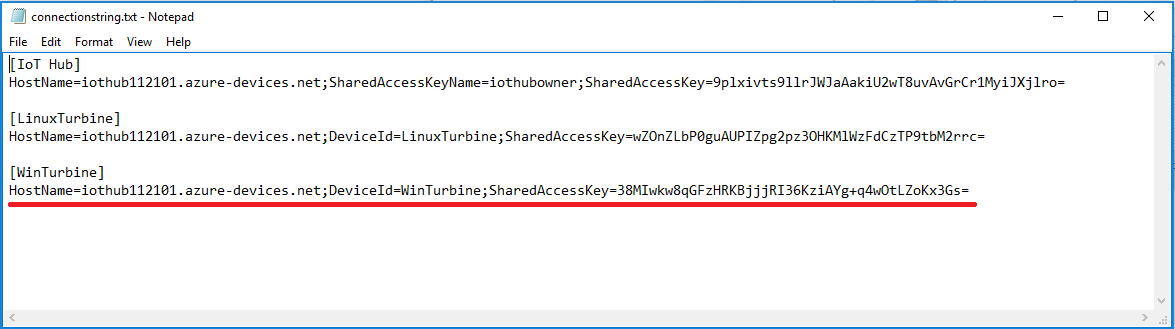
* Create a device for **WinTurbine** from the Device Explorer (please reference the step 4 of HOL-2, also the device name must be fixed as **WinTurbine**)





* Copy the connection string of device and keep it for later used.

