# Getting Started with Microsoft Azure IoT Suite on Moxa UC-8100-LX Using NodeJS SDK

*Jimmy Chen (陳永達)*
*Field Application Engineer*
*jimmy.chen@moxa.com*

# Contents



 Released on January 18, 2017

**About Moxa**

Moxa is a leading manufacturer of industrial networking, computing, and automation solutions. With over 25 years of industry experience, Moxa has connected more than 30 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for automation systems. Information about Moxa's solutions is available at www.moxa.com. You may also contact Moxa by email at info@moxa.com.

**How to Contact Moxa**

Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

MOXA®
Reliable Networks ▲ Sincere Service

# Running the MS Azure IoT Suite on Moxa UC-8100-LX Using NodeJS SDK

*The instructions in this tech note apply only to the UC-81XX series of computers. These instructions MAY NOT be suitable for use in other computers because of the different Debian distribution versions, hardware peripherals, and firmware versions.*

*The instructions in this document require familiarity with the Microsoft Azure Portal and Microsoft Azure IoT Suite. Additional details on the Microsoft Azure Portal and Microsoft Azure IoT Suite are available at:*
*https://azure.microsoft.com*

## 1 Background

The purpose of this document is to provide step-by-step instructions on how to run the Microsoft Azure software development kit (SDK) on the UC-8100-LX computer.

## 2 Requirement

- UC-8100-LX/ UC-8100-LX-ME
- Microsoft Azure Portal account

## 3 Overview

The UC 8100-LX computer comes preinstalled with the Debian Linux distribution. It is easy to support the NodeJS Software Development Kit (SDK) for Microsoft Azure suite on the UC 8100-LX computer by installing the required libraries and tools. This document does not go into the details of how to use the Microsoft Azure IoT Suite, instead, the focus is on how to run the NodeJS SDK provided by Microsoft to connect to the Microsoft Azure Cloud service.

Once you have installed all the libraries and tools required for this distribution, follow the instructions given under the following sections:

1. Preparing the Microsoft Azure SDK environment on the UC-8100-LX
2. Creating resources in the Microsoft Azure IoT Suite
3. Connecting to the Microsoft Azure Cloud service

## 3.1 Preparing the Microsoft Azure Environment on the UC-8100-LX

### 3.1.1 Libraries and Tools

You will need to first install NodeJS SDK and all the relevant libraries on the UC-8100-LX to be able to run the source code. To download the libraries and tools, make sure that the UC-8100-LX is connected to the Internet and run the following commands with **root** authentication:

1. **#apt-get update**
2. **#apt-get upgrade**
3. **#apt-get install git -y**
4. **#apt-get install curl -y**

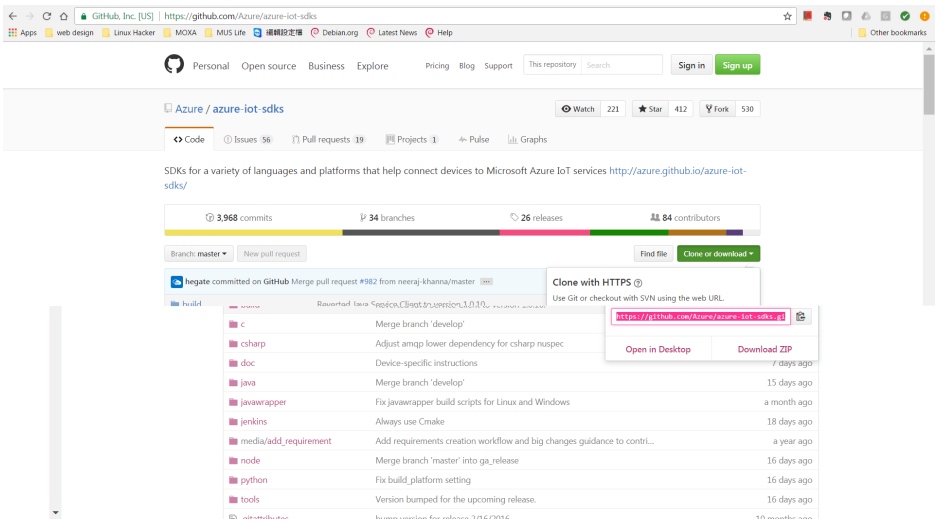
Microsoft Azure SDK requires the NodeJS language and NPM packages to be installed on the UC-8100-LX. To install these packages, run the following commands:

1. **#apt-get install nodejs -y**
2. **#update-alternatives --install /usr/bin/node nodejs /usr/bin/nodejs 100**
3. **#curl https://www.npmjs.com/install.sh | sh**

### 3.1.2 Building the C Source Code

Microsoft Azure IoT SDK is available free of cost at:
https://github.com/Azure/azure-iot-sdks

**Figure 1**



Download the Microsoft Azure IoT SDK and then use the **Git** tool to download the source code and dependent libraries to the **UC-8100-LX/home/moxa** directory on your UC-8100-LX using the following commands:

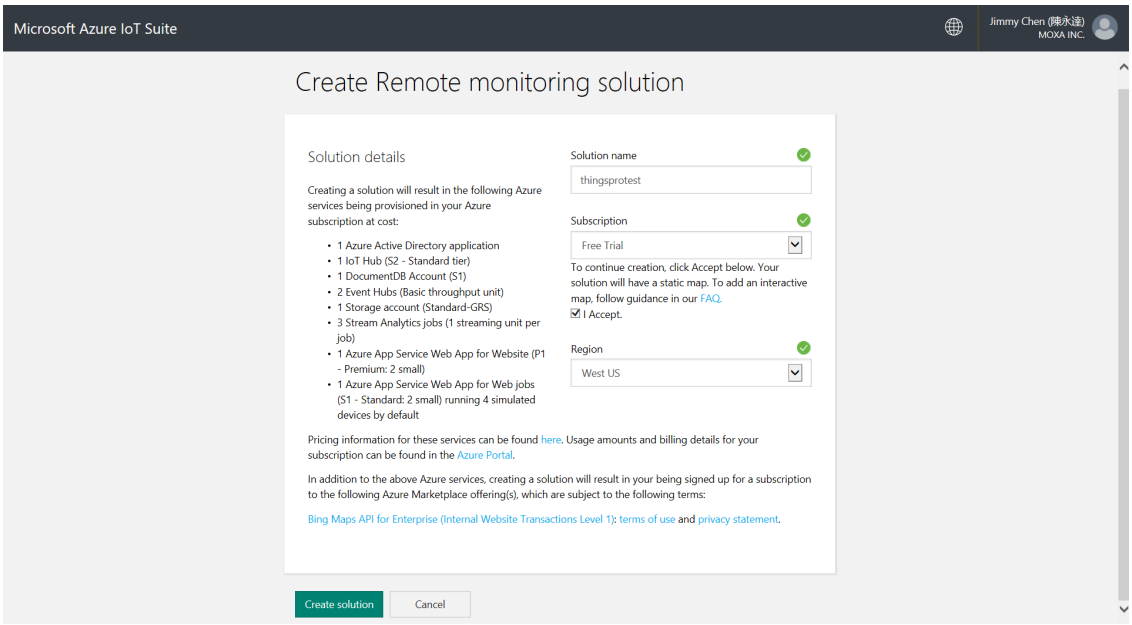
1. **#cd /home/moxa**
2. **#npm install azure**
3. **#git clone https://github.com/Azure/azure-iot-sdks.git**
4. **#cd *[PATH_OF_SOURCE_CODE]*/azure-iot-sdks/node/device/samples**
5. **#npm install**

## 3.2 Creating Resources in the Microsoft Azure IoT Suite

Once you set up an account on https://www.azureiotsuite.com/ and log in, you will be able to see detailed instructions and a wizard that will guide you through the process of creating a remote monitoring solution (Figure 2 to Figure 5).

