IoT Application with HANI board and Azure

Creator: Grzegorz Jakubowski
Revision 1.3, 11.10.2018
©2018 by ARROW

All rights reserved. No part of this manual shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, desktop publishing, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this document, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. All terms mentioned in this manual that are known to be trademarks or service marks are listed below. In addition, terms suspected of being trademarks or service marks have been appropriately capitalized. ARROW cannot attest to the accuracy of this information. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.
## Revision History

<table>
<thead>
<tr>
<th>Revision, Date</th>
<th>Editor</th>
<th>Subject (major changes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision 1.1</td>
<td>Grzegorz Jakubowski</td>
<td>Initial Document</td>
</tr>
<tr>
<td>Revision 1.2</td>
<td>Kristóf Féja</td>
<td>Edits</td>
</tr>
<tr>
<td>Revision 1.3</td>
<td>Grzegorz Jakubowski</td>
<td>Edits</td>
</tr>
</tbody>
</table>
Table of Contents

Revision History ........................................................................................................3
Table of Contents ......................................................................................................4
Overview ..................................................................................................................5
  Description ............................................................................................................5
  Kit contents .........................................................................................................5
System overview ......................................................................................................6
  Block diagram ....................................................................................................6
  Main devices .......................................................................................................7
Requirements ..........................................................................................................8
  Hardware ............................................................................................................8
  Software ............................................................................................................8
  Configuration .....................................................................................................8
    Download and install MCUXpresso ...............................................................8
    Debug probes ..................................................................................................8
Microsoft Azure Configuration ...............................................................................10
  Getting Microsoft Azure subscription ..............................................................10
Preparing HANI board application .......................................................................16
  Prerequisites .....................................................................................................16
  Setting up the HANI board ...............................................................................19
  Building the application on the HANI board ....................................................20
Data visualization in Azure ....................................................................................25
Overview

This guide will show an example how to demonstrate an IoT application with HANI board connecting to Microsoft Azure.

Description

HANI is a board focused on HMI (Human Machine Interface) supporting multiple display sizes and connectivity protocols with a multi-protocol wireless module, a Wi-Fi module, a NFC reader, CAN, and Ethernet and USB interfaces. On-board sensors make the HANI board a robust IoT kit.

Azure IoT Hub

Azure IoT Hub is a fully managed Azure service. This service enables reliable and secure bi-directional communications between millions of Internet of Things (IoT) devices and a solution back end. One of the biggest challenges that IoT projects face is how to reliably and securely connect devices to the solution back end. To address this challenge, IoT Hub:

- Offers reliable device-to-cloud and cloud-to-device hyper-scale messaging.
- Enables secure communications using per-device security credentials and access control.
- Includes device libraries for the most popular languages and platforms.

(Azure IoT Hub get started tutorials - Microsoft)

Kit contents

The following items are included in the box:

- 1x HANI board
- 1x 5V 2A power wall adapter (input: 100-240 V, 50/60 Hz)
- 4x 10mm plastic spacers mounted on the board, 8x additional 20mm plastic spacers
System overview

Block diagram

An overview of the functional blocks of HANI board is shown in the figure below.
Main devices

HANI board main features are summarized in the picture below.
Requirements

Hardware

- HANI board
- 5V 2A power wall adapter (input: 100-240 V, 50/60 Hz)
- Segger J-link with USB cable
- Wi-Fi access point

Software

- MCUXpresso Integrated Development Environment (IDE) v10.1.0 or later release [http://www.nxp.com/mcuxpresso/ide](http://www.nxp.com/mcuxpresso/ide)
- MCUXpresso SDK for LPC54618J512 v2.3.0 or later release [https://mcuxpresso.nxp.com/en/welcome](https://mcuxpresso.nxp.com/en/welcome)
- Microsoft Azure subscription
- Device Explorer from Microsoft ([link](#))
- Segger J-link driver
- Application source code

Configuration

**Download and install MCUXpresso**
Download and install MCUXpresso Integrated Development Environment (IDE) ([http://www.nxp.com/mcuxpresso/ide](http://www.nxp.com/mcuxpresso/ide)). It requires to create an account on NXP web page.