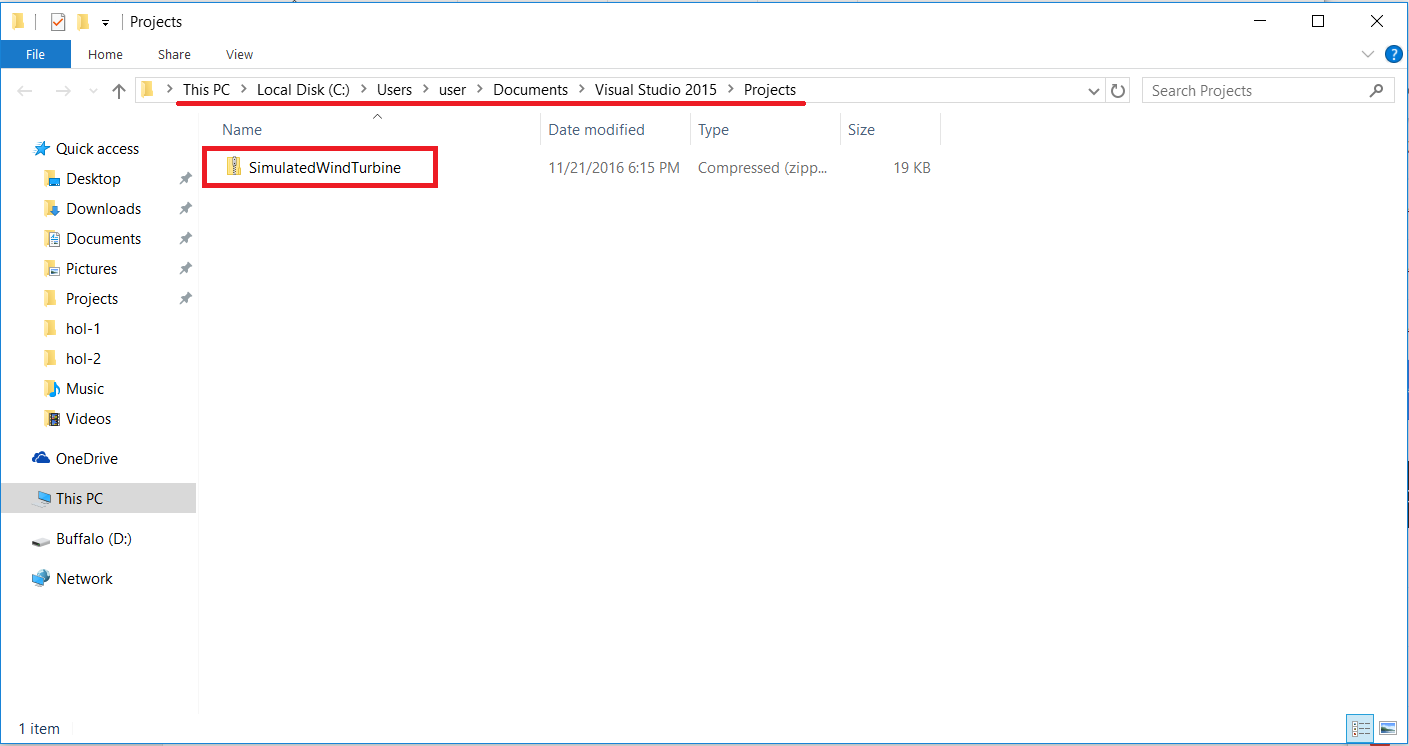


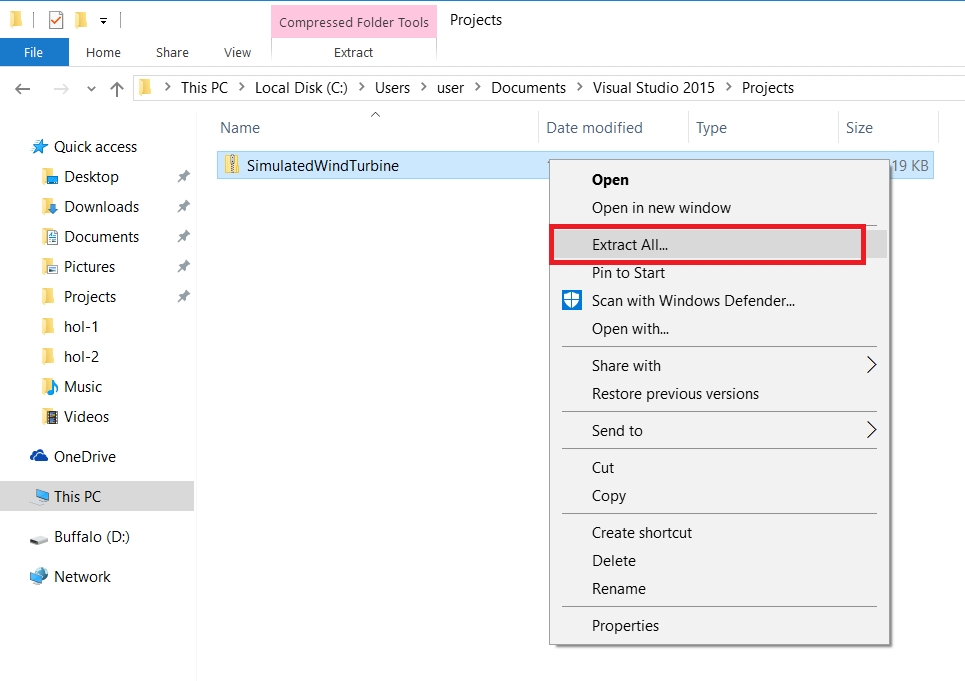


## Step 5: Build the C# simulated wind turbine (WinTurbine)

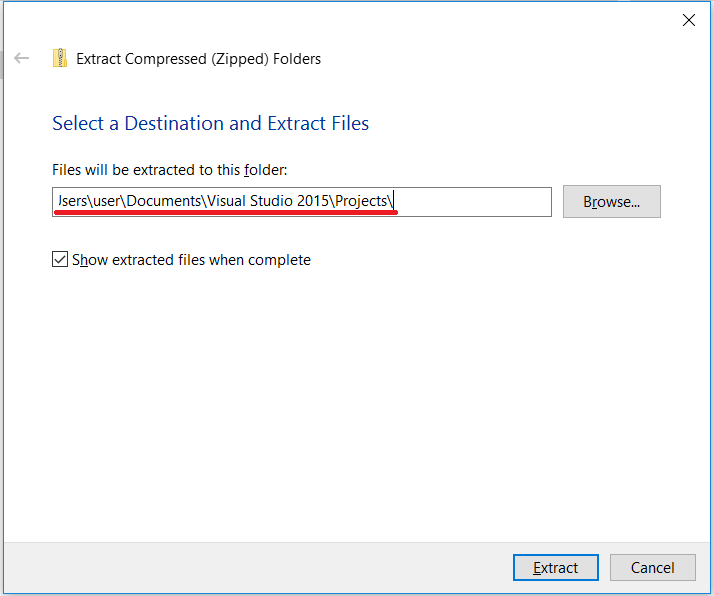
* Copy and extract the simulated wind turbine of C# project (**SimulatedWindTurbine.zip**) to the Projects of Visual Studio (VS).