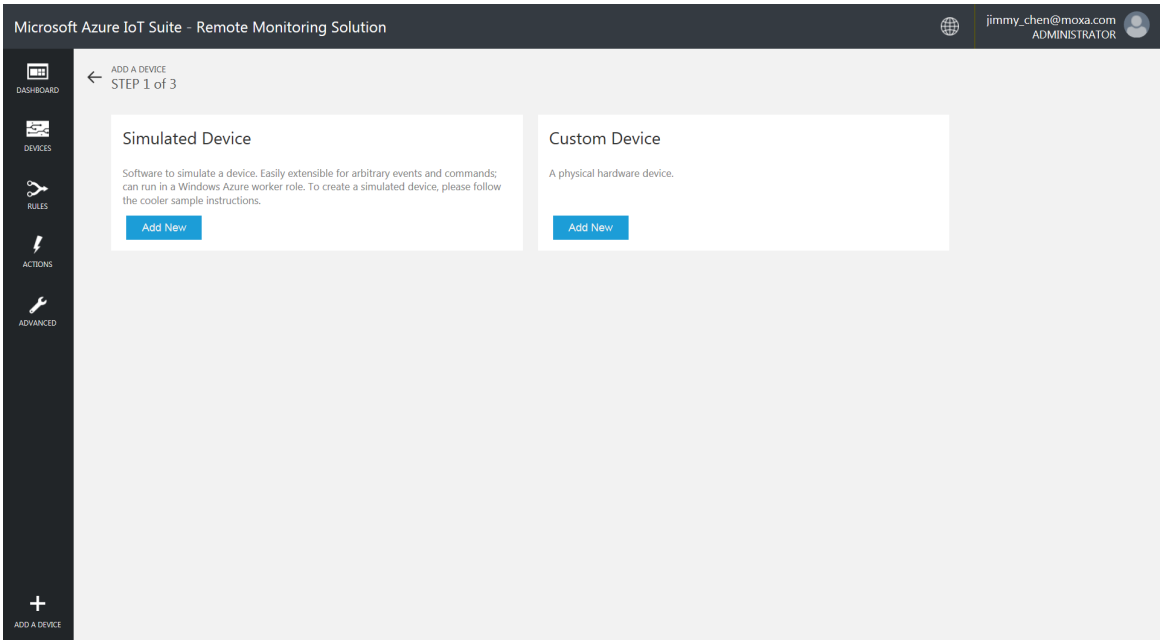
1. Enter the **Solution name** "thingsprotest", specify the **Subscription** and **Region** details, and click **Create solution**.

**Figure 2**



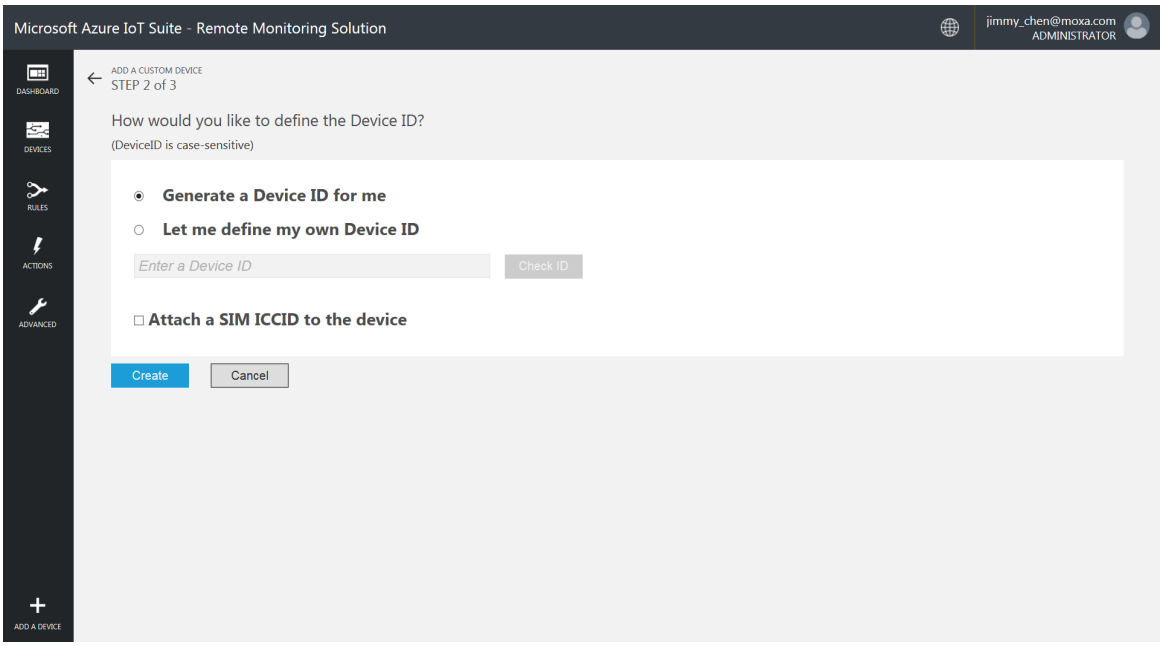
2. Select **Custom Device** and click **Add New**.