**Debug probes**
An external debug probe is required to flash and debug the software. Any device supported by NXP MCUXpresso should work properly.
During the lab Segger JLink will be used.
1. Please **download JLink V6.33f or later release and install**. The software is available on Segger website: [https://www.segger.com/downloads/jlink/](https://www.segger.com/downloads/jlink/)

### J-Link Software and Documentation Pack

- All-in-one debugging solution
- Can be downloaded and used free of charge by any owner of a SEGGER JLink, J-Trace or Flasher model. Not all features of it may be available on all J-Link / J-Trace / Flasher models.
- Updated frequently
- Release Notes
- More information

[Click for downloads](#)
2. After installation is complete, please open **Window → Preferences** menu in **MCUXpresso** and select the proper **path for J-link Server executable**:

![Preferences menu](image)

3. Check also that discovery of **Segger J-Link probes** is enabled in **Window → Preferences → MCUXpresso IDE → Debug Probe Discovery**.

![Debug Probe Discovery](image)
Microsoft Azure Configuration

Getting Microsoft Azure subscription

To start with this hands-on. An Azure subscription is required. An Azure free trial and its feature can be obtained here.

Creating Azure IoT hub resources

In the following steps, you will create a Resource Group that collects all the resources you will use in Azure for this hands-on. One of this resource is the IoT Hub that is basically the cloud application itself.

1. Sign in to the Azure portal with your account
   Select Create a resource ➔ Internet of Things ➔ IoT Hub
2. In the IoT hub pane, enter following information

**Subscription:** Name of your subscription

**Resource group:** Create a resource group or use an existing one

**Region:** Choose closest location

**Name:** Name of your IoT hub, if the name is valid, a green check mark appears. Remember this name since it will be used in the next section

**Review + create:** Click to create
3. In the **IoT hub pane** review the settings than click **Create**!

4. When the deployment is finished You will be taken to the newly created IoT Hub.
Locating the connection key

In order to connect a device to Azure you will need to use the custom key that was generated for your application (IoT Hub resource). You can consider this as a password each end device that you want to connect has to know. In this case scenario it is not the physical HANI board that will be connected to Azure but a twin device (a logical equivalent of the HANI that you will create in the next chapter). The Hani board will connect to the twin device.

1. Click “All resources” and select the IoT Hub you just created (named “hani1” in this example)!
2. Go to “Shared access policies”, select “iothubowner” and copy the “Connection string – primary value”!
There are different access levels provided by Microsoft. The “iothubowner” level provides all the permissions.

Creating the twin device

1. Double-click on “DeviceExplorer.exe”!
2. Open the “Configuration” tab and insert the “Connection string – primary key” copied in the above step!
3. Click “Update”!

4. Open the “Management” tab and click on “Create”!
5. **Give an ID** as your “**Device ID**” (for example `hanidev1`) and make sure you have the settings as shown below!

6. **Click “Create” then “Done”!**

Now we need to create a token. Only hardware with the right token can connect to the twin device.

7. Still **remaining** in the “**Management**” tab, click on the “**SAS Token...**” button!

8. **Select** the “**Device ID**” you just created and **click “Generate”**!
   (In a later stage you will need to add the last part of the token string to the application source code.)
Preparing HANI board application

Before moving further with this step, you should have already setup a HUB IoT and device in the Microsoft Azure portal.

Prerequisites

The following development tools should be installed before trying to either create a project or use any example based on HANI board:

- MCUXpresso Integrated Development Environment (IDE) v10.1.0 or later release
- MCUXpresso SDK for LPC54618J512 v2.3.0 or later release

1. Click “Select Development Board”!
2. Type “LPC54618” in the Search by Name field and select processor LPC54618J512! Click on “Build MCUXpresso SDK”!