Path: **C:\Users\user\Documents\Visual Studio 2015\Projects\**

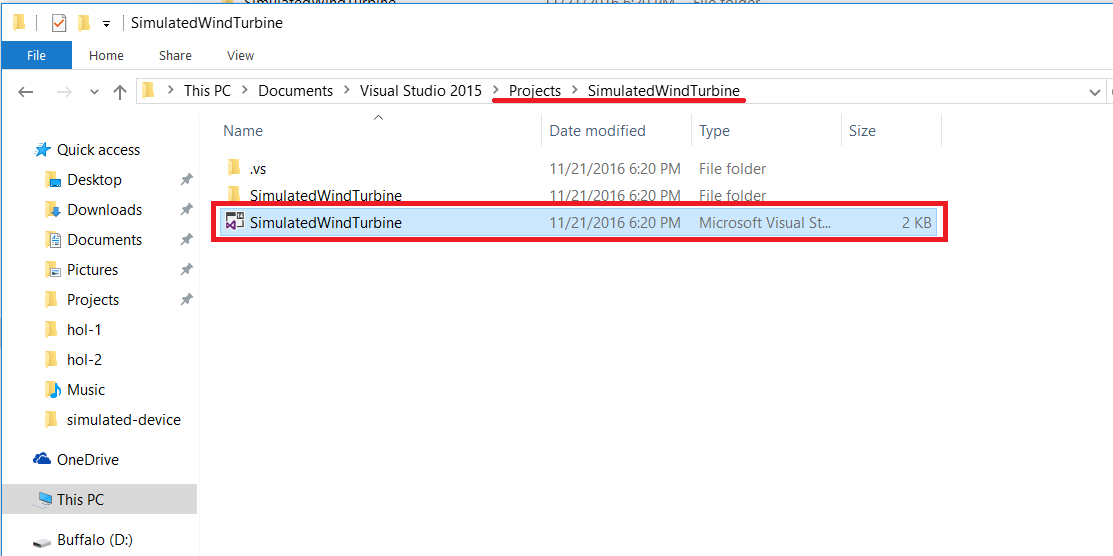




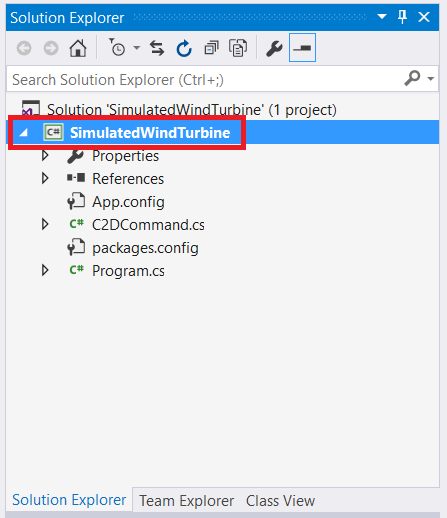
* The files should be extracted to **C:\Users\user\Documents\Visual Studio 2015\Projects\**