**Figure 3**




3. Select the **Let me define my own Device ID**.option and click **Create**.

**Figure 4**



4. Copy the device credentials and click on **Done** to complete the wizard.

> **NOTE** Copy the information in the **Device ID**, **IoT Hub Hostname**, and **Device Key** fields to a configuration file on your device so that it is available when you change your sample code.

**Figure 5**

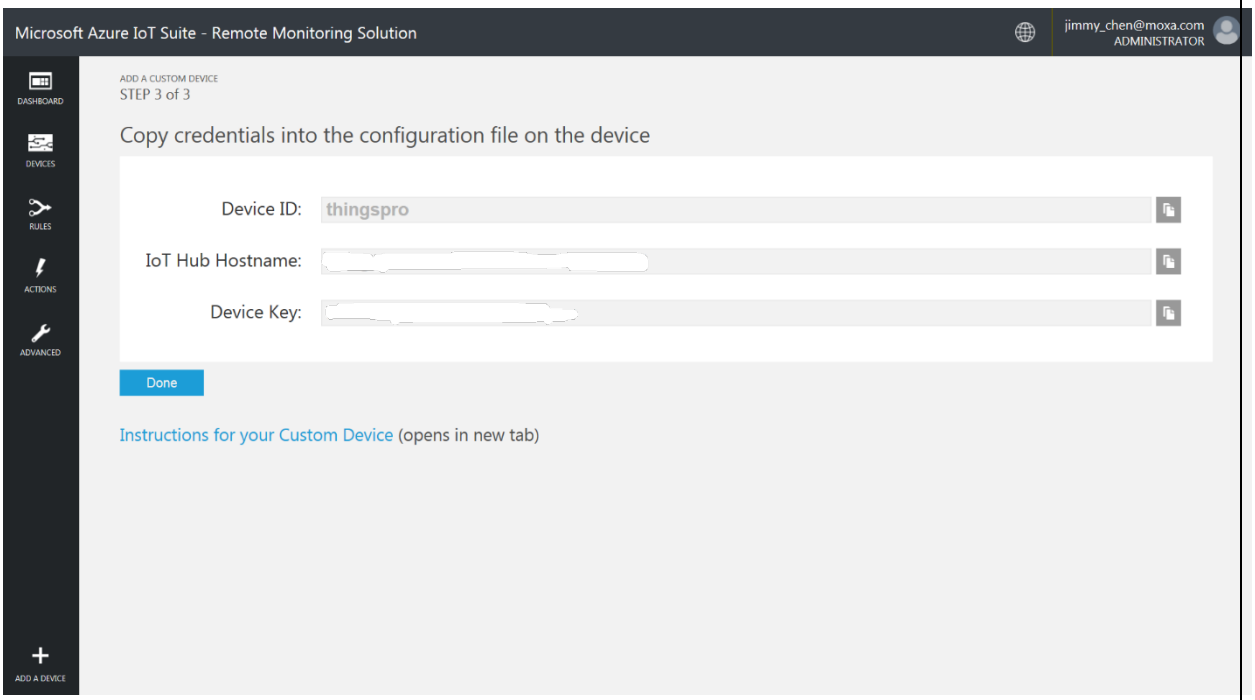
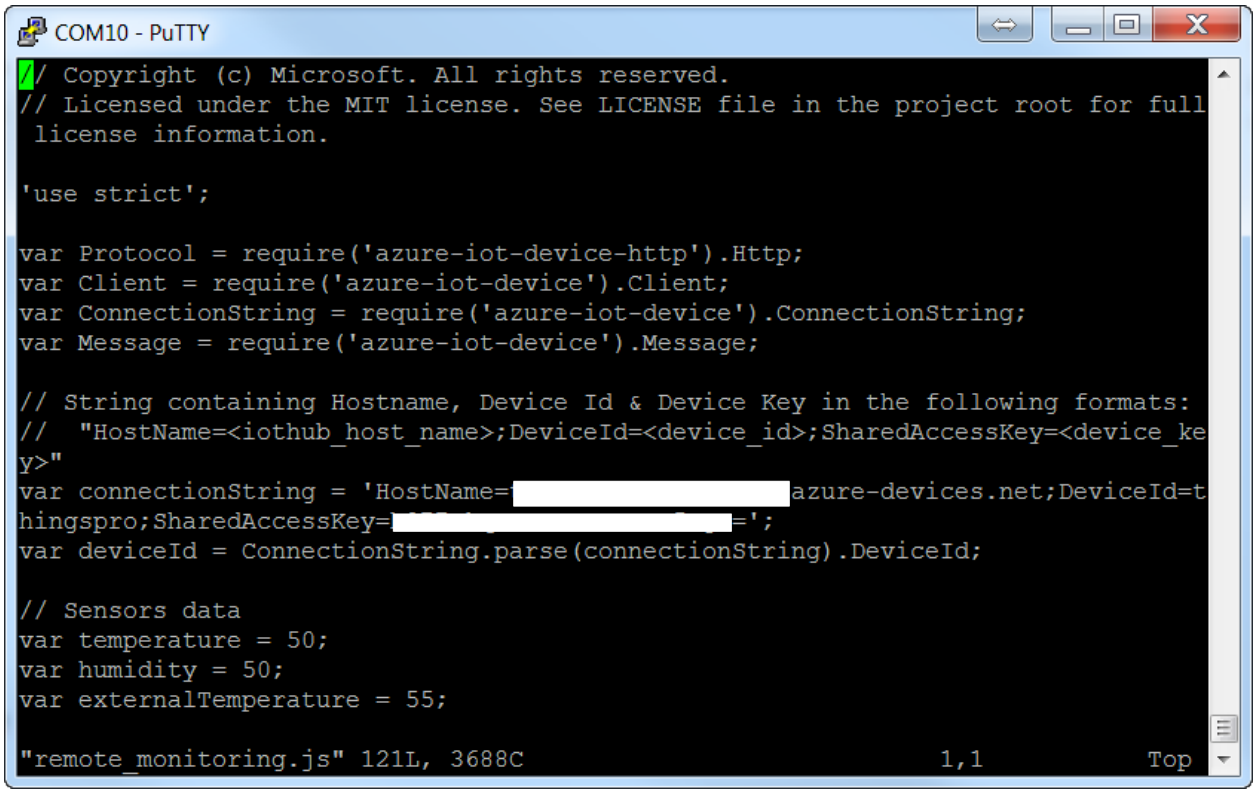