3. Click on “Add software component”!

4. Open the “items selected” drop-down menu, scroll to the bottom and make sure that “Amazon-FreeRTOS Kernel” is selected! Click on “Save Changes”!
5. **Click on “Request Build” button!**

6. Once built, **click on the “Download SDK archive” button!**
7. The SKD has to be installed from within MCUXpresso. **Drag and drop** the SDK archive zip file into the `Installed SDKs` view. (See below).
Setting up the HANI board

HANI Board need to be properly configured before starting to use it.

1. Connect Segger J-Link to HANI board J21 connector, pay attention to the cable polarization (see below).
2. Connect power supplier to HANI Board (see below).
Building the application on the HANI board

In the next step you will open the project file of this hands-on and modify certain settings to have the application working with your local WiFi access point, with your Azure IoT Hub and twin device.

1. **Unzip** the “hani_azure.zip” archive into the work folder you created when opening MCUXpresso IDE.
2. Go to File → Open Projects from File System...

3. **Click** on the “Directory…” button and **select** the extracted “hani_azure” folder! **Click OK**!
4. **Verify** that the **hani_azure** project is detected! **Click “Finish”!**

5. Once the project is imported, **open** the “wifi_task_cfg.h” file! (**Project Explorer window → hani_azure → source**)  
6. Go to lines # 46 & 47 and modify “WIFI_TASK_SSID” and “WIFI_TASK_PASSWORD” values to reflect the local WiFi settings!
7. Next open “wifi_task.c” file and modify lines #58-60 according to the below table!

<table>
<thead>
<tr>
<th>Edit line</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>#58</td>
<td>Instead of devid1 put your Device Id.</td>
</tr>
<tr>
<td>#59</td>
<td>Instead of hani1 put your IoT Hub.</td>
</tr>
<tr>
<td>#60</td>
<td>Instead of sr=hani1.azure-devices.net%2FDevices%2Fdevid1&amp;sig=j064hA1jEuxUKe%2BwJtu7Q%3D&amp;se=15693236 put your SAS Token generated during creating device in Device Explorer. Take a part marked in green.</td>
</tr>
</tbody>
</table>

8. **Connect** your J-Link probe to your **laptop** and to the **board’s J21 header**, then provide power to the board through **J1 power connector**!

9. **Click Debug** in the **Quickstart Panel**!
10. **Select** your J-Link probe then **click OK**!

11. **Click** the **Resume All** button to run the application!
12. If You followed the instructions properly You shall see sensor data getting streamed to the twin device in your Device Explorer. Go to the **Data** tab of your **Device Explorer**, select your **Device ID** and click the **Monitor button**!
Data visualization in Azure

For this purpose we will use the Time Series Insights service from Azure. It is a fully managed analytics, storage, and visualization service for managing IoT-scale time-series data in the cloud.

1. **Open** the Management tab in Azure (menu on the left side)!
2. **Click** on Create a resource button, select Internet of Things, then click on Time Series Insights!
3. Enter the following information, click Create:

- **Environment name:** hani_tsi
- **Subscription:** Choose your subscription
- **Resource group:** Choose your resource
- **Click:** Create
4. Go to your Azure resources, open the just created “hani_tsi” resource. Open Data Access Policies and click on the Add button!
5. **Click Select user**, fill your e-mail address used in Azure if not visible in the list.
6. **Select your account name**, click the **Select** button!

7. **Click Select role**, select Reader and Contributor options!
8. **Click OK** and **OK** again!
9. **Switch** from Data Access Policies to **Event Sources**! **Click** on **Add**!

10. **Enter** the following **information** in the **New event source** panel, **click Create**:

    - **Event source name**: hanisource
    - **Source**: Iot Hub
    - **Iot hub name**: Hani1

    ![New event source panel](image)
11. **Switch** form Event Sources to the **Overview panel** and **open** the link (marked on the below picture)!

![Overview panel with link](https://Insights.timeseries.azure.com/?environmentId=...)

12. A new webpage should appear and data from the HANI board should be displayed on the chart! Data shown on the chart can be changed by choosing value from **measure** drop down menu.
Additional charts can be added by clicking Add button.