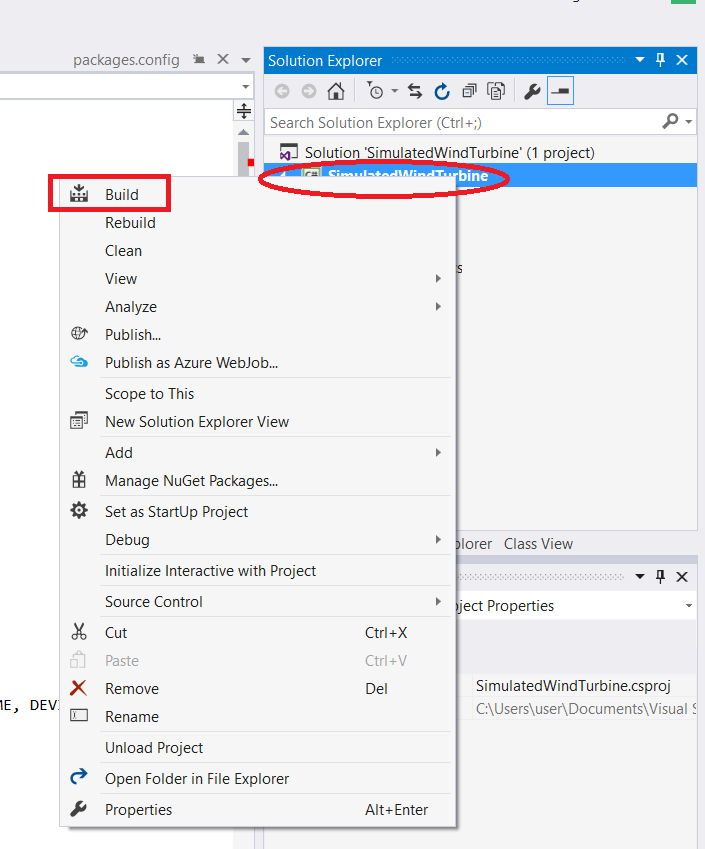


* Open the **SimulatedWindTurbine** solution (.sln file) of VS

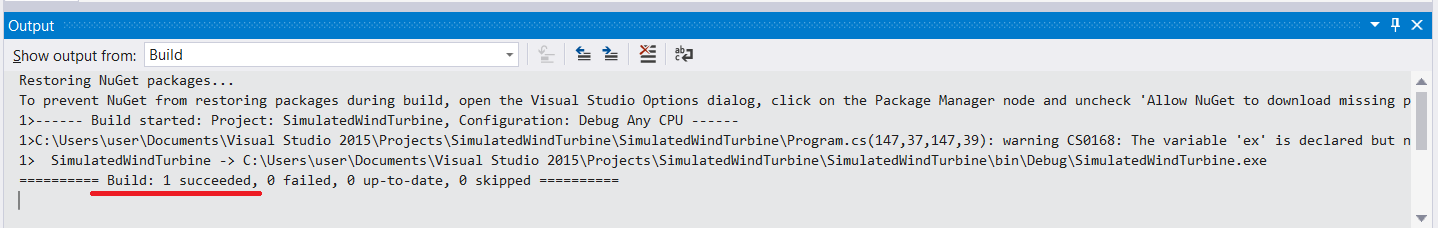


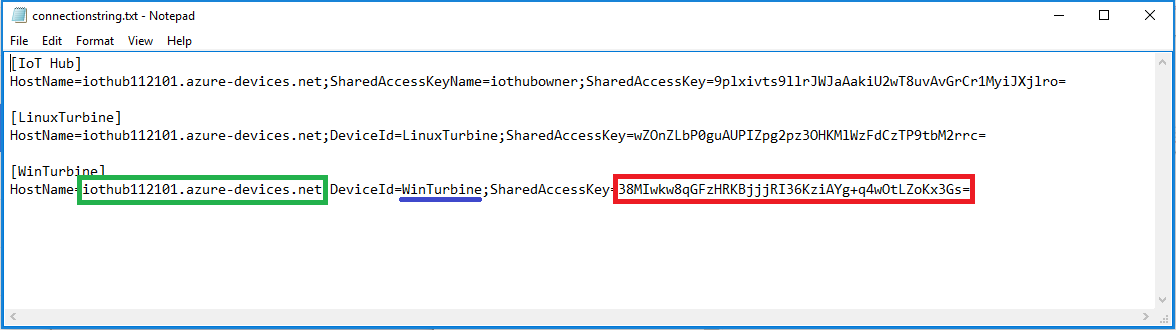
* Build the dependences of package
  + Right-click the selected project to build