Figure 6 shows how you can change the variable `connectionString` in the NodeJS sample program, **remote_monitoring.js**.

Figure 6

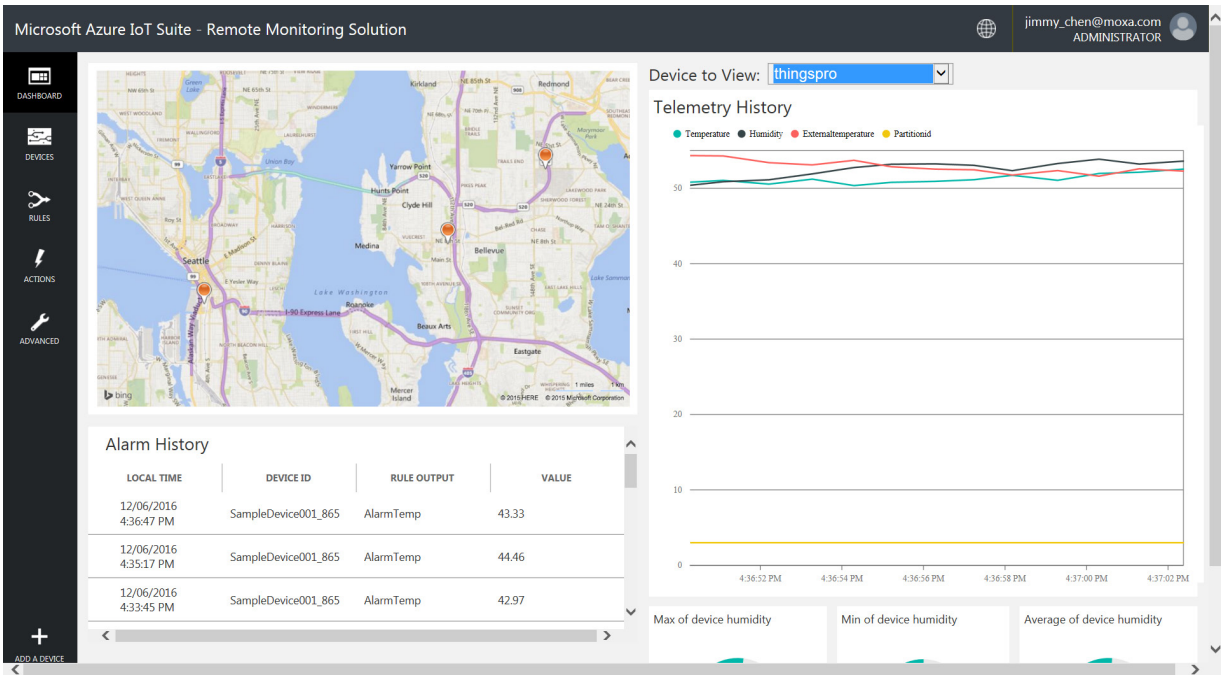



## 3.3 Connecting to the Microsoft Azure Cloud Service

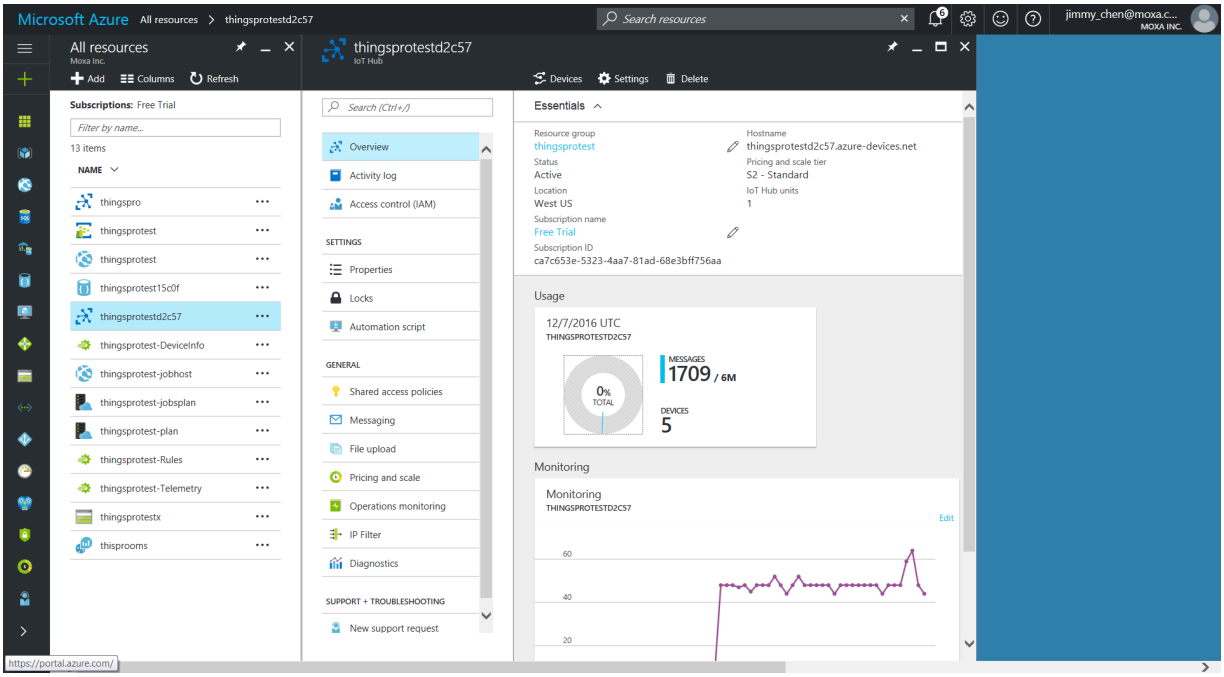
After installing all necessary tools and libraries and setting up the resources in Microsoft Azure Cloud, it is time to run the remote monitoring program using the following commands:

1. **#cd *[PATH_OF_SOURCE_CODE]*/azure-iot-sdks/node/device/samples**
2. **#export NODE_PATH="'$(npm root -g)'"**
3. **#nodejs remote_monitoring.js**

In the Microsoft Azure IoT Suite solution window, click on the **Dashboard** item and choose **thingsprotest** in the **Device to View** field to see a history graph as shown below:




Now, log in to the Microsoft Azure Portal using the device credentials to monitor the device. You will see a dashboard. Wait for the dashboard to update. You are all set when the values on the dashboard start to change.



# 4 Additional Reading

- https://github.com/Azure/azure-iot-sdks
- https://www.azureiotsuite.com/
- https://portal.azure.com
- http://www.moxa.com/product/uc-8100.htm