* Check the output of build



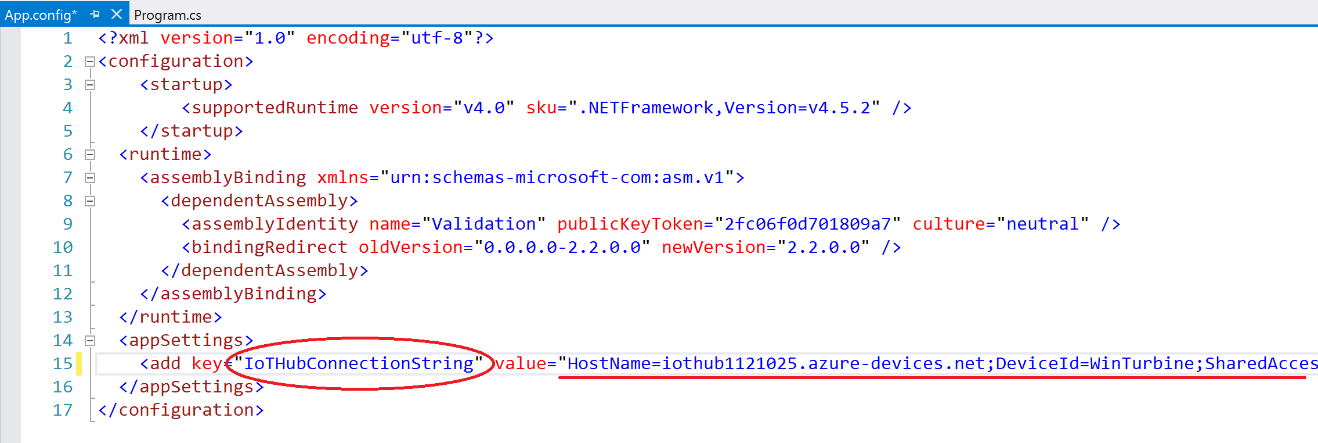
* Get the **ConnectionString** of WinTurbine device from your noted as previous.
* Update the code in App.config (Paste the device connection string)

Replace it:

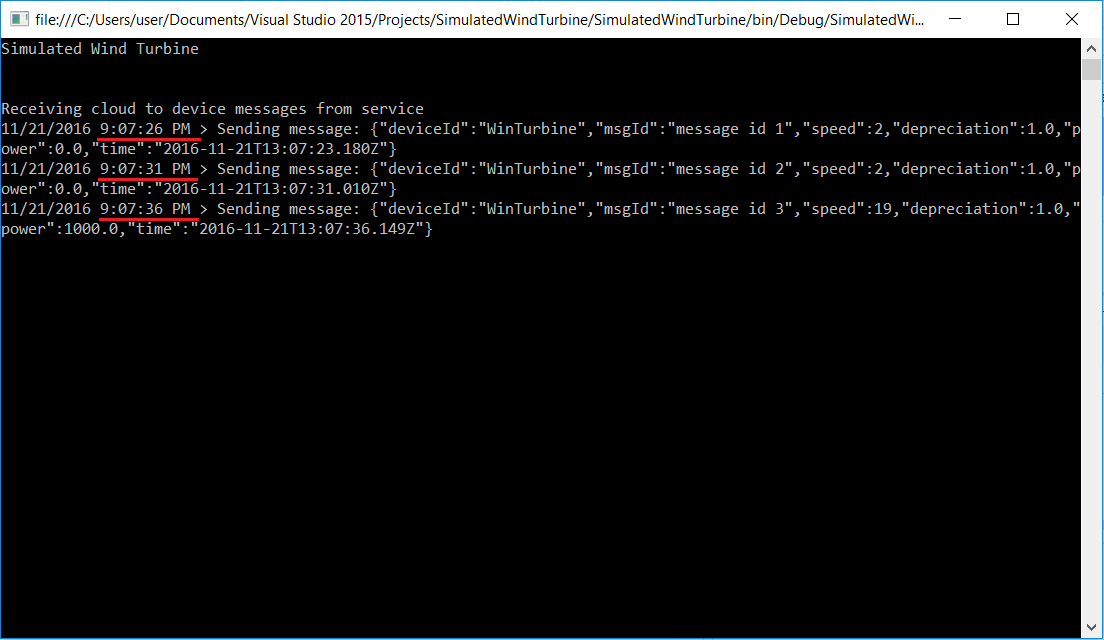
<add key="DeviceConnectionString" value="[Please place your device connection string]" />

To:

<add key="DeviceConnectionString" value="HostName=iothub112101.azure-devices.net;DeviceId=WinTurbine;SharedAccessKey=eyrYWTiqbImI6JEeYDykmzkZ6BT+Q1zbyWnHN3yKCHU=" />

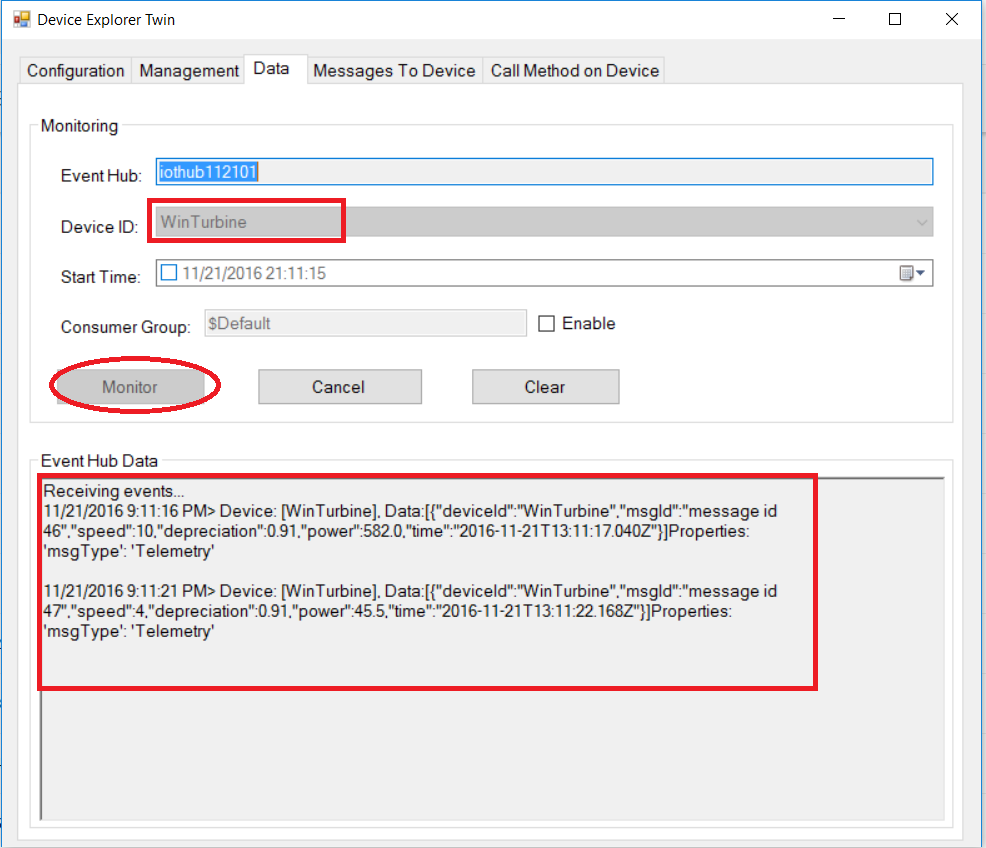


* Press the **F5** to start debugging and check the output of console
  + Send the telemetry data every 5 seconds



Step 6: Monitor the C# simulated device (WinTurbine)

* Open the Device Explorer and check the receiving events



Step 7: Test the Cloud to Device messages

* LinuxTurbine Commands (see the **c2d-linux-command.txt** under 03-HOL folder)

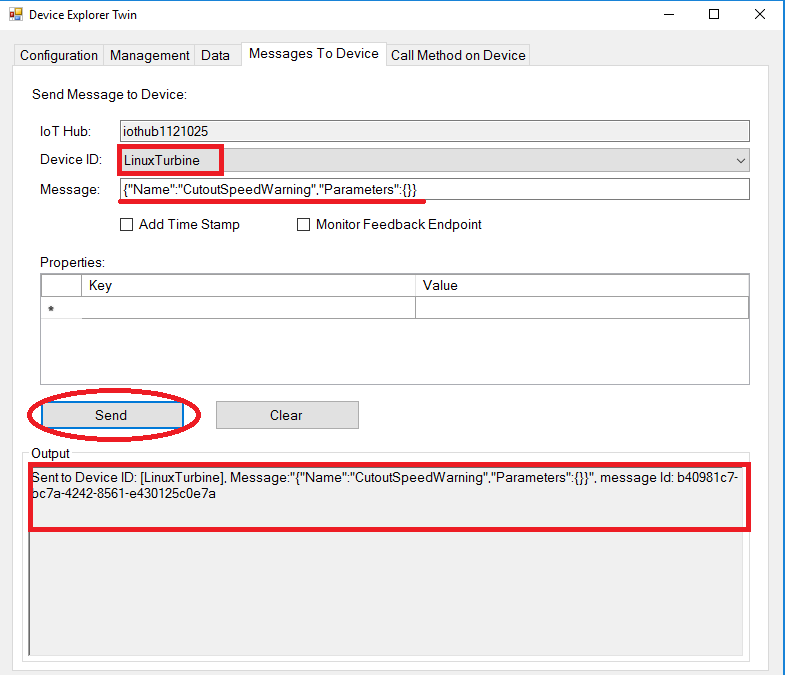
{"Name":"CutoutSpeedWarning","Parameters":{}}

{"Name":"RepairWarning","Parameters":{}}

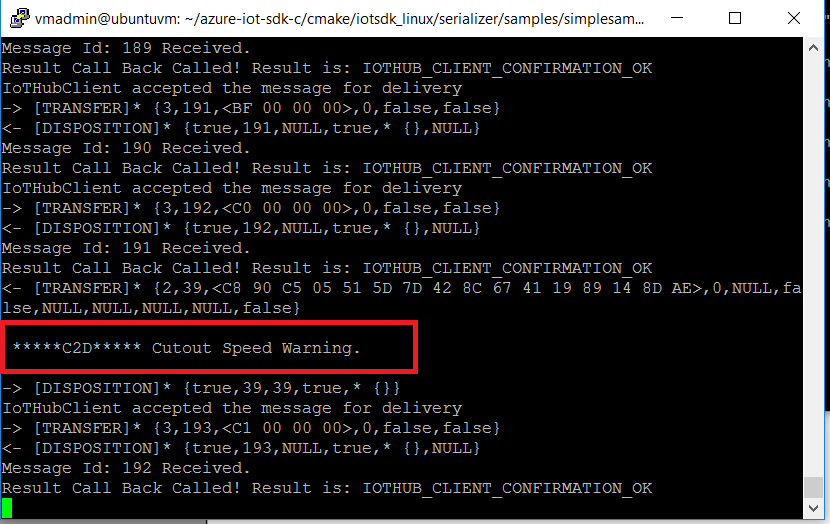
{"Name":"TurnOnOff","Parameters":{"On":1}}

{"Name":"ResetDepreciation","Parameters":{"Depreciation":1}}

* Device Explorer Tool



* PuTTY console



* WinTurbine (see the c2d-windows-command.txt under 03-HOL folder)

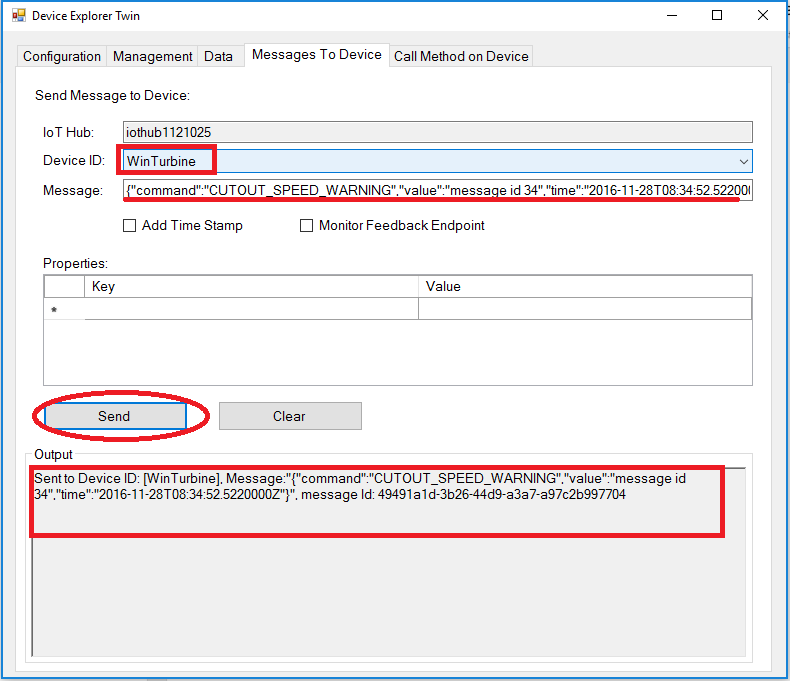
{"command":"CUTOUT\_SPEED\_WARNING","value":"message id 34","time":"2016-11-28T08:34:52.5220000Z"}

{"command":"REPAIR\_WARNING","value":"message id 34","time":"2016-11-28T08:34:52.5220000Z"}

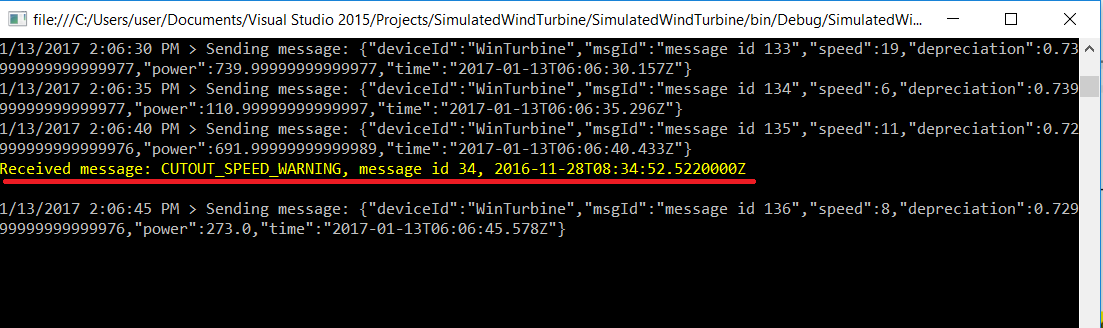
{"command":"TURN\_ONOFF","value":"1","time":"2016-11-28T08:34:52.5220000Z"}

{"command":"RESET\_DEPRECIATION","value":"1","time":"2016-11-28T08:34:52.5220000Z"}

* Device Explorer Tool



* Windows Console App



* *The HOL 3 has been completed. Now you have one Windows and one Linux simulated devices and they can send and receive the messages for the use case of wind turbine. In the next hands-on lab, we will learn how to process the IoT data as historic data by Azure Stream Analytics.*